Three days training programme on

Empowering Agriculture through Artificial Intelligence and Digital Technologies

August (20-22), 2024

Training Mode: Online

Platform: NARES - Blended Learning Platform (BLP)

Brief About the Programme:

This training program is organized to enhance participants' expertise in leveraging AI, Machine Learning, and digital technologies in the agricultural domain. It encompasses foundational Python programming, advanced AI applications in crop protection, improvement, and animal science, as well as the integration of technologies such as GIS, Deep Learning, and AR/VR, to foster innovation and efficiency in agriculture.

Training Objectives

- Learn about digital transformations and AI applications under NAHEP and in agriculture.
- Gain foundational skills in Python programming and key libraries used in AI.
- Study how AI is applied to crop protection, animal, and plant sciences.
- Understand the integration of AI with GIS for accurate yield forecasting in agriculture.
- Get hands-on experience with deep learning models applied to agricultural datasets.
- Explore generative AI like ChatGPT and AR/VR applications in agriculture.

Course Coordinators

Dr Madhu
Scientist,
Division of Computer Applications,
ICAR-IASRI, New Delhi

Dr Sapna Nigam
Scientist,
Division of Computer Applications,
ICAR-IASRI, New Delhi

Mr Akshay Dheeraj
Scientist,
Division of Computer Applications,
ICAR-IASRI, New Delhi

Course Directors

Dr. Sudeep Marwaha
PI, NAHEP Component 2
& RAES Project,
ICAR-IASRI, New Delhi

Dr. Rajender Parsad
Director,
ICAR-IASRI, New Delhi

Scan here to register

कृत्रिम बुद्धिमत्ता और डिजिटल प्रौद्योगिकियों के माध्यम से कृषि को सशक्त बनाना

Empowering Agriculture through Artificial Intelligence and Digital Technologies

20 – 22 August 2024

Reference Manual

Course Coordinators:
Dr. Madhu, Dr. Sapna Nigam, Mr. Akshay Dheeraj

ICAR- Indian Agricultural Statistics Research Institute
Library Avenue, New Delhi – 110012

Editors:

Dr. Madhu

Dr. Sapna Nigam

Mr. Akshay Dheeraj

Disclaimer: The information contained in this reference manual has been taken from various web resources. Respective URLs are mentioned in the content.

Citation:

Madhu, Nigam, S., Dheeraj, A., (2024). **Empowering Agriculture through Artificial Intelligence and Digital Technologies**. Reference Manual, ICAR-Indian Agricultural Statistics Research Institute, New Delhi.

प्रस्तावना

भा°कृ°अनु°प°–भारतीय कृषि सांख्यिकी अनुसंधान संस्थान (भा°कृ°सां°अ°सं°), सांख्यिकीय विज्ञान (सांख्यिकी, संगणक अनुप्रयोग, जैव सूचना विज्ञान) में प्रासंगिकता और कृषि अनुसंधान की गुणवत्ता को समृद्ध करने और सूचित नीति निर्णय लेने के लिए कृषि विज्ञान में उनके विवेकपूर्ण संलयन का एक प्रमुख संस्थान है। संस्थान ने कृषि अनुसंधान के लिए उपयोगी विभिन्न सॉफ्टवेयर, वेब एप्लिकेशन, मोबाइल ऐप और अब आर्टिफिशियल इंटेलिजेंस (एआई) आधारित टूल, तकनीक और कार्यप्रणाली विकसित करने में अग्रणी भूमिका निभाई है।

"आर्टिफिशियल इंटेलिजेंस और डिजिटल प्रौद्योगिकियों के माध्यम से कृषि को सशक्त बनाना" पर प्रशिक्षण कार्यक्रम का उद्देश्य कृषि क्षेत्र में आर्टिफिशियल इंटेलिजेंस, मशीन लर्निंग और डिजिटल प्रौद्योगिकियों का लाभ उठाने में प्रतिभागियों की विशेषज्ञता को बढ़ाना है। यह प्रशिक्षण कार्यक्रम प्रतिभागियों को एआई के गतिशील क्षेत्र में सफलता प्राप्त करने के लिए आवश्यक कौशल, ज्ञान और संसाधनों से लैस करने के मुख्य उद्देश्य से बनाया गया है। यह प्रशिक्षण कार्यक्रम आचार्य एनजी रंगा कृषि विश्वविद्यालय, गुंटूर, आंध्र प्रदेश के संकाय सदस्यों के लिए सोच-समझकर डिजाइन किया गया है ताकि प्रतिभागियों को आवश्यक कौशल, ज्ञान और संसाधनों के साथ सशक्त बनाया जा सके जो आर्टिफिशियल इंटेलिजेंस के तेजी से विकसित हो रहे क्षेत्र में उत्कृष्टता प्राप्त करने के लिए महत्वपूर्ण हैं।

हमने प्रतिभागियों को एनएएचईपी के तहत और कृषि में डिजिटल परिवर्तनों और एआई अनुप्रयोगों के बारे में जानने के लिए, पायथन प्रोग्रामिंग और एआई में उपयोग की जाने वाली प्रमुख लाइब्रेरी में मूलभूत कौशल हासिल करने के लिए, फसल सुरक्षा, पशु और पौधों पर एआई को कैसे लागू किया जाता है, इसका अध्ययन करने के लिए इस प्रशिक्षण कार्यक्रम को डिजाइन किया है। विज्ञान, कृषि में सटीक उपज पूर्वानुमान के लिए जीआईएस के साथ एआई के एकीकरण को समझना, कृषि डेटासेट पर लागू गहन शिक्षण मॉडल के साथ व्यावहारिक अनुभव प्राप्त करना और कृषि में चैटजीपीटी और एआर/वीआर अनुप्रयोगों जैसे जेनरेटिव एआई का पता लगाना।

हम इस अवसर पर संस्थान के संकाय को धन्यवाद देना चाहते हैं जिन्होंने इस पाठ्यक्रम को सफल बनाने में अपना बहुमूल्य समय दिया। उनके सहयोग के बिना इस मैनुअल को समय पर पूरा करना संभव नहीं होता। हम इस प्रशिक्षण कार्यक्रम में अपने कर्मचारियों को तैनात करने के लिए आचार्य एन जी रंगा कृषि विश्वविद्यालय के भी आभारी हैं। हम डॉ. राजेंद्र प्रसाद, निदेशक, भा°कृ°अनु°प° – भा°कृ°सां°अ°सं° और डॉ. सुदीप मारवाह, प्रमुख, संगणक अनुप्रयोग प्रभाग के उनके बहुमूल्य मार्गदर्शन और पाठ्यक्रम के सुचारू संचालन के लिए सभी आवश्यक सुविधाएं उपलब्ध कराने के लिए आभारी हैं। हम उन सभी के आभारी हैं जिन्होंने इस प्रशिक्षण मैनुअल को तैयार करने के लिए प्रत्यक्ष या अप्रत्यक्ष रूप से हमारा समर्थन किया है।

(डॉ मधु)
पाठ्यक्रम समन्वयक

(डॉ सपना निगम)
पाठ्यक्रम समन्वयक

(श्री अक्षय धीरज)
पाठ्यक्रम समन्वयक

PREFACE

ICAR- Indian Agricultural Statistics Research Institute (ICAR-IASRI) is a premier Institute of relevance in Statistical Sciences (Statistics, Computer Applications and Bioinformatics) and their judicious fusion in agricultural sciences for enriching quality of agricultural research and informed policy decision making. The Institute has taken a lead in developing various Software, Web Applications, Mobile apps and now Artificial Intelligence (AI) based tools, techniques, and methodologies useful for Agricultural Research.

The objective of the training program *on "Empowering Agriculture through Artificial Intelligence and Digital Technologies"* is to enhance participants' expertise in leveraging Artificial Intelligence, Machine Learning, and digital technologies in the agricultural domain. This training program is designed with the core aim of equipping participants with essential skills, knowledge, and resources critical for achieving success in the dynamic field of AI. This training program has been thoughtfully designed for faculty members of Acharya N G Ranga Agricultural University, Guntur, Andhra Pradesh to empower participants with the essential skills, knowledge, and resources that are crucial for excelling in the rapidly evolving field of Artificial Intelligence.

We have designed this training program to learn the participants about digital transformations and AI applications under NAHEP and in agriculture, to gain foundational skills in Python programming and key libraries used in AI, to study how AI is applied to crop protection, animal, and plant sciences, to understand the integration of AI with GIS for accurate yield forecasting in agriculture, to get hands-on experience with deep learning models applied to agricultural datasets and to explore generative AI like ChatGPT and AR/VR applications in agriculture.

We would like to take this opportunity to thank the faculty of the institute who have spared their valuable time in making this course successful. Without their cooperation timely completion of this manual would not have been possible. We are also thankful to the Acharya N G Ranga Agricultural University for deputing their employees in this training programme. We are grateful to Dr Rajender Parsad, Director, ICAR-IASRI and Dr. Sudeep Marwaha, Head, Division of Computer Applications for their valuable guidance and making all necessary facilities available for smooth conduct of the course. We are thankful to everyone who has supported us directly or indirectly for preparing this training manual.

Madhu 24/9/24

(Dr Madhu)

Course Coordinator

Sapna Nigam 24/09/24

(Dr Sapna Nigam)

Course Coordinator

24/09/24

(Mr. Akshay Dheeraj)

Course Coordinator

Content

Training Programme
On

"Empowering Agriculture through Artificial Intelligence and Digital Technologies"

Tentative Training Schedule: 20-22 August, 2024

Dates	09:30-10:30	10:30-11:15	11:30-1:00	LUNCH BREAK	2:15-3:30		3:45-5:30
20/08/2024 Tuesday	Inaugural Session	Digital Initiatives under NAHEP (Dr. Sudeep Marwaha)	Introduction to AI & Machine Learning in Agriculture (Dr. Sapna Nigam)		Basics of Python (Dr. Madhu)		Python Libraries (Dr. Sanchita Naha)
21/08/2024 Wednesday	AI Applications in the area of Crop Protection (AI-DISC) (Dr. Md. Ashraful Haque)	AI Application in Crop Improvement (Dr. Alka Arora)	AI Initiatives in Animal Science Domain (Dr. Chandan/Dr. Sanchita)		AI in Agricultural Education (Dr. Shashi Dahiya)	GIS & AI, AI for Yield Forecasting (Dr. Anshu Bharadwaj)	Augmented Reality/Virtual Reality (AR/VR) (Dr. Samarth Godara)
22/08/2024 Thursday	Deep Learning in Agriculture (Mr. Akshay Dheeraj)	Hands-on on Deep Learning in Agriculture (Dr. Md. Ashraful Haque)	Generative AI: ChatGPT (Dr. Soumen Pal)		NARES-Blended Learning Platform (Dr. Madhu)		Valedictory Session

Digital Initiatives under NAHEP

Dr. Sudeep Marwaha
Division of Computer Applications
ICAR-Indian Agricultural Statistics Research Institute, New Delhi – 110012
sudeep@icar.gov.in

The National Agricultural Research and Education and Extension System (NAREES) of India is one of the largest agricultural education systems with 4 Deemed-to-be Universities, 3 Central Agricultural Universities, 4 Central Universities with Agricultural Faculties, 65 State Agricultural Universities (SAUs), 114 institutions (including Agricultural Technology Application Research Institute- ATARI) and 731 Krishi Vigyan Kendra (KVK) under the aegis of Indian Council of Agricultural Research. These universities and institutions offer a wide range of courses, including UG, PG, and Doctoral programs in agriculture, horticulture, animal husbandry, fisheries, and other related fields and also conduct research in agricultural sciences, provide extension services to farmers and other stakeholders.

The Indian Council of Agricultural Research (ICAR), established in 1929, is the apex body responsible for coordinating and promoting agricultural education and research in India. ICAR plays a pivotal role in formulating policies, setting standards, and developing curricula to enhance agricultural education across the country. ICAR's mission is to "promote agricultural research, education, and extension for sustainable development." It coordinates with various agricultural universities and institutions to ensure the dissemination of quality education to aspiring agricultural professionals.

The Changing World of Education and the Need for New Initiatives in Agricultural Education

Education, as a whole, has undergone a significant transformation in recent years. The integration of technology, interactive learning platforms, and real-world experiences are enhancing the effectiveness of education, promoting critical thinking, problem-solving skills, and innovation.

The evolution of digital education has been a remarkable journey. From its humble beginnings as a mere concept, it has rapidly transformed into a global phenomenon that has revolutionized the way knowledge is imparted and acquired.

Recognizing the pressing need for transformation, agricultural education has embraced the digital revolution fascinatingly.

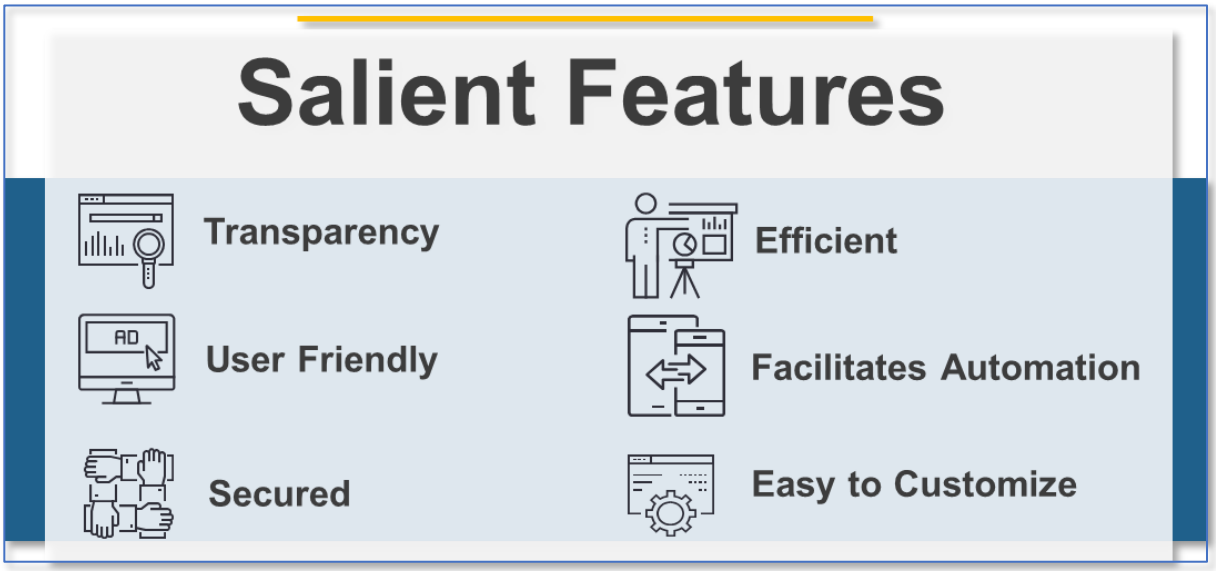
Embracing Change: ICAR's Commitment to Innovation

Recognizing the potential of strategic interventions in revolutionizing agricultural education, ICAR has embraced the digital revolution in education. This commitment stemmed from the understanding that strategic interventions aided by technology can bridge gaps, foster inclusive education, and enable the efficient dissemination of knowledge existing in agricultural education. By leveraging digital tools and platforms, ICAR aims to ensure that agricultural education remains relevant, dynamic, and resilient on the face of adversities. To keep the system dynamic and resilient, ICAR launched RAES (Resilient Agricultural Education System) under NAHEP (National Agricultural Higher Education Project) which is a robust three-tiered digital framework that has been put in place to strengthen digital infrastructure, enhance digital capacity and create robust, relevant digital content for system wide consumption. Through this initiative, ICAR is paving the way for a more inclusive and transformative learning experience for students and educators.

Digital Infrastructure

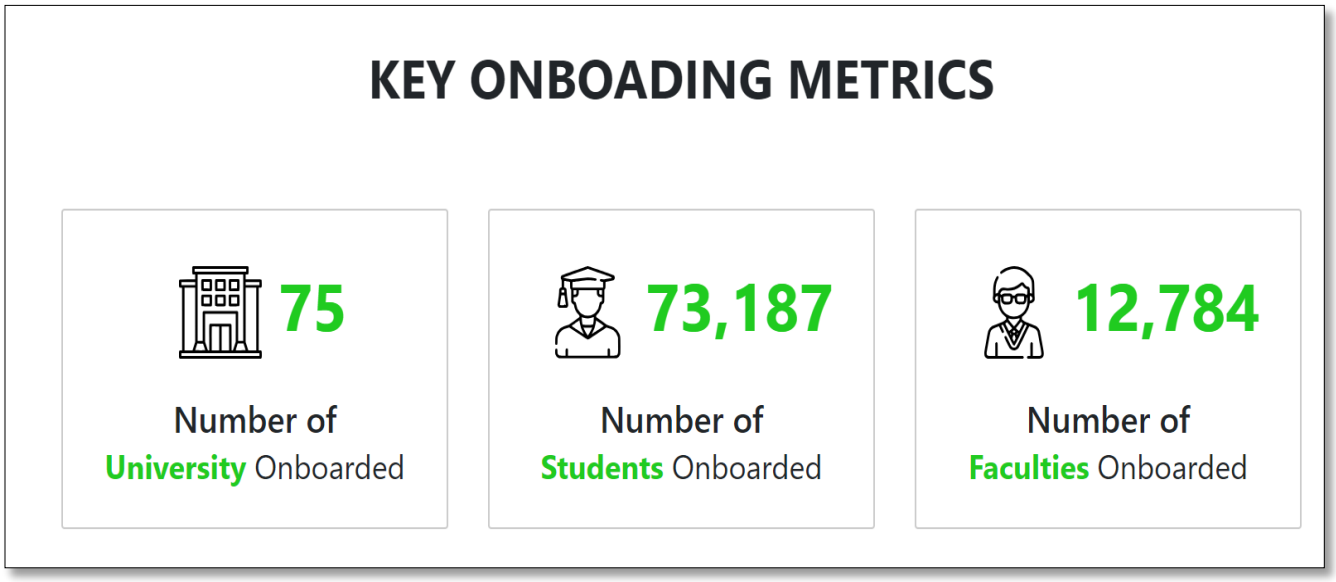
Digital infrastructure is revolutionizing the education system in the field of agricultural education by providing a platform for transformative learning experiences. Digital infrastructure is the key enabler and foundation of Resilient Agriculture Education System (RAES) to achieve the end goal of providing an exceptional learning experience to various stakeholders through enabling a robust digital eco-system. Under digital infrastructure, ICAR has launched various initiative:

Academic Management System (AMS): The Academic Management System is an integrated digital platform that streamlines administrative processes, student management, and facilitates effective communication between universities, colleges, and ICAR. Agricultural Universities and Institutions have not only welcomed this initiative but have also successfully completed their day-to-day tasks with it. A total of 63 universities are using the AMS system for their regular day to day administrative tasks. This also includes the 417 Registered Institutes/Colleges, 62,784 registered students, 7911 Registered faculty members under the universities.



Krishi Megh: The Krishi Megh was launched to meet the growing IT needs of the NARES system with the employment of applications such as e-Office, ICAR-ERP, Education Alumni Portal, e-Courses etc. More than 80 applications and websites are hosted on the Krishi Megh, which acts as a digital backbone for digital infrastructure.

Blended Learning Platform (BLP): The Blended Learning Platform facilitates a hybrid learning approach. It combines traditional classroom instruction with online modules and resources, allowing students to learn at their own pace, access quality content, and engage in interactive learning experiences. The platform is being implemented in all agricultural universities to equip students with the necessary knowledge and competencies to excel in the agricultural sector while promoting innovation and preparing them for the challenges of the digital age.



Agricultural Education Portal: The Agricultural Education Portal provides students, educators, and researchers with access to a wide range of resources, including information about agricultural universities, institutes, schemes, digital initiatives, e-learning modules and collaborative tools related to agricultural education.

ICAR has taken a holistic approach in developing educational initiatives that encompass various aspects. These initiatives include organizing hackathons through the Kritagya Hackathon Portal, establishing a network for alumni engagement known as KVC ALNET, and implementing AI-based disease identification in crops through AI-DISC. Alongside these efforts, ICAR's digital infrastructure initiatives are focused on providing stakeholders with the finest digital facilities available through effective utilization of digital resources. This approach ensures that the goal of delivering top-notch digital services to users of the portal is successfully achieved. It covers 3 Central AUs, 4 ICAR deemed universities, 63 State AUs, 4 Central University with Agriculture Faculty, 2,48,443 Unique Student ID generated.

Digital Content

ICAR has taken the lead in introducing numerous pioneering initiatives aimed at elevating agricultural education and establishing an enriching learning atmosphere for students. These ICAR initiatives, falling under the umbrella of digital content, are strategically designed to harness the power of technology, foster collaborative partnerships, and grant access to state-of-the-art resources.

Some of the key initiatives under Digital Content by ICAR include:

E-Learning Portal: The E-Learning Portal provides students and faculty with access to a diverse range of online courses aligned with the agricultural curriculum. The dynamic system of the portal ensures high quality content creation by the renowned faculty of SAUs and deemed universities. The platform is becoming popular amongst students as they are accessing content online. The institution offers 74 postgraduate courses that have been downloaded 33,923 times, and 163 undergraduate courses that have been downloaded 41,432 times.

***Agri-DIKSHA**:* The Agri-DIKSHA is an online repository of agricultural educational resources. Under Agri-Diksha, Lectures are delivered across multiple disciplines through video recordings, video repositories, interactive and personalized adaptive learning, online assessments etc. The platform showcases input from 74 contributing universities, consisting of 2,575 resources in the public library section. Additionally, a total of 7,108 videos have been created, amounting to a total duration of 5,067 hours.

E-Krishi Shiksha: The e-Krishi Shiksha portal, is a digital platform dedicated to agricultural education. It provides online courses, video lectures, e-books, and interactive learning resources to students and educators. The portal has successfully facilitated access to quality agricultural education, reaching a wide audience across the country and abroad.

Virtual Reality Experience Labs: ICAR has established Virtual Reality Experience Labs to provide immersive learning experiences. VR Labs are the integration of immersive technology with research and education. A total of 74 Virtual Reality Experience Labs have been established in the country by ICAR with about 2066 students have used it since its launch.

Introduction to Artificial Intelligence & Machine Learning in Agriculture

Dr. Sapna Nigam

Division of Computer Applications
ICAR-Indian Agricultural Statistics Research Institute, New Delhi - 110 012

sapna.nigam@icar.gov.in

1. Overview of Artificial Intelligence and Machine Learning

Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) are interconnected fields that are revolutionizing various industries, including agriculture. While they are related, each has distinct characteristics, methodologies, and applications. This section provides a detailed overview of these concepts, their interrelationships, and their significance.

1.1 Artificial Intelligence (AI)

Artificial Intelligence (AI) refers to the simulation of human intelligence by machines, particularly computer systems. It encompasses a wide range of techniques that allow computers to perform tasks that typically require human intelligence, such as visual perception, speech recognition, decision-making, and language translation. AI is a broad field of computer science focused on creating systems that can perform tasks that typically require human intelligence. These tasks include reasoning, learning, problem-solving, perception, and language understanding. AI systems aim to mimic cognitive functions such as recognizing patterns, making decisions, and understanding natural language. The concept of AI dates back to the mid-20th century, with pioneers like Alan Turing and John McCarthy. Turing's famous question, "Can machines think?" laid the foundation for AI research. Over the decades, AI has evolved from rule-based systems to more sophisticated techniques like neural networks and reinforcement learning.

Key Areas of AI

- Natural Language Processing (NLP): Enables machines to understand, interpret, and generate human language. Applications include chatbots, voice assistants, and language translation.
- Computer Vision: Allows machines to interpret and process visual information from the world, enabling applications like facial recognition, object detection, and image classification.
- Robotics: AI is integral to robotics, enabling machines to perform tasks autonomously, such as manufacturing, surgery, and agriculture.
- Expert Systems: AI systems that mimic human experts' decision-making abilities in specific domains, like medical diagnosis or financial forecasting.

Types of AI:

- **Narrow AI (Weak AI):** AI systems designed to perform a specific task, such as playing chess or recognizing faces. These systems do not possess general intelligence.

- **General AI (Strong AI):** Hypothetical AI that possesses the ability to understand, learn, and apply intelligence across a wide range of tasks, similar to human cognition.
- **Super intelligent AI:** An AI that surpasses human intelligence in all aspects. This remains speculative and is a topic of ongoing research and debate.

1.2 Machine Learning (ML)

ML is a subset of AI focused on developing algorithms that enable computers to learn from and make predictions or decisions based on data. Unlike traditional programming, where explicit instructions are provided, ML algorithms learn patterns from data to perform tasks such as classification, regression, clustering, and anomaly detection.

Types of Machine Learning:

- **Supervised Learning:** In supervised learning, the model is trained on labeled data, where the input-output pairs are provided. The goal is to learn a mapping from inputs to outputs. Common algorithms include Linear Regression, Decision Trees, and Support Vector Machines. Example: Predicting crop yield based on historical weather and soil data.
- **Unsupervised Learning:** In unsupervised learning, the model is trained on unlabeled data, and the goal is to find hidden patterns or structures in the data. Common techniques include clustering (e.g., K-means) and dimensionality reduction (e.g., PCA). Example*:* Grouping similar crops based on their growth patterns without prior labels.
- **Reinforcement Learning:** In reinforcement learning, an agent learns to make decisions by interacting with an environment and receiving feedback in the form of rewards or penalties. The agent's goal is to maximize cumulative rewards over time. Example: Training a robot to navigate a field and perform tasks like planting or weeding.

Machine Learning Model Development Process

a. *Data Collection:* Gathering relevant data for the problem at hand, such as sensor readings, images, or historical records.
b. *Data Preprocessing:* Cleaning and transforming raw data into a suitable format for modeling. This may involve handling missing values, scaling features, and encoding categorical variables.
c. *Model Training:* Using the pre-processed data to train a machine learning model. The model learns patterns in the data and optimizes its parameters to minimize prediction errors.
d. *Model Evaluation:* Assessing the model's performance on unseen data using metrics like accuracy, precision, recall, and F1-score.
e. *Model Deployment:* Integrating the trained model into a production environment where it can make predictions or decisions on new data.

1.3 Deep Learning (DL)

DL is a subset of ML that focuses on neural networks with many layers (hence the term "deep"). These deep neural networks are capable of learning complex patterns and representations from large amounts of data. DL has been particularly successful in fields like computer vision, natural language processing, and speech recognition.

1.4 Interrelationships and Synergy

- *AI as the Umbrella Term:* AI encompasses both ML and DL. AI includes any technology that enables machines to mimic human intelligence, whereas ML and DL are specific approaches to achieving AI.
- *ML as a Subset of AI:* ML is a key method used within AI to create intelligent systems. It focuses on creating models that learn from data to make decisions or predictions.
- *DL as a Subset of ML:* DL is a specialized area within ML that deals with neural networks, particularly those with many layers. While traditional ML models might require manual feature engineering, DL models automatically learn representations from raw data.

2. Importance of AI and ML in Agriculture

2.1 Challenges in Modern Agriculture

With the growing global population, there is an increasing demand for food production. However, challenges such as climate change, pest infestations, and resource constraints make it difficult to meet these demands.

- Climate Change: Adverse effects on crop production, altered pest dynamics, and soil degradation.
- Population Growth: Rising food demand and shrinking arable land due to urbanization.
- Environmental Impact: Deforestation, biodiversity loss, and water pollution from unsustainable practices.
- Economic Instability: Farmers face fluctuating market prices, high costs, and limited access to technology.
- Technological Barriers: Slow adoption of precision agriculture and the digital divide in rural areas.

2.2 Opportunities with AI and ML

- Precision Agriculture: AI-driven technologies enable precise monitoring and management of crops, ensuring optimal use of resources like water, fertilizers, and pesticides.
- Predictive Analytics: ML models can predict crop yields, disease outbreaks, and weather patterns, allowing farmers to make informed decisions and take preventive measures.
- Automation: AI-powered robots and drones can automate labor-intensive tasks such as planting, harvesting, and monitoring, reducing the reliance on manual labor.

3. Key AI and ML Techniques in Agriculture

- Computer Vision: Involves the use of image processing and ML techniques to analyze visual data from fields and crops. Applications include detecting plant diseases, estimating crop yield, and monitoring plant growth. Example: Drones equipped with cameras capture images of fields, which are then analyzed by AI models to identify areas affected by pests or diseases.

- Predictive Modeling: Uses historical data to predict future events. In agriculture, predictive models are used for yield forecasting, disease prediction, and climate impact analysis. Example: An ML model trained on weather data and soil conditions can predict the best planting times and forecast potential crop failures.
- Natural Language Processing (NLP): Enables AI systems to understand and process human language. In agriculture, NLP can be used in chatbots and virtual assistants to provide farmers with real-time advice and information. Example: A WhatsApp-based chatbot that answers farmers' queries related to crop care and pest management.
- Robotics and Automation: AI-driven robots are used for tasks such as planting, weeding, and harvesting. These robots use ML algorithms to navigate fields and perform tasks with precision. Example: Autonomous tractors equipped with AI algorithms can plow fields and plant seeds with minimal human intervention.

4. Applications of AI and ML in Agriculture

- Crop Monitoring and Management: AI systems monitor crops using sensors, drones, and satellites. They analyze data on soil health, moisture levels, and crop growth to provide recommendations on irrigation, fertilization, and pest control. Example: A smart irrigation system that adjusts water levels based on real-time soil moisture data, reducing water waste and improving crop yield.
- Soil Health Monitoring: ML models analyze soil samples to determine nutrient levels and recommend appropriate fertilizers. AI can also predict soil degradation and suggest preventive measures. Example: An AI system that analyzes soil data to recommend crop rotation strategies to maintain soil fertility.
- Disease and Pest Detection: AI-driven computer vision systems detect diseases and pests early by analyzing images of crops. This allows for timely intervention, reducing crop losses. Example: An AI model trained on images of infected plants can identify diseases and suggest treatments, helping farmers take quick action.
- Yield Prediction: ML models use historical data on weather, soil, and crop conditions to predict crop yields. This helps farmers plan their activities and resources more effectively. Example: A yield prediction model that forecasts wheat production based on historical weather patterns and soil data.
- Supply Chain Optimization: AI systems optimize the agricultural supply chain by predicting demand, managing inventory, and reducing waste. This ensures that fresh produce reaches consumers efficiently. Example: An AI-powered platform that connects farmers with buyers, optimizing pricing and reducing food wastage by predicting demand.

5. Challenges and Limitations

- Data Quality and Availability: The effectiveness of AI and ML models depends on the quality and quantity of data. In agriculture, data collection can be challenging due to the variability in environmental conditions and farming practices. Example: Inconsistent data from different regions can lead to inaccurate predictions, making it difficult to develop reliable AI models.
- High Initial Costs: Implementing AI and ML technologies in agriculture requires significant investment in hardware, software, and training. Small-scale farmers may find it difficult to afford these technologies. Example: The cost of deploying drones and AI-powered sensors may be prohibitive for smallholder farmers.

- Limited Technical Expertise: Farmers may lack the technical knowledge to implement and manage AI and ML systems. Training and support are essential to ensure the successful adoption of these technologies. Example: Farmers may need assistance in interpreting data from AI-driven crop monitoring systems.
- Ethical and Environmental Concerns: The use of AI in agriculture raises ethical questions about data privacy, the displacement of labor, and the environmental impact of technology. Example: The automation of farming tasks could lead to job losses in rural areas, raising concerns about the social impact of AI

6. Future Trends in AI and ML in Agriculture

- Integration of AI with IoT: The Internet of Things (IoT) enables the seamless integration of AI with various agricultural devices, creating smart farms where all operations are automated and optimized. Example: A farm where AI-driven systems control irrigation, fertilization, and pest management in real time based on data from IoT sensors.
- Personalized Farming Solutions: AI systems will offer personalized recommendations tailored to individual farms based on specific conditions, such as soil type, climate, and crop variety. Example: A personalized AI assistant that provides crop management advice based on the unique characteristics of a farm.
- Enhanced Decision Support Systems: AI-driven decision support systems will provide farmers with real-time insights, enabling them to make informed decisions that improve productivity and sustainability. Example: A decision support system that integrates weather forecasts, market trends, and crop data to recommend optimal planting strategies.
- Sustainable Agriculture: AI and ML will play a crucial role in promoting sustainable agricultural practices by optimizing resource use, reducing waste, and minimizing environmental impact. Example: AI-driven systems that optimize the use of water and fertilizers to reduce environmental pollution and promote sustainable farming.

7. Conclusion

AI, ML, and DL form a hierarchy of transformative technologies that are revolutionizing industries, with agriculture being a prime beneficiary. AI provides a comprehensive framework for building intelligent systems, ML offers the tools for data-driven decision-making, and DL brings advanced capabilities to handle complex and large-scale data. These technologies are driving innovation in agriculture, enabling more efficient, sustainable, and productive farming practices. From precision farming to disease detection, AI and ML are addressing some of the most pressing challenges in agriculture, enhancing productivity, reducing waste, and promoting environmental sustainability. However, the successful integration of these technologies in agriculture requires overcoming challenges such as data quality, high implementation costs, and the need for specialized technical expertise. As AI and ML continue to evolve, they will play an increasingly critical role in shaping the future of agriculture, helping to ensure food security and sustainability for future generations. By embracing these technologies, the agricultural sector can rise to meet the global challenges of feeding a growing population while safeguarding the environment.

Basics of Python

Dr. Madhu
Division of Computer Applications
ICAR-Indian Agricultural Statistics Research Institute, New Delhi – 110012
madhu1@icar.gov.in

Python is a very popular general-purpose interpreted, interactive, object-oriented, and high-level programming language. Python is dynamically-typed and garbage-collected programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).

Characteristics of Python

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.

It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python Syntax

```
print('India is my country.')
```

Variables

```
x=2
y="India"
print(x)
print(y)
```

Type Casting

```
x=str(5)
y=int(5.0)
z=float(5)
print(x)
print(y)
print(z)
```

print(type(x))
print(type(y))

Single & Double Quotes

x="india"
y='india'
print(x)
print(y)

Multiline Strings

a = """hello,
good morning,
h r u,
all."""
print(a)

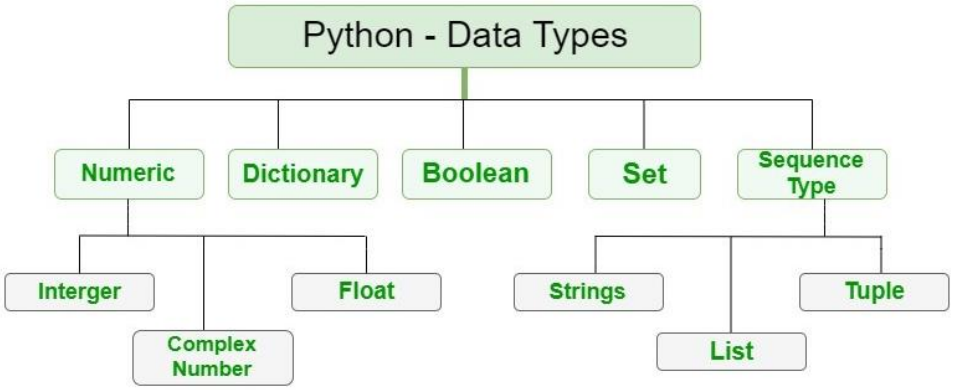
a = '''hello,
good morning,
h r u
all.'''
print(a)

Python Data Types

Data types are the classification or categorization of data items. It represents the kind of value that tells what operations can be performed on a particular data. Since everything is an object in Python programming, data types are actually classes and variables are instance (object) of these classes.

Following are the standard or built-in data type of Python:

- Numeric
- Sequence Type
- Boolean
- Set
- Dictionary



Numeric

In Python, numeric data type represent the data which has numeric value. Numeric value can be integer, floating number or even complex numbers. These values are defined as int, float and complex class in Python.

- **Integers** – This value is represented by int class. It contains positive or negative whole numbers (without fraction or decimal). In Python there is no limit to how long an integer value can be.
- **Float** – This value is represented by float class. It is a real number with floating point representation. It is specified by a decimal point. Optionally, the character e or E followed by a positive or negative integer may be appended to specify scientific notation.
- **Complex Numbers** – Complex number is represented by complex class. It is specified as *(real part) + (imaginary part)j*. For example – 2+3j

Note – type() function is used to determine the type of data type.

Float

```
x = 1.10
y = 1.0
z = -35.59

print(type(x))
print(type(y))
print(type(z))
```

int

```
x = 1
y = 3565
z = -3255522
```

print(type(x))
print(type(y))
print(type(z))

complex

x = 3+5j
y = 5j
z = -5j

print(type(x))
print(type(y))
print(type(z))

Sequence Type

In Python, sequence is the ordered collection of similar or different data types. Sequences allows to store multiple values in an organized and efficient fashion. There are several sequence types in Python –

- String
- List
- Tuple

1) String

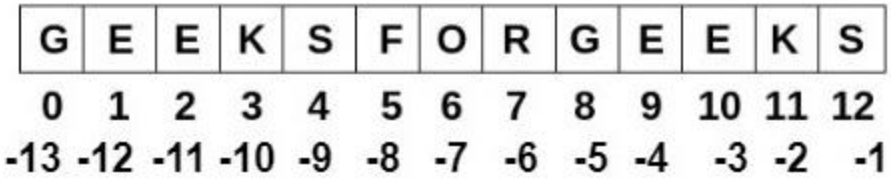
In Python, Strings are arrays of bytes representing Unicode characters. A string is a collection of one or more characters put in a single quote, double-quote or triple quote. In python there is no character data type, a character is a string of length one. It is represented by str class.

Creating String

Strings in Python can be created using single quotes or double quotes or even triple quotes.

Accessing elements of String

In Python, individual characters of a String can be accessed by using the method of Indexing. Indexing allows negative address references to access characters from the back of the String, e.g. -1 refers to the last character, -2 refers to the second last character and so on.



```
print("Hello")
print('Hello')

a = "Hello"
print(a)
```

Strings are Arrays

```
a = "Hello, World!"  # 0....n
print(a[1])
print(a[2])
print(a[7])
```

Looping Through a String

```
for x in "banana":
  print(x)
```

String Slicing

```
b = "Hello, World!"
print(b[2:5])

b = "Hello, World!"
print(b[:5])

b = "Hello, World!"
print(b[2:])

b = "Hello, World!"
print(b[-5:-2])
print(b[-1])
```

Strings Functions

```
a = "hello, World!"
print(a.upper())              #Converts a string into upper case
print(a.capitalize())         #Converts the first character to upper case
print(a.casefold())           #Converts string into lower case
```

```
print(a.split())            #Splits the string at the specified separator, and returns a list
print(a.lower())            #Converts a string into lower case
print(a.strip())            # Returns a trimmed version of the string
                           #returns "Hello, World!"
print(a.replace("h", "J"))          #Returns a string where a specified value is replaced with a specified value
print(a.isdigit())              #Returns True if all characters in the string are digits
print(a.isupper())          #Returns True if all characters in the string are upper case
print(len(a))           #len() function returns the length of a string
```

String Concatenation

```
a = "Hello"
b = "World"
c = a + b
print(c)
```

```
a = "Hello"
b = "World"
c = a + "    "    + b
print(c)
```

"""we cannot combine strings and numbers"""

```
age = 36
txt = "My name is John, I am " + age
print(txt)
```

we can combine strings and numbers by using the **format() method!**

The format() method takes the passed arguments, formats them, and places them in the string where the placeholders {}

```
age = 36
txt = "My name is John, and I am {}"
print(txt.format(age))
```

The format() method takes unlimited number of arguments, and are placed into the respective placeholders:

You can use index numbers {0} to be sure the arguments are placed in the correct placeholders.

quantity = 3
itemno = 567
price = 49.95
myorder = "I want {} pieces of item {} for {} dollars."
print(myorder.format(quantity, itemno, price))

quantity = 3
itemno = 567
price = 49.95
myorder = "I want to pay {2} dollars for {0} pieces of item {1}."
print(myorder.format(quantity, itemno, price))

2) List

Lists are just like the arrays, declared in other languages which is a ordered collection of data. It is very flexible as the items in a list do not need to be of the same type.

Creating List
Lists in Python can be created by just placing the sequence inside the square brackets[].

Accessing elements of List
In order to access the list items refer to the index number. Use the index operator [ ] to access an item in a list. In Python, negative sequence indexes represent positions from the end of the array. Instead of having to compute the offset as in List[len(List)-3], it is enough to just write List[-3]. Negative indexing means beginning from the end, -1 refers to the last item, -2 refers to the second-last item, etc
Lists are used to store multiple items in a single variable.

List items are ordered, changeable, and allow duplicate values. Lists are created using **square brackets.**

Ordered
When we say that lists are ordered, it means that the items have a defined order, and that order will not change.
If you add new items to a list, the new items will be placed at the end of the list.

Changeable
The list is changeable, meaning that we can change, add, and remove items in a list after it has been created.

Allow Duplicates
Since lists are indexed, lists can have items with the same value

thislist = ["apple", "banana", "cherry"]

```
print(thislist)
print(type(thislist))

thislist = ["apple", "banana", "cherry", "apple", "cherry"]
print(thislist)                                  # Allow Duplicates
```

Length of a List

```
thislist = ["apple", "banana", "cherry"]
print(len(thislist))                  #len function for finding the number of values in a tuple

list1 = ["apple", "banana", "cherry"]            #List items can be of any data type
list2 = [1, 5, 7, 9, 3]
list3 = [True, False, False]
print(type(list2))
print(type(list1))

list1 = ["abc", 34, True, 40, "male"]            #A list can contain different data types
print(type(list1))
```

Indexes of List items

```
thislist = ["apple", "banana", "cherry"]
print(thislist[1])

thislist = ["apple", "banana", "cherry"]
print(thislist[-1])
print(thislist[-2])
print(thislist[-3])
print(thislist[0])

thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[2:5])

thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[:4])

thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[-4:-1])
```

```
thislist = ["apple", "banana", "cherry"]
if "apple" in thislist:       # true
  print("Yes, 'apple' is in the fruits list")

thislist = ["apple", "banana", "cherry"]
if "orange" in thislist:       #false
  print("no, 'orange' is not in the fruits list")    #not executed
```

Change Item Value

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "mango"]
thislist[1:3] = ["blackcurrant", "watermelon"]                    #Change the values "banana" and "cherry" with the values "blackcurrant" and "watermelon
print(thislist)

thislist = ["apple", "banana", "cherry"]
thislist[1:3] = ["watermelon"]
print(thislist)
```

Insert Items

To insert a new list item, without replacing any of the existing values, we can use the **insert() method**.

```
thislist = ["apple", "banana", "cherry"]
thislist.insert(2, "watermelon")          # insert() method inserts an item at the specified index

thislist = ["apple", "banana", "cherry"]
thislist.append("orange")                                  #append() method add an item to the end of the list
print(thislist)

thislist = ["apple", "banana", "cherry"]
tropical = ["mango", "pineapple", "papaya"]

print(thislist+tropical)

thislist = ["apple", "banana", "cherry"]
thistuple = ("kiwi", "orange")
thislist.extend(thistuple)      #extend() method does not have to append lists only, you can add any iterable object (tuples, sets, dictionaries etc.).
```

```
print(thislist)
print(type(thistuple))

thislist = ["apple", "banana", "cherry"]
thislist.remove("banana")              #remove() method removes the specified item
print(thislist)

thislist = ["apple", "banana", "cherry"]
thislist.pop(-1)                       #pop() method removes the specified index
print(thislist)

thislist = ["apple", "banana", "cherry"]
thislist.pop()         #do not specify the index, the pop() method removes the last item
print(thislist)

thislist = ["apple", "banana", "cherry"]
del thislist[0]                        #del keyword also removes the specified index
print(thislist)

thislist = ["apple", "banana", "cherry"]
del thislist
print(thislist)                              #Delete the entire list

thislist = ["apple", "banana", "cherry"]
thislist.clear()                         #list still remains, but it has no content
print(thislist)

thislist = ["apple", "banana", "cherry"]
for x in thislist:
  print(x)

thislist = ["apple", "banana", "cherry"]
for i in range(len(thislist)):
  print(thislist[i])

thislist = ["apple", "banana", "cherry"]
i = 0
while i < len(thislist):         #until true
  print(thislist[i])
  i = i + 1
```

Sorting List

```
thislist = ["orange", "mango", "apricot", "apple", "banana"]
thislist.sort()            #sort() method that will sort the list alphanumerically, ascending, by default
print(thislist)

thislist = [100, 50, 50, 82, 23]
thislist.sort()
print(thislist)

thislist = [100, 50, 65, 82, 23]
thislist.sort(reverse = False)        #To sort descending, use the keyword argument reverse = True
print(thislist)
```

Copy a List

```
thislist = ["apple", "banana", "cherry"]
mylist = thislist.copy()      #to make a copy, one way is to use the built-in List method copy()
print(mylist)

thislist = ["apple", "banana", "cherry"]
mylist = list(thislist)     #make a copy is to use the built-in method list()
print(mylist)
```

Join Two Lists

```
list1 = ["a", "b", "c"]
list2 = [1, 2, 3]

list3 = list1 + list2       #by using the + operator
print(list3)

list1 = ["a", "b" , "c"]
list2 = [1, 2, 3]

for x in list2:
  list1.append(x)          #Another way to join two lists is by appending all the items from list2 into list1, one by one
```

print(list1)

3) Tuple

Just like list, tuple is also an ordered collection of Python objects. The only difference between tuple and list is that tuples are immutable i.e. tuples cannot be modified after it is created. It is represented by tuple class.

Creating Tuple

In Python, tuples are created by placing a sequence of values separated by 'comma' with or without the use of parentheses for grouping of the data sequence. Tuples can contain any number of elements and of any datatype (like strings, integers, list, etc.).

Tuples are used to store multiple items in a single variable.

A tuple is a collection which is ordered and unchangeable.

Tuples are written with **round brackets. ()**

Ordered

When we say that tuples are ordered, it means that the items have a defined order, and that order will not change.

Unchangeable

Tuples are unchangeable, meaning that we cannot change, add or remove items after the tuple has been created.

Allow Duplicates

Since tuples are indexed, they can have items with the same value:

```
thistuple = ("apple", "banana", "cherry")
print(thistuple)

thistuple = ("apple", "banana", "cherry", "apple", "cherry")
print(thistuple)                              #allow duplicates

thistuple = ("apple", "banana", "cherry")
print(len(thistuple))          #len function for finding the number of values in a tuple

thistuple = ("apple",)         #use comma (,) if tuple having single value otherwise it is considered as string
print(type(thistuple))
print(len(thistuple))
```

```
thistuple = ("apple")          #NOT a tuple
print(type(thistuple))

tuple1 = ("apple", "banana", "cherry")
tuple2 = (1, 5, 7, 9, 3)
print(type(tuple1))
print(type(tuple2))

tuple1 = ("abc", 34, True, 40, "male")
print(type(tuple1))

thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")
print(thistuple[2:5])          #access the elements of a tuple
```

Change Tuple Values

Once a tuple is created, you cannot change its values. Tuples are unchangeable, or immutable.

You can convert the tuple into a list, change the list, and convert the list back into a tuple

```
x = ("apple", "banana", "cherry")   #can not change the tuple
y = list(x)              #convert tuple into list
y[1] = "kiwi"              # change list
x = tuple(y)             #convert list into tuple

print(x)

thistuple = ("apple", "banana", "cherry")              #Since tuples are immutable, they do not have a build-in append()
method
y = list(thistuple)
y.append("orange")                                  # list have append method
thistuple = tuple(y)
print(thistuple)

thistuple = ("apple", "banana", "cherry")
y = ("orange","mango")
thistuple += y          # allowed to add tuples to tuples

print(thistuple)
```

```
thistuple = ("apple", "banana", "cherry")        #  they do not have a build-in remove() method
y = list(thistuple)
y.remove("apple")                          # list have remove method
thistuple = tuple(y)
print(thistuple)

thistuple = ("apple", "banana", "cherry")
del thistuple
print(thistuple) #this will raise an error because the tuple no longer exists
```

Packing & Unpacking Tuples

When we create a tuple, we normally assign values to it. This is called **"packing"** a tuple.

In Python, we are also allowed to extract the values back into variables. This is called **"unpacking".**

```
a = ("Delhi", 5000, "Agriculture")   #PACKS values into variable a

(City, student, type_ofcollege) = a     #UNPACKS values of variable a

print(City,student,type_ofcollege)
```

If the number of variables is less than the number of values, you can add an * to the variable name and the values will be assigned to the variable as a list.

```
fruits = ("apple", "banana", "cherry", "strawberry", "raspberry")

(green, yellow, *red) = fruits

print(green)
print(yellow)
print(red)
```

If the asterisk is added to another variable name than the last, Python will assign values to the variable until the number of values left matches the number of variables left.

```
fruits = ("apple", "mango", "papaya", "pineapple", "cherry")

(green, *tropic, red) = fruits
```

```
print(green)
print(tropic)
print(red)
```

Join Tuples

```
tuple1 = ("a", "b" , "c")
tuple2 = (1, 2, 3)

tuple3 = tuple1 + tuple2
print(tuple3)

fruits = ("apple", "banana", "cherry")
mytuple = fruits * 2

print(mytuple)
```

Boolean

Data type with one of the two built-in values, True or False. Boolean objects that are equal to True are truthy (true), and those equal to False are falsy (false). But non-Boolean objects can be evaluated in Boolean context as well and determined to be true or false. It is denoted by the class bool.
Note – True and False with capital 'T' and 'F' are valid booleans otherwise python will throw an error.

Booleans represent one of two values: True or False.

```
print(10 > 9)
print(10 == 9)
print(10 < 9)

print(bool("Hello"))        #Almost any value is evaluated to True if it has some sort of content.
print(bool(15))          #Any string is True, except empty strings.
print(bool(0))          # Any number is True, except 0.
print(bool(""))
print(bool(()))
print(bool([]))
print(bool({}))           # Any list, tuple, set, and dictionary are True, except empty ones.
print(bool(False))
print(bool(None))
```

Set

In Python, Set is an unordered collection of data type that is iterable, mutable and has no duplicate elements. The order of elements in a set is undefined though it may consist of various elements.

Creating Sets

Sets can be created by using the built-in set() function with an iterable object or a sequence by placing the sequence inside curly braces, separated by 'comma'. Type of elements in a set need not be the same, various mixed-up data type values can also be passed to the set.

Accessing elements of Sets

Set items cannot be accessed by referring to an index, since sets are unordered the items has no index. But you can loop through the set items using a for loop, or ask if a specified value is present in a set, by using the in keyword.

Sets are used to store multiple items in a single variable.

Set Items

Set items are unordered, unchangeable, and do not allow duplicate values.

Unordered

Unordered means that the items in a set do not have a defined order.

Set items can appear in a different order every time you use them, and cannot be referred to by index or key.

Unchangeable

Set items are unchangeable, meaning that we cannot change the items after the set has been created.

Duplicates Not Allowed

Sets cannot have two items with the same value.

Sets are written with **curly brackets.**

```
thisset = {"apple", "banana", "cherry"}
print(thisset)

thisset = {"apple", "banana", "cherry", "apple"}
```

```
print(thisset)

thisset = {"apple", "banana", "cherry"}
print(len(thisset))                # len() function used to  find length

set1 = {"abc", 34, True, 40, "male"}
print(set1)              # A set with strings, integers and boolean values
print(type(set1))          # type function return data type

thisset = {"apple", "banana", "cherry"}
for x in thisset:
  print(x)

thisset = {"apple", "banana", "cherry"}
print("banana" in thisset)
print("kiwi" in thisset)
```

Add an Item

```
thisset = {"apple", "banana", "cherry"}
thisset.add("orange")           #  to add one item to a set use the add() method.
print(thisset)

thisset = {"apple", "banana", "cherry"}
tropical = {"pineapple", "mango", "papaya"}
thisset.update(tropical)                    #To add items from another set into the current set, use the update() method
print(thisset)

thisset = {"apple", "banana", "cherry"}
mylist = ["kiwi", "orange"]
thisset.update(mylist)             #update() method does not have to be a set only, it can be any iterable object (tuples,
lists, dictionaries etc.)
print(thisset)
```

Remove Item

```
thisset = {"apple", "banana", "cherry"}
thisset.remove("apple")
thisset.remove("kiwi")           #To remove an item in a set, use the remove(), or the discard() method.
print(thisset)                  #If the item to remove does not exist, remove() will raise an error
```

```
thisset = {"apple", "banana", "cherry"}
thisset.discard("kiwi")              #If the item to remove does not exist, discard() will NOT raise an error.
print(thisset)

thisset = {"apple", "banana", "cherry"}
x = thisset.pop()                #   pop() method to remove an item,but this method will remove the last item
print(x)                        #Remember that sets are unordered, so you will not know what item that gets removed.The
return value of the pop() method is the removed item.
print(thisset)

thisset = {"apple", "banana", "cherry"}
thisset.clear()                  #   clear() method empties the set
print(thisset)

thisset = {"apple", "banana", "cherry"}
del thisset          #  del keyword will delete the set completely
print(thisset)
```

Join Two Sets

```
set1 = {"a", "b" , "c"}
set2 = {1, 2, 3}

set3 = set1.union(set2)       # union() method returns a new set with all items from both sets
print(set3)

set1 = {"a", "b" ,"b" ,"c"}
set2 = {1, 2, 3}

set1.update(set2)          #update() method inserts the items of set2 into set1.
                        #Both union() and update() will exclude any duplicate items
print(set1)
```

Both union() and update() will exclude any duplicate items.

```
x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}

x.intersection_update(y)         #intersection_update() method will keep only the items that are present in both sets.
```

```
print(x)

x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}

z = x.intersection(y)     #intersection() method will return a new set, that only contains
                          # the items that are present in both sets.

print(z)

x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}

x.symmetric_difference_update(y)      #symmetric_difference_update() method will keep only the elements that are
                                      #NOT present in both sets.

print(x)

x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}

z = x.symmetric_difference(y)          # symmetric_difference() method will return a new set,
                                       # that contains only the elements that are NOT present in both sets.

print(z)
```

Dictionary

Dictionary in Python is an unordered collection of data values, used to store data values like a map, which unlike other Data Types that hold only single value as an element, Dictionary holds key:value pair. Key-value is provided in the dictionary to make it more optimized. Each key-value pair in a Dictionary is separated by a colon :, whereas each key is separated by a 'comma'.

Creating Dictionary

In Python, a Dictionary can be created by placing a sequence of elements within curly {} braces, separated by 'comma'. Values in a dictionary can be of any datatype and can be duplicated, whereas keys can't be repeated and must be immutable. Dictionary can also be created by the built-in function dict(). An empty dictionary can be created by just placing it to curly braces{}.

Note – Dictionary keys are case sensitive, same name but different cases of Key will be treated distinctly.

Accessing elements of Dictionary

In order to access the items of a dictionary refer to its key name. Key can be used inside square brackets. There is also a method called get() that will also help in accessing the element from a dictionary.

Dictionaries are used to store data values in **key:value pairs**.

A dictionary is a collection which is ordered, changeable and do not allow duplicates.

Ordered

When we say that dictionaries are ordered, it means that the items have a defined order, and that order will not change.

Changeable

Dictionaries are changeable, meaning that we can change, add or remove items after the dictionary has been created.

Duplicates Not Allowed

Dictionaries cannot have two items with the same key.

```
thisdict = {
  "brand": "Ford",

  "model": "Mustang",
  "year": 1964
}
print(thisdict)
```

Dictionary items are presented in key:value pairs, and can be referred to by using the key name.

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
print(thisdict["brand"])
print(len(thisdict))          # len () function returns length
print(type(thisdict))         # type() function show datatype

thisdict = {
```

```
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964,
  "year": 2020          #Dictionaries cannot have two items with the same key.
                        # Duplicate values will overwrite existing values
}
print(thisdict)
```

Accessing Items

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
x = thisdict["model"]
y = thisdict.get("model")      #Get the value of the "model" key
z = thisdict.keys()        #keys() method will return a list of all the keys in the dictionary
b = thisdict.values()       #values() method will return a list of all the values in the dictionary
print(x)
print(y)
print(z)
print(b)
```

Change Dictionary Items

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict["year"] = 2018
print(thisdict)

thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict.update({"year": 2020})   #update() method will update the dictionary with the items from the given argument.
```

```
print(thisdict)
```

Adding Items

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict["color"] = "red"
print(thisdict)

thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict.update({"color": "red"})        #update() method will update the dictionary with the items from a given
argument
print(thisdict)
```

Removing Items

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict.pop("model")     #pop() method removes the item with the specified key name
print(thisdict)

thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict.popitem()        #popitem() method removes the last inserted item
print(thisdict)
```

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
del thisdict["model"]       #del keyword removes the item with the specified key name
print(thisdict)

thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
del thisdict            #del keyword can also delete the dictionary completely
print(thisdict)

thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict.clear()        #clear() method empties the dictionary
print(thisdict)
```

Loop Through a Dictionary

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
for x in thisdict:       #Print all key names in the dictionary, one by one
  print(x)
for y in thisdict:
  print(thisdict[y])     #Print all values in the dictionary, one by one
for x in thisdict.values(): #values() method to return values of a dictionary
  print(x)
for x in thisdict.keys():
  print(x)                 #keys() method to return the keys of a dictionary
for x, y in thisdict.items(): #items methods to return values for both key and values
```

```
print(x, y)
```

Copy a Dictionary

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
mydict = thisdict.copy()         # Make a copy of a dictionary with the copy() method
print(mydict)
print(thisdict)
mydict = dict(thisdict)     # to make a copy is to use the built-in function dict()
print(mydict)
```

Nested Dictionaries

A dictionary can contain dictionaries, this is called nested dictionaries.

```
myfamily = {
  "child1" : {
    "name" : "Emil",
    "year" : 2004
  },
  "child2" : {
    "name" : "Tobias",
    "year" : 2007
  },
  "child3" : {
    "name" : "Linus",
    "year" : 2011
  }
}
print(myfamily)

child1 = {
  "name" : "Emil",
  "year" : 2004
}
child2 = {
```

```
  "name" : "Tobias",
  "year" : 2007
}
child3 = {
  "name" : "Linus",
  "year" : 2011
}


myfamily = {
  "child1" : child1,
  "child2" : child2,
  "child3" : child3
}
print(myfamily)
```

Python Operators

Operators are special symbols in Python that carry out arithmetic or logical computation. The value that the operator operates on is called the operand.

1) Arithmetic operators

Arithmetic operators are used to perform mathematical operations like addition, subtraction, multiplication, etc.

+ Add two operands or unary plus x + y+ 2:

- Subtract right operand from the left or unary minus x - y- 2

* Multiply two operands x * y

/ Divide left operand by the right one (always results into float) x / y

% Modulus - remainder of the division of left operand by the right x % y (remainder of x/y)

// Floor division - division that results into whole number adjusted to the left in the number line x // y

** Exponent - left operand raised to the power of right x**y (x to the power y)
"""

```
x = 3
y = 2
print('x + y =',x+y)
print('x - y =',x-y)
print('x * y =',x*y)
print('x / y =',x/y)
print('x // y =',x//y)
print('x ** y =',x**y)
```

2) Comparison operators

Comparison operators are used to compare values. It returns either True or False according to the condition.

> Greater than - True if left operand is greater than the right x > y

< Less than - True if left operand is less than the right x < y

== Equal to - True if both operands are equal x == y

!= Not equal to - True if operands are not equal x != y

>= Greater than or equal to - True if left operand is greater than or equal to the right x >= y

<= Less than or equal to - True if left operand is less than or equal to the right x <= y

"""

```
x = 5
y = 10

print('x > y is',x>y)
print('x < y is',x<y)
print('x == y is',x==y)
print('x != y is',x!=y)
print('x >= y is',x>=y)
print('x <= y is',x<=y)
```

3) Logical operators

Logical operators are the and, or, not operators.

and True if both the operands are true x and y

or True if either of the operands is true x or y

not True if operand is false (complements the operand) not x
"""

x = True
y = False

print('x and y is',x and y) #true true

print('x or y is',x or y) #either true

print('not x is',not x)

4) Assignment operators

Assignment operators are used in Python to assign values to variables.

a = 5 is a simple assignment operator that assigns the value 5 on the right to the variable a on the left.

= x = 5 x = 5

+= x += 5 x = x + 5

-= x -= 5 x = x - 5

*= x *= 5 x = x * 5

/= x /= 5 x = x / 5

a = 21
b = 10
c = 0

c = a + b
print ("Value of c is ", c)

```
c += a
print ("Value of c is ", c)

c *= a
print ("Value of c is ", c)

c /= a
print ("Value of c is ", c)

c  = 2
c %= a
print ("Value of c is ", c)

c **= a
print ("Value of c is ", c)

c //= a
print ("Value of c is ", c)
```

5) Bitwise Operators

Bitwise operators are used to compare (binary) numbers:

& AND Sets each bit to 1 if both bits are 1

| OR Sets each bit to 1 if one of two bits is 1

^ XOR Sets each bit to 1 if only one of two bits is 1

~ NOT Inverts all the bits

```
a = 10          #1010    0101
b = 4           #0100

# Print bitwise AND operation
print("a & b =", a & b)         #0000
```

```
# Print bitwise OR operation
print("a | b =", a | b)      #1110

# Print bitwise NOT operation
print("~a =", ~a)            #  ~a = ~1010
                             #    = -(1010 + 1)
                             #    = -(1011)
                             #     = -11 (Decimal)

# print bitwise XOR operation
print("a ^ b =", a ^ b)       # Returns 1 if one of the bits is 1 and the other is 0 else returns false.
```

6) Shift Operators

These operators are used to shift the bits of a number left or right thereby multiplying or dividing the number by two respectively.

Bitwise right shift: Shifts the bits of the number to the right and fills 0 on voids left(fills 1 in the case of a negative number) as a result.

Bitwise left shift: Shifts the bits of the number to the left and fills 0 on voids right as a result.

```
a = 10
b = -10

# print bitwise right shift operator
print("a >> 1 =", a >> 1)
print("b >> 1 =", b >> 1)

a = 5
b = -10

# print bitwise left shift operator
print("a << 1 =", a << 1)
print("b << 1 =", b << 1)
```

7) Identity operators

is and is not are the identity operators in Python. They are used to check if two values (or variables) are located on the same part of the memory.

```
x1 = 5
y1 = 5
x2 = 'Hello'
y2 = 'Hello'
x3 = [1,2,3]
y3 = [1,2,3]
print(x1 is not y1)     # Output: False

print(x2 is y2)         # Output: True

print(x3 is y3)       # Output: False
```

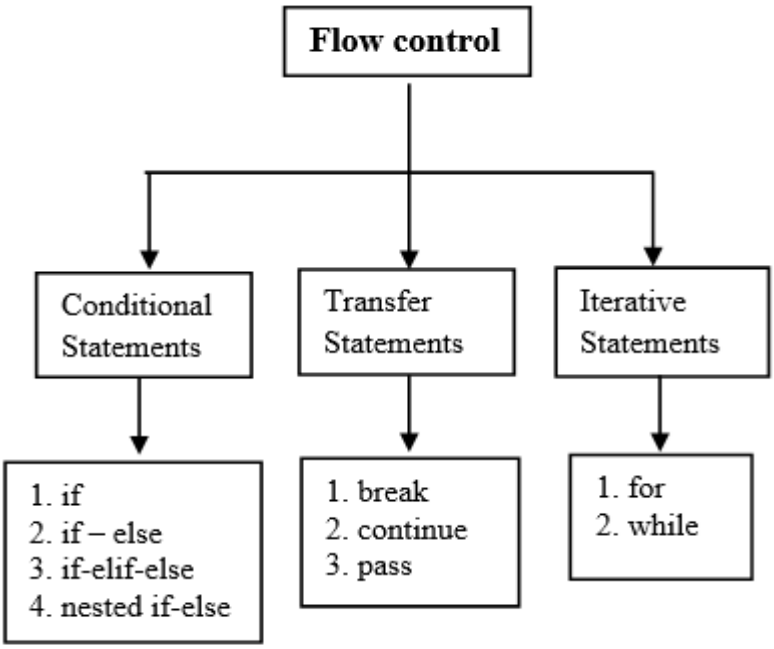
8) Membership operators

in and not in are the membership operators in Python. They are used to test whether a value or variable is found in a sequence (string, list, tuple, set and dictionary).

```
x = 'Hello world'
y = {1:'a',2:'b'}

print('H' in x)         # Output: True

print('hello' not in x)   # Output: True

print(1 in y)           # Output: True

print('a' in y)         # Output: False
```

Control Flow Statements

The flow control statements are divided into three categories
1. Conditional statements
2. Iterative statements.
3. Transfer statements



Python If ... Else

```
a = 33
b = 200
if b > a:
  print("b is greater than a")
```

```
a = 33
b = 200
if b > a:
  print("b is greater than a")
```

Elif

```
a = 33
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
```

Else

```
a = 200
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
else:
  print("a is greater than b")
```

Short Hand If

```
if a > b: print("a is greater than b")
```

Short Hand If ... Else

```
a = 2
b = 330
print("A") if a > b else print("B")
```

Nested If

```
x = 41

if x > 10:
  print("Above ten,")
  if x > 20:
    print("and also above 20!")
  else:
    print("but not above 20.")
```

The pass Statement

```
a = 33
b = 200

if b > a:
 pass
```

while Loop

```
i = 1
while i < 6:
  print(i)
  i += 1
```

break Statement

```
i = 1
while i < 6:
  print(i)
  if i == 3:
    break
  i += 1
```

continue Statement

```
i = 0     #
while i < 6:
  i += 1
  if i == 3:
    continue
  print(i)
```

For Loop

A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
```

Looping Through a String

```
for x in "banana":
  print(x)
```

range() Function

To loop through a set of code a specified number of times, we can use the range() function,

The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.

```
for x in range(6):
  print(x)

for x in range(2, 6):
  print(x)              # not including 6

for x in range(2, 30, 3):           #third parameter is the increment value
  print(x)

for x in range(6):
  print(x)
else:                 #else keyword in a for loop specifies a block of code to be executed when the loop is finished
  print("Finally finished!")

for x in range(6):
  if x == 3: break                #else block will NOT be executed if the loop is stopped by a break statement
  print(x)
else:
  print("Finally finished!")
```

Nested Loops

```
adj = ["red", "big", "tasty"]
fruits = ["apple", "banana", "cherry"]

for x in adj:
  for y in fruits:
    print(x, y)
```

Functions

You use functions in programming to bundle a set of instructions that you want to use repeatedly. That means that a function is a piece of code written to carry out a specified task.

There are three types of functions in Python:

• **User-Defined Functions (UDFs),** which are functions that users create to help them out.

• **Anonymous functions,** which are also called **lambda functions** because they are not declared with the standard def keyword.

• **Built-in functions,** such as help() to ask for help, min() to get the minimum value, print() to print an object to the terminal.

Creating a Function

```
def my_function():
  print("Hello from a function")

"""**Calling** **a** **Function** """

def my_function():
  print("Hello from a function")

my_function()

def my_function(fname):                #A parameter is the variable listed inside the parentheses in the function definition.
  print(fname + " Refsnes")

my_function("Emil")                    #An argument is the value that is sent to the function when it is called.
my_function("Tobias")
my_function("Linus")
```

Number of Arguments

```
def my_function(fname, lname):
  print(fname + " " + lname)

my_function("Emil", "Refsnes")

def my_function(*kids):                #do not know how many arguments that will be passed into your function, add a * before the parameter name in the function definition
  print("The youngest child is " + kids[2])
```

```
my_function("Emil", "Tobias", "Linus")

def my_function(child3, child2, child1):
  print("The youngest child is " + child3)

my_function(child1 = "Emil", child2 = "Tobias", child3 = "Linus")          #send arguments with the key = value syntax

def my_function(**kid):                    #number of keyword arguments is unknown, add a double ** before the parameter name
  print("His last name is " + kid["lname"])

my_function(fname = "Tobias", lname = "Refsnes")
```

Default Parameter Value

```
def my_function(country = "Norway"):
  print("I am from " + country)

my_function("Sweden")
my_function("India")
my_function()                #If we call the function without argument, it uses the default value
my_function("Brazil")
```

Passing a List as an Argument

```
def my_function(food):
  for x in food:
    print(x)

fruits = ["apple", "banana", "cherry"]

my_function(fruits)
```

Return Values

```
def my_function(x):
  return 5 * x
```

```
print(my_function(3))
print(my_function(5))
print(my_function(9))
```

Python Lambda

A lambda function is a small anonymous function.

A lambda function can take any number of arguments, but can only have one expression.

```
x = lambda a : a + 10
print(x(5))
Max = lambda a, b : a if(a > b) else b          # Example of lambda function using if-else
print(Max(1, 2))
```

Difference Between Lambda functions and def defined function

```
def cube(y):
    return y*y*y

lambda_cube = lambda y: y*y*y
print(cube(5))
print(lambda_cube(5))
```

Python Built-In Functions

all()

The python all() function accepts an iterable object (such as list, dictionary, etc.). It returns true if all items in passed iterable are true. Otherwise, it returns False. If the iterable object is empty, the all() function returns True.

```
k = [1, 3, 4, 6]          # all values true
  print(all(k))

  k = [0, False]            # all values false
  print(all(k))

  k = [1, 3, 7, 0]          # one false value
  print(all(k))
```

```
  k = [0, False, 5]        ## one true value
  print(all(k))

  k = []                   # empty iterable
print(all(k))

test1 = []
  print(test1,'is',bool(test1))
  test1 = [0]
  print(test1,'is',bool(test1))
  test1 = 0.0
  print(test1,'is',bool(test1))
  test1 = None
  print(test1,'is',bool(test1))
  test1 = True
  print(test1,'is',bool(test1))
  test1 = 'Easy string'
print(test1,'is',bool(test1))

x = 10
print('Absolute value of -40 is:', abs(x))    #abs() function is used to return the absolute value of a number
floating = -20.83
print('Absolute value of -20.83 is:', abs(floating))
y =  bin(x)  #bin() function is used to return the binary representation of a specified integer.
print (y)
s = sum([1, 2,4 ])              #sum() function is used to get the sum of numbers of an iterable, i.e., list.
print(s)
print(float(9))              # float() function change into float number
print(complex(9))              # complex() function change into complex number
```

Python Libraries

Dr. Sanchita Naha

Division of Computer Applications
ICAR-Indian Agricultural Statistics Research Institute, New Delhi - 110 012

sanchita.naha@icar.gov.in

NumPy, stands for Numerical Python, is a library designed for managing multidimensional arrays and includes a range of functions for array operations. NumPy arrays offer a performance boost, being up to 50 times faster than standard Python lists. This library supports a variety of mathematical and logical operations on arrays. This tutorial compiles different code snippets used to declare and manipulate multidimensional arrays using NumPy. To start using the NumPy library on your local machine, include the following code in your Python editor.

pip install numpy as np

After installation, run the following code to print the version of Numpy library installed in your system:

```
import numpy as np
print(np.__version__)
```

Array is a collection of homogeneous data elements. It is a linear data structure where elements are arranged sequentially. Use the following code to declare an array using the Numpy library:

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5])
print(arr)
print(type(arr))
```

Print the Datatype of an array:

```
print(arr.dtype)
```

Create arrays of float and string type:

```
arr_float = np.array([10.2,23.0,68.5,98.7,5.0])
print(arr_float)
print(arr_float.dtype)
```

 # string type array

```
arr_string = np.array(['a','b','c','d','e'])
print(arr_string)
```

arr_string.dtype

Declare array of zeroes and ones:

arr = np.zeros(5)
arr = np.zeros([2,3])

arr = np.ones(5)
print(arr)

Dimensions in Array:

One-dimensional Array (1-D Array): elements are stored one after the other as in a Row. These are the most common and basic arrays.

import numpy as np
arr = np.array([1, 2, 3, 4, 5])
print(arr)

Two-Dimensional Array (2-D Array): an array of arrays or as a matrix consisting of rows and columns is called a 2-D array. These are often used to represent matrix or 2nd order tensors.

arr = np.array([[1, 2, 3], [4, 5, 6]])
print(arr)

Three-Dimensional (3-D Array): An array that has 2-D arrays (matrices) as its elements is called 3-D array. These are often used to represent a 3rd order tensor.

arr = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])
print(arr)

NumPy Arrays provides the *ndim* attribute that returns an integer that tells us how many dimensions the array have.

arr_0D = np.array(42)
arr_1D = np.array([1, 2, 3, 4, 5])
arr_2D = np.array([[1, 2, 3], [4, 5, 6]])
arr_3D = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])

print('Dimension of arr_0D is: ',arr_0D.ndim)

print('Dimension of arr_1D is: ',arr_1D.ndim)
print('Dimension of arr_2D is: ',arr_2D.ndim)
print('Dimension of arr_3D is: ',arr_3D.ndim)

Higher Dimensional Arrays: An array can have multiple dimensions. When creating an array, you can specify the number of dimensions using the `*ndmin*` parameter.

arr = np.array([1, 2, 3, 4], ndmin=5)
print(arr)
print('number of dimensions :', arr.ndim)

Indexing of an array: Indexing refers to accessing individual array elements according to their positions. To retrieve an element, you refer to its index number. In NumPy arrays, indexing begins at 0, so the first element is accessed with index 0, the second with index 1, and so on.

arr = np.array([1, 2, 3, 4])
print(arr[0])

Get third and fourth elements from the following array and add them
arr = np.array([1, 2, 3, 4])
print(arr[2] + arr[3])

Access 2-D Arrays: To access elements from 2-D arrays we can use comma separated integers representing the dimension and the index of the element. 2-D arrays are like a matrix with rows and columns, where the first index position refers to the row and the comma separated second index represents the column.

 # Access the element on the first row, second column:
arr = np.array([[1,2,3,4,5], [6,7,8,9,10]])
print('2nd element on 1st row: ', arr[0, 1])

#Access the element on the 2nd row, 5th column:
print('5th element on 2nd row: ', arr[1, 4])

Access 3-D Arrays: To access elements in 3-D arrays, you use comma-separated integers. The first index specifies the block, while the second and third indices function similarly to those in a 2-D array.

arr_3d = np.array([[[1, 2, 3], [4, 5, 6]], [[7, 8, 9], [10, 11, 12]]])
print(arr_3d.ndim)
print(arr_3d[0, 1, 2])

 # Printing size (total number of elements) of array
print("Size of array: ", arr.size)
#size of an array refers to the total number of elements in an array

The first index denotes the initial dimension, which includes two arrays: ` [[1, 2, 3], [4, 5, 6]] ` and ` [[7, 8, 9], [10, 11, 12]] `. By selecting 0, we are left with the first array: ` [[1, 2, 3], [4, 5, 6]] `. The second index pertains to the second dimension, which also contains two arrays: ` [1, 2, 3] ` and ` [4, 5, 6] `. By selecting 1, we obtain the second array: ` [4, 5, 6] `. The third index corresponds to the third dimension, which includes three values: 4, 5, and 6. By selecting 2, we are left with the third value: 6.

Negative Indexing: Negative indexing is used to access an array from right to left.

#negative indexing in 1D array
import numpy as np
array_1d=np.array([1,2,3,4,5,6,7,8,9,10])
print(array_1d)
print(array_1d[-1], array_1d[-2], array_1d[-3], array_1d[-4], array_1d[-5], array_1d[-6], array_1d[-7], array_1d[-8], array_1d[-9], array_1d[-10])

 # negative indexing in 2D array
arr = np.array([[1,2,3,4,5], [6,7,8,9,10]])
print('Last element from 2nd dim: ', arr[1, -1])

Array Slicing: In Python, NumPy array slicing refers to extracting elements from a specified start index to an end index. The slicing syntax is ` [start : end] `, where you can also include a step parameter as ` [start : end : step] `. If the start index is omitted, it defaults to 0. If the end index is omitted, it defaults to the length of the array in that dimension. If the step is not provided, it defaults to 1. Note that the slicing operation includes the start index but excludes the end index.
arr = np.array([1, 2, 3, 4, 5, 6, 7])
print(arr[1:5])
 # this will print elements at positions 1,2,3,4

Negative Slicing: Use the minus operator to refer to an index from the end.

import numpy as np
arr = np.array([1, 2, 3, 4, 5, 6, 7])
print(arr[-3:-1])

STEP: use the step value to determine the step of the slicing

arr = np.array([1, 2, 3, 4, 5, 6, 7])

print(arr[1:5:2])
#Return every other element from the entire array:
arr = np.array([1, 2, 3, 4, 5, 6, 7])
print(arr[: : 2])

Slicing 2-D Arrays: from the second element, slice elements from index 1 to index 3

arr = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])
print(arr[1, 1:4])

Creating Arrays with a user defined Data Type: We use the array() function to create arrays, this function can take an optional argument: dtype that allows us to define the expected data type of the array elements:
arr = np.array([1, 2, 3, 4], dtype='S')
print(arr)
print(arr.dtype)

arr = np.array([1, 2, 3, 4], dtype='i4')
print(arr)
print(arr.dtype)

Converting Data Type of Existing Arrays: The best way to change the data type of an existing array is to make a copy of the array with the astype() method. The astype() function creates a copy of the array and allows you to specify the data type as a parameter. The data type can be specified using a string, like 'f' for float, 'i' for integer etc. or you can use the data type directly like float for float and int for integer.
arr = np.array([1.1, 2.1, 3.1])
newarr = arr.astype('i')
print(newarr)
print(newarr.dtype)

arr = np.array([1, 0, 3])
newarr = arr.astype(bool)
print(newarr)
print(newarr.dtype)

Shape of an array: The shape of an array is the number of elements in each dimension. NumPy arrays have an attribute called shape that returns a tuple with each index having the number of corresponding elements.
arr = np.array([[1, 2, 3, 4], [5, 6, 7, 8]])
print(arr.shape)

#Create an array with 5 dimensions using ndmin using a vector with values 1,2,3,4 and verify that last dimension has value 4:

```
arr = np.array([1, 2, 3, 4], ndmin=5)
print(arr)
print('shape of array :', arr.shape)
```

Reshaping arrays: Reshaping refers to changing the shape of an array. By reshaping we can add or remove dimensions or change the number of elements in each dimension. But the total number of elements should be the same in both the arrays.

```
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
print(arr.shape)
newarr = arr.reshape(4, 3)
print(newarr)
```

Reshape from 1-D to 3-D:

```
#Convert the following 1-D array with 12 elements into a 3-D array.
#The outermost dimension will have 2 arrays that contains 3 arrays, each with 2 elements:
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
print(arr)
newarr = arr.reshape(2, 3, 2)
print(newarr)
newarr1 = arr.reshape(-1, 1, 2)
print(newarr1)
```

Array Iteration:

```
arr = np.array([[1, 2, 3], [4, 5, 6]])
for x in arr:
  for y in x:
    print(y)
```

Iterating on each scalar element:

```
arr = np.array([[[1, 2], [3, 4]], [[5, 6], [7, 8]]])
for x in np.nditer(arr):
  print(x)
```

Searching Arrays: You can search an array for a certain value and return the indexes that get a match. To search an array, use the where() method.

```
arr = np.array([1, 2, 3, 4, 5, 4, 4])
x = np.where(arr == 4)
print(x)
```

Addition, Subtraction, Division of elements of Matrix:

```
import numpy

# initializing matrices
x = numpy.array([[1, 2], [4, 5]])
y = numpy.array([[7, 8], [9, 10]])

print(x)
print(y)

# using add() to add matrices
print ("The element wise addition of matrix is : ")
print (numpy.add(x,y))

# using subtract() to subtract matrices
print ("The element wise subtraction of matrix is : ")
print (numpy.subtract(x,y))

# using divide() to divide matrices
print ("The element wise division of matrix is : ")
print (numpy.divide(x,y))
```

Array Multiplication:

```
import numpy

# initializing matrices
x = numpy.array([[1, 2], [4, 5]])
y = numpy.array([[7, 8], [9, 10]])

# using multiply() to multiply matrices element wise
print ("The element wise multiplication of matrix is : ")
print (numpy.multiply(x,y))

# using dot() to multiply matrices
print ("The product of matrices is : ")
print (numpy.dot(x,y))
```

Matrix transpose:

```
print ("The transpose of given matrix is : ")
print (x.T)
```

Matrix Multiplication:

```
# creating two matrices
p = [[1, 2], [2, 3]]
q = [[4, 5], [6, 7]]
print("Matrix p :")
print(p)
print("Matrix q :")
print(q)

# computing product
result = np.dot(p, q)

# printing the result
print("The matrix multiplication is :")
print(result)
```

References:

1. Idris, I. (2015). NumPy: Beginner's Guide. Packt Publishing Ltd.
2. Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. Nature, 585(7825), 357-362.
3. https://www.w3schools.com/python/numpy/default.asp
4. https://www.geeksforgeeks.org/numpy-tutorial/
5. https://numpy.org/doc/stable/user/quickstart.html

AI Applications in the area of Crop Protection (AI-DISC)

Dr. Md. Ashraful Haque

Division of Computer Applications
ICAR-Indian Agricultural Statistics Research Institute, New Delhi – 110012
ashraful.haque@icar.gov.in

What is AI-DISC?

- **AI-DISC** (Artificial Intelligence based Disease Identification System for Crops) is an AI-enabled android mobile application for automatic identification of diseases through image
- Developed under **NAHEP Component 2** and **NASF Project** (Artificial Intelligence based mobile app for identification and advisory of maize diseases and Insect Pests)
- Images and advisories have been collected from **ICAR-IIMR**, **ICAR-IARI** and **11** SAUs
- Imagebase and Knowledgebase has been maintained in **NIBPP** (National Image Base for Plant Protection)
- Deployed and Hosted on **Krishi-Megh** Cloud Infrastructure
- Provides real time crop protection solution using **AI** and **Image-processing**
- Expert forum of AI-DISC facilitates Expert-Users interaction through chat for complex multifactor plant protection problems

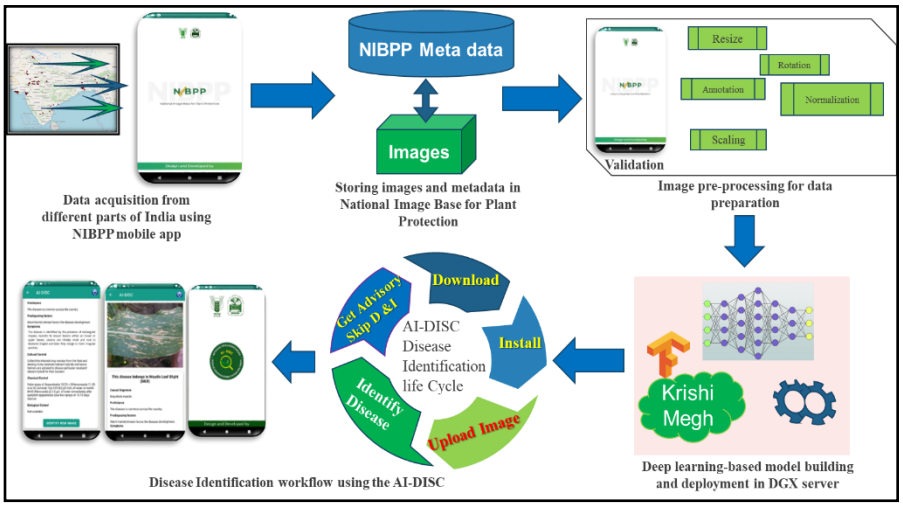


Figure: Developmental journey of AI-DISC

Simple Steps to Identify Crop Diseases

- Download AI-DISC android mobile app (https://play.google.com/store/apps/details?id=com.ai.ai_disc)
- Upload images with visible symptoms

- Get the disease and advisory automatically

Models

- Deep learning models over 19 crops (Rice, Wheat, Maize, Tomato, Mustard, Cotton, Brinjal, Apple, Peach, Kinnow, Mandarin, Assam Lemon, Chickpea, Green gram, Cluster bean, Moth bean, Chilli, Coriander etc.)
- Trained over 1.5 lakh images
- Developed and deployed in NVDIA GPU server

Features

- AI-enabled disease identification within fraction of seconds
- Expert consultations facility via text/video chat
- Real time reporting system for disease infestation across India

Figure: AI-DISC Mobile Home Screen

Types of Accounts in AI-DISC

Multiple Type of users with different privileges

Extension Worker/ Farmer

Image based disease identification

Report location wise disease occurrence

Domain Expert Account

Provide domain advisory to the users through chat and video call

Administrator

Overall control through NIBPP

Facilities available in AI-DISC: Identification

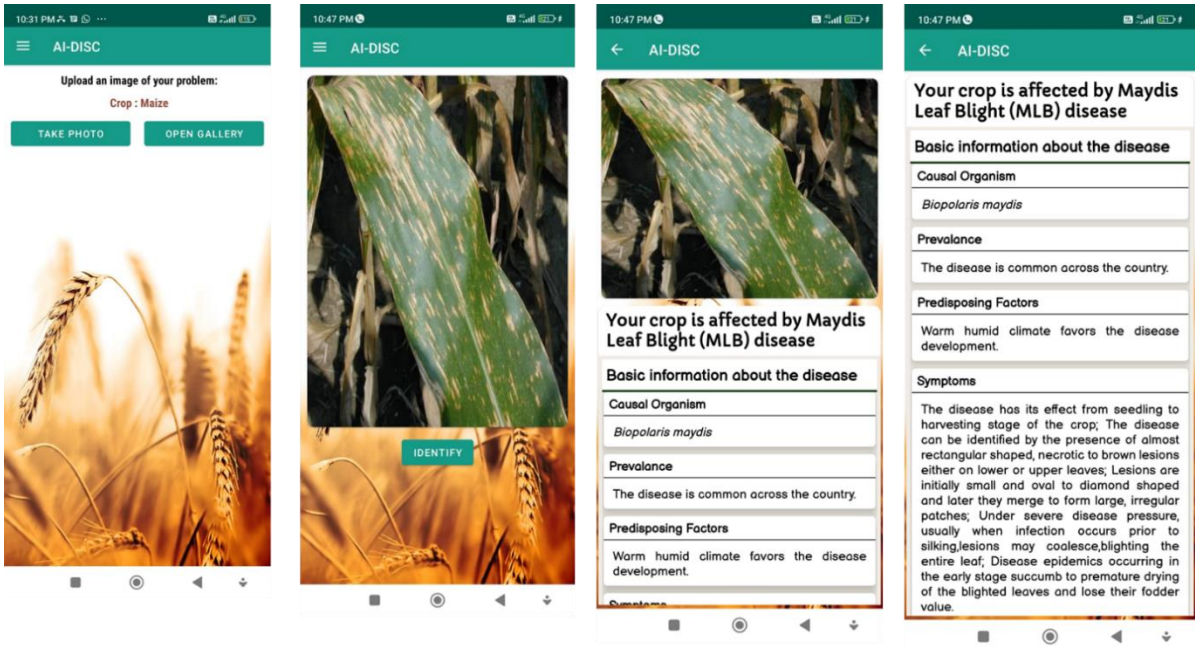


Figure: Identification Module of AI-DISC

Facilities available in AI-DISC: Reporting

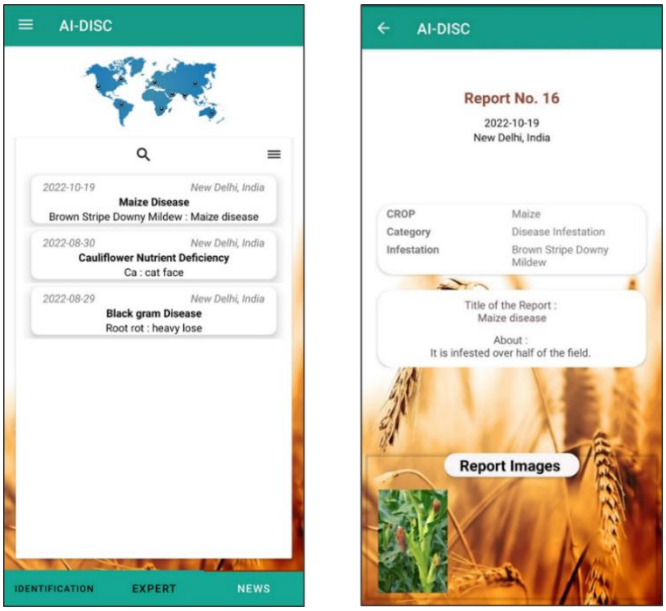


Figure: Reporting Module of AI-DISC

Report

Location wise report of plant diseases and related information along with images

Facilities available in AI-DISC: Expert Forum

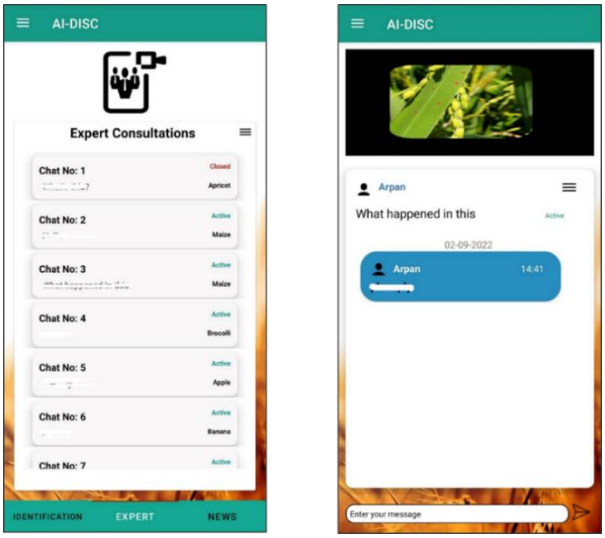


Figure: Expert forum of AI-DISC

Expert Forum

Query facility for plant protection problems

Expert-User Chat

Technology Used

App Development

Java

Android SDK

MS SQL Server

Model Programming

Python 3.6 and above

Packages

Tensorflow, Keras, Scikit-learn, Numpy, Pandas, Matplotlib

Model Train and Deploy

NVIDIA DGX GPU Clusters

Configuration

System: **NVidia DGX Server**

Operating system: **Ubuntu 18.04.3 LTS**

CPU processor: **Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20 GHz**

Graphics processor unit (GPU): **Tesla V100-SXM2- 32 GB**

RAM: **528 GB**

Deep learning framework : **PyTorch, TensorFlow**

Deep learning environment: **Jupyter Notebook**

Programming language: **Python**

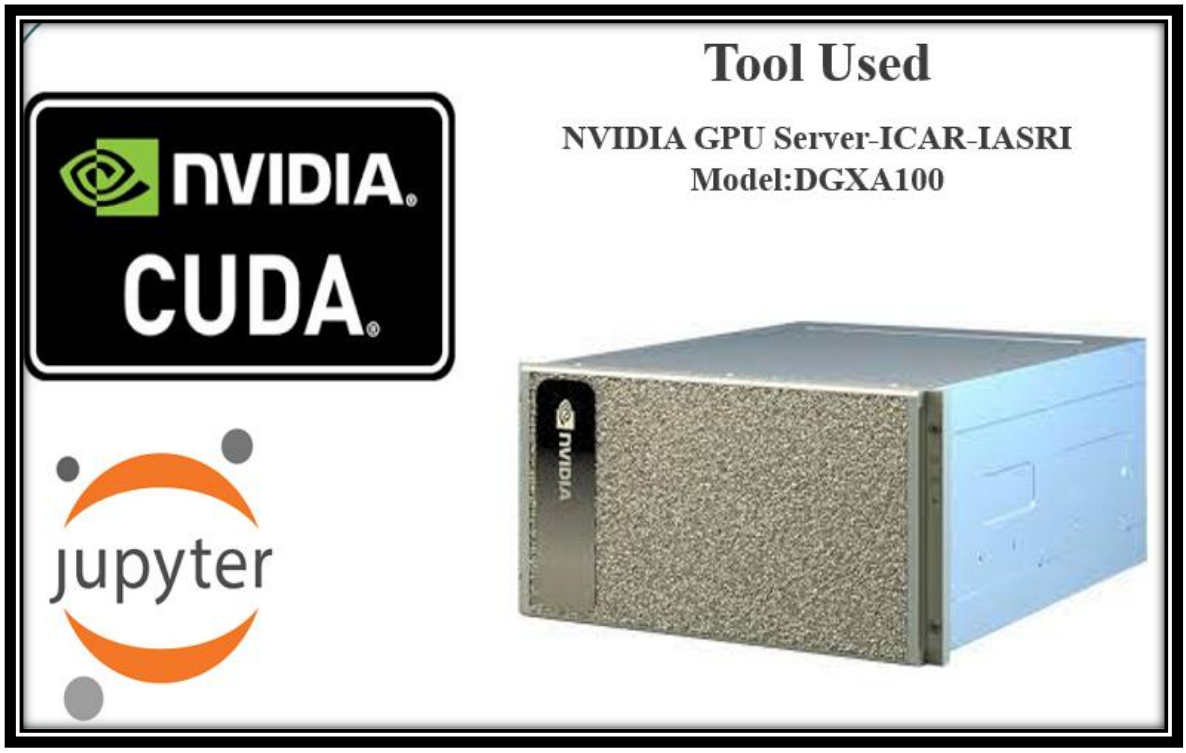


Figure: Tool Used for model development

AI Initiatives in Animal Science Domain

Dr. Chandan Kumar Deb and Dr. Sanchita Naha

Division of Computer Applications
ICAR-Indian Agricultural Statistics Research Institute, New Delhi – 110012
chandan.deb@icar.gov.in & sanchita.naha@icar.gov.in

1. Artificial Intelligence Based Disease Identification for Animal (AI-DISA)

The primary objective of the AI-DISA mobile application is to provide livestock disease identification at the field level using artificial intelligence (AI) technology. This application aims to empower extension workers and progressive farmers by offering accurate disease diagnosis and corresponding remedies in real-time, directly on their mobile devices.

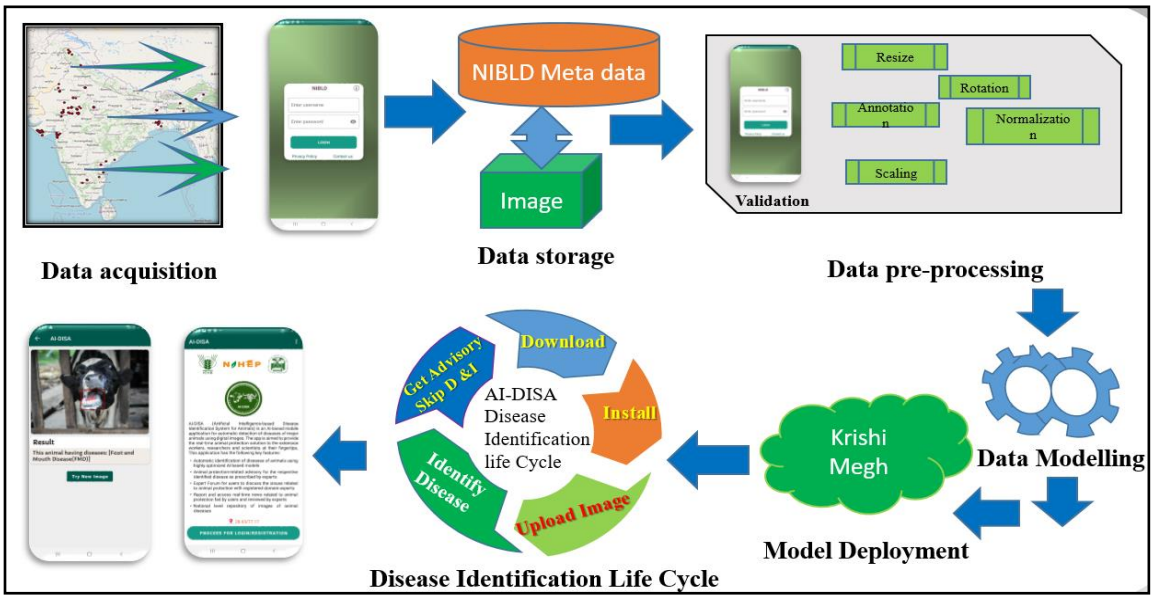


Figure: Life cycle of AI-DISA

The purpose of AI-DISA is to bridge the gap between livestock farmers and veterinary experts by providing an AI-driven platform for disease identification and advisory services. The application facilitates timely intervention, helping to mitigate losses in livestock production due to diseases. By offering accurate and swift disease identification at the field level, AI-DISA ensures that farmers can take immediate corrective actions, reducing the spread of diseases and minimizing economic losses.

India, being a vast country with a diverse livestock population, faces significant challenges in delivering expert veterinary care to every corner. The scarcity of trained experts and the logistical difficulties in reaching remote areas make it difficult to provide timely disease diagnosis and treatment. Improper identification and delayed response to

livestock diseases can lead to substantial losses and improper use of veterinary medicines, contributing to environmental hazards.

AI-DISA addresses this critical need by providing an accessible, AI-powered tool for accurate disease identification and expert advisory services. The application ensures that even in the absence of physical veterinary support, farmers and extension workers can receive reliable information and guidance, thus preventing disease outbreaks and ensuring the well-being of livestock.

Figure: Home Screen of AI-DISA app

The AI-DISA app leverages the National Image Base for Livestock Disease (NIBLD) as its primary knowledge and image repository. This database contains a comprehensive collection of images and data on various livestock diseases, making it a valuable resource for the AI models integrated into the app. The entire infrastructure is hosted on Krishi Megh, a cloud platform dedicated to agricultural data and applications, ensuring scalability, reliability, and security.

Features / Services Included

Disease Identification Module (DIM): The DIM is a core component of AI-DISA, providing an interface for identifying diseases across multiple livestock. It utilizes deep learning models trained on vast datasets to analyze

images uploaded by users and identify diseases with high accuracy. The module also offers tailored management practices and remedies based on the identified disease, aiding extension workers in providing informed advice.

Expert Forum Module (EFM): The EFM fosters communication between users (primarily extension workers) and subject matter experts from various ICAR institutes and State Agricultural Universities (SAUs). This module enables users to consult experts on complex disease-related issues, ensuring that they receive accurate and up-to-date information for disease management.

News Feeds Module (NFM): This module provides real-time updates on livestock disease outbreaks and related issues across the country. Verified users can report new incidents, which are reviewed by experts before being disseminated through the app. This feature ensures that users are always informed about the latest developments in livestock disease management.

Technology Used:

- **Platform:** AI-DISA is a native Android app, built using the latest versions of Java and Android SDK.
- **Database:** MS SQL Server is used as the backend database server.
- **AI Models:** The app incorporates deep learning models built using Python libraries and frameworks like TensorFlow and Keras. These models are trained, tested, and deployed on NVIDIA DGX GPU Clusters, ensuring high performance and accuracy.
- **Compatibility:** The app is designed to work on Android phones running versions 7 (Nougat) to 12 (R).

Key Features of AI-DISA:

- **Automatic Disease Identification:** The app provides automatic image-based identification of livestock diseases using AI models, enabling quick and accurate diagnosis.
- **Advisory Services:** Users receive crop protection-related advice from experts based on the identified disease.
- **Expert Forum:** Users can discuss other crop protection issues with registered domain experts.
- **Real-time News:** The app provides access to real-time news about livestock disease outbreaks across the country, reported and verified by experts.

AI-DISA stands out from existing livestock disease management apps by offering a comprehensive, AI-driven solution that integrates disease identification, expert consultation, and real-time information sharing in one platform. While other apps may provide similar services, AI-DISA's use of advanced deep learning models and its integration with the National Image Base for Livestock Disease (NIBLD) make it a superior choice for livestock disease management.

To ensure the AI-DISA application remains effective and accurate, regular updates and upgrades are necessary. These updates will include improvements to the AI models, enhancements to the user interface, and the addition of new features based on user feedback. Users are encouraged to provide feedback through the app, which will be used to guide future development efforts, ensuring that the application continues to meet the evolving needs of its users.

2. Development and assessment of Conversational Virtual Agents 'Chatbots' for improving livestock, Pet, and Poultry health & production

- Dairy SHRIA (Smart Heuristic Response based Intelligent Assistant): Designed and developed a conversational virtual agent, 'Chatbot', to address various health and production management concerns regarding cattle and buffalo. In collaboration with ICAR-Indian Veterinary Research Institute (ICAR-IVRI), a dataset comprising 40,000 and 42,000 question-answer pairs, respectively, for buffalo and cattle has been curated. These pairs were collected from diverse sources, such as incoming query calls to IVRI veterinary polyclinic and Kisan Call Center. Leveraging artificial intelligence-based pretrained Large Language Models (LLMs) like BERT, RoBERTa, DistilBERT, and XLNet have been fine-tuned on the dairy dataset to generate query embedding matrices. When users input queries, these matrices are used to identify the most similar Q&A entry, and the corresponding answer is provided as the chatbot response. We evaluated the performance of these models using metrics such as exact matches (EMs) and BLEU scores and ultimately deployed the best-performing model, BERT, for chatbot development. The bot is hosted on the central server of the ICAR-Data Center and is publicly accessible at https://dairyshria.icar.gov.in/chatbot/api.To track query statistics and user feedback on the chatbot response, a web-based dashboard has been developed. Additionally, an Android-based mobile application has been developed to offer a user-friendly interface for the chatbot. This application is available on the Google Play Store in 10 different languages, including English, Hindi, Bengali, and Tamil. The application can be downloaded from https://play.google.com/store/apps/details?id=com.ivriapp.ivri_chatbot.ivri_chatbot&hl=en-IN.
- Sheep and Goat SHRIA (Smart Heuristic Response based Intelligent Assistant): Developed a new chatbot tailored specifically for sheep and goats, employing a methodology like the one utilized in the development of dairy SHRIA. The training dataset, comprising 90,000 Q&A pairs evenly split between sheep and goats, forms the backbone of this methodology. Recognizing the unique health and production management challenges faced by sheep and goats, this chatbot is crafted to address these specific concerns. To facilitate accessibility, an Android-based mobile application has been developed, which is accessible via https://play.google.com/store/apps/details?id=com.ivriapp.ivri_chatbot.ivri_chatbot&hl=en_IN&gl=US. This application boasts support for 10 different languages. Thus far, it has successfully resolved over 6,500 queries pertaining to sheep and goats. Additionally for monitoring and administrative purposes, a web-based dashboard accessible at https://shria-sg.icar.gov.in has been developed. This dashboard presents a statistical summary of input queries, allows for admin-level monitoring of deployed models, and compiles user feedback for each resolved query.

Figure: Conversation through Dairy SHRIA

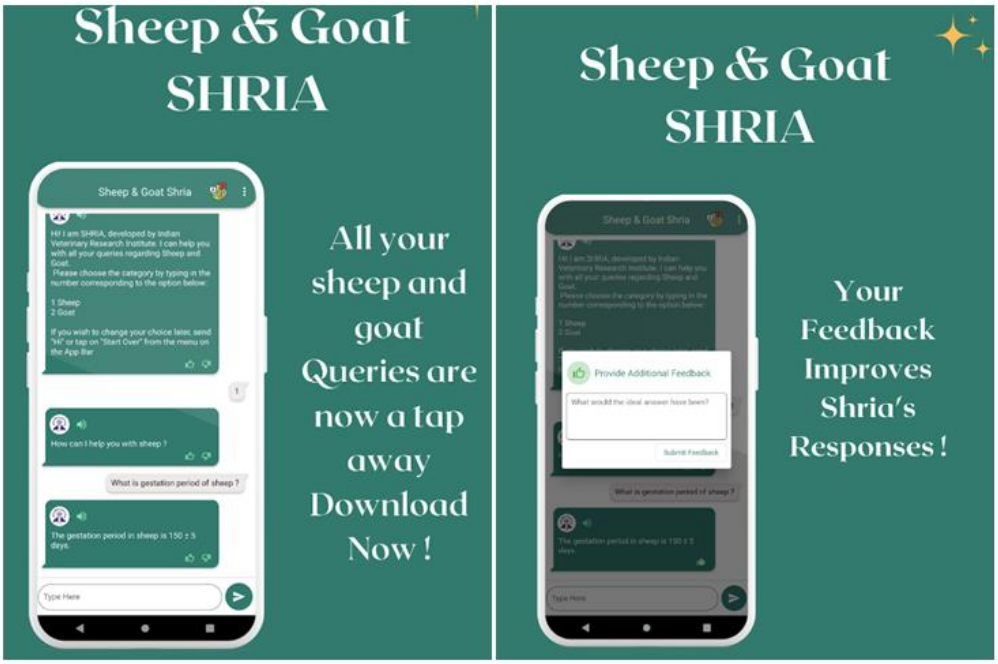


Figure: Conversation through Dairy Sheep & Goat SHRIA

GIS: A way to Spatially Intelligent Agriculture

Dr. Anshu Bharadwaj

Division of Computer Applications
ICAR-Indian Agricultural Statistics Research Institute, New Delhi – 110012
anshu.bharadwaj@icar.gov.in

1. Introduction

Agricultural sustainability has the highest priority in all countries, whether developed or developing. Use and analysis of geospatial technologies is gaining importance for sustainable agricultural management and development. Agriculture is more sophisticated than it ever was. Farmers do lots of planning and analysis. Information like soil type, soil characteristics, water sources and climate are important for strategic planning. Soil fertility and historic crop yield are important for precision farming purposes. The solution for providing food security to all people of the world without affecting the agro ecological balance lies in the adaptation of new research tools and combining them with conventional as well as frontier technologies like geospatial data, remote sensing, Geographic Information Systems (GIS) etc. Together they are called as Geospatial Infrastructure. The Geospatial Infrastructure brings the people, data, workflows, process together in a single platform as in the common framework, in terms of talking through Maps and Apps. At the micro level implementation of geospatial tools/technology is mainly used for mapping of ground water resources, drainage patterns, variable rate application and management of fertilizers, pesticides and insecticides. Geospatial technologies play a significant role in enabling the location intelligence and supporting us in many of our decisions making process and how easily it can disseminate the information to various users. In the agriculture sector Geospatial technologies plays a significant role by increasing yields, managing of resources, prediction of outcomes, in assisting in improved farm practices such as precision agriculture and many more and these are some other use cases. By using GIS in agriculture, farms can be more profitable because informed farmers can achieve higher crop yields and can have reduced waste. A Geographic Information System (GIS) is a tool that creates visual representations of data and performs spatial analyses in order to make informed decisions. It is a technology that combines hardware, software, and data. The data can represent almost anything imaginable so long as it has a geographic component.

2. What is a GIS?

"GIS" is an acronym meaning Geographic Information System. A Geographic Information System is basically a computer-based tool for mapping and analyzing geographic information. GIS can be thought of as maps on a computer but remember that it is more than just a map. It's possible to combine many maps or "layers" of information and then analyze and manipulate them to create a new map. These new maps are often used for making decisions about land use, resources, transportation, real estate, retailing, etc.

A Geographic Information System is a multi-component environment used to create, manage, visualize and analyze data and its spatial counterpart. It's quite interesting to know that most datasets that we come across in our lifetime can all be assigned a spatial location whether on the earth's surface or within some arbitrary coordinate system (such as a farmer's field or a gridded ground). Thus, we can say that, any dataset can be represented in a Geographical

Information System (GIS). Next question arises that “does it need to be analyzed in a GIS environment?” The answer to this question depends on the purpose of the analysis.

Maps are ubiquitous: we can find them online and offline in various printed version. The important part of a map is its boundaries, and we need to know how the boundaries of the map features are encoded in a computing environment. When we require that a software should assist us in the analysis, the spatial aspect /elements of that data should be readily available in a digital form. That makes us realize that simple tables or spreadsheets are not enough for the analysis of this data and a more specific and complex data storage mechanism is required. This is the core of a GIS environment: a spatial database that facilitates the storage and retrieval of spatial data which defines the spatial boundaries, lines or points of the entities that are to be analysed. This may seem trivial, but without a spatial database, most spatial data exploration and analysis would not be possible.

This ability to incorporate spatial data, manage it, analyze it, and answer spatial questions is the distinctive characteristic of a geographic information system. Map making and geographic analysis is not new, but GIS makes it possible to do this type of work faster and more efficiently because of the power and ease of using modern computers. It allows virtually anyone to create a map to help explain historic events, plan for the future, and predict outcomes. The following explains the three components of a Geographic Information System.

Geographic: This is the part of GIS that explains "spatially" where things are such as the location of nations, states, counties, cities, schools, roads, rivers, lakes, and the list can go on and on. Spatially means where on the earth's surface an object or feature is located. This can be as simple as the latitude and longitude of a feature. The geographic feature or object can be anything of interest.

Information: GIS information is the "data" or "attribute" information about specific features that we are interested in. The name of the feature, what the feature is, the location of the feature, and any other information that is important. An example could be the name of a city, where it is located, how big it is in square feet (area), its population, its population in the past, and any other information that is important.

System: The system in GIS is the computer and the software that is written to help people analyze the data, look at the data and combine it in various ways to show relationships or to create geographic models. A GIS can be made up of a variety of software and hardware tools, as long as they are integrated to provide a functional geographic data processing tool.

A GIS is a particular form of Information System applied to geographical data. It uses geographically referenced data as well as non-spatial data and includes operations which support spatial analysis

- in GIS, the common purpose is decision-making, for managing use of land, resources, transportation, retailing, oceans or any spatially distributed entities
- the connection between the elements of the system is geography, e.g. location, proximity, spatial distribution

In context of the above, a geographical information system can be defined as a computer- based tool for mapping and analyzing geographic phenomenon that exist, and events that occur, on Earth. GIS technology integrates common databases operations such as query and statistical analysis with the unique visualization and geographic analysis benefits

offered by maps. These abilities distinguish GIS from any other information system and make it valuable to a wide range of public and private enterprises for explaining events, predicting outcomes, and planning strategies. Map making and geographical analysis are not new, but a GIS performs these tasks faster and with more sophistication than traditional manual methods.

Overall, GIS should be viewed as a technology, not simply as a computer system.

A geographic information system, commonly referred to as a GIS, is an integrated set of hardware and software tools used for the manipulation and management of digital spatial (geographic) and related attribute data.

3. Feature Representation in GIS

3.1 Data Types in GIS

The basic data type in GIS reflects traditional data found on a map. Accordingly, GIS technology utilizes two basic types of data. These are:

1. **Spatial data**: These are datasets that have been prepared through field surveys or remote sensing data that is referenced on the earth's surface. Spatial data is the spatially referenced data that acts as the model of reality. Spatial data represents the absolute and relative geographical location of features e.g. points, lines, area etc.
2. **Non-Spatial Data**: They are attributes as complimentary to the spatial data and describe what is at a point, along a line or in a polygon and also socio-economic characteristics from other sources. Describes characteristics of the spatial features. These characteristics can be quantitative and/or qualitative in nature. Attribute data is often referred to as tabular data.

3.2 Nature of Geographical Data

- Geographical position (spatial location) of a spatial object is represented by 2-, 3-, or 4 dimensional coordinates in a geographically referenced system (latitudes and longitudes)
- Attributes are the descriptive information about the specified spatial objects. They have no direct information about the spatial location but can be linked to spatial objects they describe. Therefore they are often referred to as " nonspatial" or "aspatial" information.
- Spatial relationship specifies inter-relationalship between spatial objects. (e.g. distance between object A and object B, direction of object A in relation to object B, whether object A encloses object B etc.)
- Time records the time stamp of data acquisition, specifies the life of the data and identifies the locational and attribute changes of the spatial objects.

3.3 Spatial Data Structures: Vector and Raster

To work in a GIS environment, real world observations (objects or events that can be recorded in 2D or 3D space) need to be reduced to spatial entities. These spatial entities can be represented in a GIS as a **vector data model** or a **raster data model**.

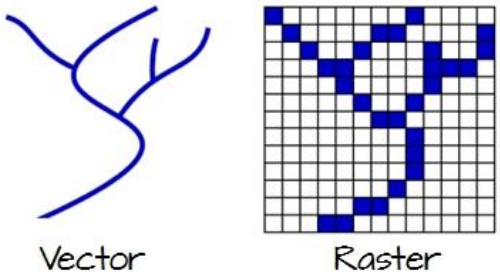


Figure 1: Vector and raster representations of a river feature.

3.3.1 Vector

Vector features can be decomposed into three different geometric primitives: **points**, **polylines** and **polygons**.

3.3.1.1 Point

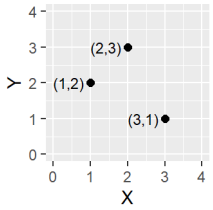


Figure 2: Three point objects defined by their X and Y coordinate values.

A point is composed of one coordinate pair representing a specific location in a coordinate system. Points are the most basic geometric primitives having no length or area. By definition a point can't be "seen" since it has no area; but this is not practical if such primitives are to be mapped. So points on a map are represented using *symbols* that have both area and shape (e.g. circle, square, plus signs).

We seem capable of interpreting such symbols as points, but there may be instances when such interpretation may be ambiguous (e.g. is a round symbol delineating the area of a round feature on the ground such as a large oil storage tank or is it representing the point location of that tank?).

3.3.1.2 Polyline

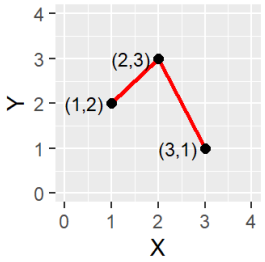


Figure 3: A simple polyline object defined by connected vertices.

A polyline is composed of a sequence of two or more coordinate pairs called vertices. A vertex is defined by coordinate pairs, just like a point, but what differentiates a vertex from a point is its explicitly defined relationship with neighboring vertices. A vertex is connected to at least one other vertex.

Like a point, a true line can't be seen since it has no area. And like a point, a line is symbolized using shapes that have a color, width and style (e.g. solid, dashed, dotted, etc…). Roads and rivers are commonly stored as polylines in a GIS.

3.3.1.3 Polygon

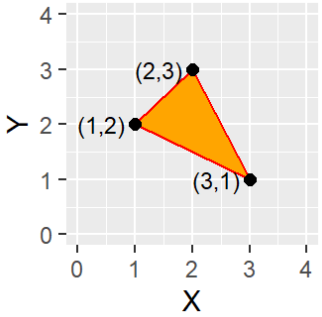


Figure.4: A simple polygon object defined by an area enclosed by connected vertices.

A polygon is composed of three or more line segments whose starting and ending coordinate pairs are the same. Sometimes you will see the words *lattice* or *area* used in lieu of 'polygon.' Polygons represent both length (i.e. the perimeter of the area) and area. They also embody the idea of an inside and an outside; in fact, the area that a polygon encloses is explicitly defined in a GIS environment. If it isn't, then you are working with a polyline feature. If this does not seem intuitive, think of three connected lines defining a triangle: they can represent three connected road segments (thus polyline features), or they can represent the grassy strip enclosed by the connected roads (in which case an 'inside' is implied thus defining a polygon).

3.3.2 Raster

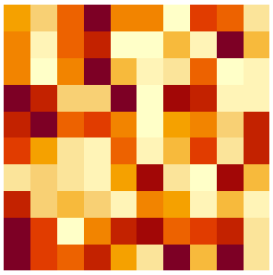


Figure 5: A simple raster object defined by a 10x10 array of cells or pixels.

A raster data model uses an array of cells, or pixels, to represent real-world objects. Raster datasets are commonly used for representing and managing imagery, surface temperatures, digital elevation models, and numerous other entities.

A raster can be thought of as a special case of an area object where the area is divided into a regular grid of cells. But a regularly spaced array of marked points may be a better analogy since rasters are stored as an array of values where each cell is defined by a single coordinate pair inside of most GIS environments.

Implicit in a raster data model is a value associated with each cell or pixel. This is in contrast to a vector model that may or may not have a value associated with the geometric primitive.

Also note that a raster data structure is square or rectangular. So, if the features in a raster do not cover the full square or rectangular extent, their pixel values will be set to no data values (e.g. NULL or NoData).

3.4 Coordinate Reference System

A Coordinate Reference System (CRS) defines with the help of coordinates , how the two dimensional, projected map in GIS is related to real places on the earth. There are various map projection and coordinate reference systems which are used, depending upon the regional extnt of the area under study, analysis to be undertaken and the data that is available.

3.5 Map Projection

Map Projection is used to represent the 3- dimensional surface of the earth or other round body on a 2-dimensional plane (map making or cartography). It is a generally a mathematical procedure but some methods are graphically based.

A map projection is a system in which locations on a curved surface of the earth are displayed on a flat sheet or surface according to some set of rules.

3.6 Scale

How one chooses to represent a real-world entity will be in large part dictated by the **scale** of the analysis. In a GIS, scale has a specific meaning: it’s the ratio of distance on the map to that in the real world. So a **large scale** map implies

a relatively large ratio and thus a small extent. This is counter to the layperson's interpretation of *large scale* which focuses on the scope or extent of a study; so a large scale analysis would imply one that covers a *large* area.

The following two maps represent the same entity: the Boston region. At a small scale (e.g. 1:10,000,000), Boston and other cities may be best represented as points. At a large scale (e.g. 1:34,000), Boston may be best represented as a polygon. Note that at this large scale, roads may also be represented as polygon features instead of polylines.

Figure 6: Map of the Boston area at a 1:10,000,000 scale. Note that in geography, this is considered small scale whereas in layperson terms, this extent is often referred to as a large scale (i.e. covering a large area).

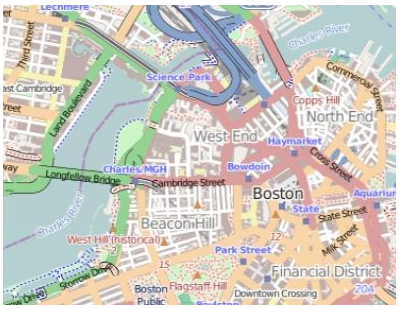


Figure 7: Map of the Boston area at a 1:34,000 scale. Note that in geography, this is considered large scale whereas in layperson terms, this extent is often referred to as a small scale (i.e. covering a small area).

3.7 Attribute Tables

Non-spatial information associated with a spatial feature is referred to as an **attribute**. A feature on a GIS map is linked to its record in the attribute table by a unique numerical identifier (ID). Every feature in a layer has an identifier. It is important to understand the one-to-one or many-to-one relationship between feature, and attribute record. Because features on the map are linked to their records in the table, many GIS software will allow you to click on a map feature and see its related attributes in the table.

Raster data can also have attributes only if pixels are represented using a small set of unique integer values. Raster datasets that contain attribute tables typically have cell values that represent or define a class, group, category, or membership. NOTE: not all GIS raster data formats can store attribute information; in fact most raster datasets you will work with in this course will not have attribute tables.

3.7.1 Measurement Levels

Attribute data can be broken down into four **measurement levels**:

Nominal data which have no implied order, size or quantitative information (e.g. paved and unpaved roads)

Ordinal data have an implied order (e.g. ranked scores), however, we cannot quantify the difference since a linear scale is not implied.

Interval data are numeric and have a linear scale, however they do not have a true zero and can therefore not be used to measure *relative* magnitudes. For example, one cannot say that 60°F is twice as warm as 30°F since when presented in degrees °C the temperature values are 15.5°C and -1.1°C respectively (and 15.5 is clearly not twice as big as -1.1).

Ratio scale data are interval data with a true zero such as monetary value (e.g. $1, $20, $100).

3.7.2 Data type

Another way to categorize an attribute is by its **data type**. ArcGIS supports several data types such as **integer, float, double and text**. Knowing your data type and measurement level should dictate how they are stored in a GIS environment. The following table lists popular data types available in most GIS applications.

Type	Stored values	Note
Short integer	-32,768 to 32,768	Whole numbers
Long integer	-2,147,483,648 to 2,147,483,648	Whole numbers
Float	-3.4 * E-38 to 1.2 E38	Real numbers
Double	-2.2 * E-308 to 1.8 * E308	Real numbers
Text	Up to 64,000 characters	Letters and words

While whole numbers can be stored as a float or double (i.e. we can store the number 2 as 2.0) doing so comes at a cost: an increase in storage space. This may not be a big deal if the dataset is small, but if it consists of tens of thousands of records the increase in file size and processing time may become an issue.

While storing an integer value as a float may not have dire consequences, the same cannot be said of storing a float as an integer. For example, if your values consist of 0.2, 0.01, 0.34, 0.1 and 0.876, their integer counterpart would be 0, 0, 0, and 1 (i.e. values rounded to the nearest whole number). This can have a significant impact on a map as shown in the following example.

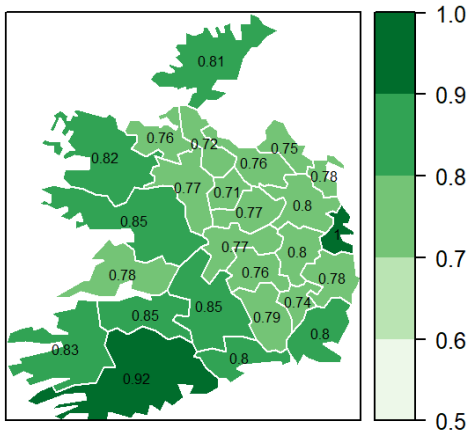


Figure 8: Map of data represented as decimal (float) values

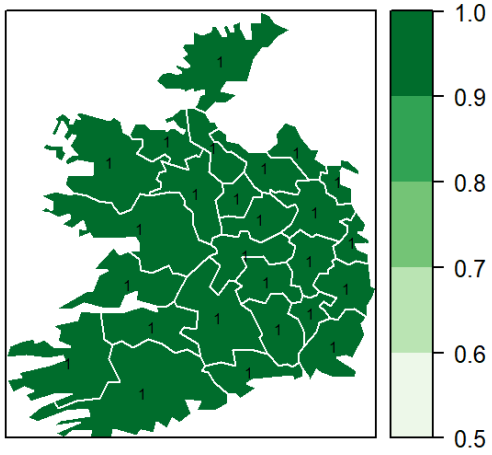


Figure 9: Map of same data represented as integers instead of float.

3.8 Spatial Analysis

Spatial Analysis is the process of manipulating spatial information to extract new information and meaning from the original data. Usually spatial analysis is carried out with a GIS . A GIS provided manipulation and spatial analysis tolls for calculating feature statistics and carryibg out geoprocessing activities as data interpolation. For example: the tool can compute an area of land under cultivation, proximity to farmed land to the sensitive ecosystems and update layers etc.

3.9 GIS software

Many GIS software applications are available–both commercial and open source. Two popular applications are **ArcGIS** and **QGIS**.

3.9.1 ArcGIS

A popular commercial GIS software is **ArcGIS** developed by ESRI (ESRI, pronounced *ez-ree*),was once a small land-use consulting firm which did not start developing GIS software until the mid 1970s. The ArcGIS desktop environment encompasses a suite of applications which include ArcMap, ArcCatalog, ArcScene and ArcGlobe. ArcGIS comes in three different license levels (basic, standard and advanced) and can be purchased with additional *add-on* packages. As such, a single license can range from a few thousand dollars to well over ten thousand dollars. In addition to software licensing costs, ArcGIS is only available for Windows operating systems; so if your workplace is a Mac only environment, the purchase of a Windows PC would add to the expense.

3.9.2 QGIS

A very capable open source (free) GIS software is **QGIS**. It encompasses most of the functionality included in ArcGIS. If you are looking for a GIS application for your Mac or Linux environment, QGIS is a wonderful choice given its multi-platform support. Built into the current versions of QGIS are functions from another open source software: **GRASS**. GRASS has been around since the 1980's and has many advanced GIS data manipulation functions however, its use is not as intuitive as that of QGIS or ArcGIS (hence the preferred QGIS alternative).

3.10 Use of GIS in Agriculture

- **Offering a huge potential for the scientist to use in the planning of crop management through effective collection of data on soil type, plant phenology and topography that is vital for maximizing crop yield.**
- Interpolate soil values for t he non-sampled using the collected and analysed figures.
- Mapping production areas, protected areas, land use pattern, crop yield estimation, water shed management, irrigation management. A
- Strategic planning of crop production areas, crop suitability analysis based on agro-ecologocal area, crop residual burning mapping.
- Scenario building of various agricultural resources like, water, soil, weather etc.

3.11 Challenges of Agriculture in India

The performance of Indian agriculture depends on many socio-economic factors. The key drivers of Indian agriculture are:

1. Government policies for funding, crops specific programmes, Education and awareness for Farmers
2. Farming and Crop Technologies, Cropping pattern for profitability
3. Environmental factors such as water availability, soil degradation and climate change

4. Market forces such as public-private ecosystems
5. Infrastructure such as irrigation, electricity, storage and raw materials such as seeds
6. Improper land use as there is huge fragmentation of land due to traditional land holding pattern
7. Suitable agricultural skills development and Availability of labour
8. Valuation and Benefit Realization for Farmers

Information technology, Geographical information systems and a complete ecosystem of services for agriculture sector can make a big difference in improving the performance of the sector.

Geospatial Artificial Intelligence or Geo AI

Artificial Intelligence (AI) has become a buzzword that symbolizes the next stage of innovative technological transformations and how various domains would be driven in future. Currently artificial intelligence is undergoing a momentous transformation and in terms of using intelligent algorithms, data classification and smart predictive analysis, AI is proving its potential application in a large number of sectors.

In the area of geographical data, the subcategory of AI that combines the exactitude of GIS with the razor-sharp analysis and solution-based approach of AI is termed Geospatial AI, or simply Geo.AI. Geospatial AI is the machine learning that deals with geographical data or the geographic component of a data set.

Machine Learning (ML)

Machine learning is a subfield of Artificial Intelligence (AI) that uses algorithms trained on data sets to create self-learning models that are capable of predicting outcomes and classifying information without human intervention. Machine learning is used today for a wide range of application areas, including suggesting products to consumers based on their past purchases, predicting stock market fluctuations, and translating text from one language to another.

Although commonly, the terms “machine learning” and “artificial intelligence” are often used interchangeably with one another, but, the two terms are meaningfully distinct. While Artificial Intelligence refers to the general attempt to create machines capable of human-like cognitive abilities, machine learning specifically refers to the use of algorithms and data sets to do so.

Machine learning in GIS is like giving the world’s most powerful magnifying glass to a person handling spatial data. It enables to uncover hidden patterns in geographic data, transforming landscapes into libraries of spatial intelligence. In simple terms, machine learning makes sense out of noisy data finding patterns that anyone would never thought existed.

Use Machine Learning (Artificial Intelligence) in GIS

Now, even when we know about **machine learning (ML), we aren’t** not exactly sure how to use it in the context of GIS.

Simply, machine learning makes sense out of noisy data finding patterns that we'd never think existed. It's like a software that writes software. Instead of applying a pre-built function, ML gains experience through historical data and builds a model to apply in future unseen situations.

Types of Machine Learning (ML)

The two broad categories of machine learning are **supervised** and **unsupervised**. And they both can apply to GIS applications in various ways. First, what's the difference between the two?

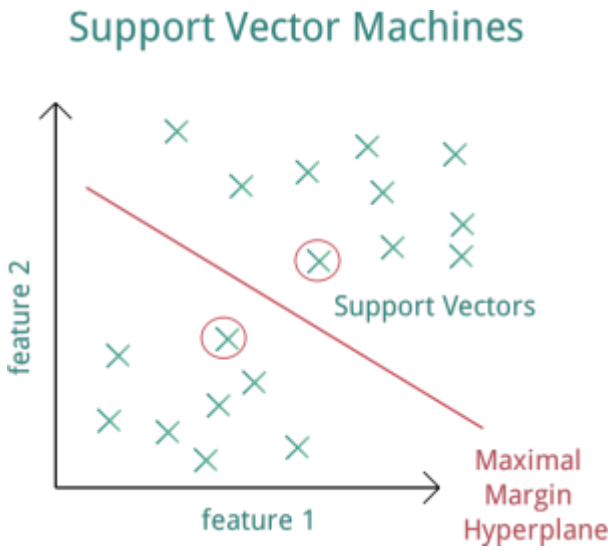
SUPERVISED LEARNING is just fitting data to a function for prediction. For example, if you plot millions of sample points in a graph, you can fit a line to approximate a function.

UNSUPERVISED LEARNING recognizes what the data is using patterns from unlabelled data. For example, it takes millions of images and runs them through a training algorithm. After trillions of linear algebra operations, it can take a new picture and segment it into clusters.

Most importantly, machine learning is about optimally solving a problem. So it automatically **learns on its own** and **improves from experience**. The application of AI or ML in GIS is many fold, but mostly, in GIS they have been extensively applied in areas such as classification, prediction and segmentation.

Image Classification (Support Vector Machine)

When you look at a satellite image, it's not always easy to know if you are looking at trees or grass… or roads vs buildings. So imagine how hard it would be for a computer to know.



Support Vector Machine (SVM) is a machine learning technique that takes classified data and looks at the extremes. Next, it draws a decision boundary line based on the data called a **"hyperplane"**. And the data points that the "hyperplane" margin pushes up against are the **"support vectors"**.

And “support vectors” are what’s important because they are the data points that are closest to the opposing classes. Because these points are the only ones considered, **all other training points can be ignored in the model**. Essentially, you feed SVM training samples of trees and grass. Based on this training data, it builds the model generating a decision boundary of its own.

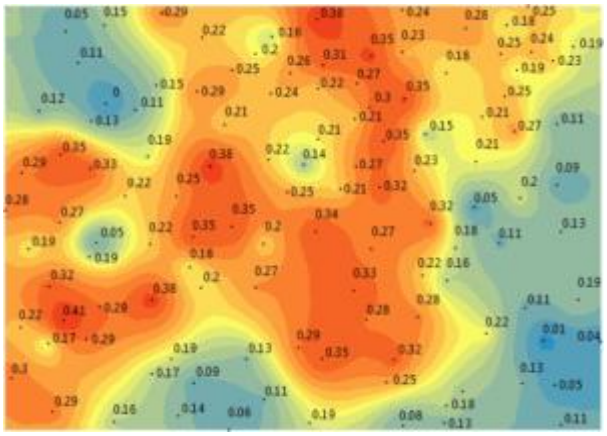
Now, the results of this supervised classification aren’t perfect and algorithms still have a lot more learning to do. We still need work on features like roads, wetlands and buildings. As algorithms get more training data, it will eventually improve to classify anywhere.

Prediction Using Empirical Bayesian Kriging (EBK)

One of the frequently used GIS tools is interpolation. Regression analysis in spatial data is for interpolation because we want to predict the unknown values in areas between the points.

The commonly used interpolation tool is krigning. To interpolate the points using ML, Empirical Bayesian Kriging (EBK) is used. When Conventional Krigning only uses a single variogram model to predict the unknown values, Empirical Bayesian Kriging (EBK) predicts values using multiple semivariograms and the Bayesian Rule.

Krigning Interpolation predicts unknown values based on spatial pattern. It estimates weights based on the variogram. The quality of the estimate surface is reflected in the quality of the weights. More specifically, weights that give an unbiased prediction and the smallest variance are required.



Unlike kriging that fits one whole model for an entire data set, EBK kriging simulates at least one hundred **local models by sub-setting the whole data set**. Because the model can morph itself locally to fit each individual semi-variogram using kriging methodology, it overcomes the challenge of stationarity.

In **Empirical Bayesian Kriging**, it predicts over and over again using a variety of simulations up to a hundred times. Each semi-variogram varies from each other. In the end, it mixes all of the semi-variograms for a final surface. It can’t be customized as in case with traditional kriging.

Finally, it outputs what it thinks is the best solution. Like a Monte Carlo analysis, it runs it repeatedly in the background. If it's a random process, then let the random process run out over a thousand times. The trends in the resulting data can be easily seen and used that to justify the selection. This is why **EBK almost always predicts better than straight kriging**.

EBK interpolates univariate data, But dependent variables can also be inout which influsences the target variable. E.g., inputting "distance from the main road", " distance from the public facility", " criminal occurrence" and " disaster risk" may support house price interpolation using EBK. Other interpolation algorithms used for spatial interpolation are Ordinary Least Square (OLS), Regression and Geographically Weighted Regression (GWR).

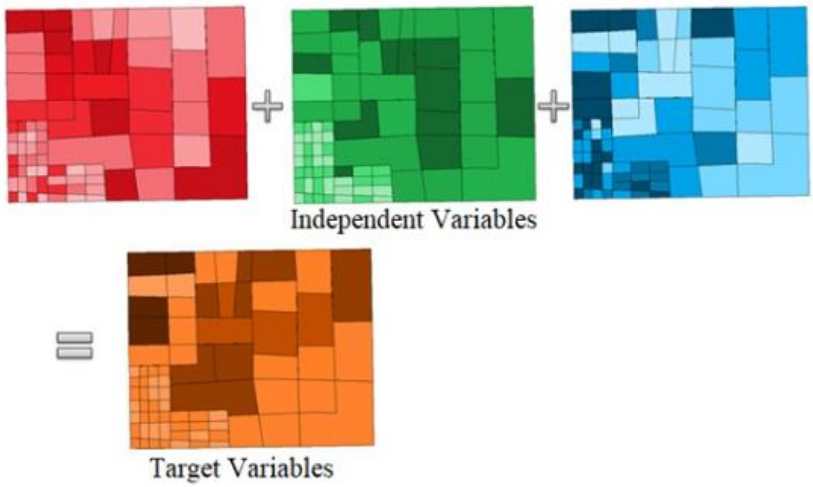


Image Segmentation and Clustering with K-means

By far, the K-means algorithm is one of the most popular methods of clustering data. In K-means segmentation, it groups unlabeled data into the **number of groups** represented by the **variable K**.

This unsupervised learning approach iteratively assigns each data point into one of the K groupings based on similarity of features. For example, similarity can be based on spectral characteristics and location.

In an unsupervised classification, the k-means algorithm first segments the image for further analysis. Next, each cluster is assigned a land cover class.

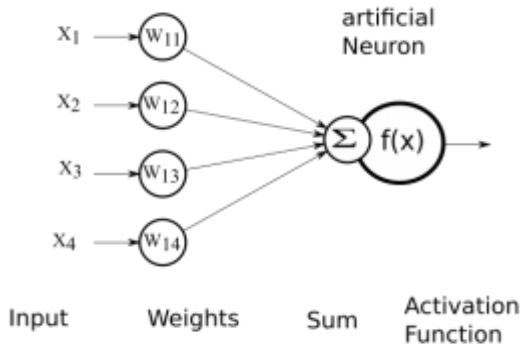
However, GIS can use clustering in other unique ways. For example, data points could represent crime and you may want to cluster **hot and low spots** of crime. Alternatively, one may want to segment based on socioeconomic, health or environmental (like pollution) characteristics.

It is now evident that sML builds a predictive model from regression, classification, and clustering tasks. Spatial data, unlike tabular data, have all observations related spatially to one another. ML for spatial data analysis builds a model to predict, classify, or cluster unknown locations according to lnown locations in the training datasets by taking the spatial attributes into account. Table below shows the Machine Leaning Techniques for Spatial Analysis as compared to Conventional Machine Learning.

	Conventional Machine Learning	Machine Learning for Spatial Analysis
Regression	• Linear regression • Decision Tree regression • Random Forest regression • Gradient Boosting regression • Support Vector Machine regression • K Nearest Neighbor	• Empirical Bayesian Kriging • EBK with independent variables • Ordinary Least Squares (OLS) Regression • Geographically Weighted Regression (GWR) • Areal Interpolation
Classification	• Decision Tree classification • Random Forest classification • Support Vector Machine classification • Maximum Likehood	• Decision Tree classification • Random Forest classification • Support Vector Machine classification • Maximum Likehood
Clustering	• Hyerarchical clustering • K-means • PAM • DBSCAN	• Multivariate clustering • Spatially Constrained Multivariate Clustering • Hot Spot Analysis • Density-based clustering • Image Segmentation • Space Time Pattern Mining

The Process of Deep Learning and Training for Big Data

Whether its GIS or another field, machine learning is all the buzz these days. It's about distilling big data sets. Because if we can let the computer detect the features, it will show us things we have never noticed.



Because there's too much data, we can uncover inherent patterns from it. And the end result is a trained neural network with just a set of weighted values.

When we train big data, this is when we are going to need all the firepower we can get. But once we have the model trained, it's just a model with a set of weights in a file… And that's why machine learning is a form of artificial intelligence – because we can train our data and then apply it to something entirely new and predict what it is.

Overall, GIS uses machine learning for prediction, classification and clustering. AI and ML is still a growing field with a lot of framework still being developed daily.

References

Barasa Bernard, Ojammbo Emmanuel and Patrick Muganga (2017). GIS Appliction in agricultural development (2017): Training Manual based on QGIS Software (version 2.18.3). Kilimo Trust.

Congedo Luca, Munafo Michele, Macchi Silvia (2013). Investigating the relationship between Land Cover amd Vulnerability to Climate Change in Dar es Salaam. Working Paper, Rome: Sapienza University.

Du, P., Bai, X., Tan, K. et al. Advances of Four Machine Learning Methods for Spatial Data Handling: a Review. J geovis spat anal 4, 13 (2020). https://doi.org/10.1007/s41651-020-00048-5

Goodchild, M.F. and Janelle, D.G., 2004. Thinking spatially in the social sciences. Spatially Integrated Social Science, 3–17.

Grisso, R.D., Alley, M.m., Philips, S.b., and McClellan, P. (2005). Interpreting Yield Maps: "I gotta yield map, now what?"

Lavallin, Abigail and Downs, Joni A. (2021). Machine learning in geography–Past, present, and future. Geography Compass, Volume15, Issue5, May 2021.https://doi.org/10.1111/gec3.12563

https://mgimond.github.io/Spatial/introGIS.html

https://smallfarms.cornell.edu/2017/04/use-of-gis/

https://www.analyticsvidhya.com/blog/2021/03/introducing-machine-learning-for-spatial-data-analysis/

https://heartbeat.comet.ml/working-with-geospatial-data-in-machine-learning-ad4097c7228d

AR-VR Devices: Augmented and Virtual Reality Devices for Transforming Education through Blended Learning Platforms

Dr. Samarth Godara
Division of Computer Applications
ICAR-Indian Agricultural Statistics Research Institute, New Delhi – 110012
samarth.godara@icar.gov.in

Introduction:

In recent years, Augmented Reality (AR) and Virtual Reality (VR) technologies have emerged as powerful tools with the potential to revolutionize various industries. In the field of education, these immersive technologies offer new ways of engaging students and enhancing learning experiences. When integrated into a blended learning platform, AR and VR devices create a dynamic and interactive environment that bridges the gap between traditional and digital learning methods.

AR and VR Devices in Education:

1. Immersive Learning Environments:

AR and VR devices transport students to virtual worlds or overlay digital information onto the real world, creating immersive learning environments. This enables students to explore historical sites, conduct virtual experiments, or even travel through space—all from the confines of their classroom.

2. Enhanced Engagement:

Traditional teaching methods often struggle to captivate students' attention. AR and VR devices provide a novel and engaging approach to learning, making it more enjoyable and memorable. Visualizing complex concepts in 3D or interacting with virtual simulations increases student engagement and understanding.

3. Personalized Learning:

Blended learning platforms powered by AR and VR allow for personalized learning experiences. Adaptive content and simulations can be tailored to students' individual learning styles, pace, and preferences, catering to a diverse range of students within the same classroom.

4. Real-world Application:

AR and VR technologies bring theoretical concepts to life by providing practical, real-world applications. Students can simulate tasks related to their field of study, such as medical procedures, architectural designs, or scientific experiments, gaining valuable hands-on experience in a risk-free environment.

Blended Learning Platform:

1. Combining Traditional and Digital Learning:

Blended learning platforms integrate traditional classroom teaching with online resources and interactive technologies. AR and VR devices play a crucial role in this integration by offering a seamless transition between physical and digital learning experiences.

2. Flexible Learning Models:
Blended learning caters to diverse learning styles and preferences by offering flexibility in how students' access and engage with educational content. AR and VR devices contribute to this flexibility by providing interactive content that complements traditional lectures and textbooks.

3. Data-driven Insights:
Blended learning platforms leverage data analytics to track students' progress and performance. AR and VR technologies enable the collection of detailed information on students' interactions within virtual environments, allowing educators to tailor their teaching strategies based on individual and collective data.

Implications for Education:
1. Increased Accessibility:
Blended learning platforms with AR and VR devices make education more accessible, breaking down geographical barriers and providing students with equal opportunities for high-quality learning experiences.

2. 21st-century Skill Development:
AR and VR technologies foster the development of essential 21st-century skills such as critical thinking, problem-solving, and collaboration. Students exposed to these immersive technologies are better equipped to thrive in a technology-driven society.

3. Preparation for Future Careers:
Integrating AR and VR into education prepares students for the digital workplaces of the future. Familiarity with these technologies enhances students' technological literacy and makes them more adaptable to rapidly evolving industries.
Augmented Reality (AR) and Virtual Reality (VR) devices have become emblematic of the cutting-edge advancements in technology, providing users with immersive experiences that bridge the gap between the physical and digital worlds. Behind the captivating visual and interactive experiences lie intricate internal workings and an array of sensors that enable these devices to deliver seamless and realistic encounters.

Internal Components of AR-VR Devices:
1. Processor Unit:
At the heart of every AR-VR device is a powerful processor unit responsible for handling the vast amount of data required for rendering immersive content in real-time. This unit executes complex algorithms to ensure smooth interactions and low latency, crucial for preventing motion sickness in users.

2. Graphics Processing Unit (GPU):
The GPU is dedicated to rendering high-quality graphics, ensuring that virtual environments and objects appear realistic and respond seamlessly to user interactions. Its processing power is instrumental in creating the visually stunning landscapes and detailed 3D models that define immersive experiences.

3. Display Technology:
AR-VR devices employ advanced display technologies, including OLED (Organic Light-Emitting Diode) and LCD (Liquid Crystal Display), to provide high-resolution visuals with low latency. Some AR devices utilize transparent displays to overlay digital information onto the real world, enhancing the augmented experience.

4. Tracking Systems:
Precise tracking is essential for maintaining the illusion of immersion. AR-VR devices incorporate tracking systems that monitor the user's movements in real-time. This includes both positional tracking (tracking the device's location in physical space) and rotational tracking (tracking the device's orientation).

Sensors in AR-VR Devices:
1. Accelerometer:
The accelerometer measures changes in velocity and direction, providing information about the device's movement. This sensor is crucial for tracking head movements in VR devices, allowing users to look around and interact with virtual environments.

2. Gyroscope:
Working in conjunction with the accelerometer, the gyroscope measures the device's orientation and rotation. This enables a more accurate representation of movement, contributing to the overall realism of the virtual experience.

3. Magnetometer:
The magnetometer helps determine the device's orientation in relation to the Earth's magnetic field. This is particularly important for maintaining accurate directional information, ensuring that virtual and real-world orientations align seamlessly.

4. Depth Sensors:
Some AR-VR devices incorporate depth sensors, such as time-of-flight or structured light cameras. These sensors enable the device to perceive and understand the depth of the surrounding environment, facilitating the integration of virtual objects into the real world in AR applications.

5. Infrared Sensors:
Infrared sensors are often used for hand and gesture tracking. They detect infrared light emitted or reflected by the user's hands or other objects, allowing for natural and intuitive interactions within virtual environments.

In an era where technological advancements are reshaping industries, agriculture stands as no exception. Augmented Reality (AR) and Virtual Reality (VR) modules are proving to be invaluable tools in agricultural education, offering innovative ways to train the next generation of farmers and agribusiness professionals. These immersive technologies bring a new dimension to learning by providing hands-on experiences and practical insights, ultimately fostering a more informed and skilled workforce in the agricultural sector.

1. Interactive Learning Environments:
AR-VR modules in agricultural education transport students from traditional classrooms to interactive and dynamic learning environments. Virtual simulations allow students to explore the entire agricultural process—from planting and harvesting to pest management—in a risk-free setting. This hands-on approach enhances comprehension and retention of key concepts.

2. Realistic Field Training:
One of the key advantages of AR-VR modules is the ability to simulate real-world scenarios. In agriculture, this means replicating the challenges farmers face in the field. Students can engage in virtual field training, practicing equipment operation, crop management, and decision-making in various agricultural scenarios. These realistic simulations prepare students for the complexities of real-world farming.

3. Precision Agriculture Insights:
AR-VR modules provide insights into precision agriculture, a data-driven approach that optimizes crop yields while minimizing resource use. Students can use virtual reality to explore precision agriculture techniques, such as sensor-based monitoring, GPS-guided machinery, and variable rate technology. This exposure prepares them for the modern, technology-driven landscape of agriculture.

4. Crop Modelling and Planning:
Virtual reality modules enable students to experiment with crop modeling and planning. They can simulate different environmental conditions, assess the impact of various farming practices, and make informed decisions about crop selection and management strategies. This practical experience enhances students' ability to make data-driven decisions in their future agricultural careers.

5. Global Perspectives and Collaboration:
AR-VR modules can connect students to agricultural practices worldwide. Virtual field trips to farms in different regions allow students to observe diverse farming methods and challenges. Collaborative projects in virtual environments facilitate knowledge exchange and collaboration among students from different geographical locations, providing a global perspective on agriculture.

6. Risk Mitigation and Sustainable Practices:
Agriculture faces various risks, including weather-related challenges, diseases, and market fluctuations. AR-VR modules allow students to explore risk mitigation strategies and sustainable farming practices in a controlled environment. This knowledge equips them to address real-world challenges and contribute to the development of resilient and sustainable agricultural systems.

Conclusion:
AR and VR devices, when integrated into blended learning platforms, represent a transformative force in education. The combination of immersive experiences, personalized learning, and data-driven insights not only enhances student engagement but also equips them with the skills needed for success in the 21st century. As educators continue to explore

innovative approaches, the integration of AR and VR technologies is poised to play a pivotal role in shaping the future of education.

The internal workings of AR-VR devices are a marvel of technological innovation, seamlessly blending hardware components and sophisticated sensors to create immersive experiences. As advancements in technology continue, we can anticipate even more refined and powerful devices that push the boundaries of what is possible in the realm of augmented and virtual reality. Understanding the intricate dance of processors, sensors, and display technologies provides a glimpse into the complexity behind the magic of AR-VR experiences.

The integration of AR-VR modules into agricultural education represents a transformative shift in the way we prepare future professionals in the field. These technologies not only enhance learning experiences by providing immersive and interactive content but also equip students with practical skills and insights crucial for the rapidly evolving agricultural landscape. As the agricultural sector continues to embrace innovation, AR-VR modules are proving to be indispensable tools in cultivating a knowledgeable, skilled, and forward-thinking workforce for the future of agriculture.

Deep Learning in Agriculture

Mr. Akshay Dheeraj
Division of Computer Applications
ICAR-Indian Agricultural Statistics Research Institute, New Delhi – 110012
akshay.dheeraj@icar.gov.in

1. Artificial intelligence

Artificial intelligence was born in the 1950s when a handful of pioneers from the nascent field of computer science started asking whether computers could be made to "think". A concise definition of the field would be as follows: the effort to automate intellectual tasks normally performed by humans. As such, AI is a general field that encompasses machine learning and deep learning, but that also includes many more approaches that don't involve any learning. Early chess programs, for instance, only involved hardcoded rules crafted by programmers and didn't qualify as machine learning. For a fairly long time, many experts believed that human-level artificial intelligence could be achieved by having programmers handcraft a sufficiently large set of explicit rules for manipulating knowledge. This approach is known as symbolic AI, and it was the dominant paradigm in AI from the 1950s to the late 1980s. It reached its peak popularity during the expert systems boom of the 1980s. Although symbolic AI proved suitable to solve well-defined, logical problems, such as playing chess, it turned out to be intractable to figure out explicit rules for solving more complex, fuzzy problems, such as image classification, speech recognition, and language translation. A new approach arose to take symbolic AI's place that is machine learning.

2. Machine learning

Machine learning arises from this question: could a computer go beyond "what we know how to order it to perform" and learn on its own how to perform a specified task? Could a computer surprise us? Rather than programmers crafting data-processing rules by hand, could a computer automatically learn these rules by looking at data? This question opens the door to a new programming paradigm. In classical programming, the paradigm of symbolic AI, humans input rules (a program) and data to be processed according to these rules, and outcome answers (Figure 1). With machine learning, humans input data as well as the answers expected from the data, and outcome of the rules. These rules can then be applied to new data to produce original answers.

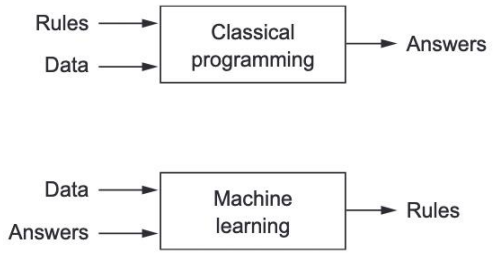


Figure 1: Machine learning new paradigm

A machine-learning system is trained rather than explicitly programmed. It's presented with many examples relevant to a task, and it finds statistical structure in these examples that eventually allows the system to come up with rules for automating the task. Although machine learning only started to flourish in the 1990s, it has quickly become the most popular and most successful subfield of AI, a trend driven by the availability of faster hardware and larger datasets. Machine learning is tightly related to mathematical statistics, but it differs from statistics in several important ways. Unlike statistics, machine learning tends to deal with large, complex datasets (such as a dataset of millions of images, each consisting of tens of thousands of pixels) for which classical statistical analysis such as Bayesian analysis would be impractical. As a result, machine learning, and especially deep learning, exhibits comparatively little mathematical theory—maybe too little—and is engineering-oriented. It's a hands-on discipline in which ideas are proven empirically more often than theoretically.

3. Learning representations from data

Machine learning discovers rules to execute a data-processing task, giving examples of what's expected. So, to do machine learning, the following things are required:

- Input data points —For instance, if the task is speech recognition, these data points could be sound files of people speaking. If the task is image tagging, they could be pictures.
- Examples of the expected output —In a speech-recognition task, these could be human-generated transcripts of sound files. In an image task, expected outputs could be tags such as "dog," "cat," and so on.
- A way to measure whether the algorithm is doing a good job —This is necessary in order to determine the distance between the algorithm's current output and its expected output. The measurement is used as a feedback signal to adjust the way the algorithm works. This adjustment step is what we call learning.

A machine-learning model transforms its input data into meaningful outputs, a process that is "learned" from exposure to known examples of inputs and outputs. Therefore, the central problem in machine learning and deep learning is to meaningfully transform data : in other words, to learn useful representations of the input data at hand—representations that get us closer to the expected output.

4. The "deep" in deep learning

Deep learning is a specific subfield of machine learning: a new take on learning representations from data that puts an emphasis on learning successive layers of increasingly meaningful representations. The deep in deep learning isn't a reference to any kind of deeper understanding achieved by the approach; rather, it stands for this idea of successive layers of representations. How many layers contribute to a model of the data is called the depth of the model. Other appropriate names for the field could have been layered representations learning and hierarchical representations learning. Modern deep learning often involves tens or even hundreds of successive layers of representations— and they're all learned automatically from exposure to training data. Meanwhile, other approaches to machine learning tend to focus on learning only one or two layers of representations of the data; hence, they're sometimes called shallow learning. In deep learning, these layered representations are (almost always) learned via models called neural networks, structured in literal layers stacked on top of each other. The term neural network is a reference to neurobiology, but although some of the central concepts in deep learning were developed in part by drawing inspiration from our understanding of the brain, deep-learning models are not models of the brain. There's no evidence that the brain implements anything like the learning mechanisms used in modern deep-learning models.

What do the representations learned by a deep-learning algorithm look like? Let's examine how a network several layers deep (Figure 2) transforms an image of a digit in order to recognize what digit it is.

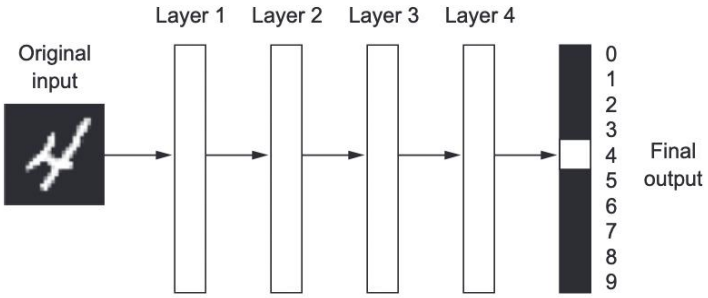


Figure 2: Digit Classification example

As you can see in Figure, the network transforms the digit image into representations that are increasingly different from the original image and increasingly informative about the final result. You can think of a deep network as a multistage information-distillation operation, where information goes through successive filters and comes out increasingly purified (that is, useful with regard to some task).

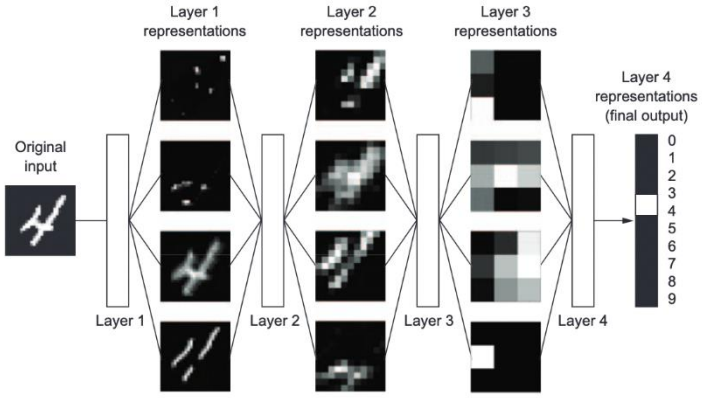


Figure 3: Deep representations learned by a digit classification model

So that's what deep learning is, technically: a multistage way to learn data representations. It's a simple idea—but, as it turns out, very simple mechanisms, sufficiently scaled, can end up looking like magic.

5. Understanding how deep learning works

The specification of what a layer does to its input data is stored in the layer's weights, which in essence are a bunch of numbers. Weights are also sometimes called the parameters of a layer. In this context, learning means finding a set of values for the weights of all layers in a network, such that the network will correctly map example inputs to their associated targets. But here's the thing: a deep neural network can contain tens of millions of parameters. Finding the

correct value for all of them may seem like a daunting task, especially given that modifying the value of one parameter will affect the behaviour of all the others!

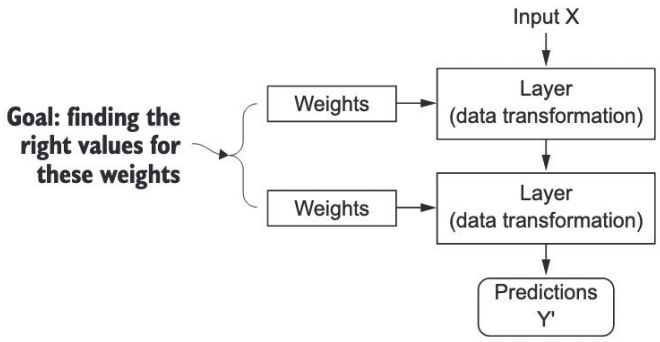


Figure 4: A neural network is parameterized by its weights

The loss function takes the prediction s of the network and the true target (what you wanted the network to output) and computes a distance score, capturing how well the network has done on this specific example (figure 5).

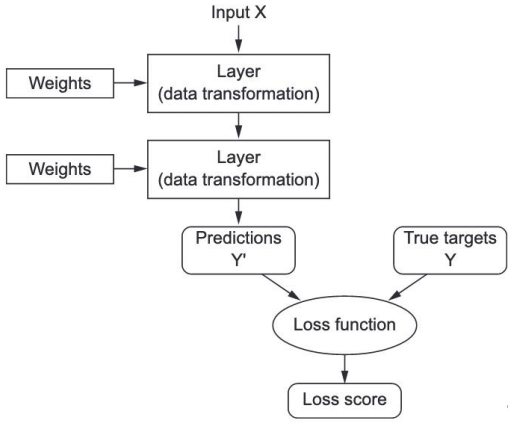


Figure 5: A loss function measures the quality of the network's output

The fundamental trick in deep learning is to use this score as a feedback signal to adjust the value of the weights a little, in a direction that will lower the loss score for the current example This adjustment is the job of the optimizer, which implements what's called the Backpropagation algorithm: the central algorithm in deep learning. The next chapter explains in more detail how backpropagation works.

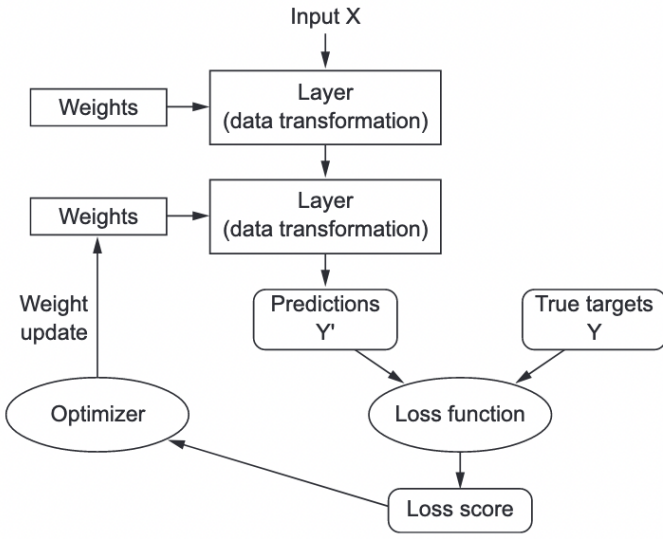


Figure 6: The loss score is used as a feedback signal to adjust the weights

Initially, the weights of the network are assigned random values, so the network merely implements a series of random transformations. Naturally, its output is far from what it should ideally be, and the lo ss score is accordingly very high. But with every example the network processes, the weights are adjusted a little in the correct direction, and the loss score decreases. This is the training loop, which, repeated a sufficient number of times (typically tens of iterations over thousands of examples), yields weight values that minimize the loss function. A network with a minimal loss is one for which the outputs are as close as they can be to the targets: a trained network. Once again, it's a simple mechanism that, once scaled, ends up looking like magic.

6. What deep learning has achieved so far

In particular, deep learning has achieved the following breakthroughs, all in historically difficult areas of machine learning:

Near-human-level image classification
Near-human-level speech recognition
Near-human-level handwriting transcription
Improved machine translation

Improved text-to-speech conversion
Digital assistants such as Google Now and Amazon Alexa
Near-human-level autonomous driving
Improved ad targeting, as used by Google, Baidu, and Bing

Improved search results on the web
Ability to answer natural-language questions
Superhuman Go playing

7. What makes deep learning different

Deep learning, on the other hand, completely automates this step: with deep learning, you learn all features in one pass rather than having to engineer them yourself. This has greatly simplified machine-learning workflows, often replacing sophisticated multistage pipelines with a single, simple, end-to-end deep-learning model.

In practice, there are fast-diminishing returns to successive applications of shallow-learning methods, because the optimal first representation layer in a three-layer model isn't the optimal first layer in a one-layer or two-layer model. What is transformative about deep learning is that it allows a model to learn all layers of representation jointly, at the same time, rather than in succession (greedily, as it's called). With joint feature learning, whenever the model adjusts one of its internal features, all other features that depend on it automatically adapt to the change, without requiring human intervention. Everything is supervised by a single feedback signal: every change in the model serves the end goal. This is much more powerful than greedily stacking shallow models because it allows for complex, abstract representations to be learned by breaking them down into long series of intermediate spaces (layers); each space is only a simple transformation away from the previous one.

These are the two essential characteristics of how deep learning learns from data: the incremental, layer-by-layer way in which increasingly complex representations are developed, and the fact that these intermediate incremental representations are learned jointly, each layer being updated to follow both the representational needs of the layer above and the needs of the layer below. Together, these two properties have made deep learning vastly more successful than previous approaches to machine learning.

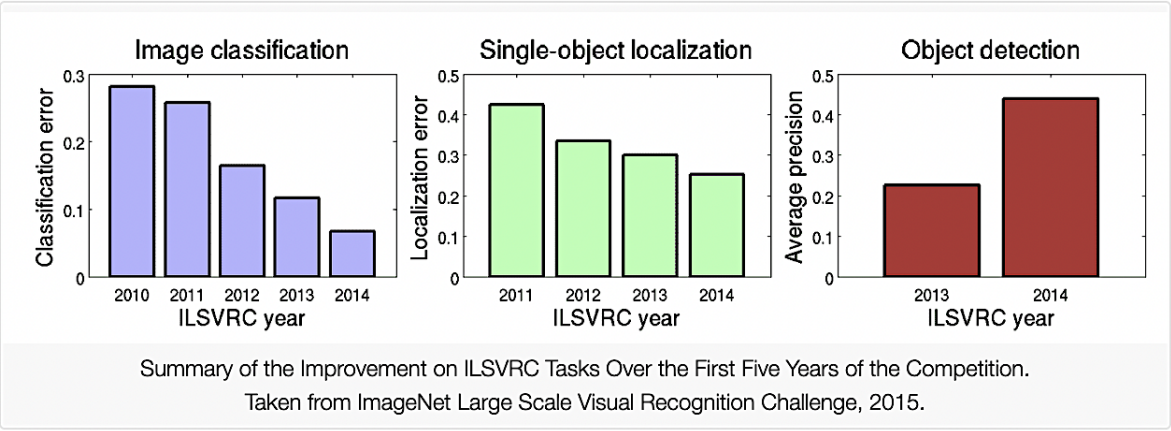
8. The limitations of deep learning

The space of applications that can be implemented with deep learning is nearly infinite. And yet, many applications are completely out of reach for current deep-- learning techniques—even given vast amounts of human-annotated data. Say, for instance, that you could assemble a data set of hundreds of thousands—even mil- lions—of English-language descriptions of the features of a software product, written by a product manager, as well as the corresponding source code developed by a team of engineers to meet these requirements. Even with this data, you could not train a deep-learning model to read a product description and generate the appropriate codebase. That's just one example among many. In general, anything that requires reasoning—like programming or applying the scientific method—long-term planning, and algorithmic data manipulation is out of reach for deep-learning models, no matter how much data you throw at them. Even learning a sorting algorithm with a deep neural network is tremendously difficult.

9. Convolutional Neural Networks Architectures

ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. The data is available for free to researchers for non-commercial use. The **ImageNet** dataset contains 14,197,122 annotated images according to the WordNet hierarchy. Since 2010 the dataset is used in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), a benchmark in image classification and object detection. The ImageNet Large Scale Visual Recognition Challenge or ILSVRC for short is an annual competition helped between 2010 and 2017 in which challenge tasks use subsets of the ImageNet dataset. The goal of the challenge was to both promote the development of better computer vision techniques and to benchmark the state of the art. The annual challenge focuses on multiple tasks

for "*image classification*" which includes both assigning a class label to an image based on the main object in the photograph and "*object detection*" which involves localizing objects within the photograph. State-of-the-art accuracy has improved significantly from ILSVRC2010 to ILSVRC2014, showcasing the massive progress that has been made in large-scale object recognition over the past five years.



Summary of the Improvement on ILSVRC Tasks Over the First Five Years of the Competition.
Taken from ImageNet Large Scale Visual Recognition Challenge, 2015.

A Convolutional Neural Network (**CNN**, or **ConvNet**) are a special kind of multi-layer neural networks, designed to recognize visual patterns directly from pixel images with minimal pre-processing. It consist of following architectures:

a. LeNet-5 (1998)

LeNet is the first CNN architecture. It was developed in 1998 by Yann LeCun, Corinna Cortes, and Christopher Burges for handwritten digit recognition problems. The model has five convolution layers followed by two fully connected layers. LeNet was the beginning of CNNs in deep learning for computer vision problems. However, LeNet could not train well due to the vanishing gradients problem. To solve this issue, a shortcut connection layer known as max-pooling is used between convolutional layers to reduce the spatial size of images which helps prevent overfitting and allows CNNs to train more effectively. The ability to process higher resolution images requires larger and more convolutional layers, so this technique is constrained by the availability of computing resources. Convolutional layers use a subset of the previous layer's channels for each filter to reduce computation and force a break of symmetry in the network. The subsampling layers use a form of average pooling.

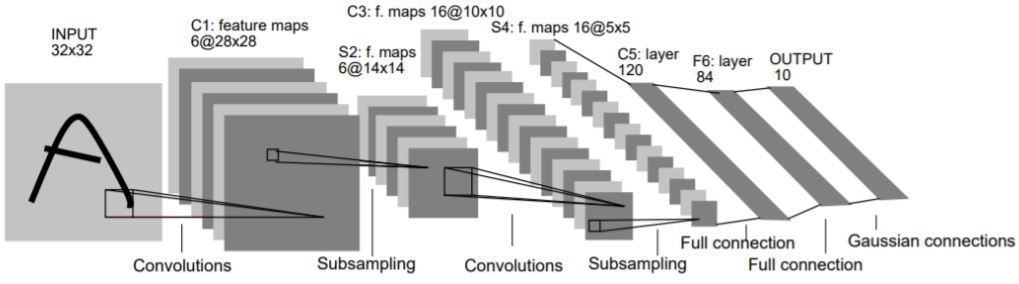


Figure 7: LeNet Architecture

b. AlexNet (2012)

AlexNet was designed by the SuperVision group in 2012 , consisting of Alex Krizhevsky, Geoffrey Hinton, and Ilya Sutskever to compete in the ImageNet competition. The general architecture is quite similar to LeNet-5, although this model is considerably larger. In 2012, AlexNet significantly outperformed all the prior competitors and won the challenge by reducing the top-5 error from 26% to 15.3%. The Alexnet model has eight CNN layers and three fully-connected layers. It was the first CNN model to have over 100 million parameters with a 60MB training set, which is considered large for deep learning models at that time. The network had a very similar architecture as LeNet by Yann LeCun et al but was deeper, with more filters per layer, and with stacked convolutional layers. It consisted 11x11, 5x5,3x3, convolutions, max pooling, dropout, data augmentation, ReLU activations, SGD with momentum. It attached ReLU activations after every convolutional and fully-connected layer.

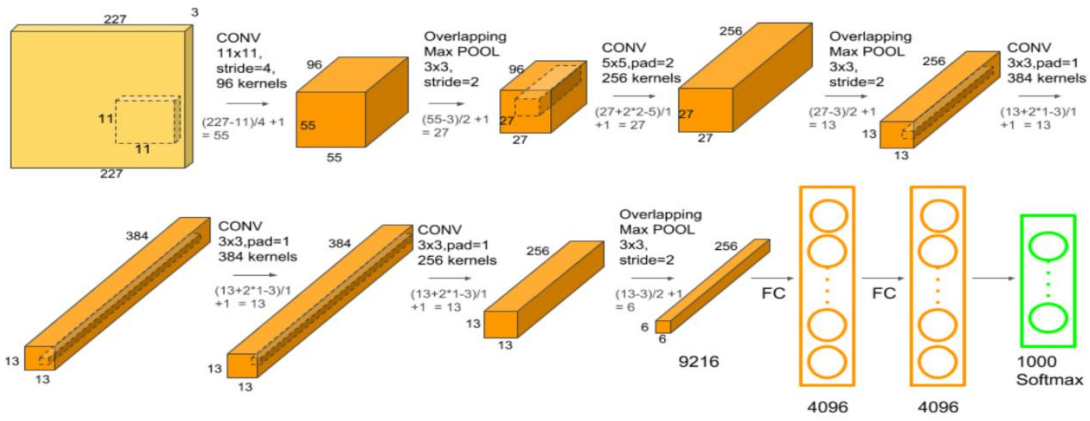


Figure 8: AlexNet Architecture

c. VGGNet (2014)

VGGNet is the CNN architecture that was developed by Karen Simonyan, Andrew Zisserman et al. at Oxford University. VGGNet is a 16-layer CNN with up to 95 million parameters and trained on over one billion images (1000 classes). It can take large input images of 224 x 224-pixel size for which it has 4096 convolutional features. CNNs with such large filters are expensive to train and require a lot of data, which is the main reason why CNN architectures like GoogLeNet (AlexNet architecture) work better than VGGNet for most image classification tasks where input images have a size between 100 x 100-pixel and 350 x 350 pixels. Real-world applications / examples of VGGNet CNN architecture include the ILSVRC 2014 classification task, which was also won by GoogleNet CNN architecture. The VGG CNN model is computationally efficient and serves as a strong baseline for many applications in computer vision due to its applicability on numerous tasks including object detection. The runner-up at the ILSVRC 2014 competition is dubbed VGGNet by the community and was developed by Simonyan and Zisserman. Its deep feature representations are used across multiple neural network architectures like YOLO, SSD etc. VGGNet consists of 16 convolutional layers and is very appealing because of its very uniform architecture. Similar to AlexNet, only 3x3 convolutions, but lots of filters. It is currently the most preferred choice in the

community for extracting features from images. The weight configuration of the VGGNet is publicly available and has been used in many other applications and challenges as a baseline feature extractor.

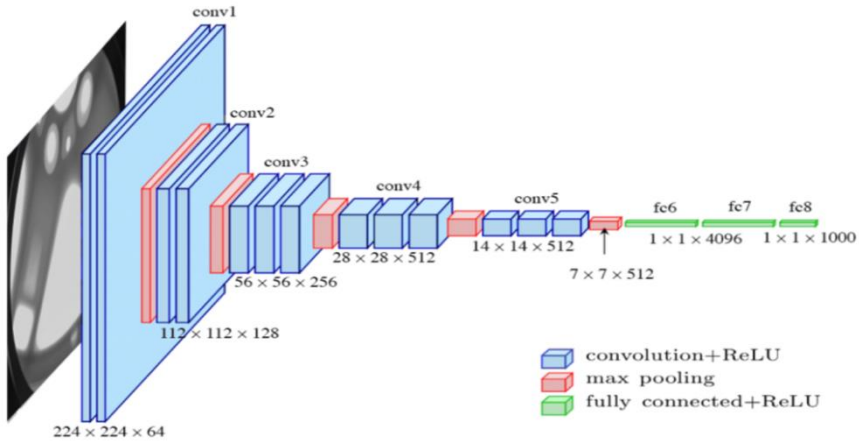


Figure 9: VGGNet Architecture

d. GoogLeNet/Inception(2014)

The winner of the ILSVRC 2014 competition was GoogLeNet(a.k.a. Inception V1) from Google. It achieved a top-5 error rate of 6.67%! This was very close to human level performance which the organisers of the challenge were now forced to evaluate.The network used a CNN inspired by LeNet but implemented a novel element which is dubbed an inception module. Their architecture consisted of a 22 layer deep CNN but reduced the number of parameters from 60 million (AlexNet) to 4 million. GoogLeNet is the CNN architecture used by Google to win ILSVRC 2014 classification task. It was developed by Jeff Dean, Christian Szegedy, Alexandro Szegedy et al.. It achieves deeper architecture by employing a number of distinct techniques, including 1×1 convolution and global average pooling. GoogleNet CNN architecture is computationally expensive. To reduce the parameters that must be learned, it uses heavy unpooling layers on top of CNNs to remove spatial redundancy during training and also features shortcut connections between the first two convolutional layers before adding new filters in later CNN layers. Real-world applications / examples of GoogLeNet CNN architecture include Street View House Number (SVHN) digit recognition task, which is often used as a proxy for roadside object detection.

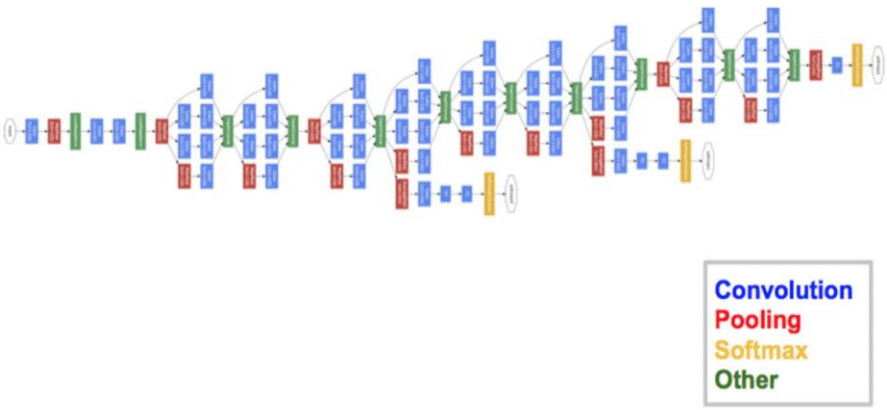


Figure 10: GoogleNet Architecture

e. ResNet(2015)

ResNet is the CNN architecture that was developed by Kaiming He et al. to win the ILSVRC 2015 classification task with a top-five error of only 15.43%.The network has 152 layers and over one million parameters, which is considered deep even for CNNs because it would have taken more than 40 days on 32 GPUs to train the network on the ILSVRC 2015 dataset. CNNs are mostly used for image classification tasks with 1000 classes, but ResNet proves that CNNs can also be used successfully to solve natural language processing problems like sentence completion or machine comprehension. Real-life applications / examples of ResNet CNN architecture include Microsoft's machine comprehension system, which has used CNNs to generate the answers for more than 100k questions in over 20 categories. The CNN architecture ResNet is computationally efficient and can be scaled up or down to match computational power of GPUs. In the residual module, the identity mapping allows to reuse the activations of the previous layer until the adjacent layers learns the weights. This identity mapping solves the problem of vanishing gradient while training a very deep CNN network

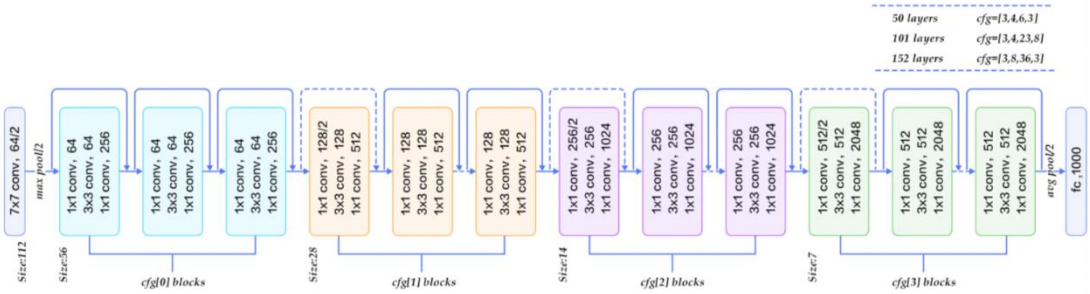


Figure 11: ResNet Architecture

Transfer Learning

Transfer Learning is a machine learning method where we reuse a pre-trained model as the starting point for a model on a new task. To put it simply—a model trained on one task is repurposed on a second, related task as an optimization that allows rapid progress when modeling the second task. By applying transfer learning to a new task, one can achieve significantly higher performance than training with only a small amount of data. ImageNet, AlexNet, and Inception are typical examples of models that have the basis of Transfer learning. Two common approaches are as follows:

- Develop Model Approach
- Pre-trained Model Approach

Develop Model Approach

- *Select Source Task.* You must select a related predictive modelling problem with an abundance of data where there is some relationship in the input data, output data, and/or concepts learned during the mapping from input to output data.

- *Develop Source Model.* Next, you must develop a skilful model for this first task. The model must be better than a naive model to ensure that some feature learning has been performed.
- *Reuse Model.* The model fit on the source task can then be used as the starting point for a model on the second task of interest. This may involve using all or parts of the model, depending on the modelling technique used.
- *Tune Model.* Optionally, the model may need to be adapted or refined on the input-output pair data available for the task of interest.

Pre-trained Model Approach

- *Select Source Model.* A pre-trained source model is chosen from available models. Many research institutions release models on large and challenging datasets that may be included in the pool of candidate models from which to choose from.
- *Reuse Model.* The model pre-trained model can then be used as the starting point for a model on the second task of interest. This may involve using all or parts of the model, depending on the modelling technique used.
- *Tune Model.* Optionally, the model may need to be adapted or refined on the input-output pair data available for the task of interest.

References:

Chollet, F. (2021). Deep learning with Python. Simon and Schuster.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

Gulli, A., & Pal, S. (2017). Deep learning with Keras. Packt Publishing Ltd.

Hands on Deep Learning:

Image classification using CNNs

Md. Ashraful Haque

Division of Computer Applications
ICAR-Indian Agricultural Statistics Research Institute, New Delhi - 110 012

ashraful.haque@icar.gov.in

Image Classification:

Image classification is the way of recognizing the instances in the image as a whole and assigning a class or label to the image. It's a one of most important subdomain of computer vision. The main aim is to categorize all the pixels of a digital image into one of the several classes. The image classification involves pre-processing of images, feature extraction and training, and classifying into predefined classes. Now-a-days, the CNNs have become the de-facto techniques for image classification/recognition tasks.

Fashion MNIST Dataset

Recently, Zalando research published a new dataset, which is very similar to the well-known MNIST database of handwritten digits. The dataset is designed for machine learning classification tasks and contains in total 60,000 training and 10,000 test images (grey scale) with each 28x28 pixel. Each training and test case is associated with one of ten labels (0–9). Up till here Zalando's dataset is basically the same as the original handwritten digits data. However, instead of having images of the digits 0–9, Zalando's data contains (not unsurprisingly) images with 10 different fashion products. Consequently, the dataset is called Fashion-MNIST dataset.

The 10 different class labels are:

- 0 T-shirt/top
- 1 Trouser
- 2 Pullover
- 3 Dress
- 4 Coat
- 5 Sandal
- 6 Shirt
- 7 Sneaker
- 8 Bag
- 9 Ankle boot

Implementation of CNN for image classification

In this section, a simple image classification model will developed using tensorflow, Keras framework using the Fashion MNIST dataset. Therefore, there are several steps that are followed during image classification which are provided below:

Step 1: Import Libraries

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
import keras
```

Step 2: Load data

```
(X_train, y_train), (X_test, y_test)=tf.keras.datasets.fashion_mnist.load_data()
#print the shape of data
X_train.shape,y_train.shape , X_test.shape,y_test.shape
#X_train - NumPy array of grayscale image data with shapes (60000, 28, 28), containing the training data.
#y_train - NumPy array of labels (integers in range 0-9) with shape (60000,) for the training data.
#X_test - NumPy array of grayscale image data with shapes (10000, 28, 28), containing the test data.
#y_test - NumPy array of labels (integers in range 0-9) with shape (10000,) for the test data.
#check the data
X_train[0]
y_train[0]

#classes of the dataset
class_labels = ["T-shirt/top", "Trouser", "Pullover", "Dress", "Coat", "Sandal", "Shirt", "Sneaker", "Bag", "Ankle boat"]

#show training images
plt.figure(figsize=(10,10))
j=1
```

```
for  i in np.random.randint(0,1000,9):
  plt.subplot(3,3,j);j+=1
  plt.imshow(X_train[i],cmap='Greys')
  plt.axis('off')
  plt.title('{} / {}'.format(class_labels[y_train[i]],y_train[i]))

#add one more dimension to the dataset as input to CNN have three dimensions
X_train=np.expand_dims(X_test,axis=1)
X_test  = np.expand_dims(X_test,-1)

#Feature Scaling (Normalise the data)
X_train = X_train/255
X_test= X_test/255

#Split the training data randomly in validation and training data (20% as validation and 80% as training data)

from sklearn.model_selection import  train_test_split
X_train,X_Validation,y_train,y_Validation=train_test_split(X_train,y_train,test_size=0.2,random_state=2023)
```

Step 3: Building the CNN model

```
model=keras.models.Sequential([

keras.layers.Conv2D(filters=32,kernel_size=3,strides=(1,1),padding='valid',activation='relu',input_shape=[28,28,1]
),
                keras.layers.MaxPooling2D(pool_size=(2,2)),
                keras.layers.Conv2D(filters=32,kernel_size=3, strides=(1,1),activation='relu'),
                keras.layers.MaxPooling2D(pool_size=(2,2)),
```

```
        keras.layers.Flatten(),
        keras.layers.Dense(units=128,activation='relu'),
        keras.layers.Dropout(0.3),
        keras.layers.Dense(units=10,activation='softmax')
])

# we have created the model with 2 convolution, 2 max pooling and 1 dense layer with 128 neuron and 1 softmax layer with 10 nodes.

#Check the summary of the developed model
model.summary()

#Compile the model with Adam optimizer with learning rate = 0.001 and sparse_categrical_crossentropy as loss function

model.compile(optimizer=keras.optimizers.Adam(lr=0.001),loss='sparse_categorical_crossentropy',metrics=['accuracy'])
#Fit the model on the training and validation data with parameter epoch and batch size

model.fit(X_train,y_train,epochs=3,batch_size=32,verbose=1,validation_data=(X_Validation,y_Validation))

#Plot the accuracy and loss curve
def accuracy_loss_plots(model):
    fig, (ax1,ax2) = plt.subplots(nrows=1,ncols=2,figsize=(12,5))
    ax1.plot(model.history.history['accuracy'], label='Training accuracy')
    ax1.plot(model.history.history['val_accuracy'], label='Validation accuracy')
    ax1.set_title('Accuracy Curve')
    ax1.set_xlabel('Epoch')
```

```
    ax1.set_ylabel('Accuracy')
    #ax1.set_ylim(0,1)
    ax1.legend()

    ax2.plot(model.history.history['loss'], label='Training loss')
    ax2.plot(model.history.history['val_loss'], label='Validation loss')
    ax2.set_title('Loss Curve')
    ax2.set_xlabel('Epoch')
    ax2.set_ylabel('Loss')
    #ax2.set_ylim(0,1)
    ax2.legend();

accuracy_loss_plots(model)

#storing predictions in variable y_pred which gives the probability value the predicted class.

y_pred = model.predict(X_test)

#evaluating model on test set to check the loss and accuracy on the test data
model.evaluate(X_test, y_test)

#Display the random images with actual and predicted label.
plt.figure(figsize=(12,12))

j=1
for i in np.random.randint(0, 1000,9):
  plt.subplot(3,3, j); j+=1
```

```
  plt.imshow(X_test[i].reshape(28,28), cmap = 'Greys')
  plt.title('Actual = {} / {} \nPredicted = {} / {}'.format(class_labels[y_test[i]], y_test[i], class_labels[np.argmax(y_pred[
np.argmax(y_pred[i])))
  plt.axis('off')


#classification report and confusion matrix
from sklearn.metrics import confusion_matrix
y_pred_labels = [ np.argmax(label) for label in y_pred ]
cm = confusion_matrix(y_test, y_pred_labels) #normalize='true'

from sklearn.metrics import classification_report
cr= classification_report(y_test, y_pred_labels, target_names=class_labels)
print(cr)

sns.heatmap(cm, annot=True, fmt='d',xticklabels=class_labels, yticklabels=class_labels)
plt.title('Fashion MNIST Confusion Matrix')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
```

Generative AI: ChatGpt

Dr. Soumen Pal
Division of Computer Applications
ICAR-Indian Agricultural Statistics Research Institute, New Delhi – 110012
soumen.pal@icar.gov.in

ChatGPT is an advanced language model developed by OpenAI, leveraging the GPT-4 architecture. It excels in natural language understanding and generation, making it suitable for various applications, including customer support, content creation, and educational tools. The journey began with GPT (Generative Pre-trained Transformer), progressing through GPT-2 and GPT-3, and now GPT-4. Each iteration has brought improvements in language understanding, contextual coherence, and response accuracy.

ChatGPT is built on transformers, a deep learning model introduced in 2017. Transformers use self-attention mechanisms to process input text, capturing intricate relationships between words and phrases. ChatGPT has been trained on a diverse dataset from the internet, encompassing a wide range of topics and language styles. This extensive training enables it to generate relevant and coherent responses across different subjects.

The key features of ChatGPT include:

(i) Natural Language Processing (NLP): Enables understanding and generating human-like text.

(ii) Contextual Awareness: Maintains context over several turns in a conversation, enhancing the relevance of responses.

(iii) Versatility: Can perform various tasks, such as answering questions, writing essays, and engaging in dialogue.

ChatGPT's architecture enables a wide range of applications, from chatbots and virtual assistants to content generation and educational tools. Some of the applications of ChatGPT are mentioned below:

(i) Customer Support: Automates responses to common inquiries, freeing human agents to handle more complex issues.

(ii) Content Creation: Assists writers by generating ideas, drafting text, and providing editorial suggestions.

(iii) Educational Assistance: Acts as a tutor, explaining concepts, answering questions, and providing study resources.

(iv) Entertainment: Engages users in interactive storytelling, role-playing games, and casual conversation. It can also generate creative content like poems, stories, and jokes.

(v) Programming Assistance: Helps developers by generating code snippets, debugging, and explaining programming concepts. It also supports learning new programming languages and frameworks.

(vi) Healthcare Support: Offers general health information and mental health support. It can also assist in scheduling appointments and providing reminders for medication.

(vii) Research and Data Analysis: Summarizes research papers, articles, and reports. It can also analyze data and generate insights for decision-making.

OpenAI is continually working to improve ChatGPT. Future developments aim to enhance accuracy, reduce biases, and expand the model's capabilities. Collaboration between AI and humans will be crucial for maximizing benefits and mitigating risks.

Reference

OpenAI. "ChatGPT". OpenAI, https://chatgpt.com. Accessed 21 June 2024.

NARES- Blended Learning Platform

Dr. Madhu
Division of Computer Applications
ICAR-Indian Agricultural Statistics Research Institute, New Delhi – 110012
madhu1@icar.gov.in

Background

While computers became part of everyday life for most in the early 2000s, education was slower to integrate computer technology. When it did, technology use was often limited to supplementing the usual teach-by-telling approach. As computers and the Internet demonstrated opportunities for connecting people in multiple locations as well as for more interaction, more visuals and greater access to information, innovation increased but in fragmented, uneven ways.

Soon, Internet connectivity and browser development allowed broader and more user-friendly resources for anyone wanting to learn. Web-based learning replaced CD-ROM materials. "Rather than having to distribute CD-ROMs to learners, organizations could simply upload material, e Learning assessments, and assignments via the web, and learners could access them with a click of a mouse button".

Today, computers, tablets and smartphones are available to the majority of the world's population, and technology-enabled learning has become more varied and accessible. More and more institutions and teachers are adding web-based learning to their delivery methods, and learners have access to many applications to support their learning. The mantra "anytime, anywhere" has been taken up to describe the new wave of education. However, this notion is being challenged by education practitioners and researchers, who know that learning competence is not universal, student skills are very different from skills needed to participate in social media, and access to broadband Internet is not evenly distributed.

Introduction

Blended learning, also known as hybrid learning, is an approach to education that combines online educational materials and opportunities for interaction online with traditional place-based classroom methods.

The simplest definition of the term *blended learning* is the use of traditional classroom teaching methods together with the use of online learning for the same students studying the same content in the same course. It is a "thoughtful fusion of face-to-face and online learning experiences. There are also *blended programmes*, in which students study some courses in face-to-face classrooms and other courses are delivered fully online.

Blended learning is sometimes called hybrid or mixed-mode learning. These systems of instructional design use many types of teaching and learning experiences and vary in design and implementation across teachers, programmes and schools. The potential variations of mixed-mode learning are virtually endless; a good way to get a sense of the range of possibilities is to consider some examples:

In one school, a few teachers create mixed-mode delivery in their individual classrooms. In another, a whole programme chooses to make blended learning its choice of delivery for all students; all teachers work together to learn how to teach in a blended delivery system.

Video recorded lectures, live video and other digitally enabled learning opportunities can be a student's primary instructional interactions with other students and the teacher. In some cases, students may work independently on online lessons, projects and assignments at home or else- where, only periodically meeting with teachers to review their learning progress, discuss their work, ask questions or receive assistance with difficult concepts. In other cases, students may spend their entire day in a traditional school building, but they will spend more time working online and independently than they do receiving instruction from a teacher.

In other words, blended learning is a term applied to the practice of providing instruction and learning experiences through some combination of both face-to-face and technology-mediated learning. During the technology-mediated components of these learning experiences, students are not required to be physically together in one place but may be connected digitally through online communities. For example, one blended learning course could involve students attending a class taught by a teacher in a traditional classroom setting while also completing online components of the course independently, outside of the classroom, on an online learning platform.
Teachers are still a key part of blended learning — teachers who have subject-matter expertise and basic technology skills, along with the new pedagogies that go with technology, such as constructivism and collaboration. Blended learning expertise provides both.

Types of blended learning models

1. Flipped model

In the flipped model, traditional classroom instruction is inverted. Learners first encounter the instructional content outside of class, typically through video lectures, readings, or online modules. Then, in-class time is dedicated to active learning activities, discussions, group projects, and exercises that reinforce and apply the pre-learned material.

The flipped model allows for more personalized and interactive learning during face-to-face sessions and provides students with the flexibility to learn at their own pace.

2. Face-to-face driver model

The face-to-face driver blended learning model is the closest to traditional classroom training, as most of the training takes place in a classroom setting under the guidance of an instructor. This approach offers individual, personalized support to learners who are struggling to grab the new concepts or are falling behind the training curriculum.

3. Rotational model

In the rotational model, learners rotate between different learning modalities, such as face-to-face instruction, online activities, small-group discussions, and independent study. These rotations can be on a fixed schedule (e.g., daily or weekly) or based on learners' progress. It offers flexibility and caters to various learning styles, allowing employees to work in the modality that best suits their needs at a particular time.

4. Flex model

The flex model provides learners with significant autonomy and control over their learning path. It combines online learning with in-person support as needed. Using an adaptive learning platform, learners have the flexibility to choose when and where they access online content and resources, making it suitable for self-paced learning. Instructors are available to assist learners when required, helping them navigate through the material and address any challenges.

5. Enriched virtual model

The enriched virtual model is primarily an online learning experience with periodic face-to-face sessions. Most of the learning occurs in a virtual environment, but learners attend physical classes, or workshops at designated times for hands-on activities, assessments, or collaborative projects.

This model combines the flexibility of online learning with the benefits of in-person interaction, ensuring learners receive both individualized instruction and opportunities for group engagement.

Benefits of Blended Learning and Virtual Learning

While there are many student and learner benefits of implementing a blended learning model, here are seven of the most impactful benefits.

1. Combination of Offline and Online

A mixture of both offline and online training approaches give you the best of both strategies. Blended learning is quite flexible and adaptable as compared to a single-method approach. It takes every type of learner into account, whether they prefer the traditional classroom, online sessions, or a mixture of both, meaning no student is left behind.

The right blended learning model helps you break the monotony of corporate training, and achieve higher employee engagement levels through a plethora of training modes including multimedia, presentations, instructor-led training, classroom workshops, real-life projects, etc.

2. Cuts costs and improves ROI

Blended learning helps you reduce your training costs as fewer trainers for less time, means fewer expenses for travel and accommodation. Furthermore, it helps to reduce the number of man-hours spent on traveling, resulting in a significant increase in productivity.

Of course, online training is not free of charge as it requires multiple resources and skilled training facilitators to develop high-quality remote training content. However, you can keep its cost at low levels by opting for many free educational technologies available today.

All in all, the blended learning approach can significantly increase your corporate training’s ROI by reducing traditional training costs and improving employee productivity.

3. Facilitates corporate training feedback

Feedback from employees is used as an indication of their performance levels. But the traditional training methods make it challenging for organizations to collect employee feedback regularly. The self-reporting surveys are unreliable most of the time, which might affect the quality of employees' performance within an organization.

An effective blended learning platform lets employees track their performance via periodic online quizzes and tests and saves you valuable time spent in collecting training feedback in a physical environment. Online blended learning platforms can help track the time taken by an employee to complete a task, the number of times they take a lesson, and the efficacy of blended learning.

Furthermore, the collaboration of multiple learning modules makes it easier for organizations to monitor and measure the effectiveness of their blended training program. The division in modules will help you collect employee feedback in small chunks that are frictionless and not intrusive.

4. Allows employees to learn at their own pace

A successful training plan is one that can work with every individual's schedule. The training program should be available whenever an employee is ready to learn.

An effective blended learning platform leaves no employee behind as it allows every employee to move through the online portion of the program at their own pace and ask queries in person during live meetings. It's a win-win for balancing busy schedules, employee preferences, and pace of learning.

Training via a blended learning approach makes your employees more active and helps them develop critical thinking because of face-to-face and technology-enhanced approaches.

5. Blended learning can be customized

Combining instructor-led training with online courses offers a unique opportunity to customize training to meet employees exactly where they are in terms of skills and knowledge.

Blended learning allows organizations to set up multiple channels that cater to every employee's learning style and demands. Information in customized training programs can be presented in different formats, based on an employee's learning preferences and goals.

The tailor-made learning program enables employees to find information through online resources, webinars, ebooks, etc. With such an approach, learners can experience the advantages of a traditional classroom, along with the flexibility of eLearning.

6. Provides ultimate flexibility

Is the online lecture moving too fast? Hit pause. Having trouble understanding a particular concept? Discuss in the face-to-face session. Whatever your employees' needs are, blended learning is flexible enough to meet them.

As discussed in the previous point, blended learning gives learners the freedom to learn at their own pace, in the comfort of their own space offering them great flexibility in learning. Also, blended learning adds the much-needed warmth of human interaction that enables learners to interact with a subject expert to clarify their doubts before they could affect their progress.

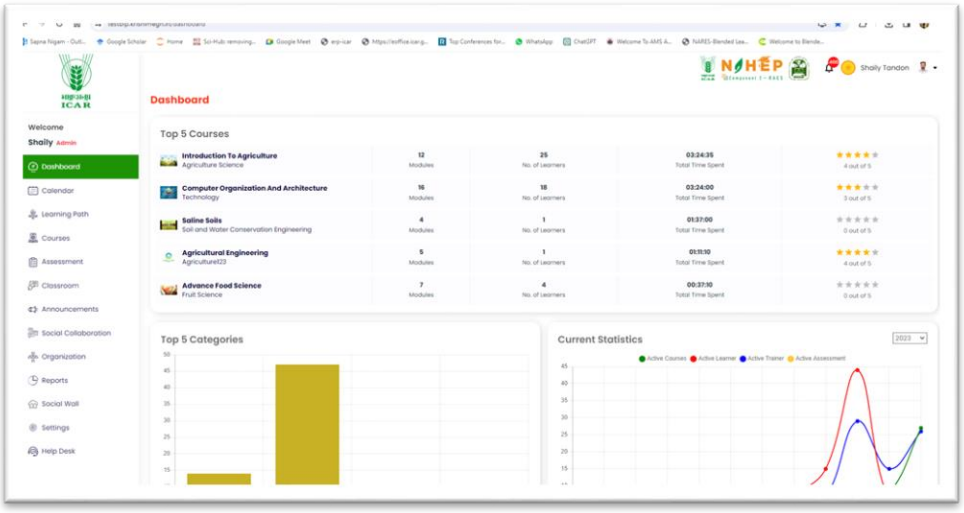
7. Increased knowledge retention

A blended learning approach helps ensure that you reach all of your employees, whether they are visual, auditory, or kinesthetic learners. It only makes sense that reinforcing training by activating more senses helps retain information longer than in a traditional approach.

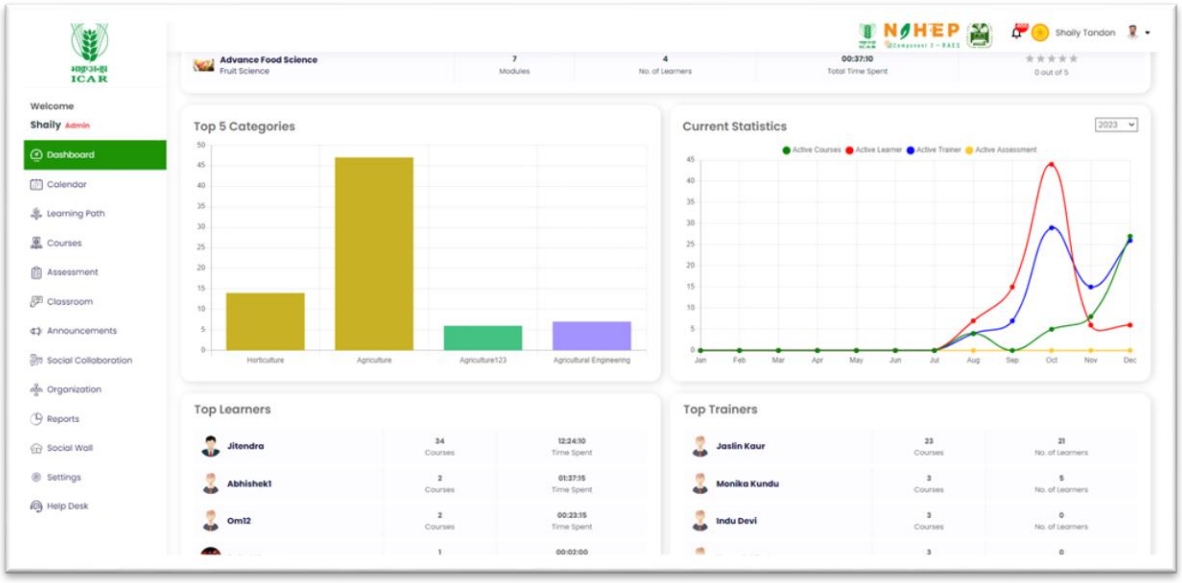
Development of Dashboard and Calendar under Faculty Module of NARES-Blended Learning Platform

After successful login as an admin, the first screen shows the Dashboard of BLP.

Step 1: Firstly, User can view their Top 5 courses along with a number of modules, learners, total time spent, and rating for each course respectively.

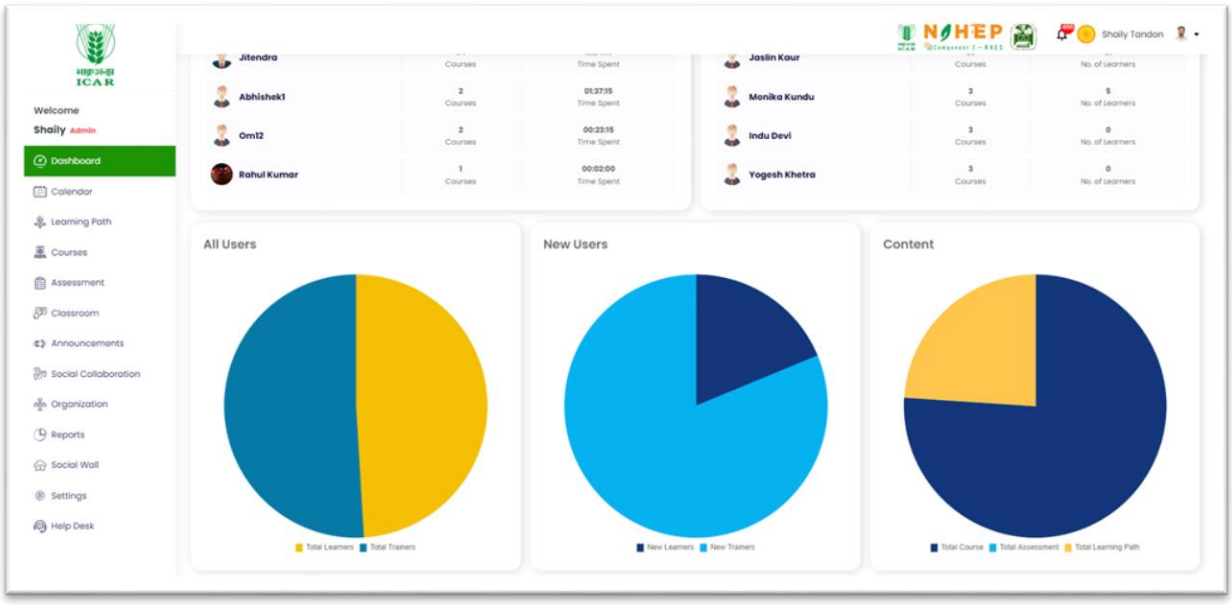


Step 2: The user can also view the Top 5 categories in which the course module has been developed and current statistics for active courses, learners, trainers, and assessments.



Step 3: The Top 5 learners/students and trainers/faculties can also be seen on the dashboard screen.

Step 4: On scrolling down the screen, the graphical representation of all users, new users, and content in BLP can be visualized by the users.



2. Calendar Module

The calendar module is the functionality that enables faculties to create events and meetings to conduct webinars, seminars, and video conferencing.

Faculties can create an event, add a schedule to the event, add an event title, assign invitees, and add a description of the event.

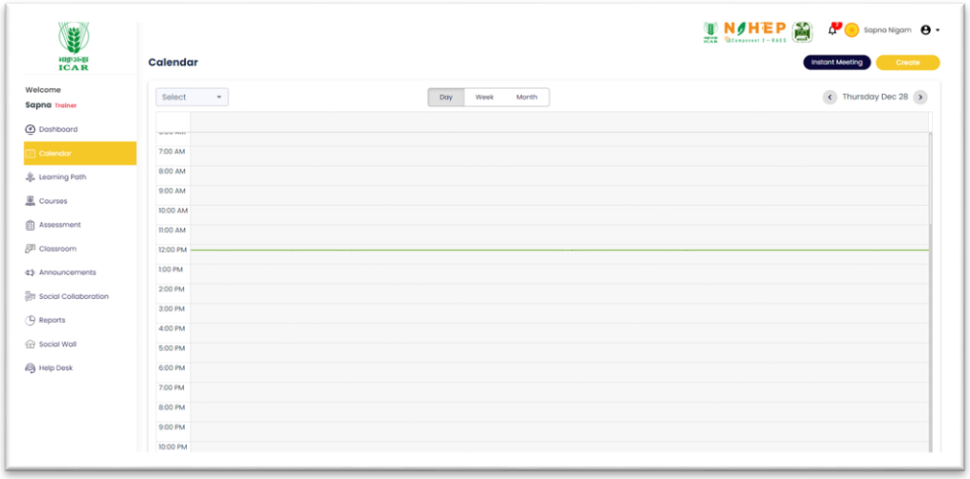
A special feature of “Instant meeting” is provided by our BLP. Users can conduct sessions/events on this integrated platform.

2.1 Calendar View

Users can select the view of the calendar as preferred. There is a table from where the users can select the view of the calendar (Day wise, week-wise & month-wise).

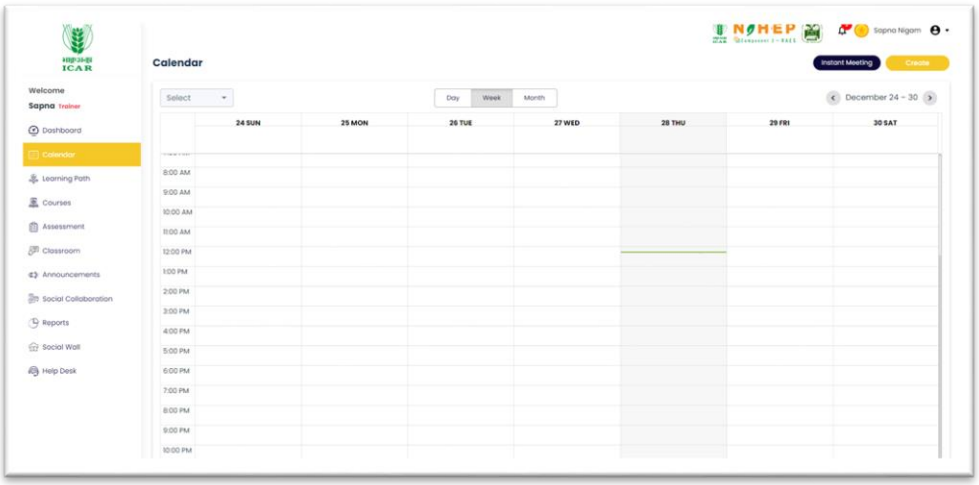
2.2 Day-wise calendar view

Select Day from the tab to view the calendar day-wise.



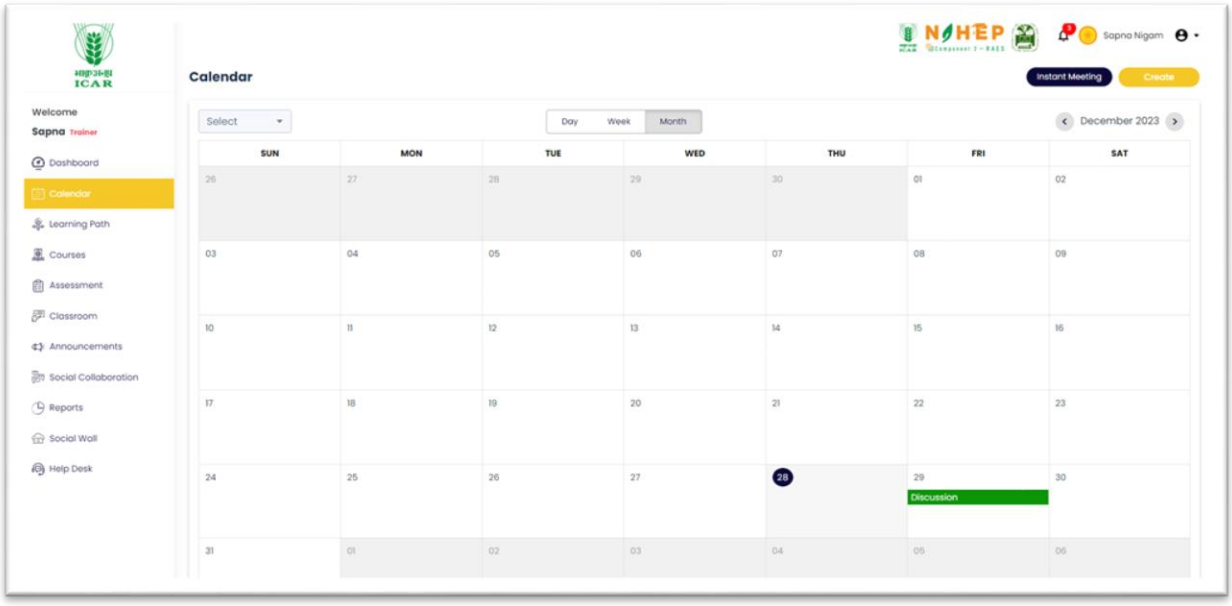
2.3 Week-wise calendar view

Select Week from the tab to view the calendar Week-wise

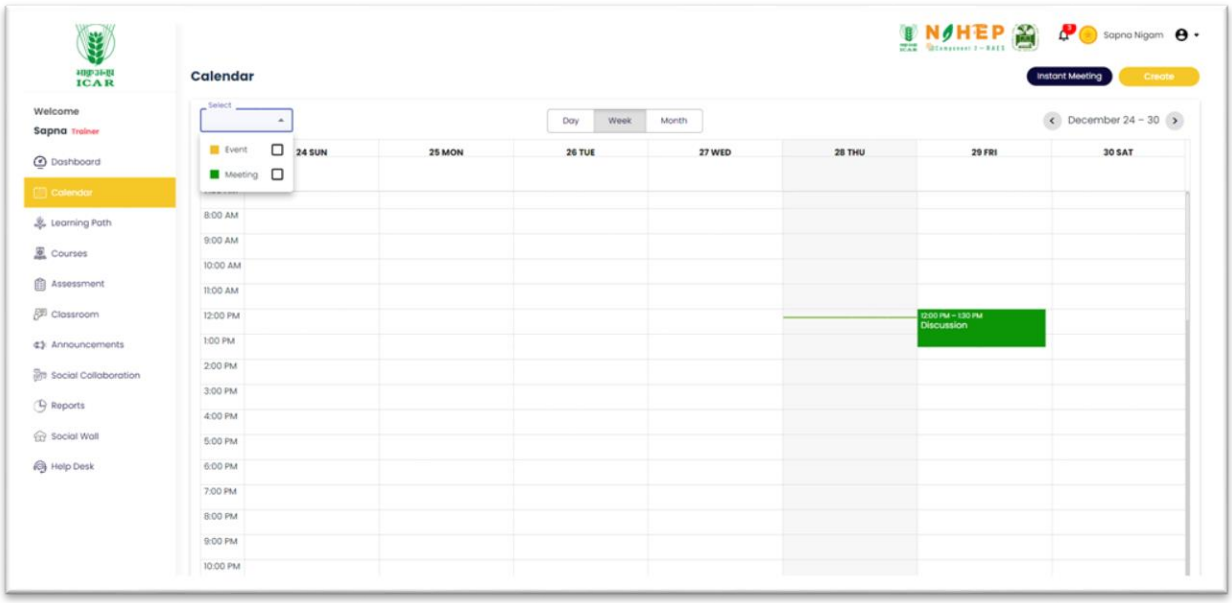


2.4 Month-wise calendar view

Select month from the tab to view the calendar month-wise.



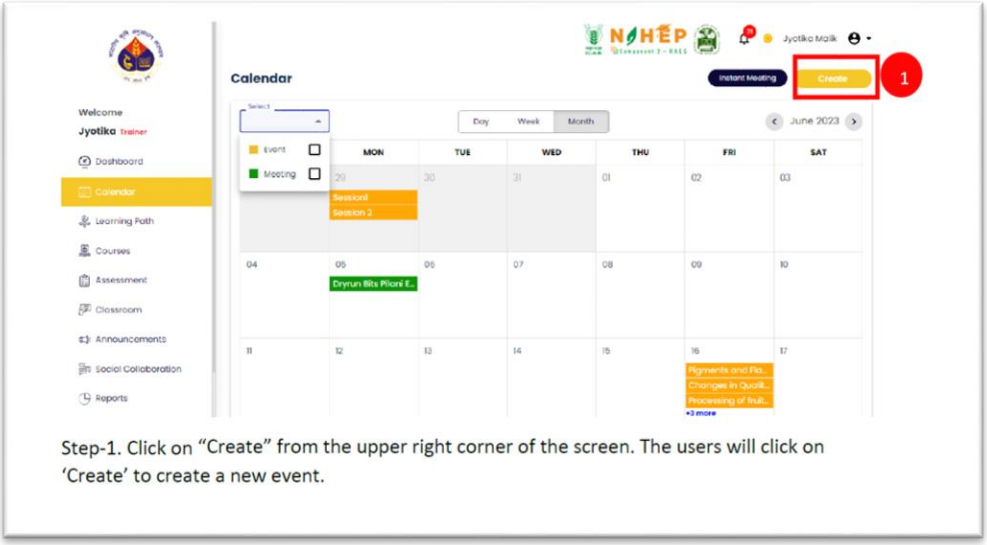
2.5 View meetings/events on the calendar



A drop-down in the upper right of the calendar gives the option to select an event or meeting. If the user selects an event from the drop-down, the user will be able to view all the scheduled events on the calendar. If the user selects meeting from the drop-down, the user will be able to view all the scheduled meetings on the calendar.

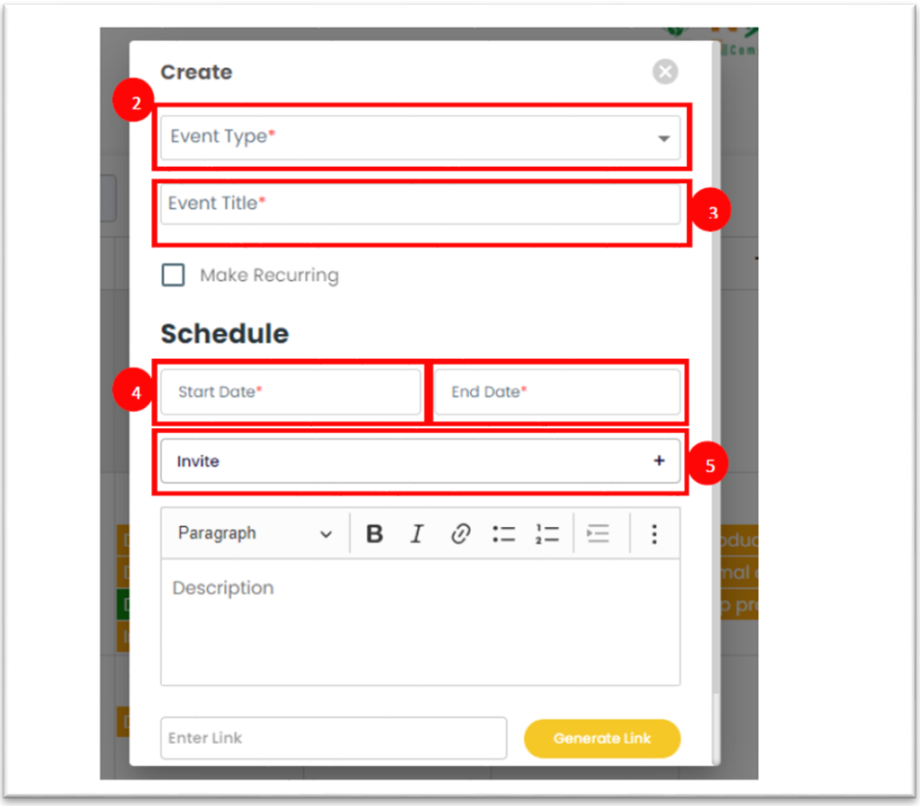
2.6 How to create an Event/Meeting?

Step-1. Click on “Create” from the upper right corner of the screen. The users will click on ‘Create’ to create a new event.



Step-1. Click on "Create" from the upper right corner of the screen. The users will click on 'Create' to create a new event.

Once the user will click on create, they will be able to see a pop-up.

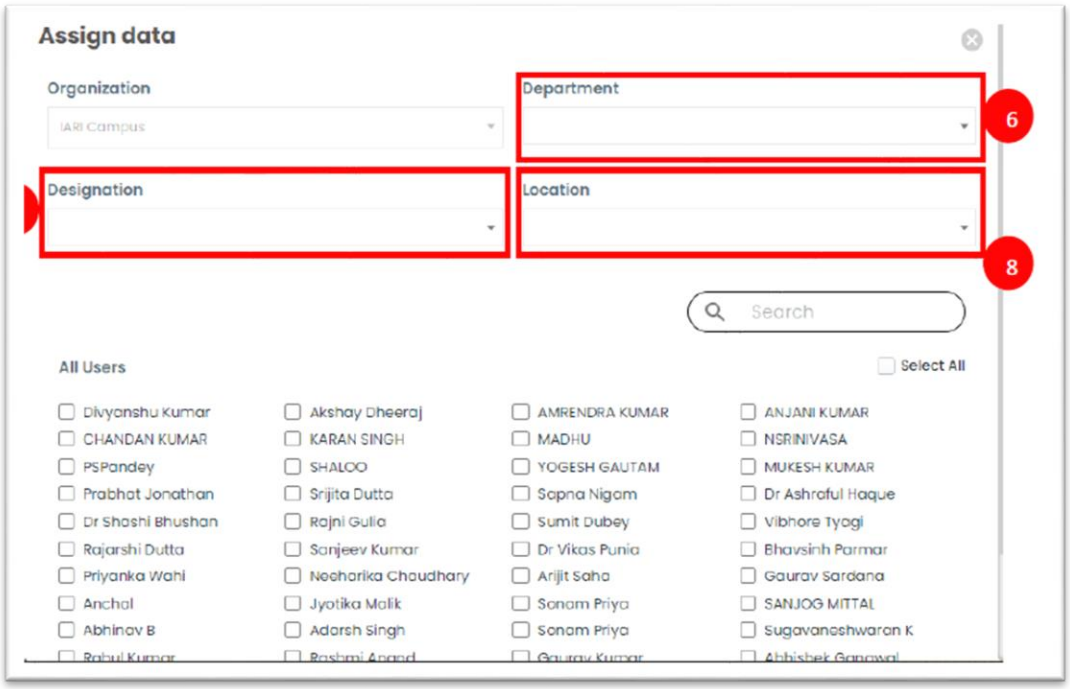


Step-2. Select "Event Type" from the drop-down menu. There will be two options i.e. event and meeting.

Step-3. Enter "Event title".

Step-4. Select "Start date" and "End date".

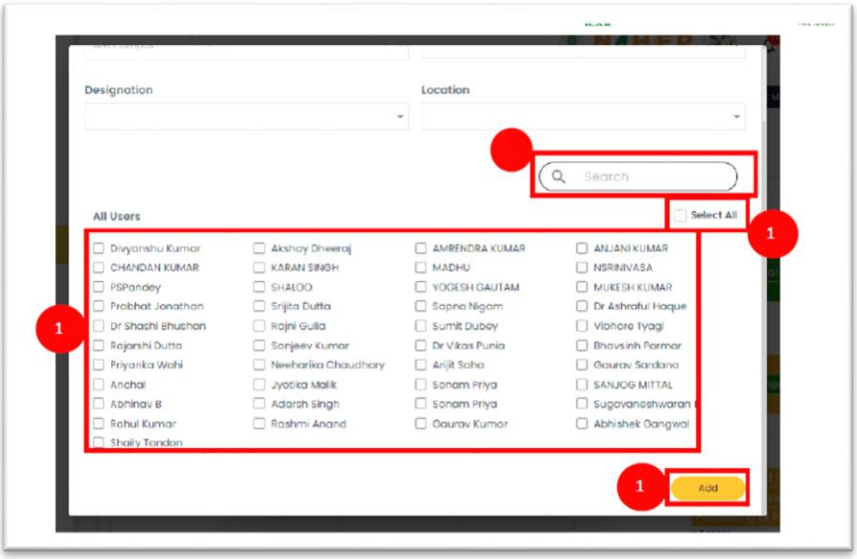
Step-5. Clicking on "+ "in the invite section. A pop-up will appear to assign students.



Step-6. Select "Department" from the drop-down menu.

Step-7. Select "Designation" from the drop-down menu.

Step-8. Select "Location" from the drop-down menu

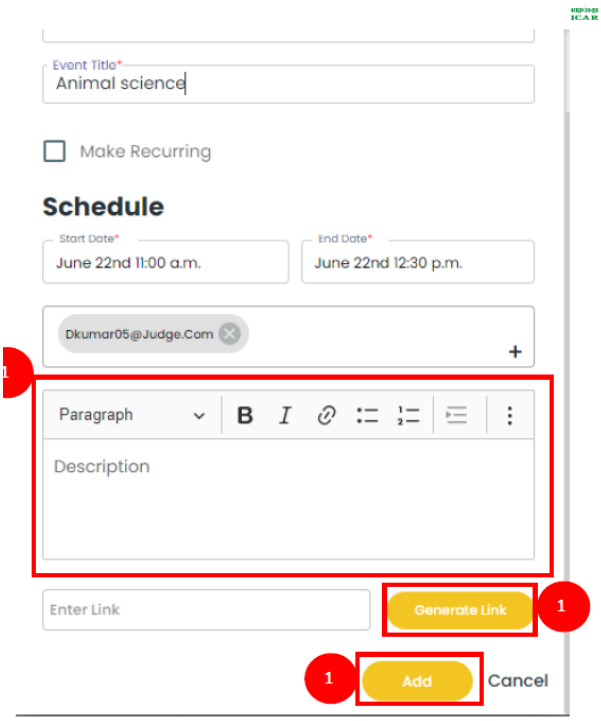


Step-9. Search the name of the student from the local search.

Step-10. Click on the "Select All" check box to select all the students.

Step-11. Click on the check box associated with the name of the student/students.

Step-12. Click on “Add”.

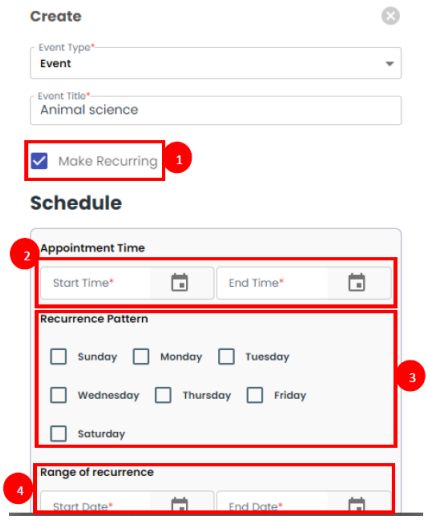


Step-13. Enter “Description”.

Step-14. Click on “Generate Link”

Step-15. Click on “Add”.

2.7 How to make Meetings recurring?



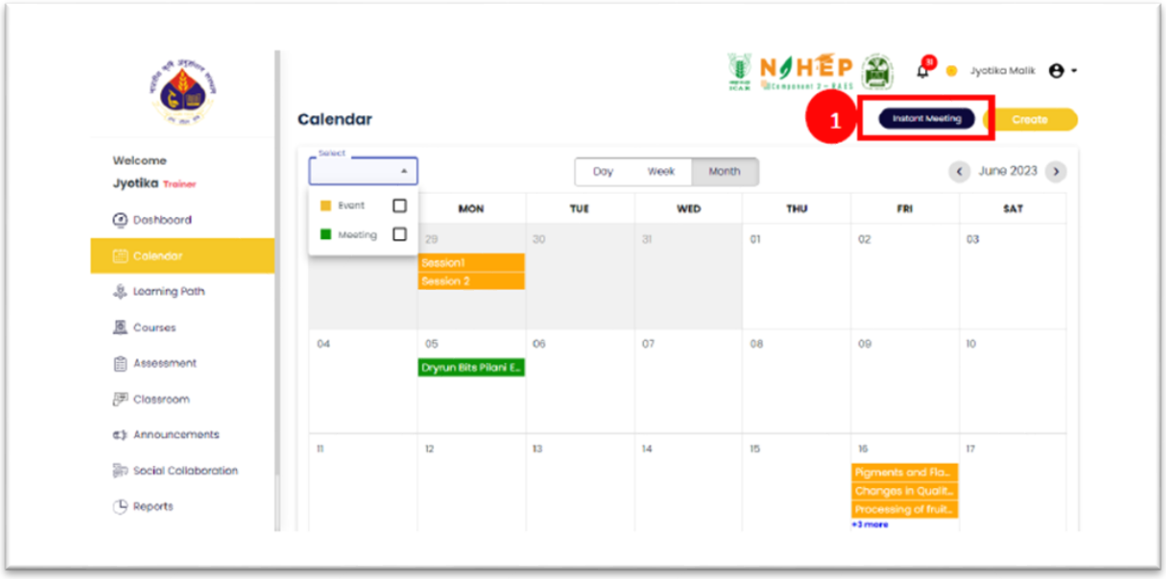
Step-1. Click on the check box “Make Recurring”.

Step-2. Select “Start Time” and “End Time” under Appointment Time.

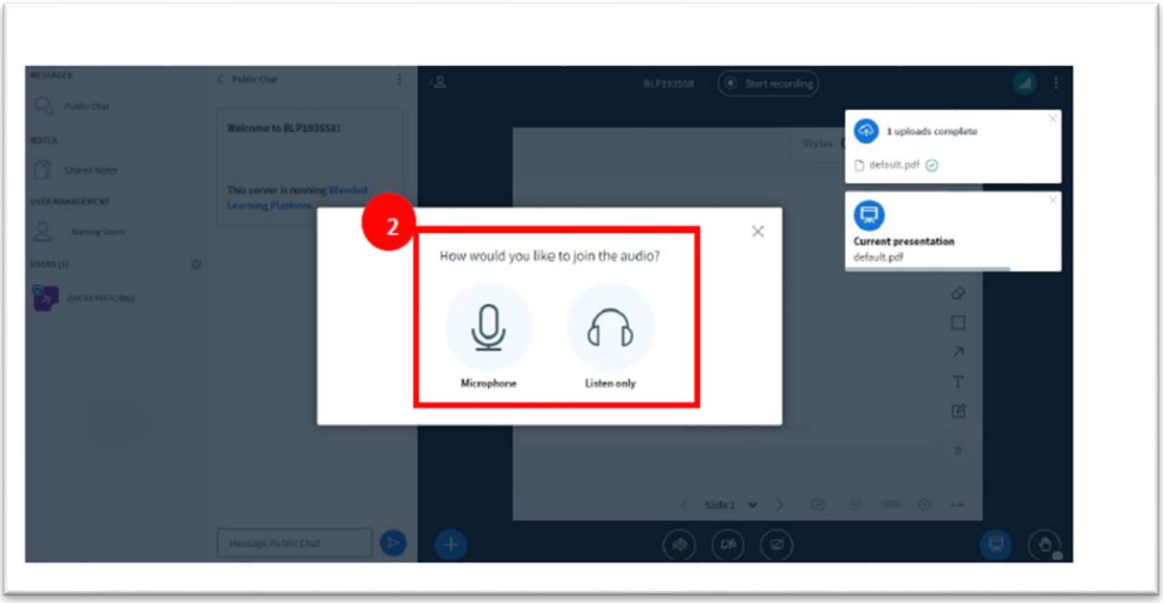
Step-3. Click on the checkboxes to select the days under the Recurrence Pattern.

Step-4. Select “Start Date” and “End Date” under Range of Recurrence.

2.8 Start an instant meeting



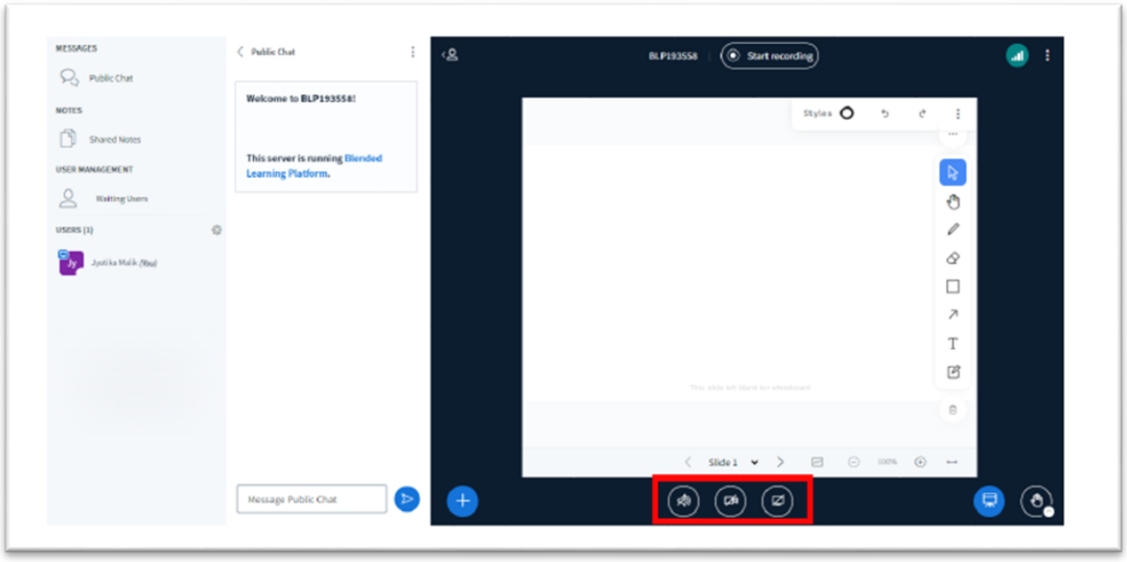
Step-1. Click on “Instant Meeting”.



..

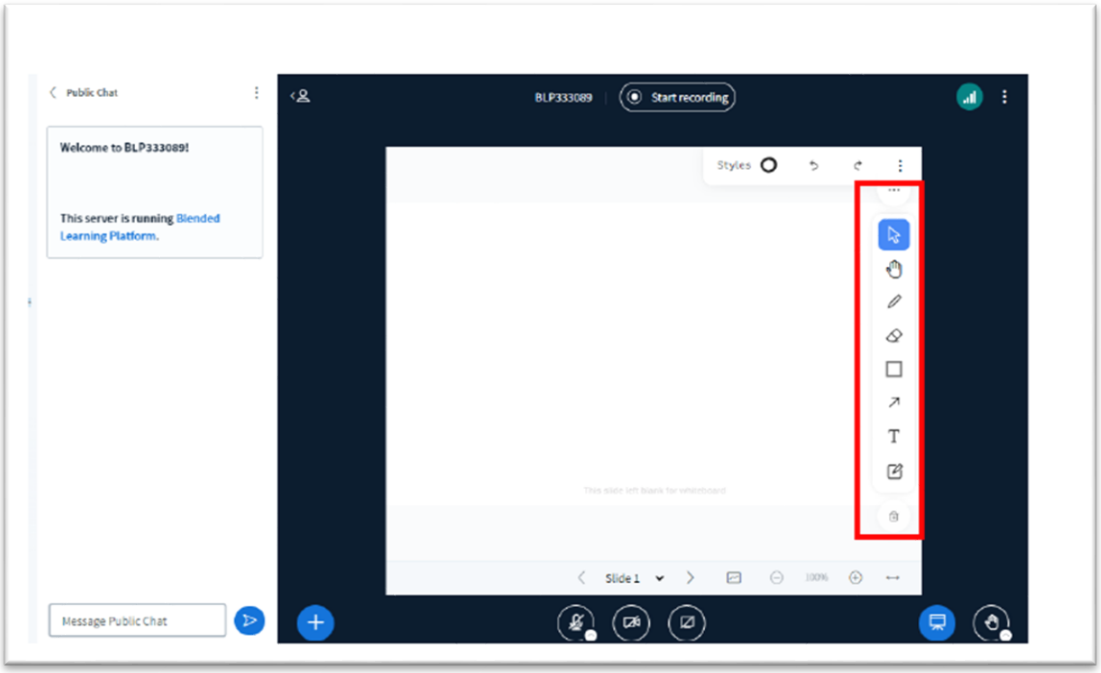
The screen will display a pop-up stating, “How would you like to join the audio?” with two options: Microphone and Listen only.

Step-2. Select Microphone or Listen only.



- Users can Mute/Unmute by clicking on the microphone icon.
- Users can On/Off video by clicking on the Camera icon.
- Users can share/unshare screen by clicking on the screen icon.

2.9 Open Whiteboard



Users have multiple features associated with a whiteboard. The features of the whiteboard are listed below:

• Select object – The users can select an object by clicking on select.

• Move object – The users can select Pan to move the board.

• Pen- The users can select a pen to draw on the board.

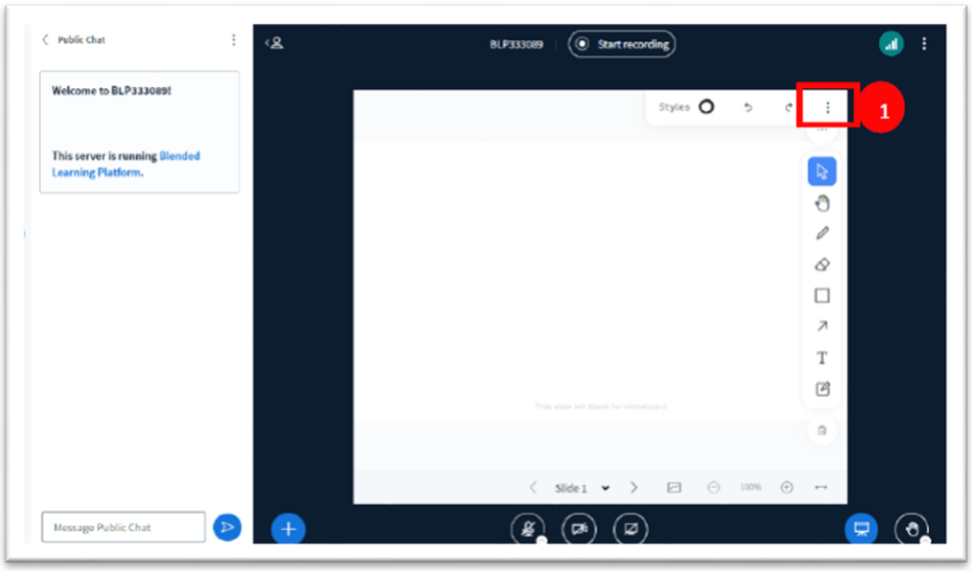
• Eraser- The users can select an eraser to erase.

• Text- The users can add text.

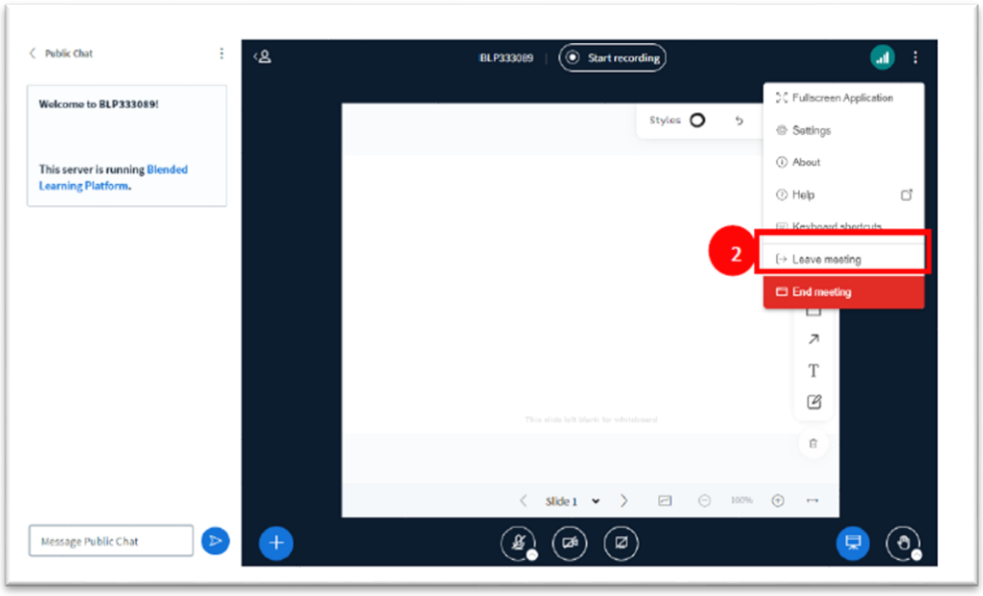
• Sticky- The users can add sticky notes.

• Delete- The users can click on delete-to-delete text.

2.10 Leave the meeting

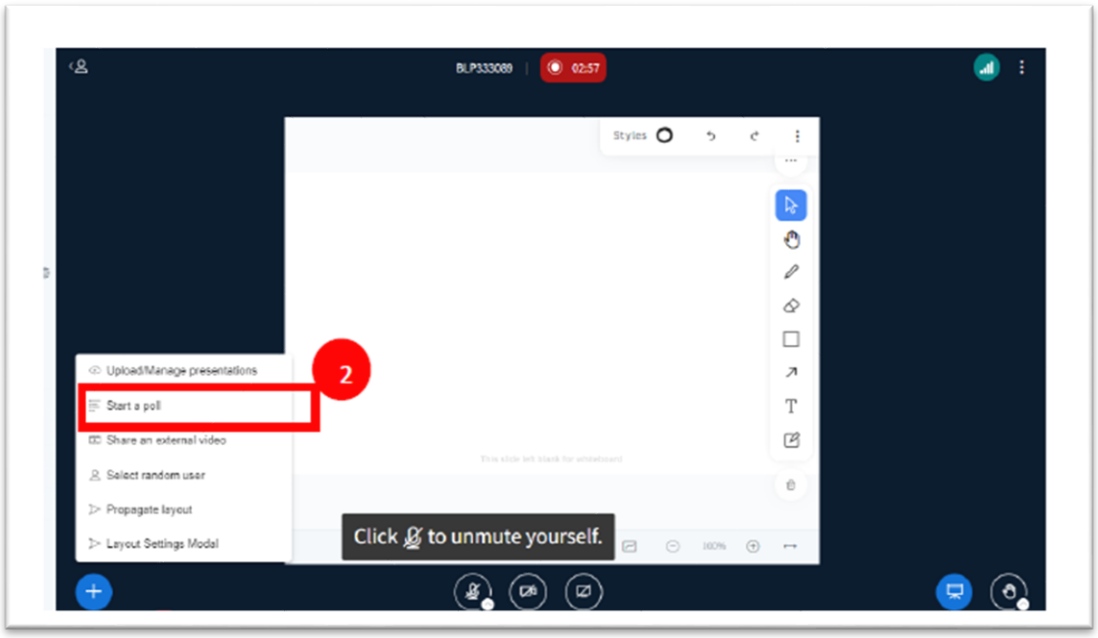


Step-1. Click on the “three dots”

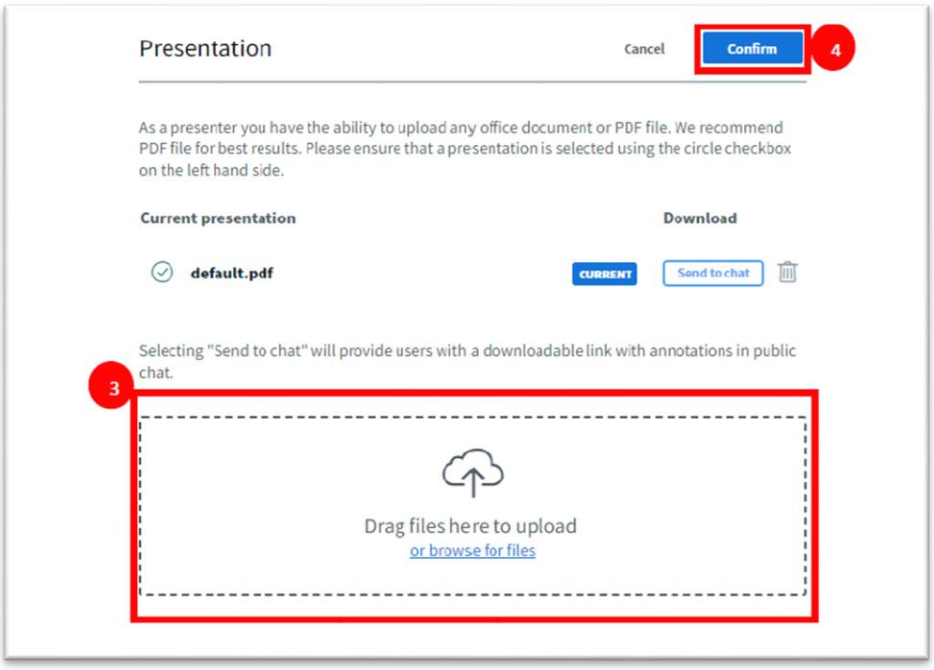


Step-2. Select ”Leave meeting”.

2.11 Upload a presentation.



Step-1. Click “+” in blue.
Step-2. Select Upload/Manage presentations.

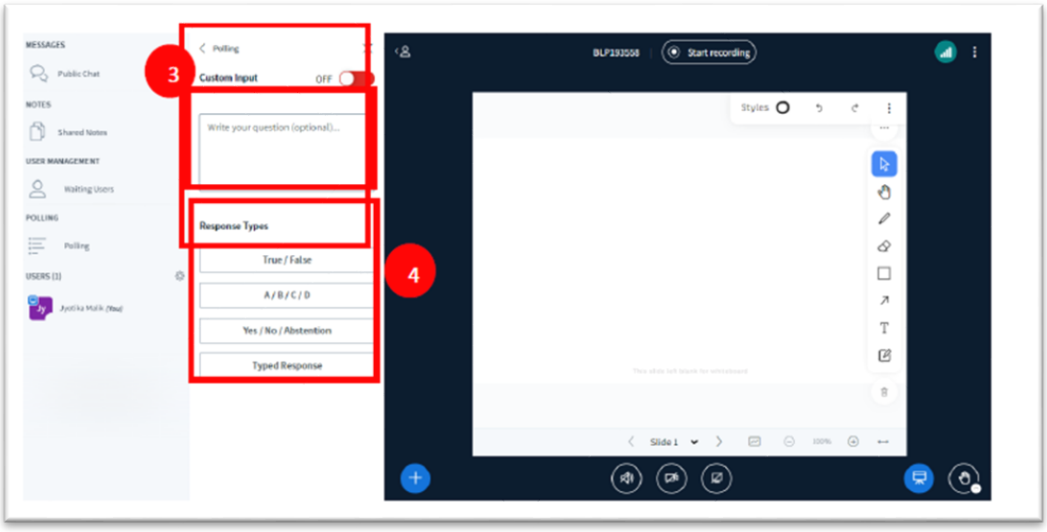


Step-3. Click on “or browse for files” or drag and drop the file in the given space.
Step-4. Click on “Confirm”.

4.12 How to start a poll?

Step-2. Select “Start a poll”.

Step-3. Type a question under “Write your question”.



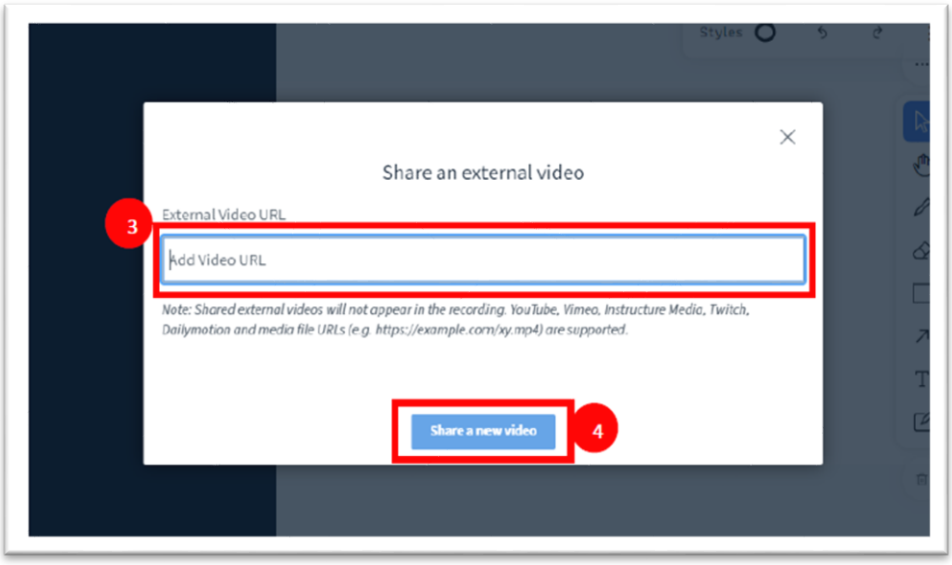
Step-4. Select “Response types”.

Step-5. Enter answers.

Step-6. Click on “Start Poll”.

4.13 Share an external video

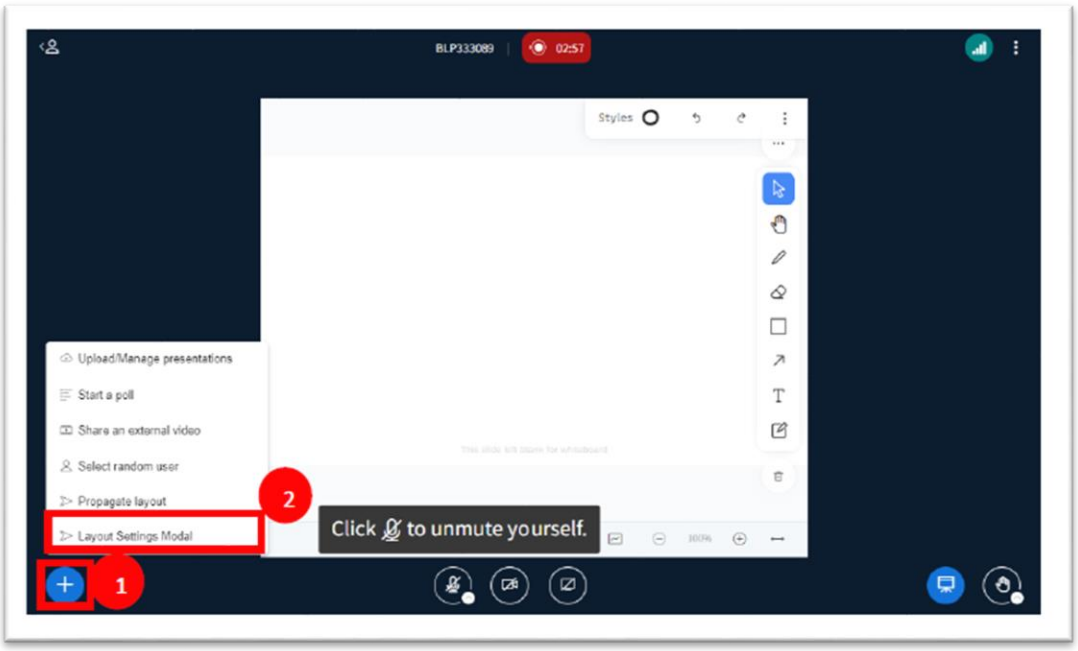
Step-1. Click “+” in blue.
Step-2. Select “Share an external video”.

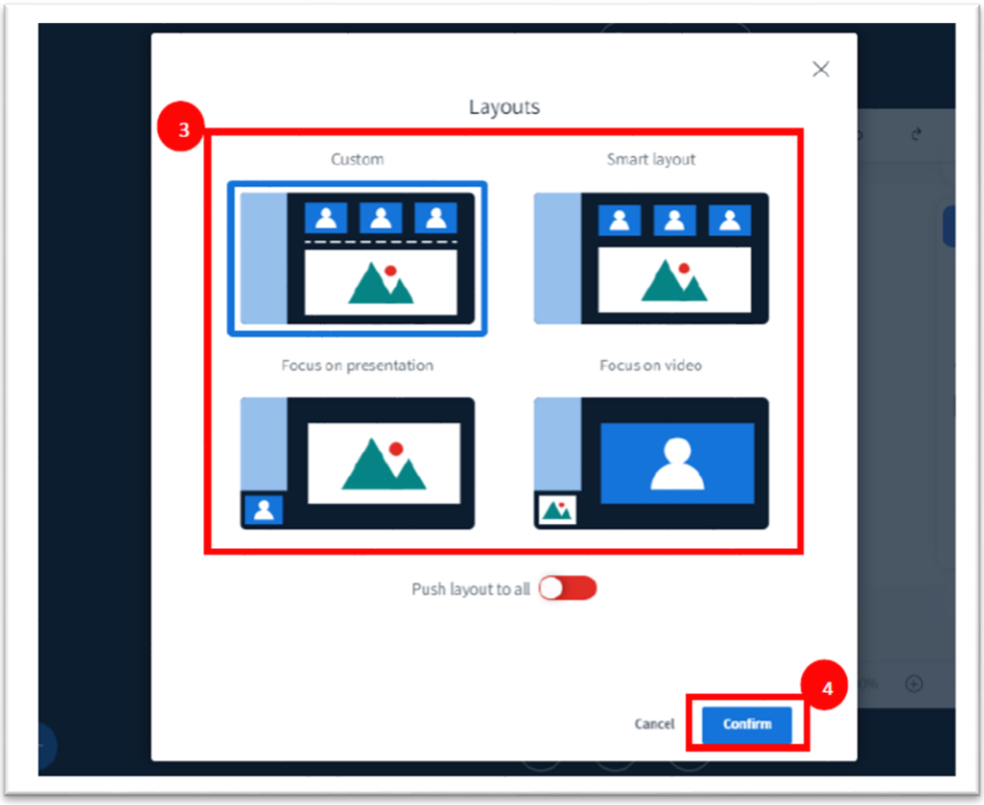


Step-3. Enter the URL, under Add Video URL.
Step-4. Click on “Share a new video”

4.14 Layout Settings

Step-1. Click “+” in blue.
Step-2. Select “Layout Settings Modal”.





Step-3. Select the layout.
Step-4. Click on “Confirm”.

Development of Courses in Faculty Module of NARES-Blended Learning Platform

Course module is visible from both faculty and student's profile. This module is to facilitate development of courses for the faculties and students can only see the courses developed by faculties but cannot create them. Course module is divided into three different segments: Category, Course Library and Course Topic.

If you are logged in as a faculty/ trainer in the BLP software, then the following functionalities can be created from your profile:

a) Create courses with additional resources (i.e. syllabus, documents, videos)

b) Can set enrolments cancellation deadline for individual courses.

c) Students can be assigned teaching materials in the form of videos for individual courses.

d) Can create courses with specific goals and milestones.

e) Create question bank comprising different questions.

Individual courses/ set of courses come under a set of categories. Universities are required to define their own set of categories according to their requirements. These categories will be defined from the admin account designated for respective universities. The categories menu is shown as following from the trainer's account:

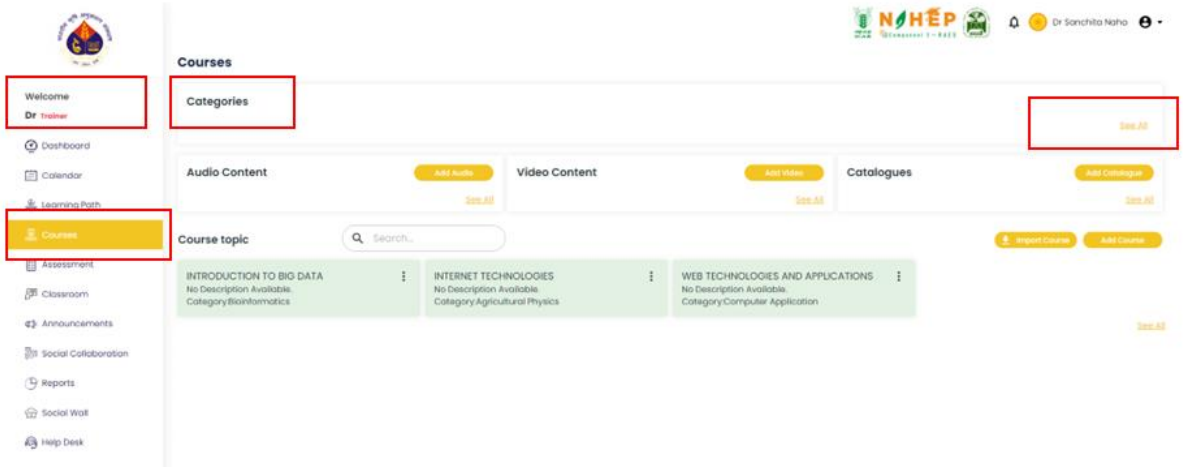


Figure 1: Screenshot of categories option under 'Courses' from faculty's account

If any faculty has already assigned courses in different categories, then by clicking on the 'See All' option in the right hand side of the screen one can see those categories, courses under each category and students enrolled in each courses as shown in the Figure 2.

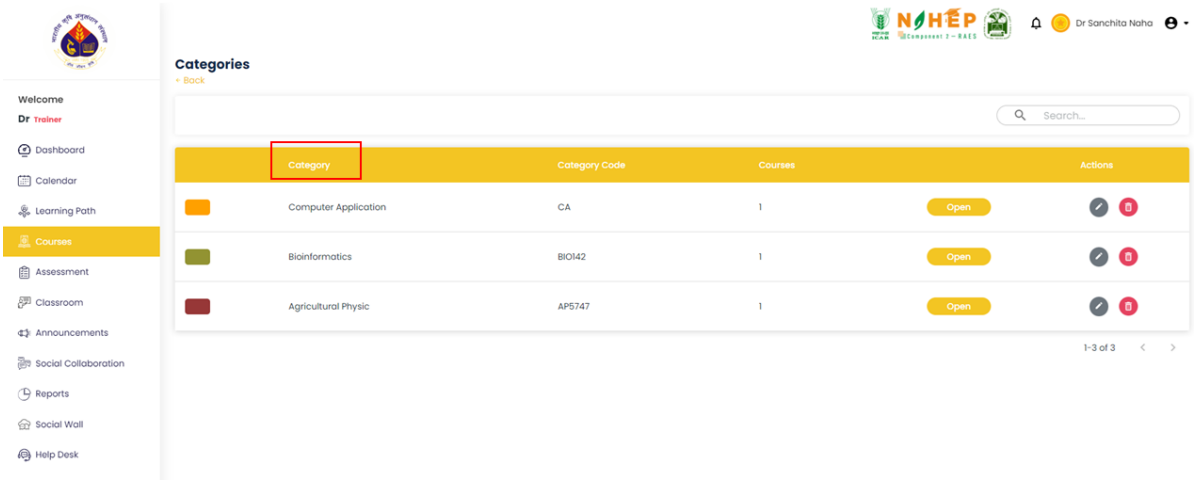
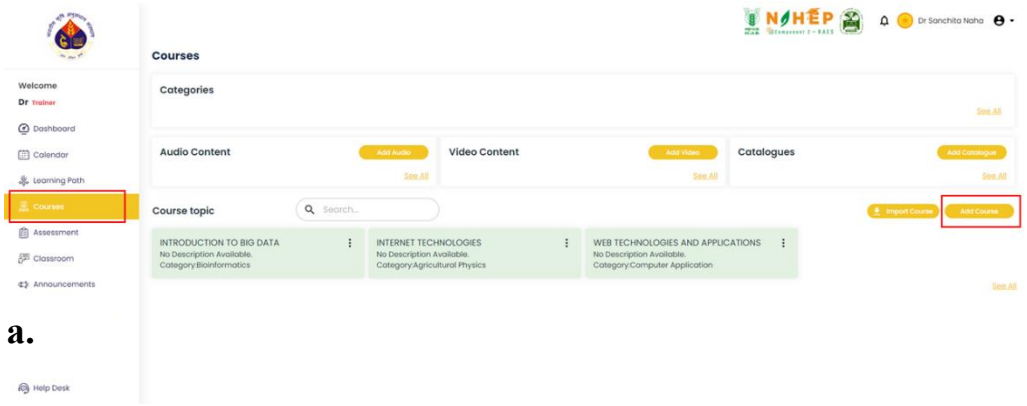


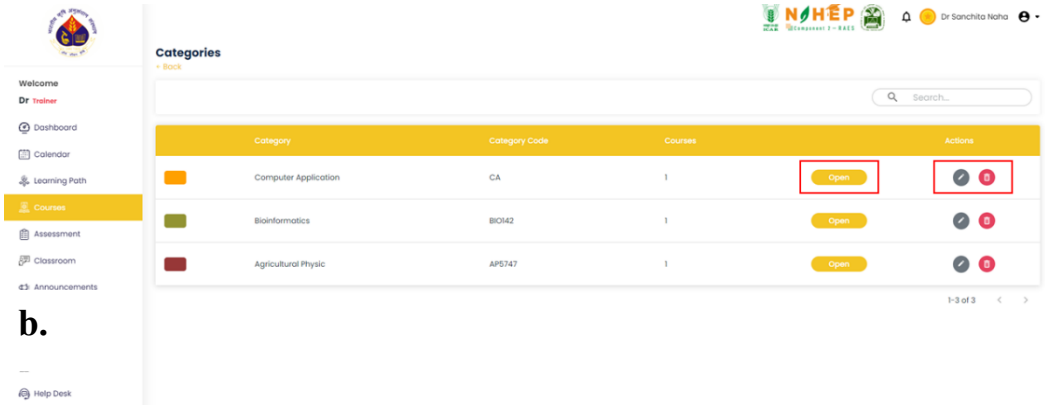
Figure 2: Screenshot of assigned list of courses under different categories

To start with adding a new course or edit any of the courses from the list shown as above under ‘categories’ option there are three options as shown below:

Option 1: Under ‘Courses’ menu click on ‘Add Course’



Option 2: Click on ‘Open’ or ‘Edit’ option under ‘categories’ menu



Option 3: Under the ‘Course Topic’ option click on ‘Edit Course’

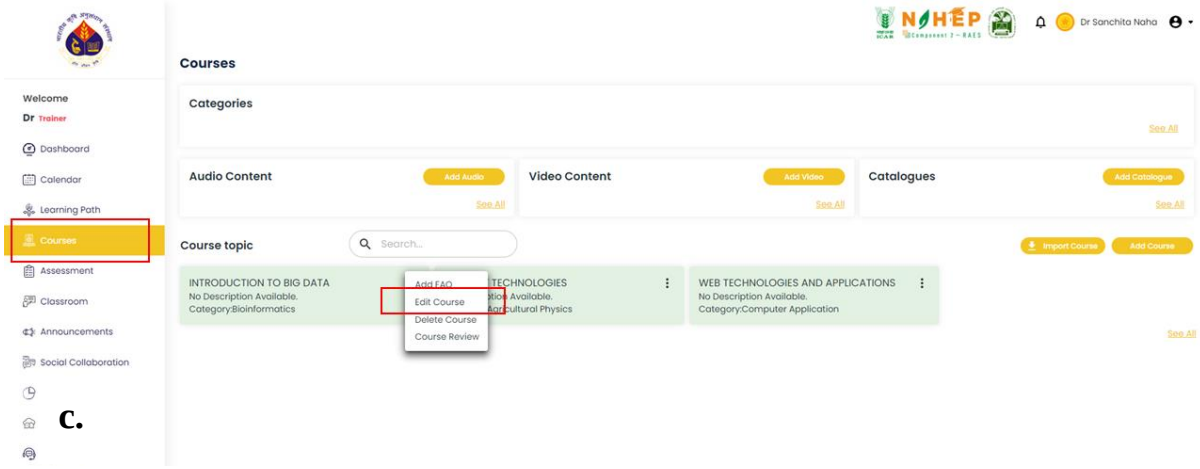
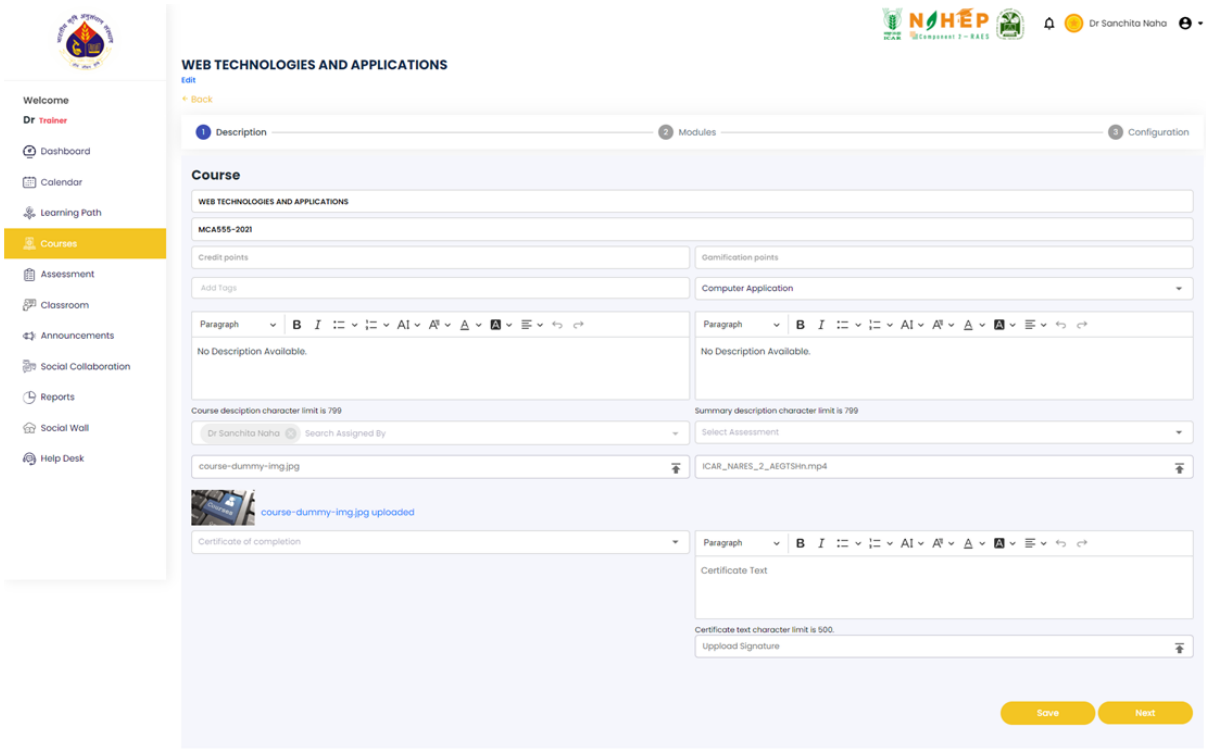


Figure3: Screenshots of options to create new courses or edit existing courses: a. Screenshot of 'Add Course' from 'Courses' menu; b. Screenshot of 'Open' or 'Edit' option under 'categories' menu; c. Screenshot of 'Edit Course' under 'Course Topic' menu

By following any one of the above 3 options, following screenshot will appear to add/ edit individual courses:



Fill the form with following relevant information:

a) Enter Course Name
b) Enter Course Code
c) Enter Credit Points for the corresponding course

d) Add relevant tags for the course
e) Add gamification points if it applies to the
f) Find the category from the drop down
g) Add description of the course
h) Add summary of the course
i) Add trainer
j) Select assessment, for which the trainer must create an assessment earlier
k) Upload any thumbnail image which will be shown next to the course
l) Upload a short introductory video
m) Add certificate
n) Add certificate text
o) Upload signature

Then click ‘save’ to save the content of the form which will let you stay on the same page, then click on ‘Save & Next’ option which will redirect you to the next page i.e. ‘Module page’.

After creating the course description, you need to add modules in the course. Step 1 – Click on ‘Add Module’ button as shown in below image.

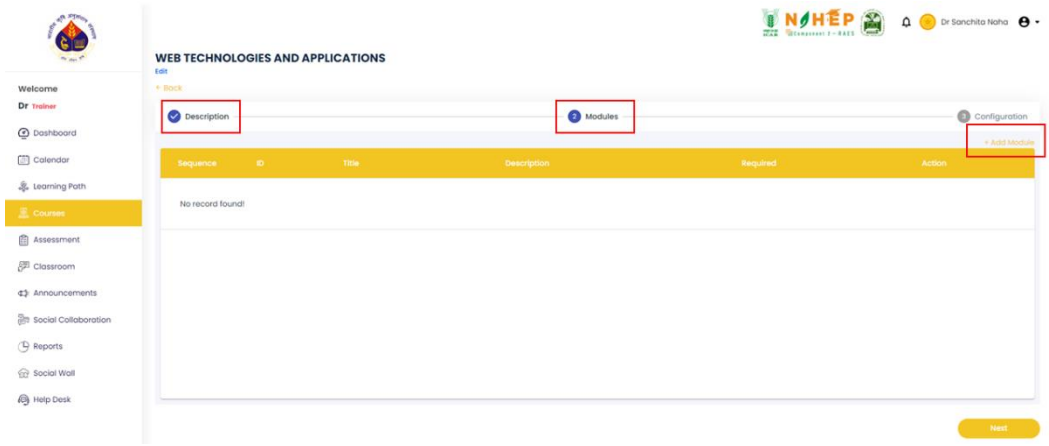


Figure 5: Screenshot to Create new module by clicking on ‘Add Module’

This will open a module description page which has to be filled in with the following information:
a) Write Module name.
b) Add Module description.
c) Enter Assessment URL.
d) Upload files.
e) Click on checkbox to make the module required

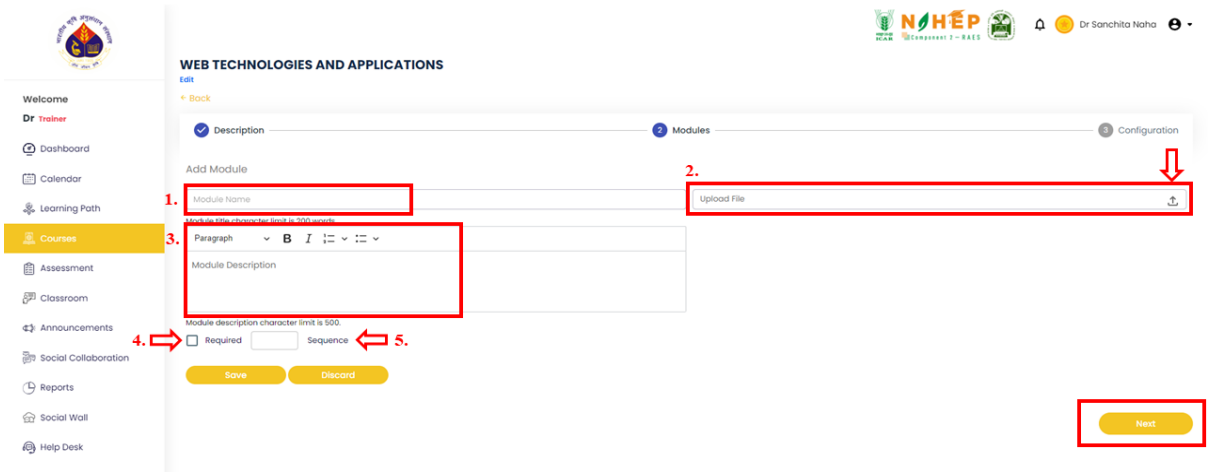


Figure 6: Screenshot to fill Module Description page from 'Add Module' option

Once you click on Upload file, below screen will appear where you upload files of different types e.g. Quiz, SCORM, Videos, Audios, Docs, PPTX, PDF, Survey.

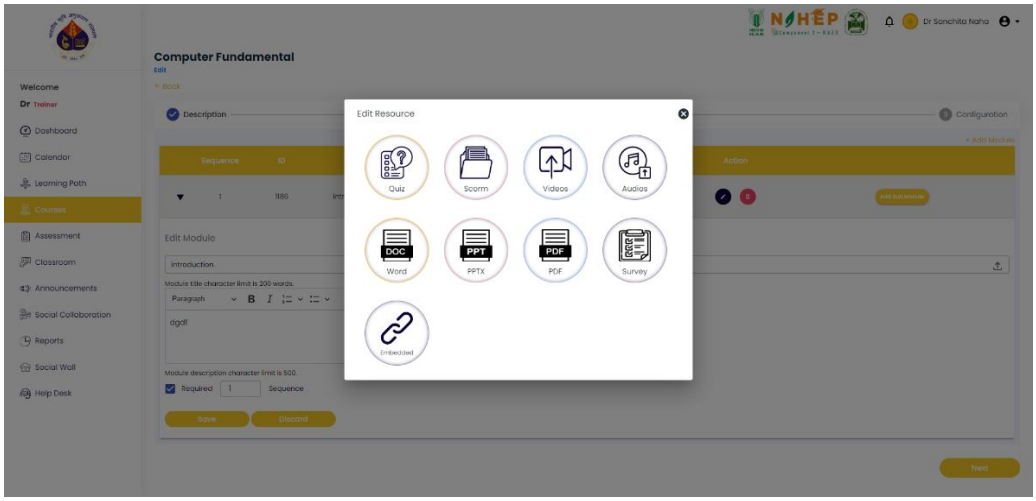


Figure 7: Screenshot to select file format from Upload File option in Add Module page

For example, if you want to upload a PDF file, select the PDF option. The following screen will appear.

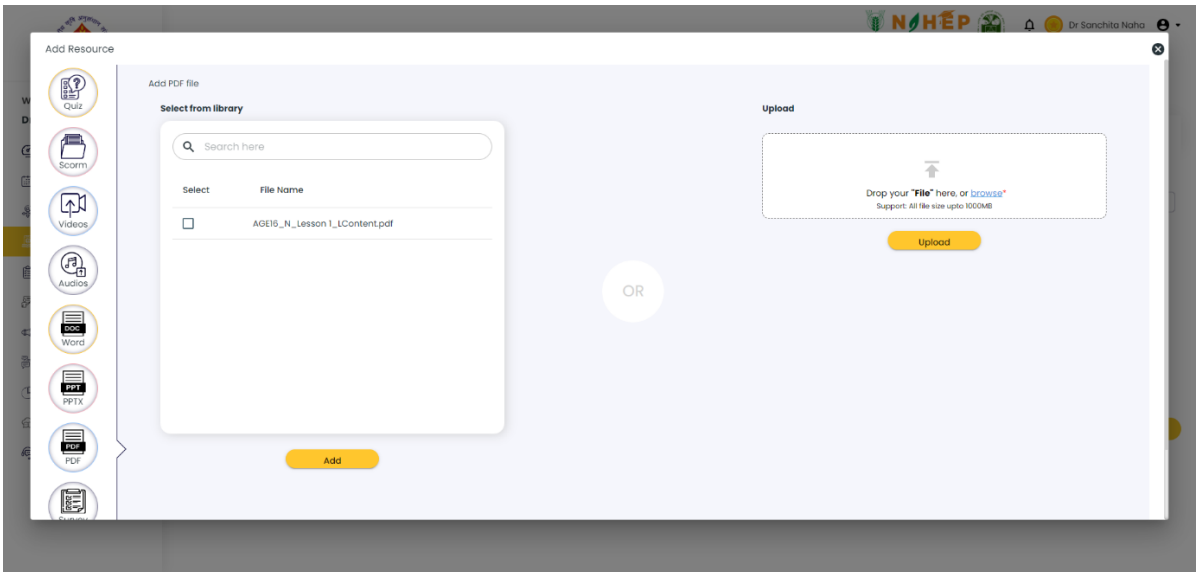


Figure 8: Screenshot to upload file from 'Add Module' page

One Module will be added as shown below. Repeat the process to add more modules. Repeat the same process to add more modules.

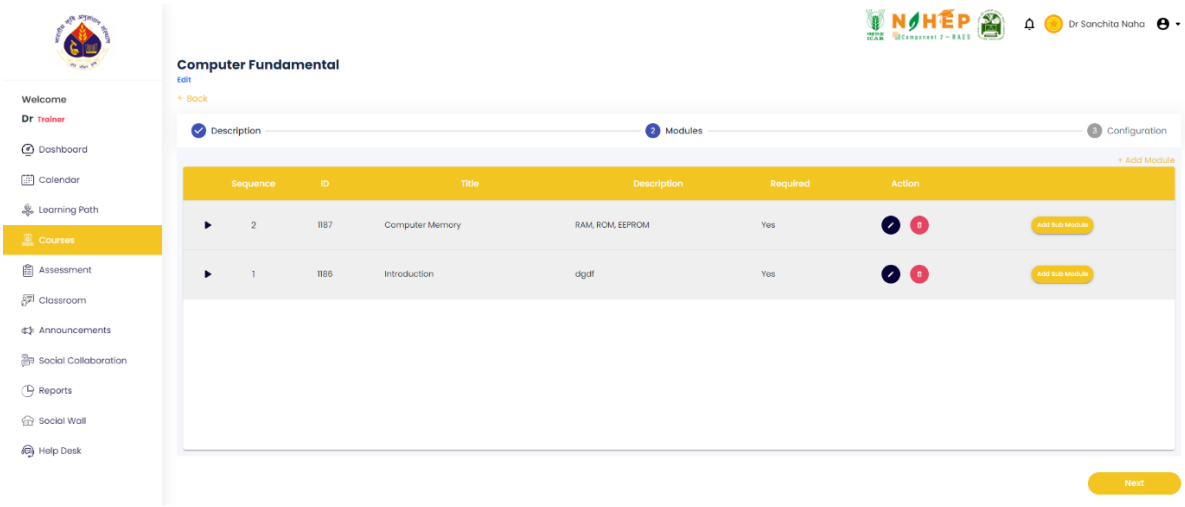


Figure 9: Screenshot of various modules under one Course

Click on the 'Next' button to go to the next page/screen which is '+Assign' to assign students. Click on 'Back' button to go to the previous page, 'Edit' icon to edit the Module, click on 'Delete' icon to delete the Module.

To assign students click on the '+Assign' button and a popup will appear to Select Organization, Department, Designation, Location and select based on the selection students name will appear. Click on the check boxes associated with the names of the students to select single or multiple learners. If you wish to select all students click on check box associated with 'Select All' option.

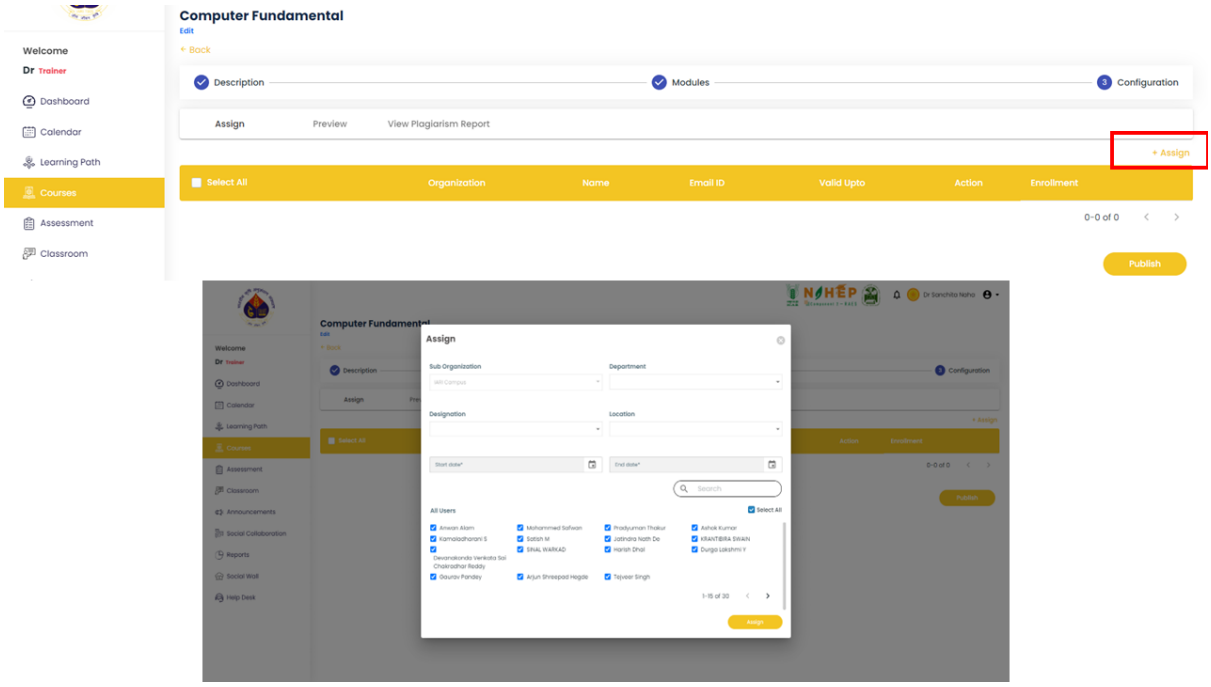


Figure 10: Screenshot to assign students by selecting Organization and Departments from '+Assign' option

Successful assignment of students to modules of the courses will look like the following:

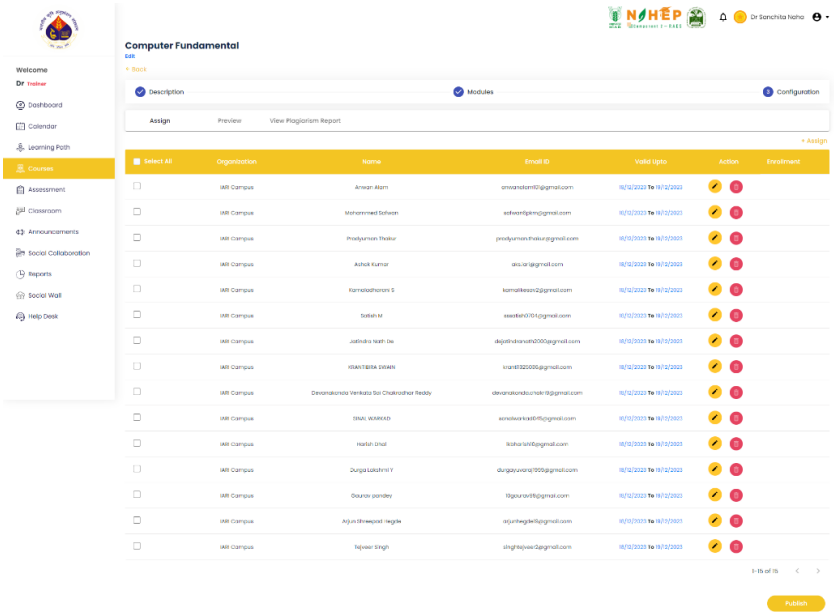


Figure 11: Screenshot of list of students assigned to modules of a course

To edit the list of students or dates click on edit icon. Follow these instructions as per requirement:

a) Click on 'Edit' icon to edit dates for students.
b) Click on 'Delete' icon to delete the details of the students
c) Click on 'Back' button to go back to the previous page.
d) Click on 'Publish' button to publish the course.

If you click on the 'Publish' button, it will prompt you to confirm as following:

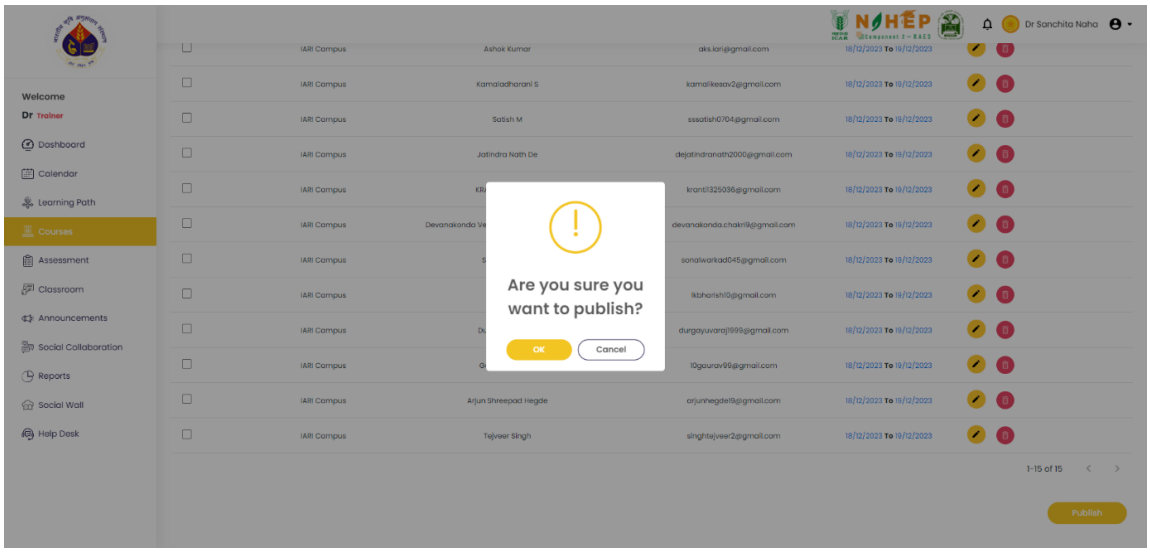


Figure 12: Screenshot to confirm before publishing any course or modules of a course

Click 'Ok' to publish the course and a success message will appear as shown below.

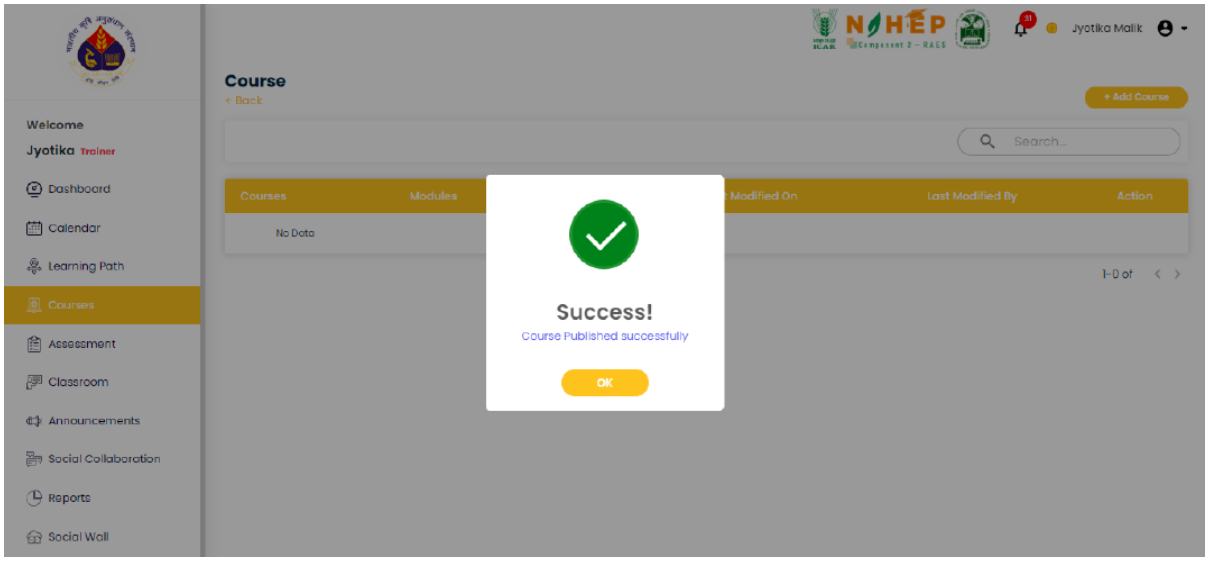


Figure 13: Screenshot to show successful publishing of one course

After you successfully publish a course Click on 'Preview' option under the 'Courses' menu to see the course preview. Scroll down a below and you will be able to see course description, Content and FAQs are available.

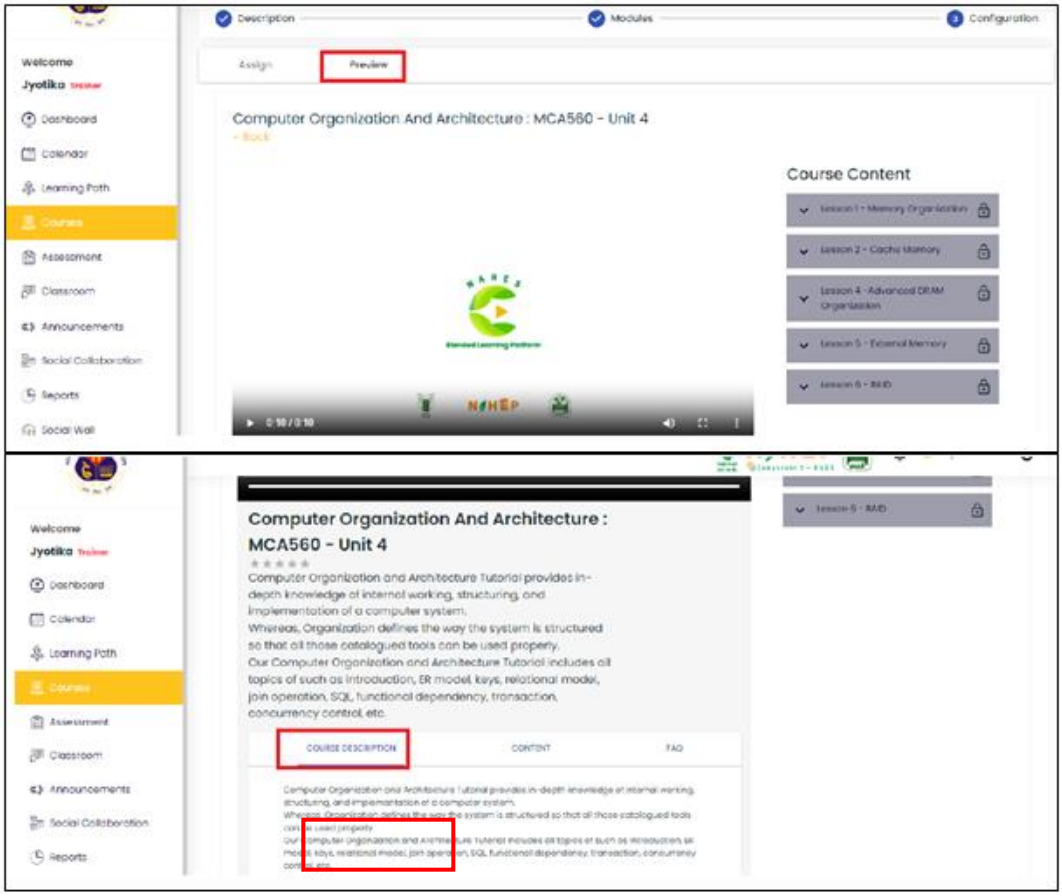


Figure 14: Screenshot of 'Preview' option under 'Course' menu

Click on 'Course Description' to see the description of the course as shown in below image. Click on 'Content' to see the course content as shown in below image. Click on 'FAQ' button to see the Frequently Asked Questions as shown in below image. Apart from creating and publishing a course, you can add Audio, Video and Catalogues. Click on Add Audio/ Add Video/ Add catalogues from the below screen. Click on Upload button (shown in Figure 15), the audio will get uploaded. To see all audio files, click on see all. Follow the same steps for video and catalogues with 'Add Video' and 'Add Catalogues' option given next to audio menu.

To edit and delete audio/ video or catalogues follow the same icons and steps as previously mentioned for adding/ editing modules. After making changes to any item, the system will prompt for confirmation, once you click 'Ok', changes will be reflected on the list of items shown.

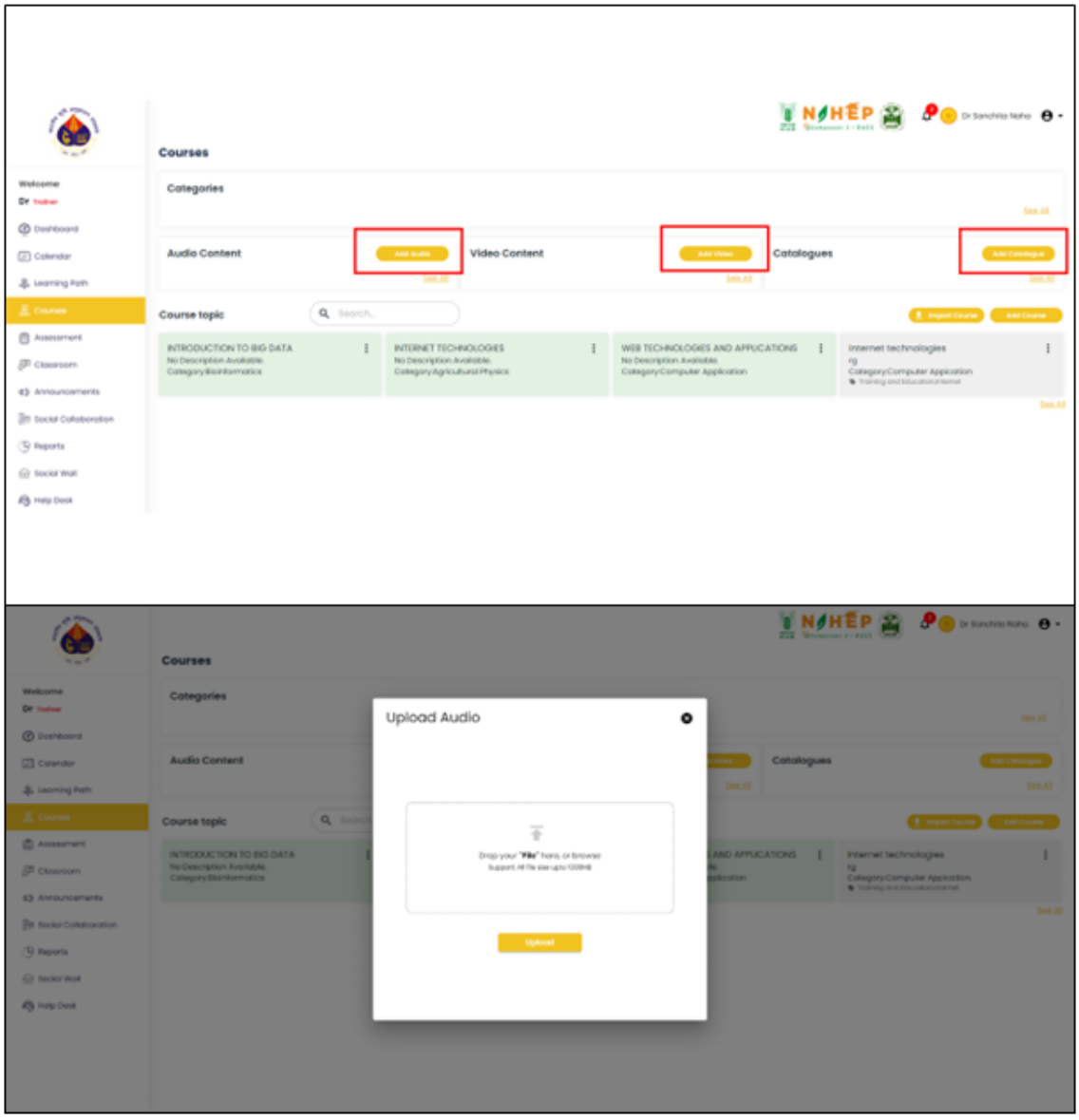
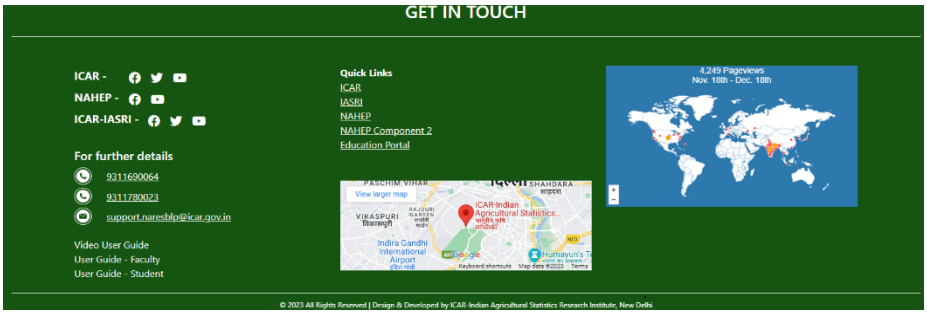


Figure 15: Screenshot to add audio, video, and catalogues

For more assistance, you can refer the instruction manuals given in the BLP website (https://naresblp.krishimegh.in/), at the bottom of the page as User Guide – faculty and User Guide – Student as shown below:



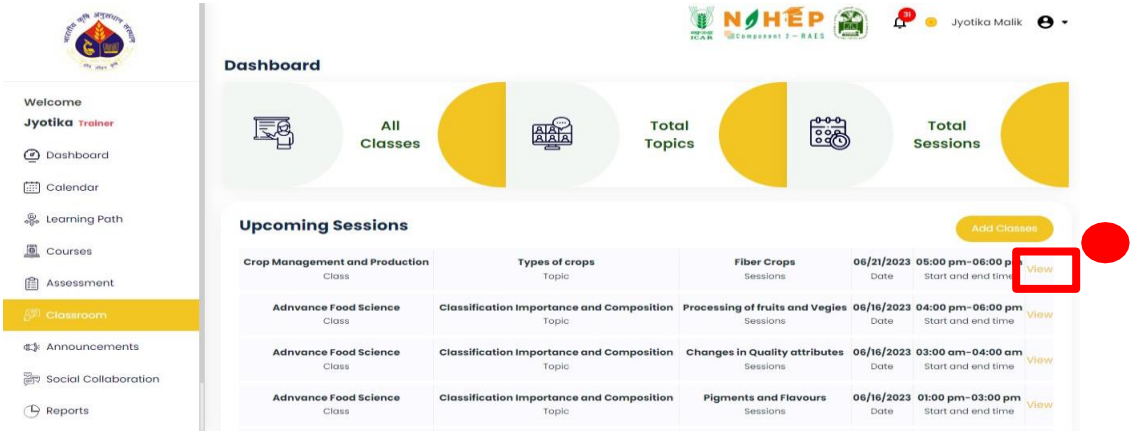
The classroom module is the functionality that enables Super Admin, Administrator, and Faculty to conduct classes, topics, and sessions. The class organizer (Super Admin/Administrator/Faculty) can also edit or delete the class. The class organizer can add credits and gamification to the topics created.

On the dashboard, every user can see a calendar with the event cards of the class or event assigned to the user. Users can also see all the classes, total topics and their scheduled sessions. On the dashboard all the on-going sessions is visible to the users. Users will have the privilege to conduct online classes through our inbuilt video conferencing feature. Also, users can conduct offline classes through this functionality.

The classroom module supports quizzes and surveys aligned with every module in a course. These surveys are conducted as feedback on the session and delivery of the session. Classroom modules also support assessment with continuous online procuring and easy question navigation for students.

View Class from the Dashboard

Users can view a listing of all the classes on the dashboard with details like Class, Topic, Trainer, Session, Session Date, Start Time, and End Time.

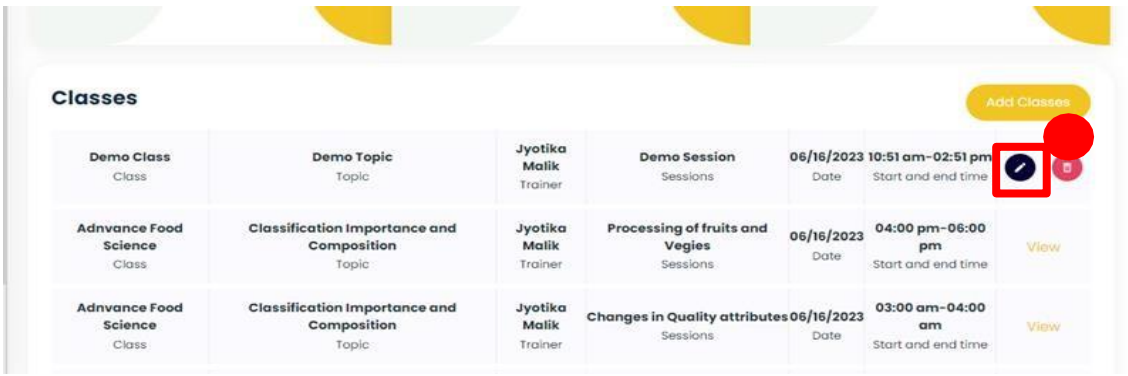


Step-1. Click on “View”.

If the session has occurred in the past, the Administrator user can view the class.

Edit class from the dashboard

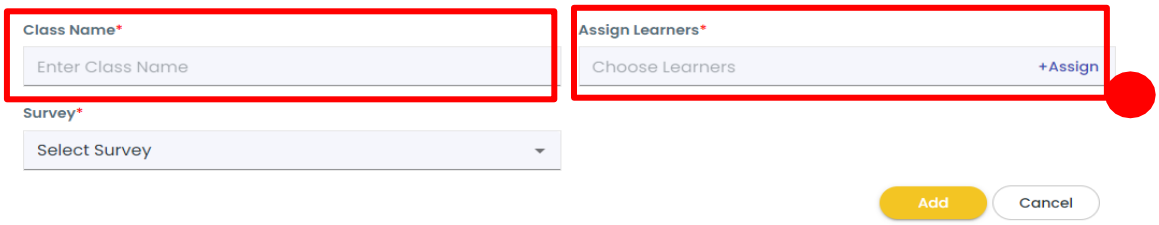
Users can edit upcoming sessions.



Step-1. Click on the “Edit Icon” associated with the class.

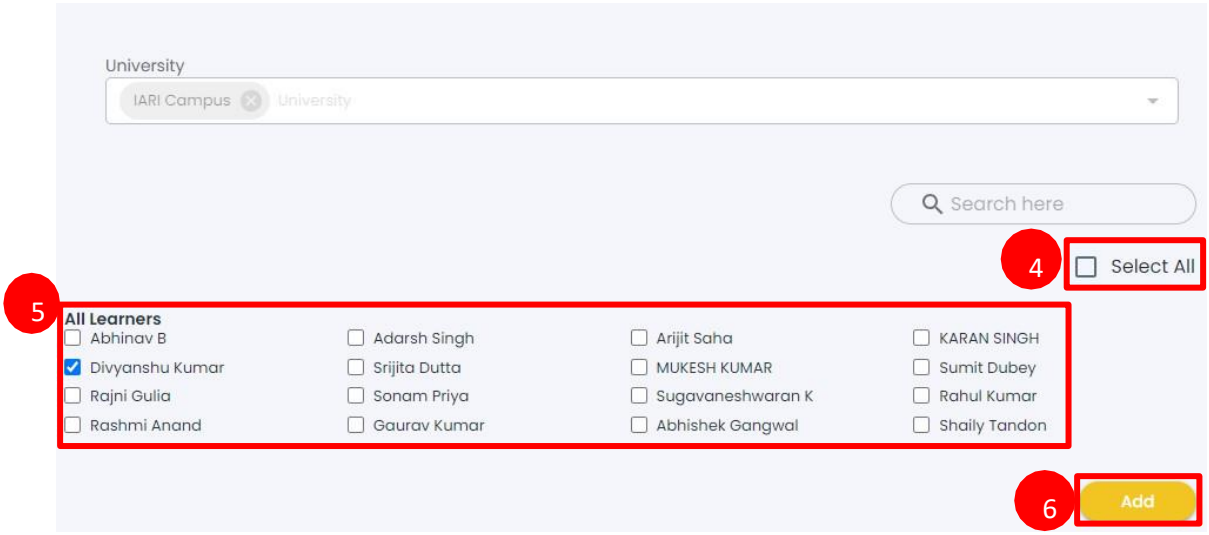
How to create a class?

Step-1. Click on “Add Classes”.



Step-2. Add “Class Name”.

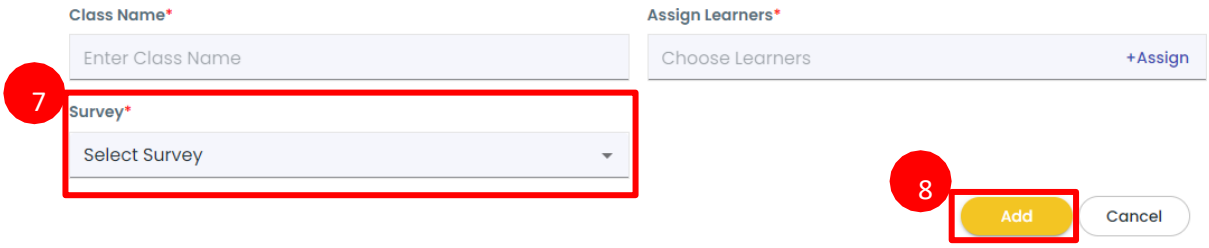
Step-3. Click on “+Assign” to assign users.



Step-4. Click on “Select All”.

Step-5. Click on the check box associated with the student’s name to select a student.

Step-6. Click on “Add”.

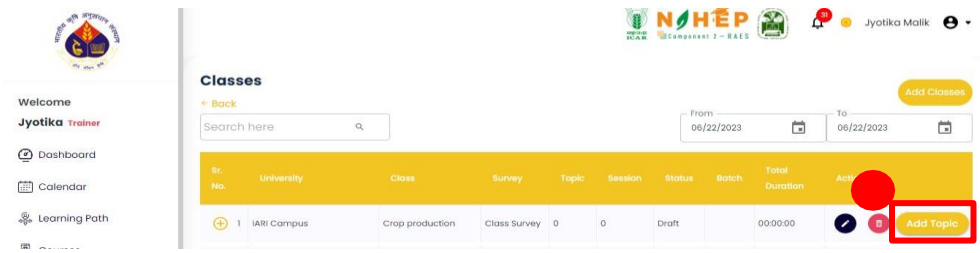


Step-7. Select “Survey” from the drop-down menu.

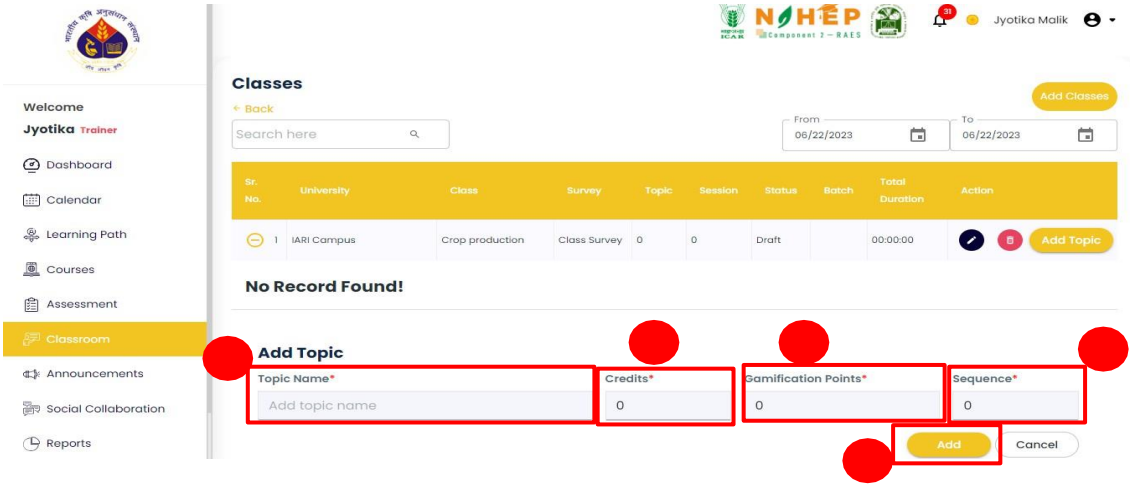
Step-8. Click on “Add”.

How to create a topic underclass?

Once the class is created Faculty can create a topic under the class. Faculty user will be able to see the screen with the complete listing of classes with the details like university name, class name, faculty name, survey name, no. of topics, no of sessions, the status of the class, learners, total duration, action (Edit/delete).



Step-1. Click on “Add topic”.



Step-2. Enter “Topic Name”.

Step-3. Add “Credits”.

Step-4. Add “Gamification Points”.

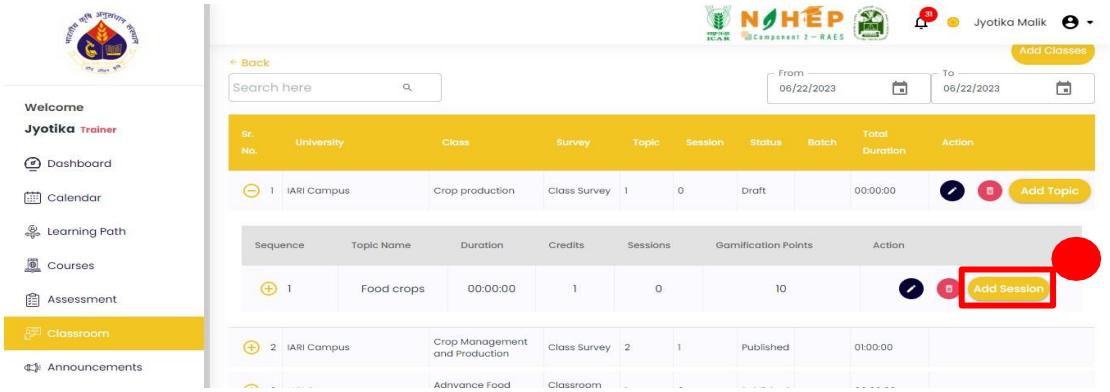
Step-5. Add “Sequence”.

Step-6. Click on “Add”.

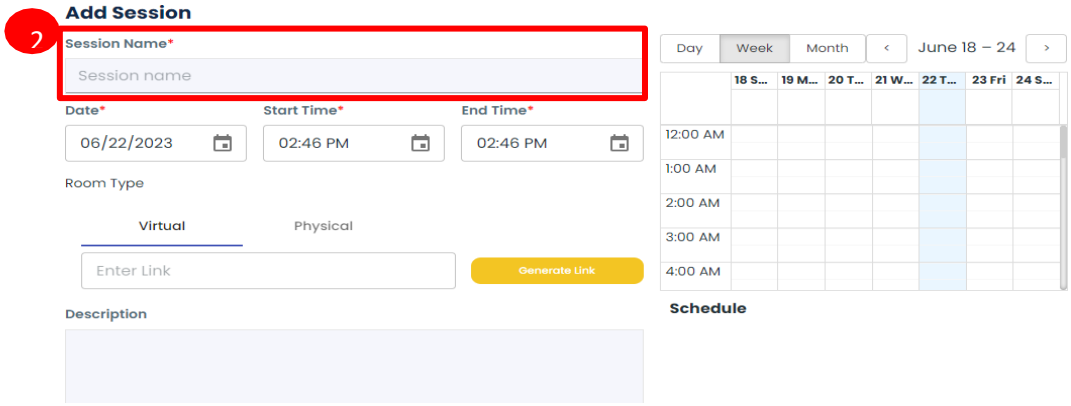
A success message will appear, “Topic created successfully”.

How to create a session under the topic?

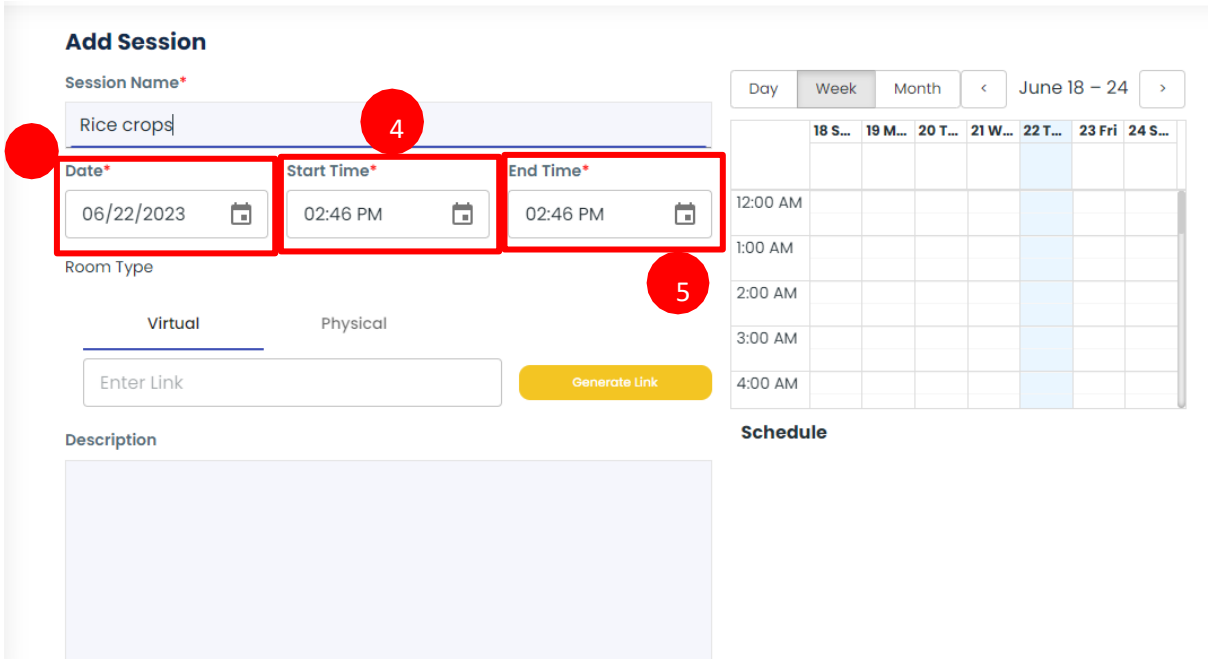
Once the topic is created, you can see the list of topics underclass.



Step-1. Click on “Add Session”.



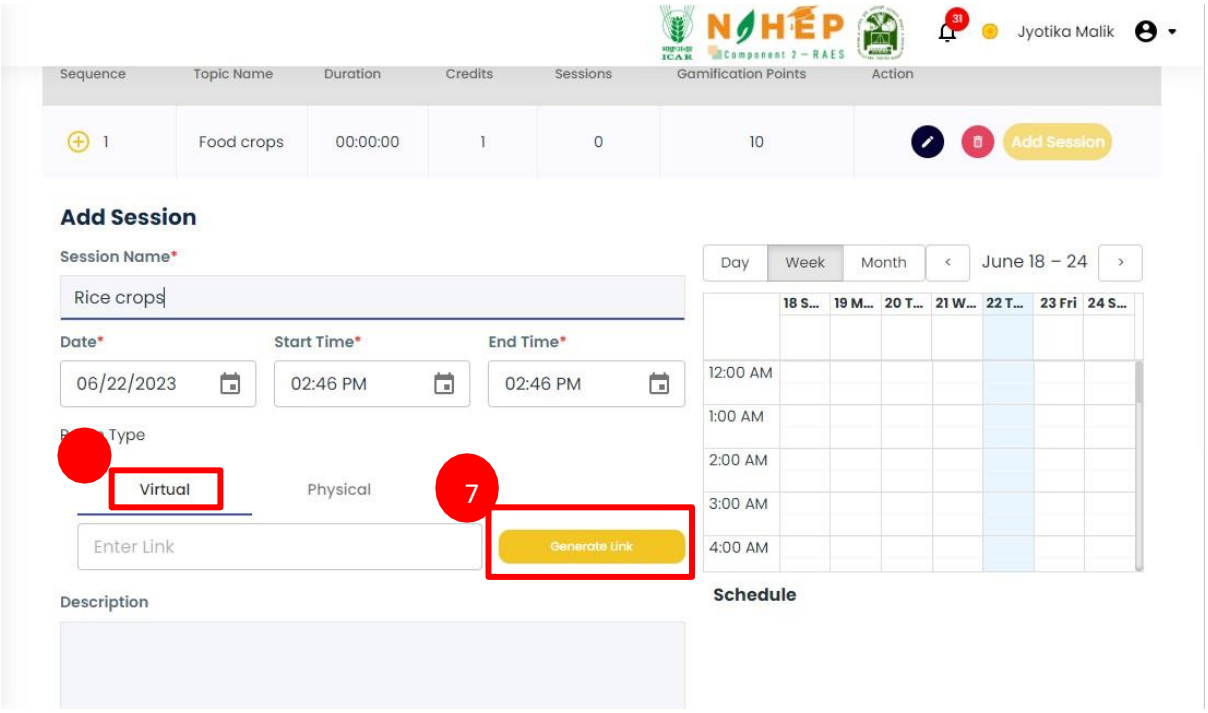
Step-2. Add Session Name



Step-3. Select “Date”.

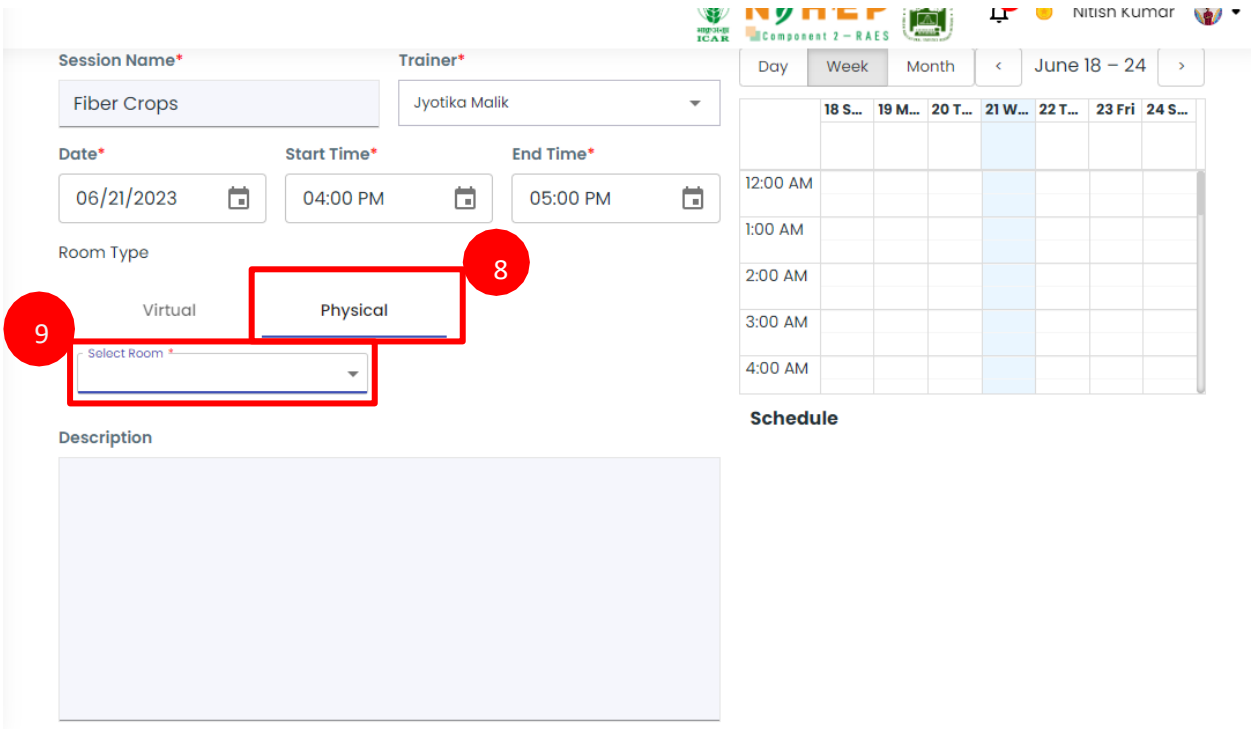
Step-4. Select “Start Time”.

Step-5. Select “End time”.



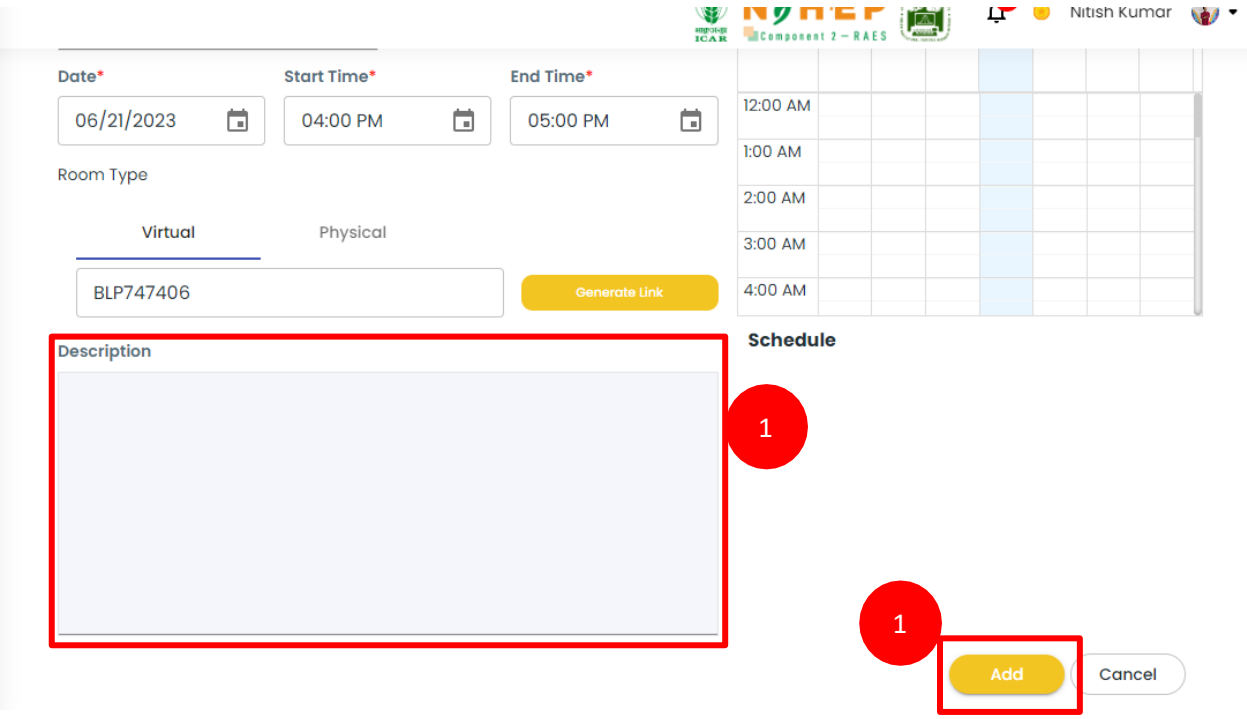
Step-6. Click on “Virtual” to create a virtual session.

Step-7. Click on “Generate Link”.



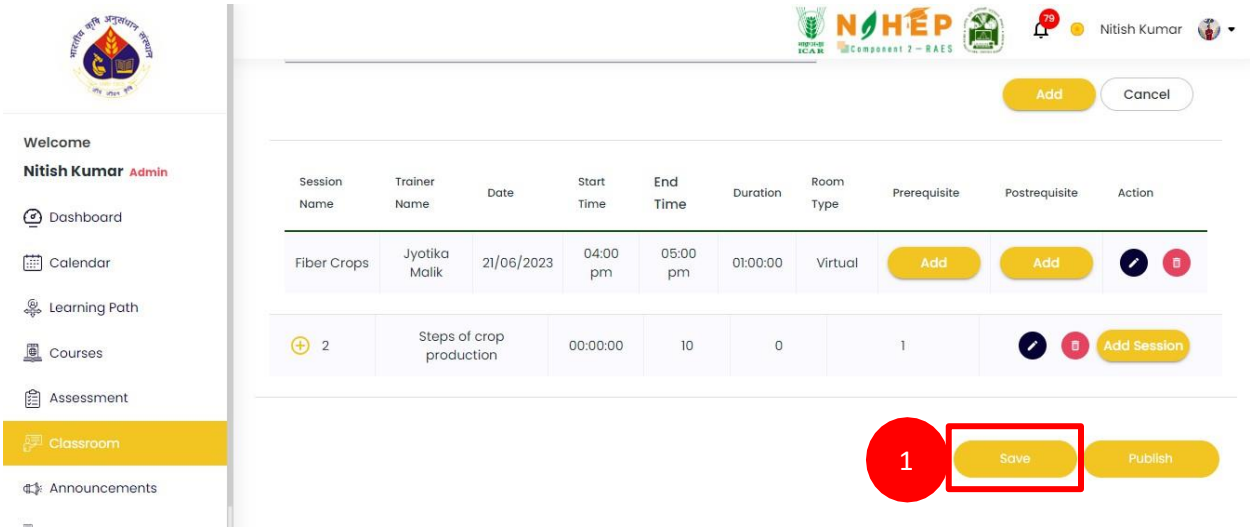
Step-8. Click on “Physical” to create a physical session.

Step-9. Select Room from the drop-down menu.



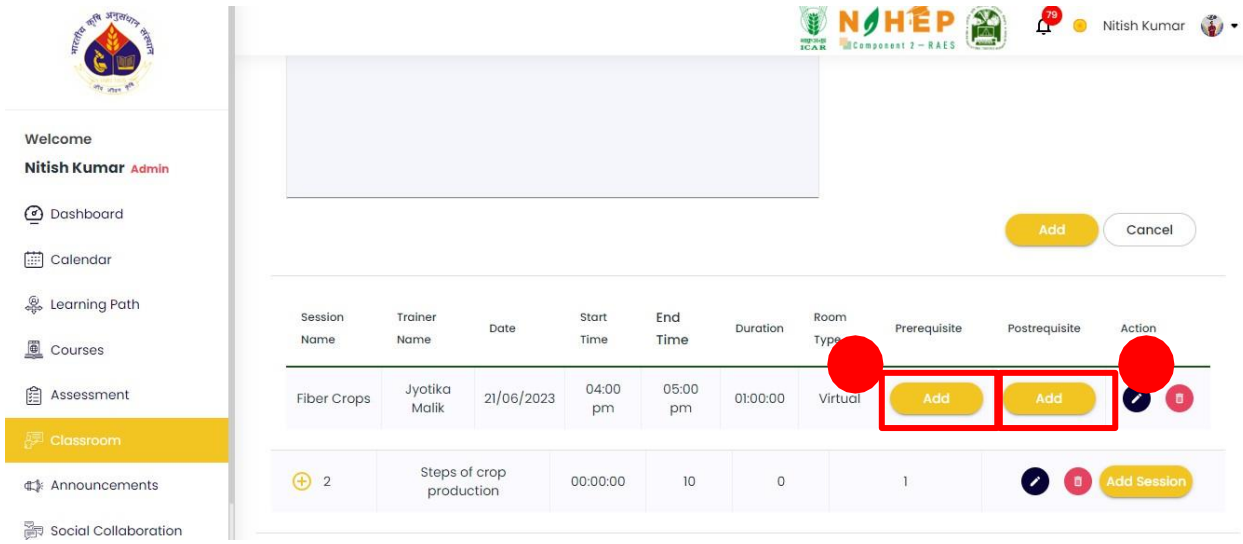
Step-10. Add “Description”.

Step-11. Click on “Add”.



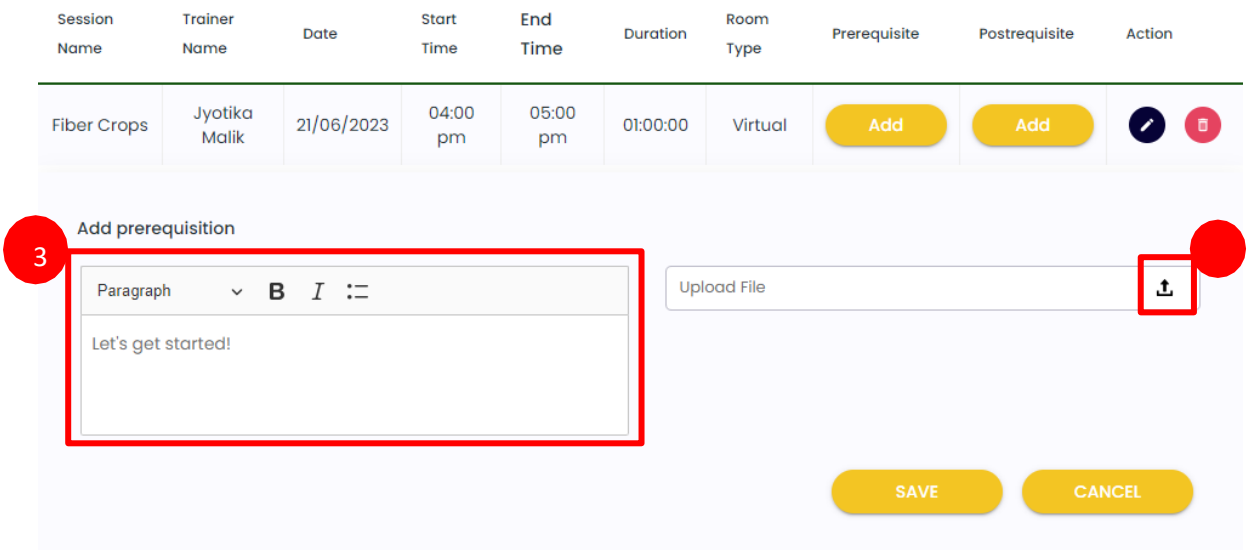
Step-12. Click on “Save”.

How to add pre-requisites and post-requisites?



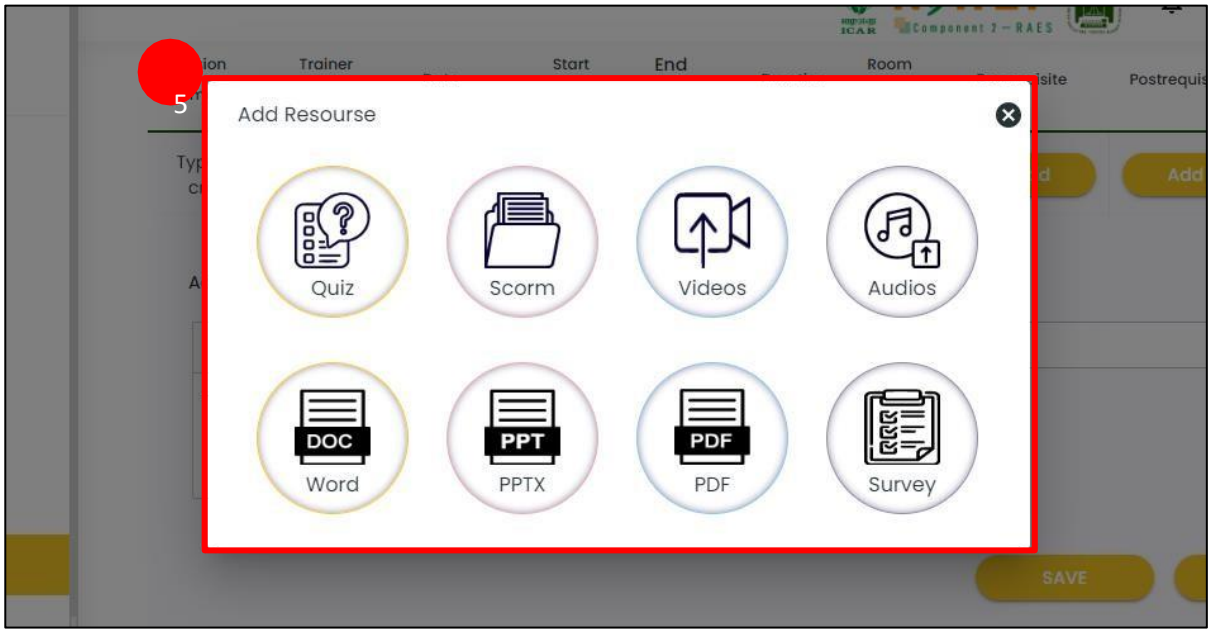
Step-1. Click on “Add” under Prerequisite.

Step-2. Click on “Add” under Post requisite.



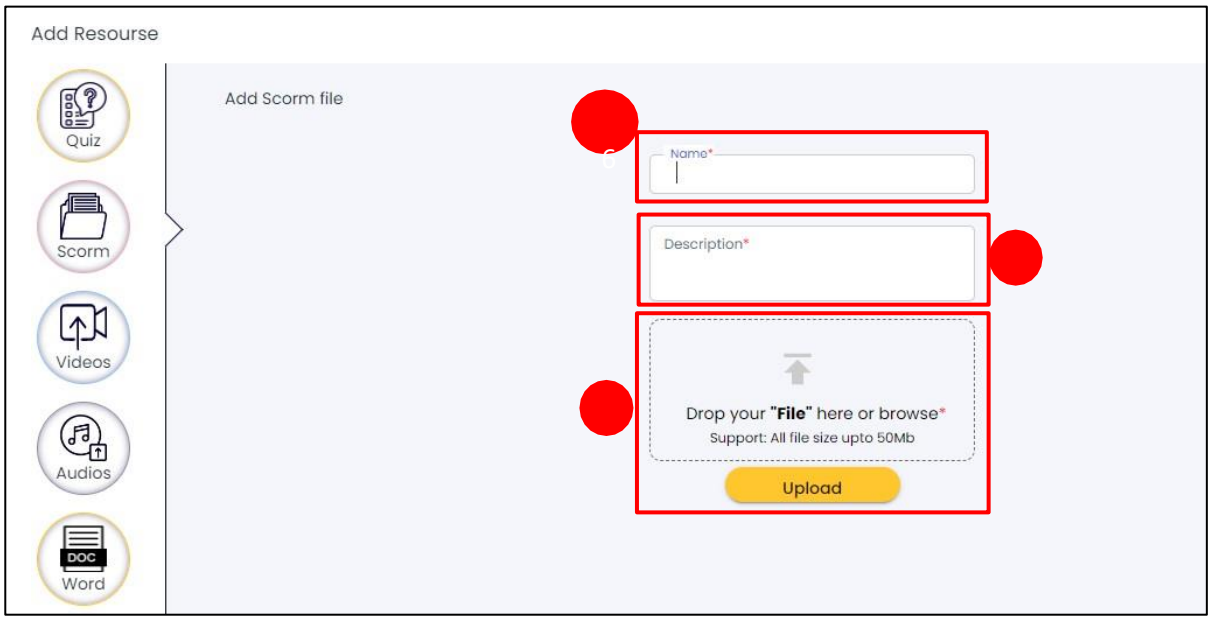
Step-3. Add Text under “Paragraph”.

Step-4. Click on “Upload Icon”.



Step-5. Select the file type. This feature supports files like:

- Quiz
- SCROM
- Videos
- Audios
- Word
- PPTX
- PDF
- Survey

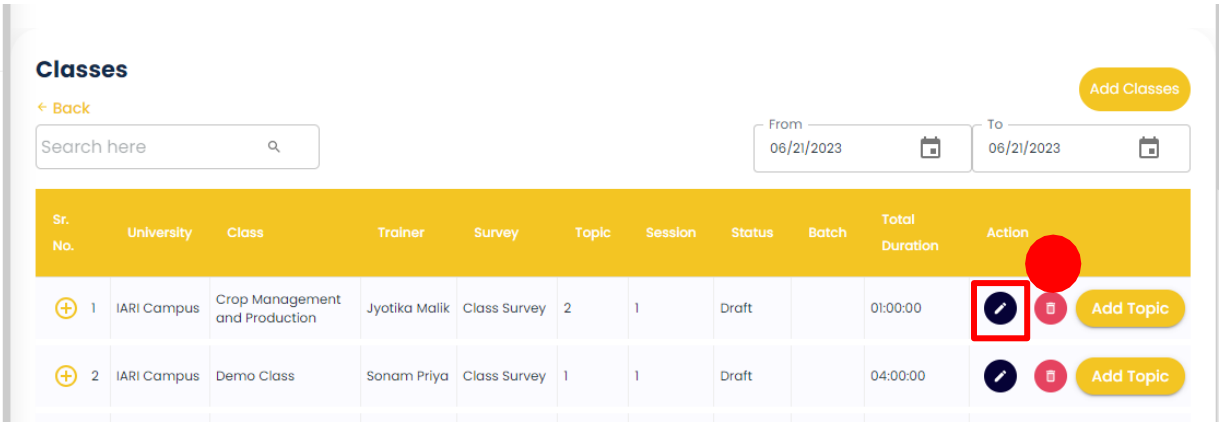


Step-6. Enter the “Name” of the file.

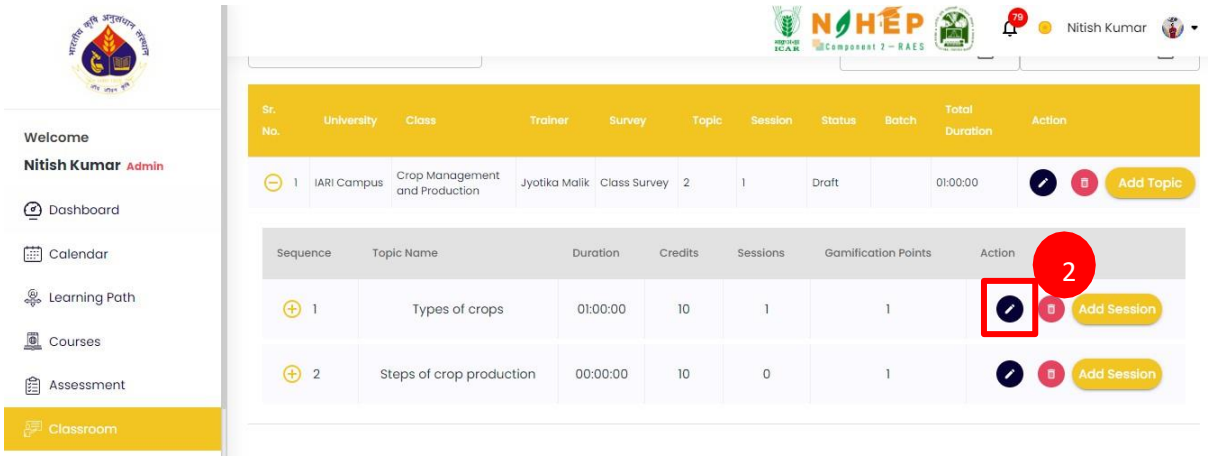
Step-7. Add “Description” to the file.

Step-8. Select the file and click on “Upload”.

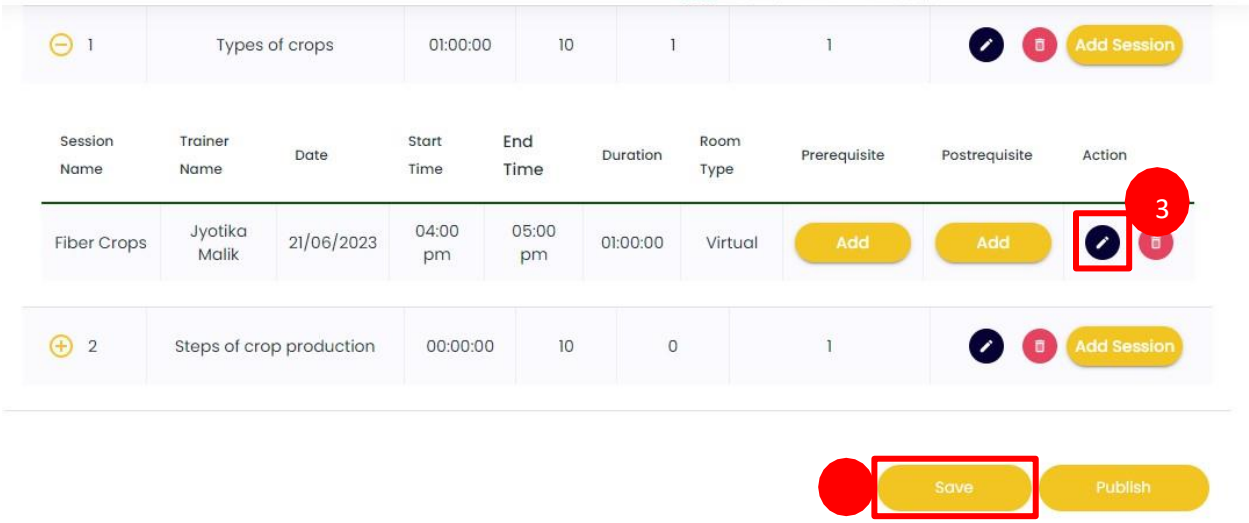
How to Edit a class, topic, or session?



Step-1. Click on the “Edit Icon” associated with the class.



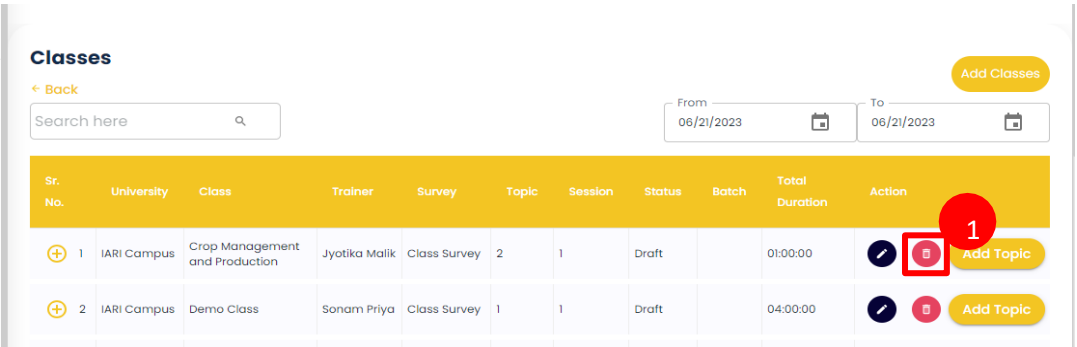
Step-2. Click on the “Edit Icon” associated with the Topic.



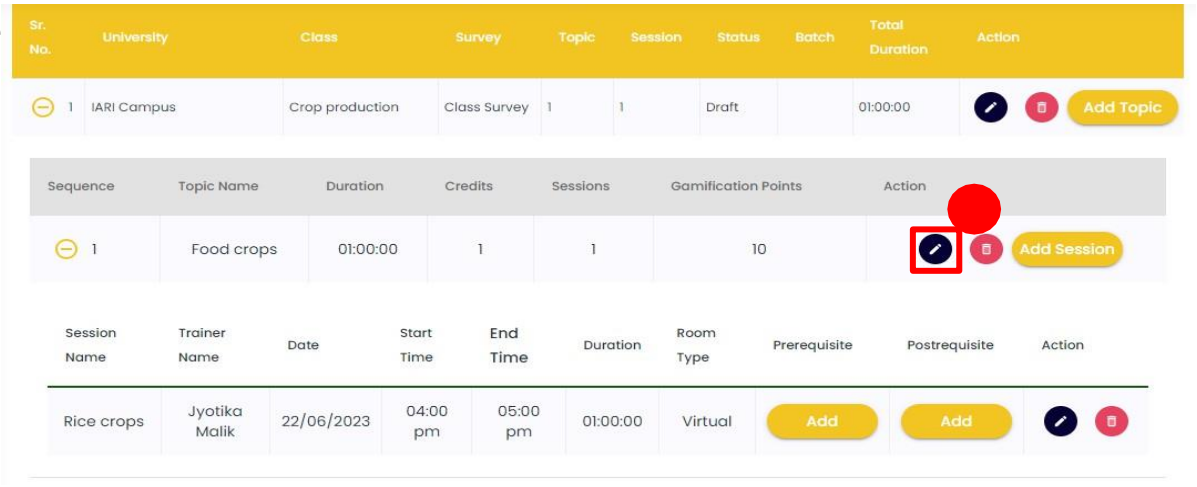
Step-3. Click on the “Edit Icon” associated with the Session.

Step-4. Click on “Save”

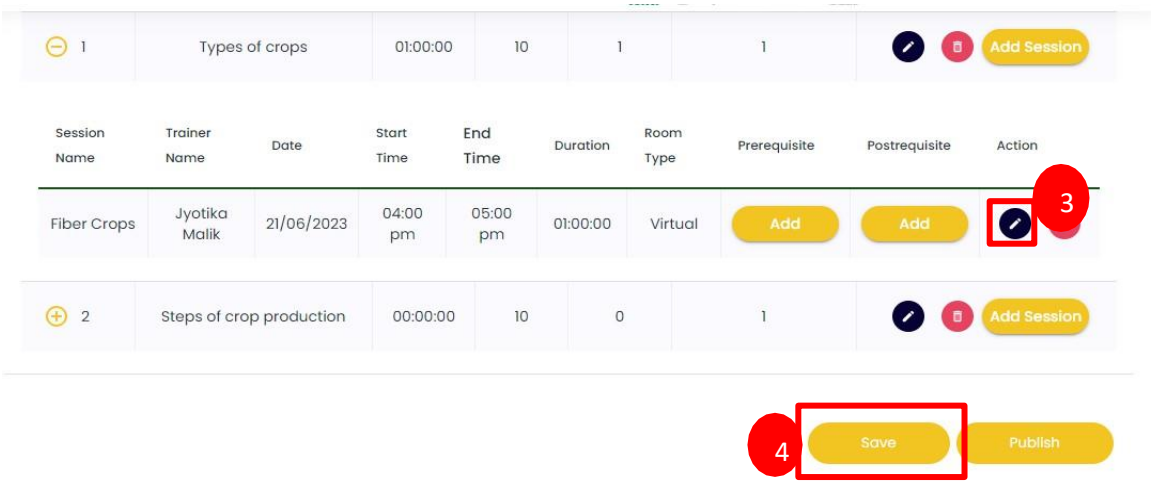
How to Delete a class, topic, or session?



Ste



Step-2. Click on the "Delete Icon" associated with the Topic.

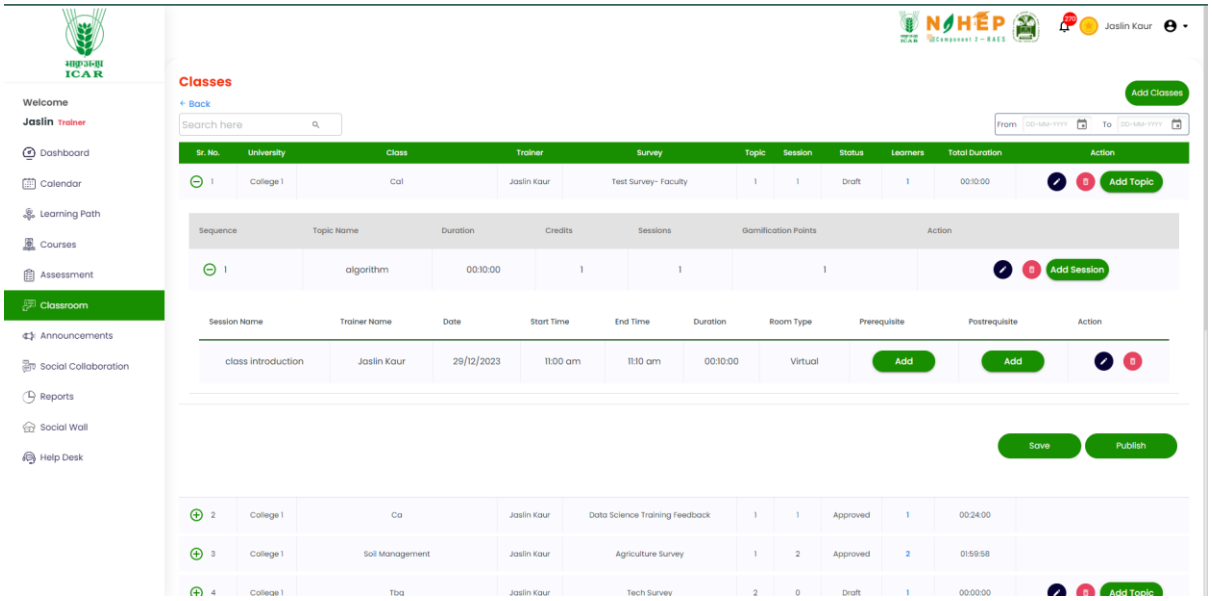


Step-3. Click on the "Delete Icon" associated with the Session.

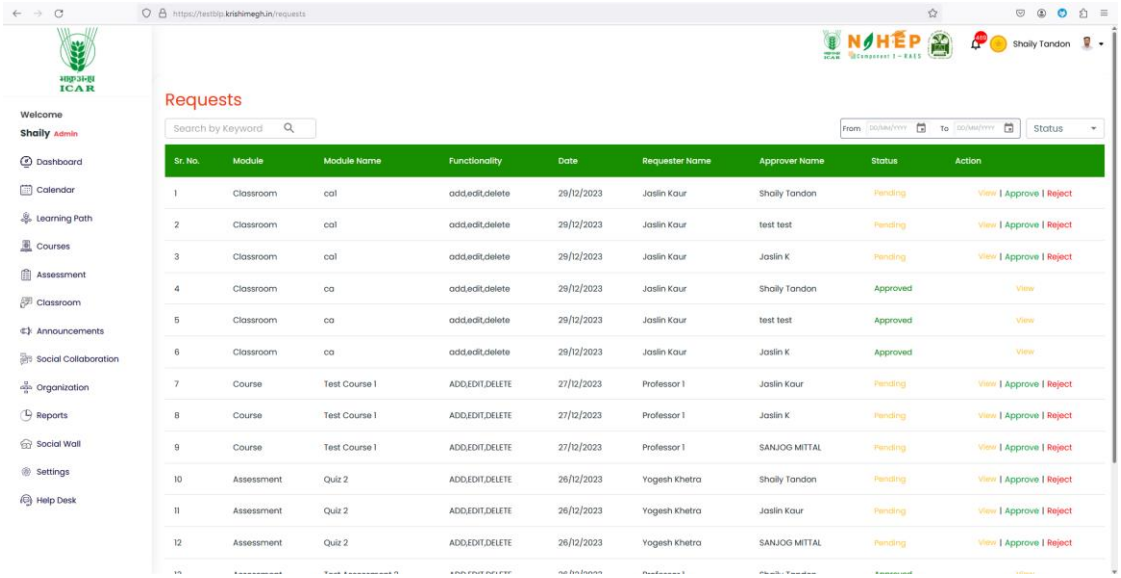
Step-4. Click on "Save".

How to Publish a SESSION?

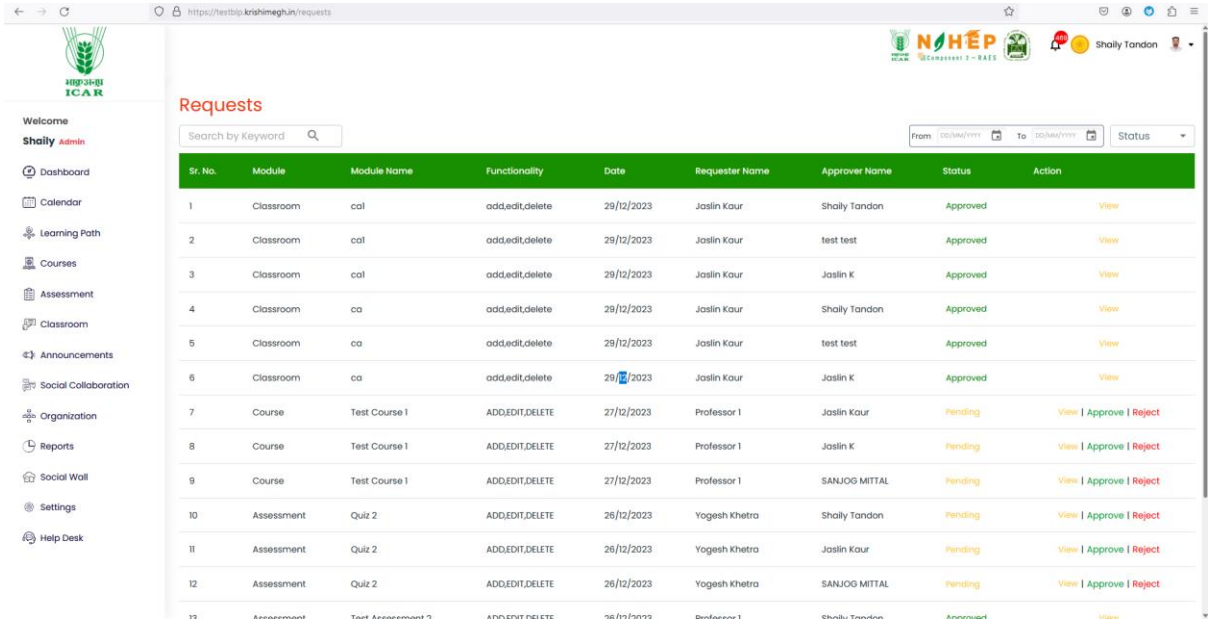
After Creating class, we have to add topic and then add session. After adding all the sessions, we have to publish the class. After publishing you have not permitted to change the class schedule. Before publish add all the topics and sessions to it. Then publish the class by clicking on "Publish icon".



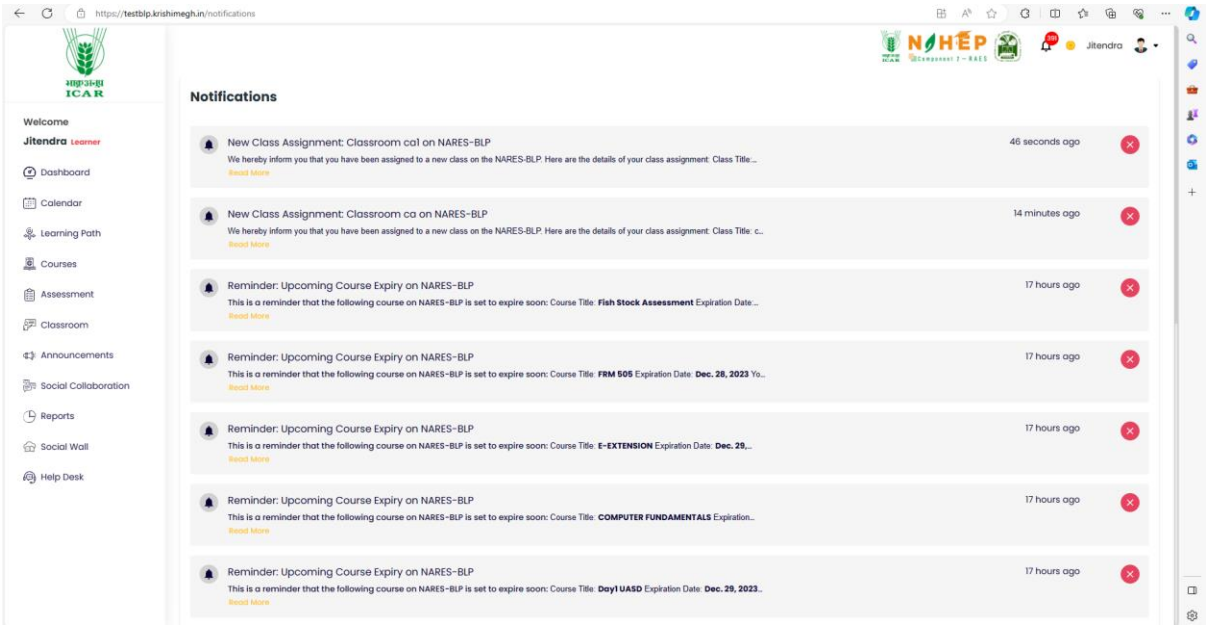
Then on the admin portal under requests Admin get the request for class approval.



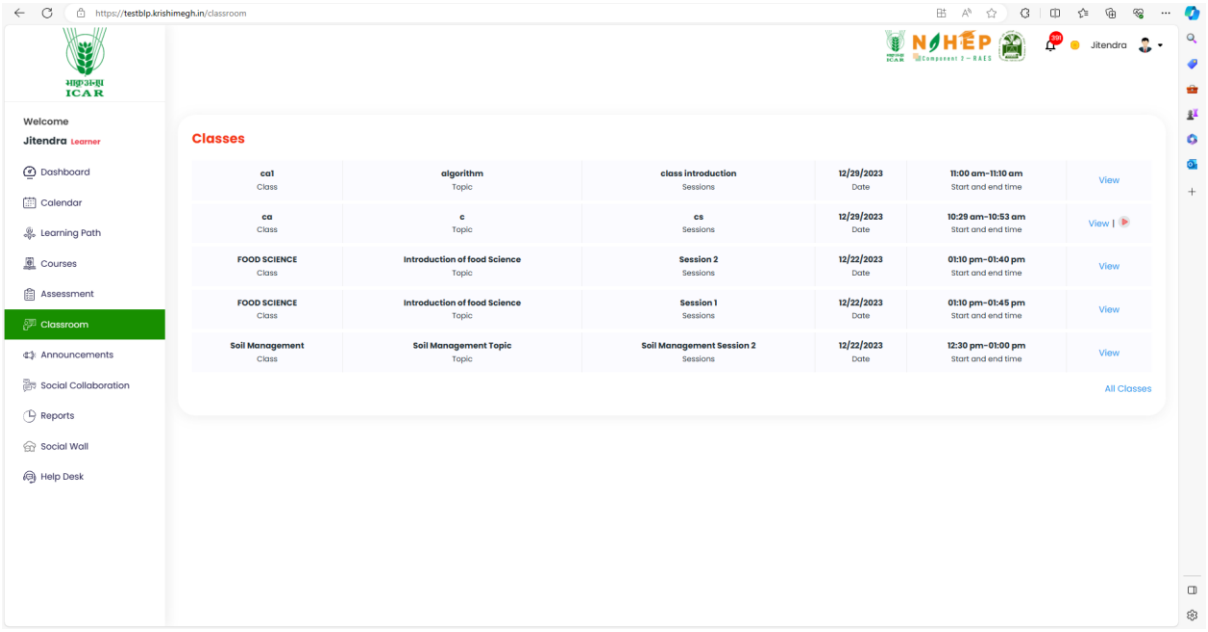
There are three level of Admin approval. A class is approved only after getting approval from all the three admin.



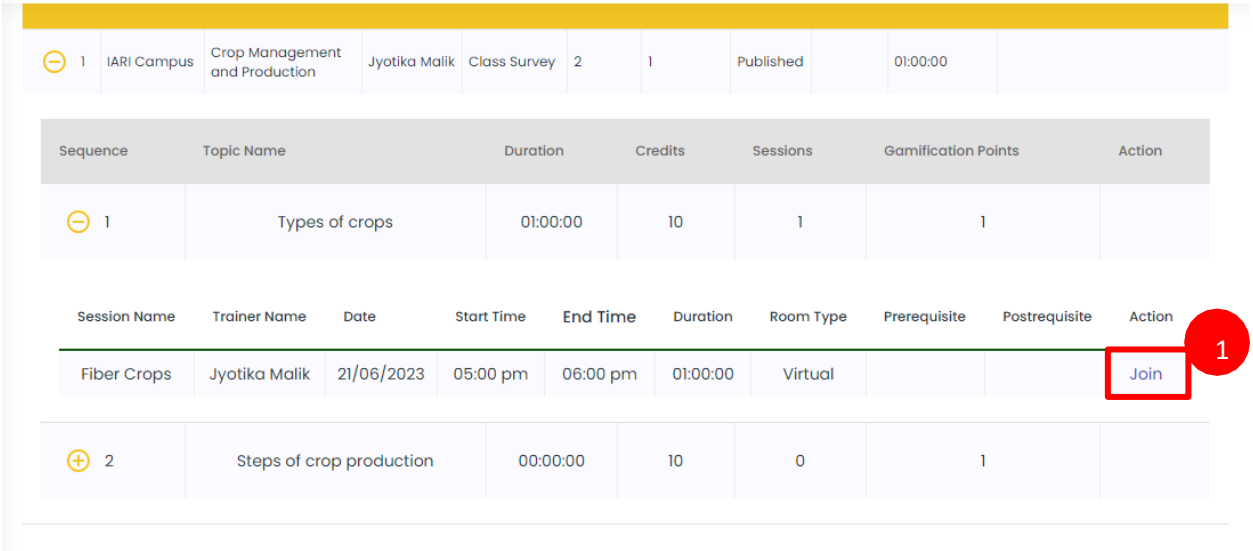
After getting approval from all the admins students get classroom notification that he/she added into that class under notification tab in their portal.



Student also get notification for new classroom on the Dashboard of classroom module.

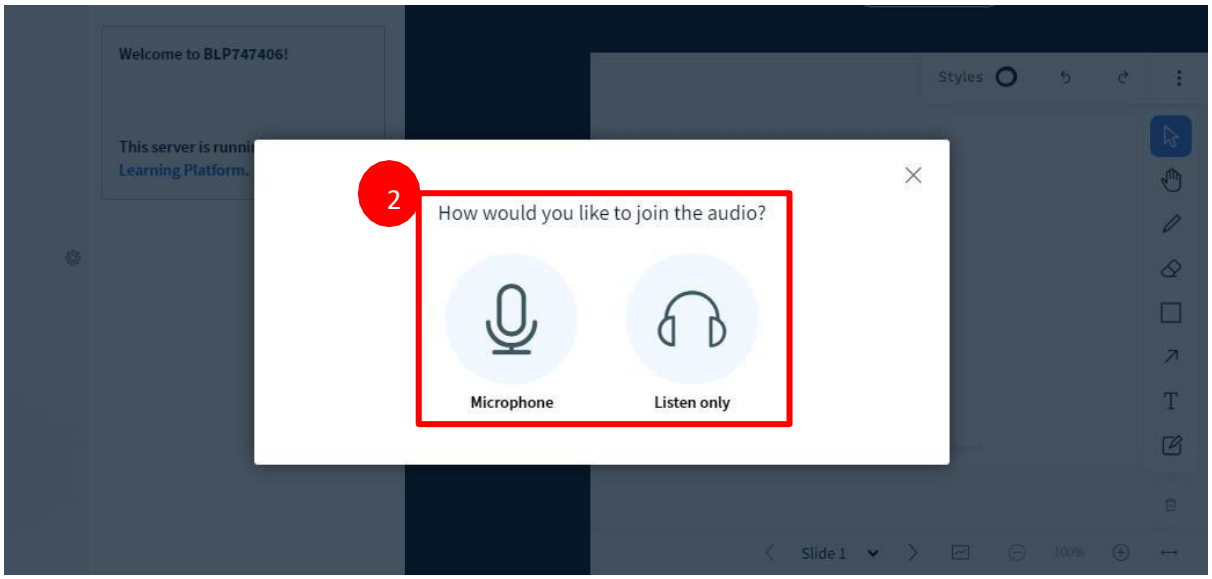


How to start a session?

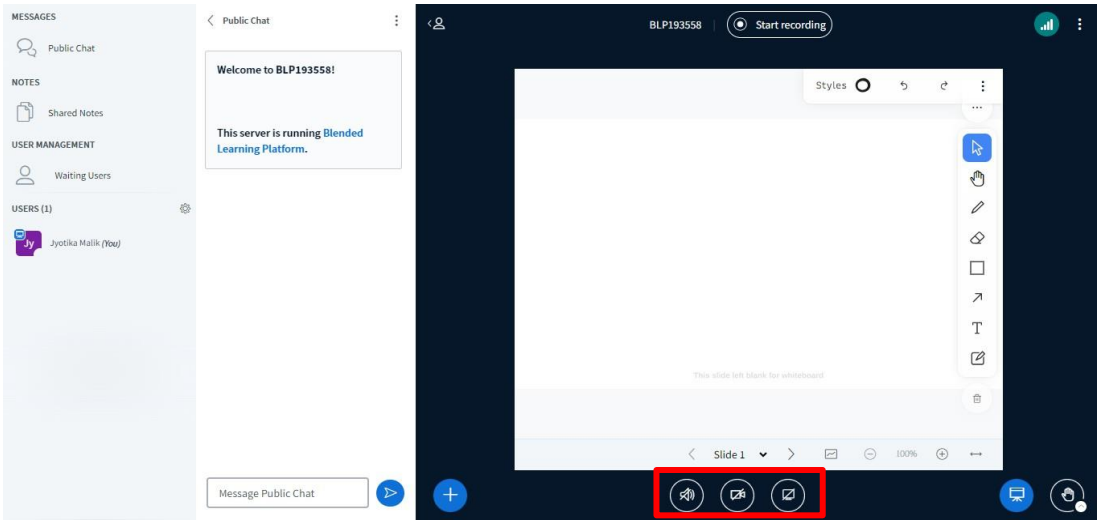


Step-1. Click on “Join”.

The screen will display a pop-up stating, “How would you like to join the audio?” with two options: Microphone and listen only.

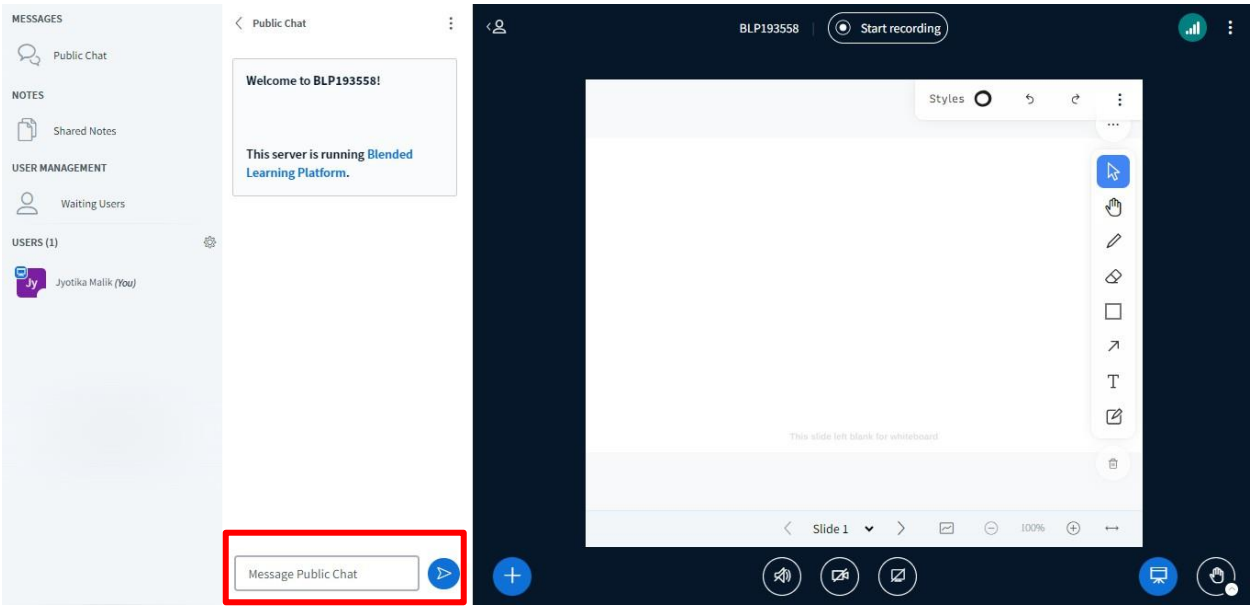


Step-2. Click on “Microphone” or “Listen only”.



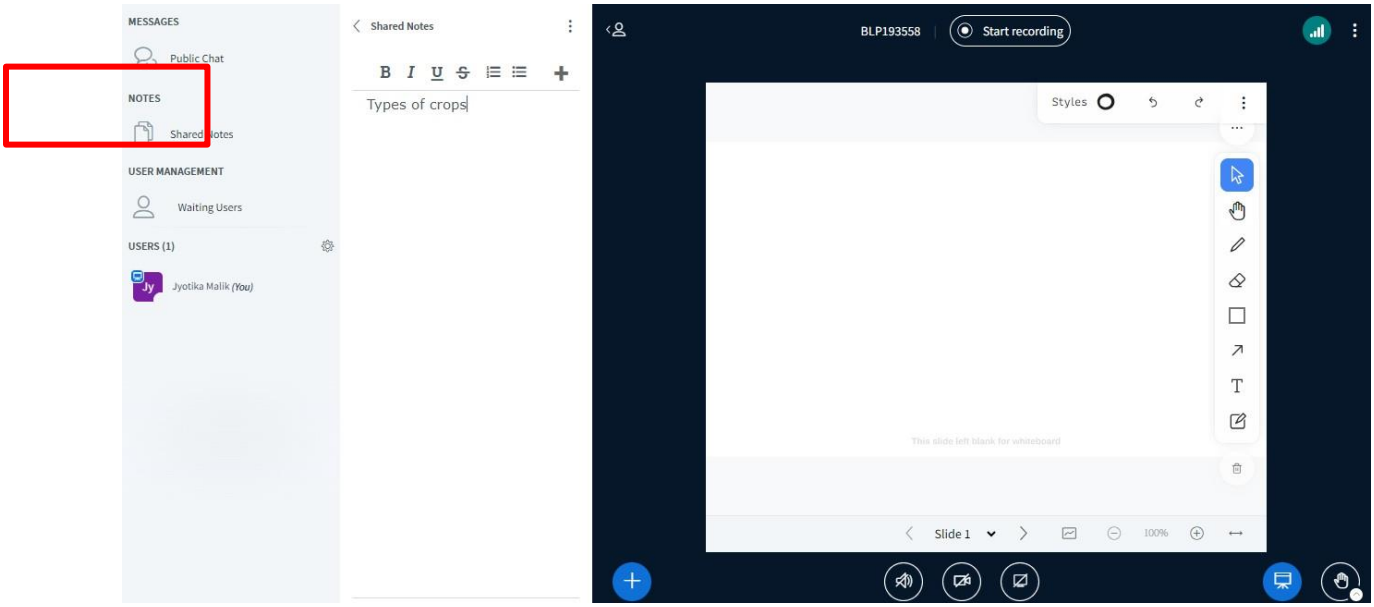
- Users can Mute/Unmute by clicking on the microphone icon.
- Users can On/Off video by clicking on the Camera icon.
- Users can share/unshare screen by clicking on the screen icon.

Public chat



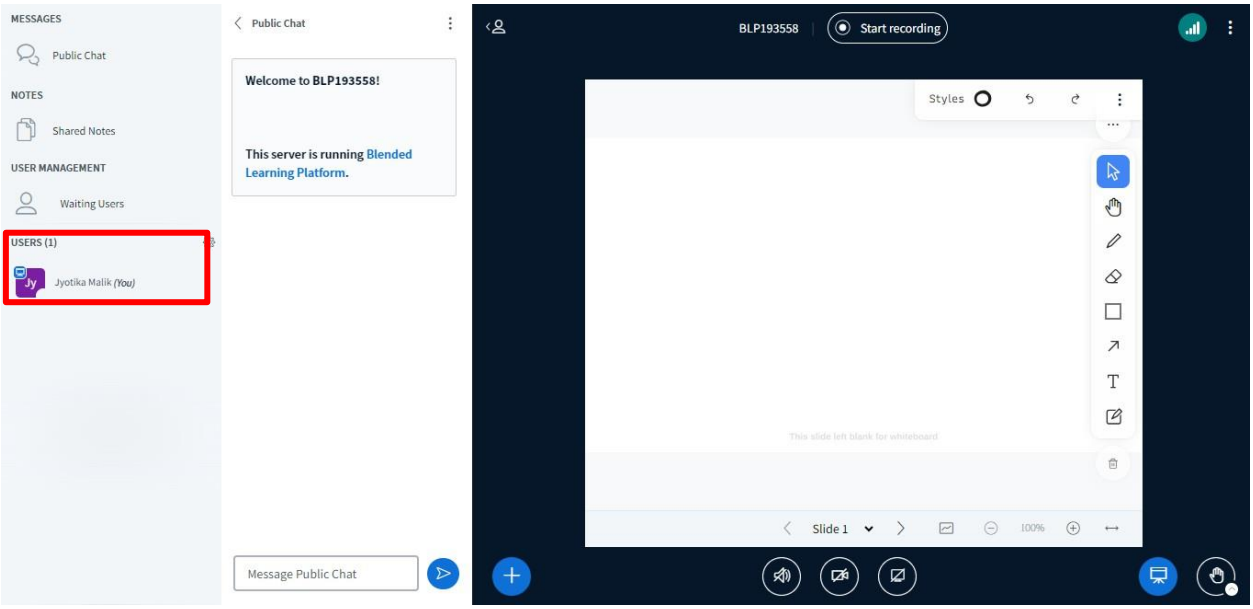
Users can also write messages in public chat, and all the other users can view the message.

Shared note



Users can write notes by clicking on shared notes. All the users will be able to view shared notes.

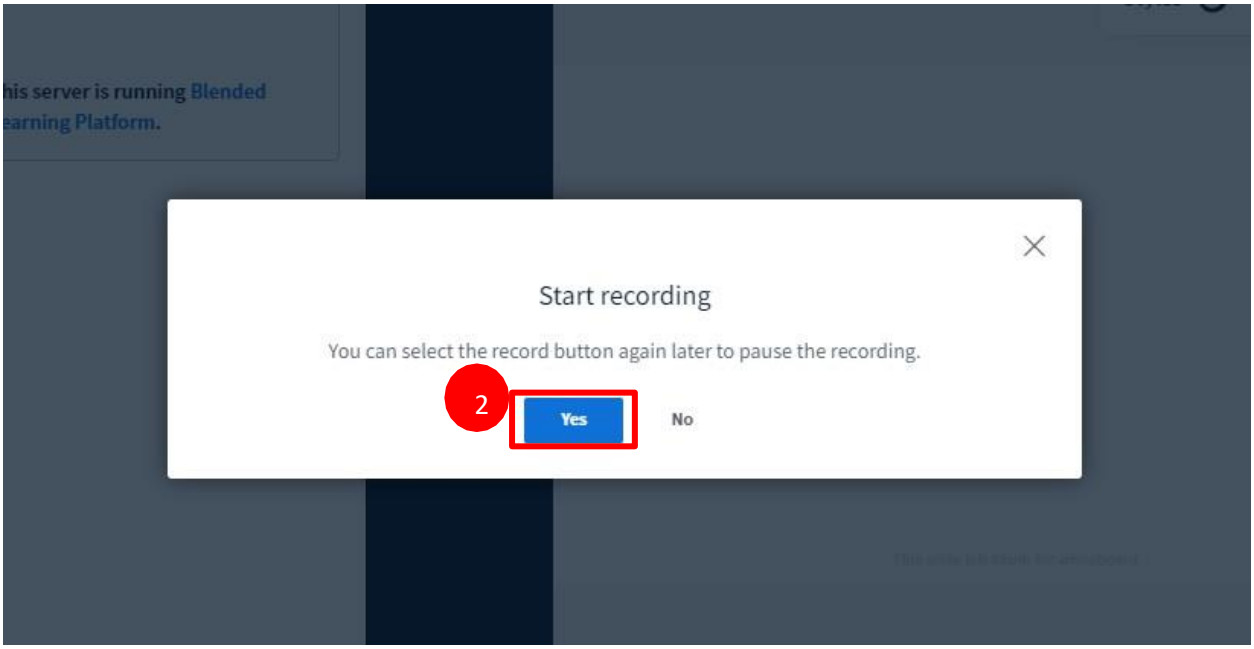
Users and Waiting Users



The list of participants and the list of waiting participants can be seen under “USER MANAGEMENT”.

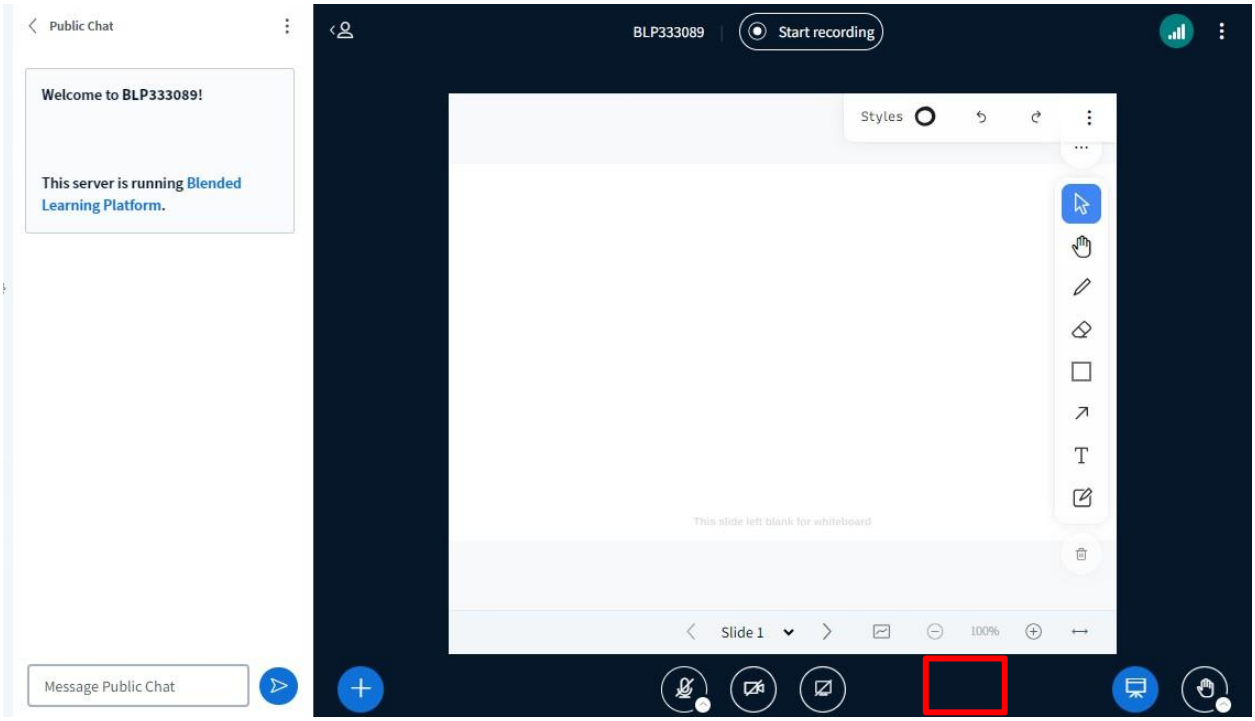
Recording

Step-1. Click on the “Start recording” button.



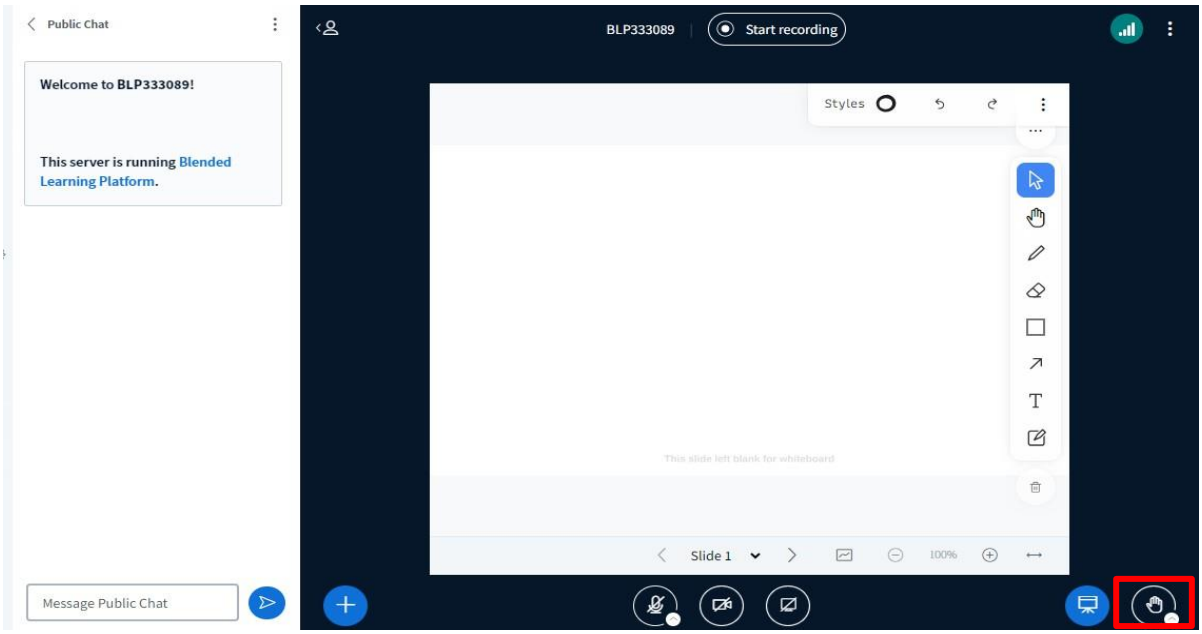
Step-2. Click on the” Yes” button.

Minimise presentation



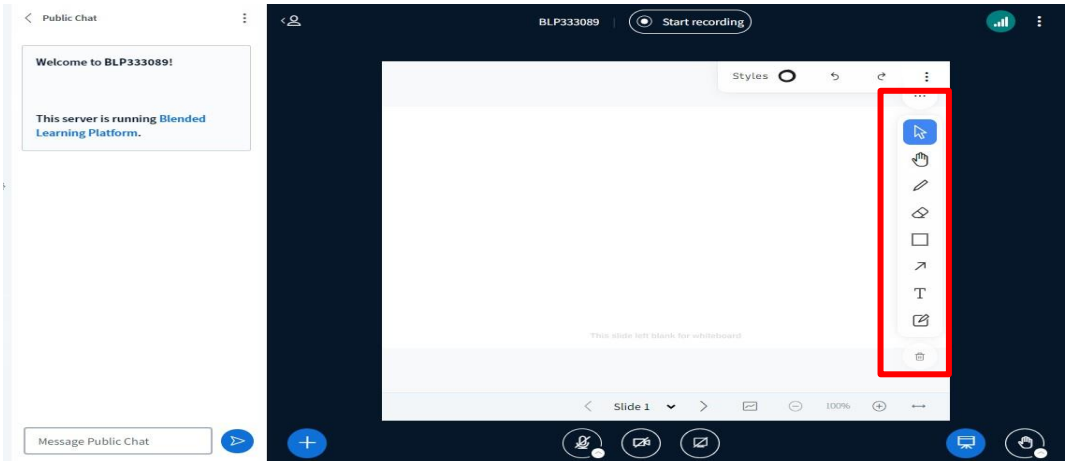
Click on the “screen Icon” in blue to minimize the presentation.

Raise hand



Click on the “raise hand” button to raise your hand.

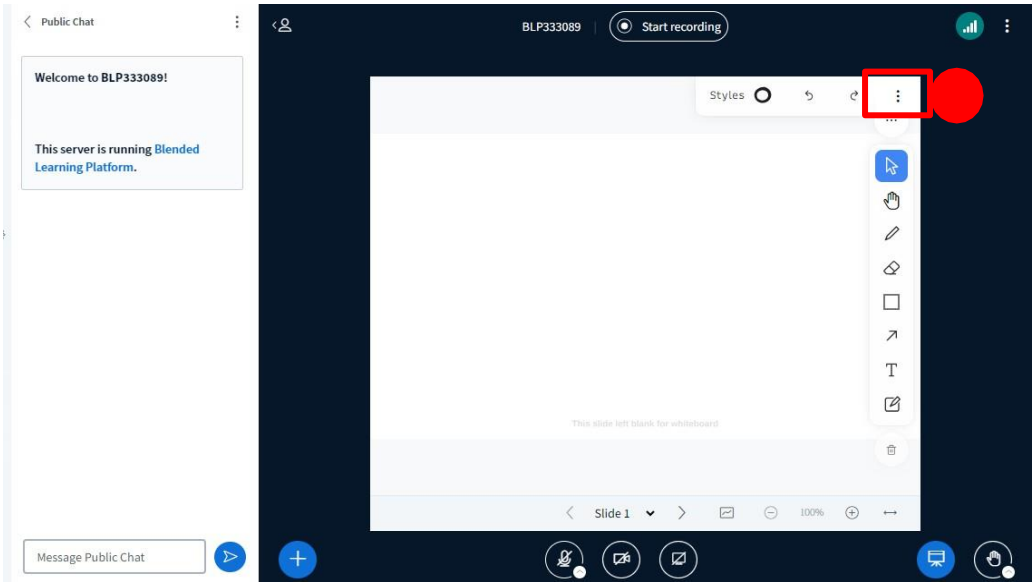
White Board



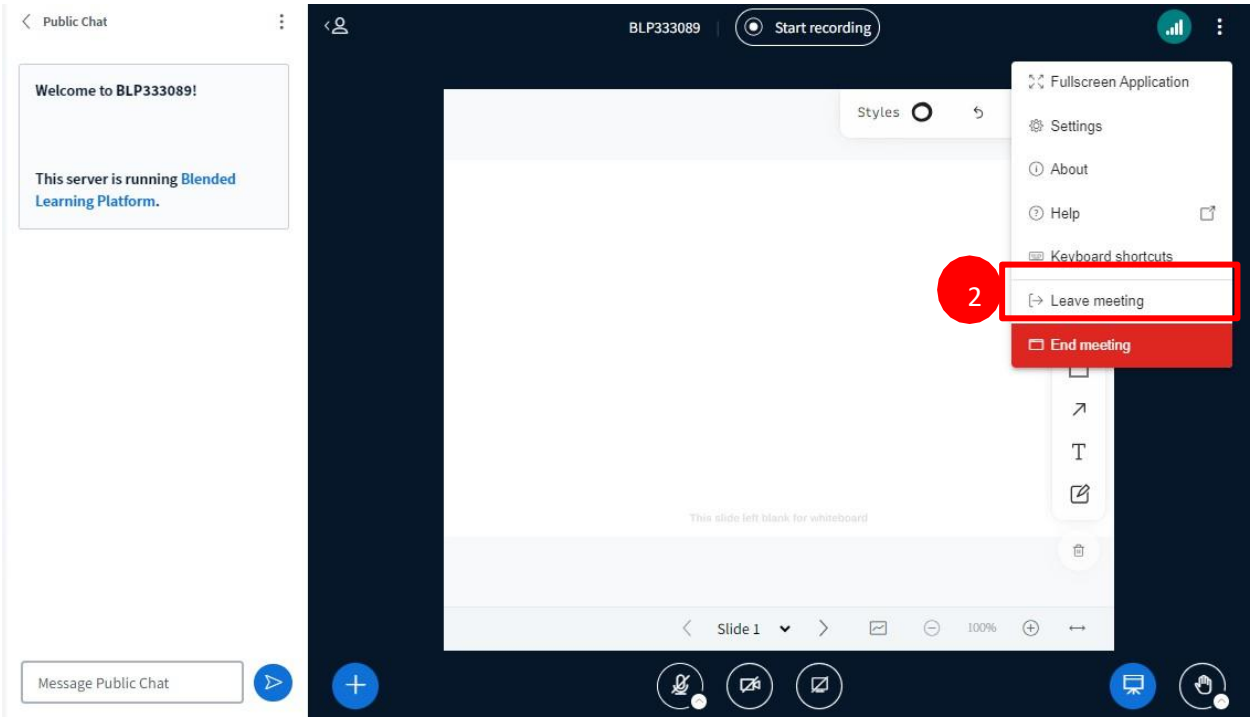
Users have multiple features associated with a whiteboard. The features of the whiteboard are listed below:

- Select object – The users can select an object by clicking on select.
- Move object – The users can select Pan to move the board.
- Pen- The users can select a pen to draw on the board.
- Eraser- The users can select an eraser to erase.
- Text- The users can add text.
- Sticky- The users can add sticky notes.
- Delete- The users can click on delete-to-delete text.

How to leave the meeting?

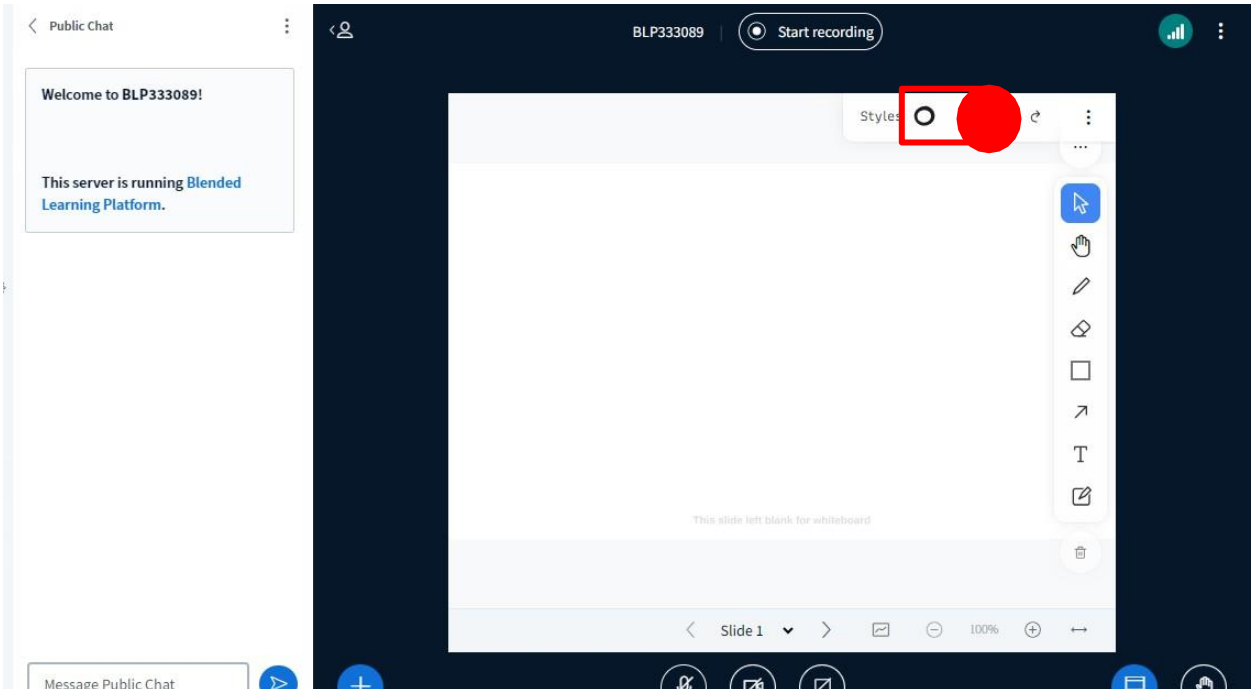


Step-1. Click on the “three dots”

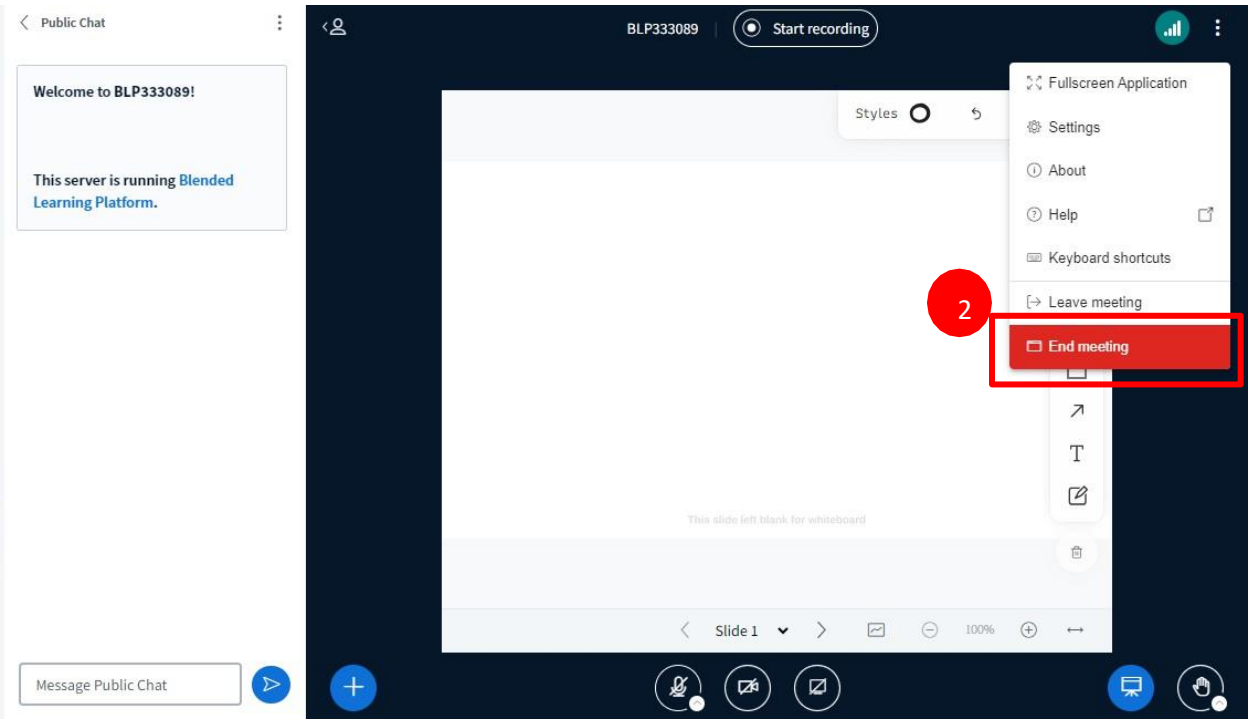


Step-2. Select ”Leave meeting”.

How to end the meeting?

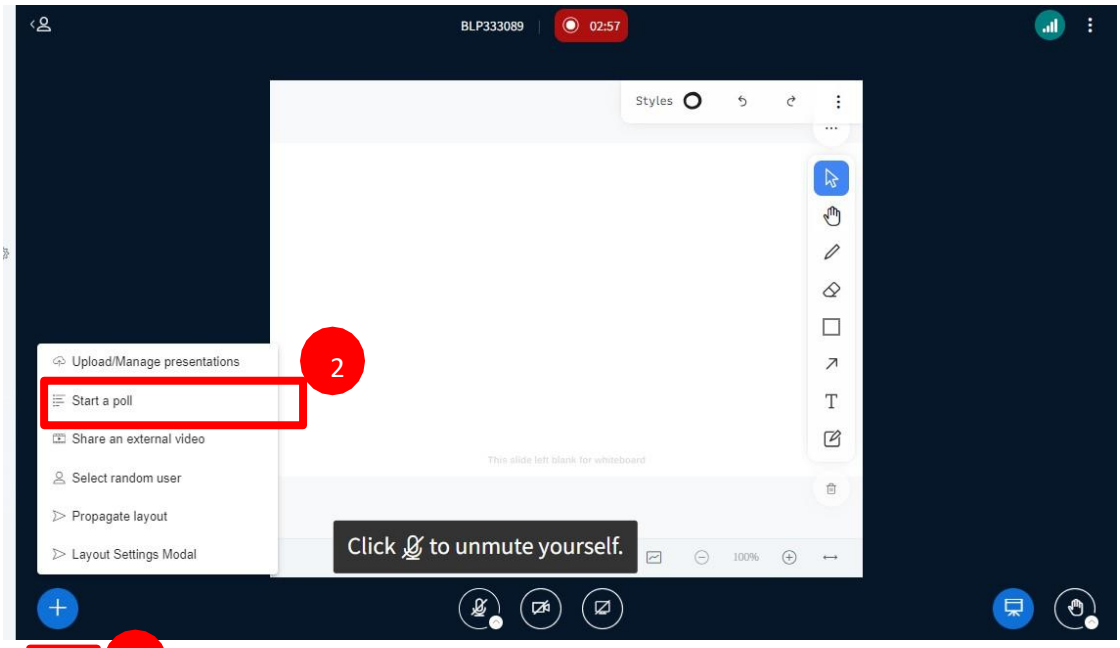


Step-1. Click on the “three dots”



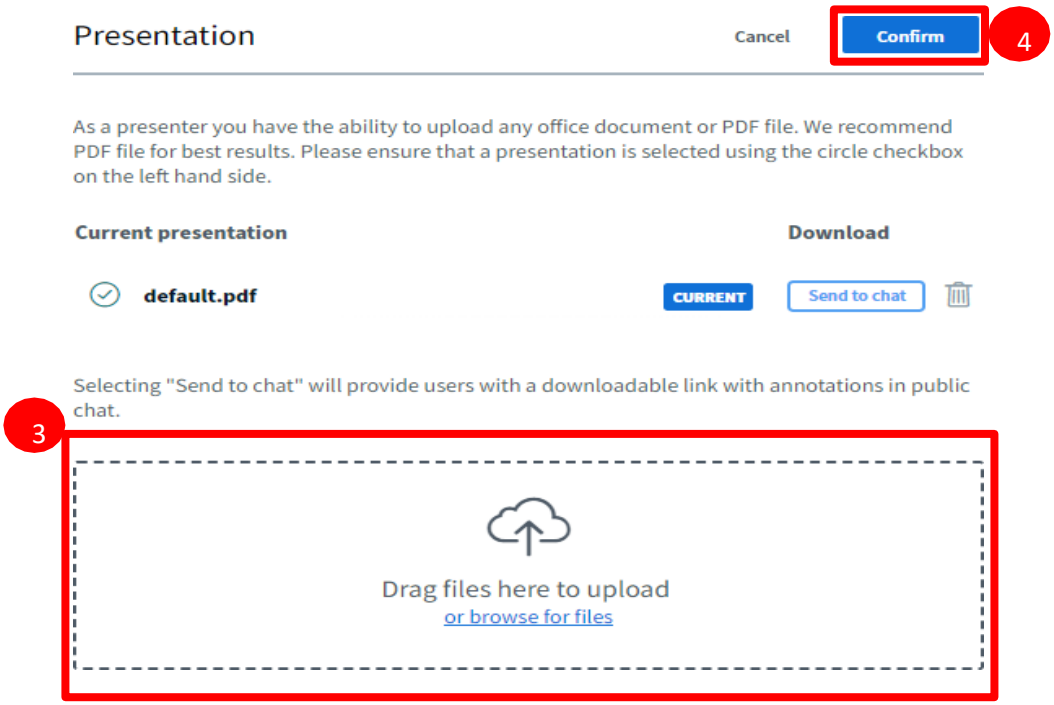
Step-2. Select “End meeting”.

How to upload a presentation?



Step-1. Click “+” in blue.

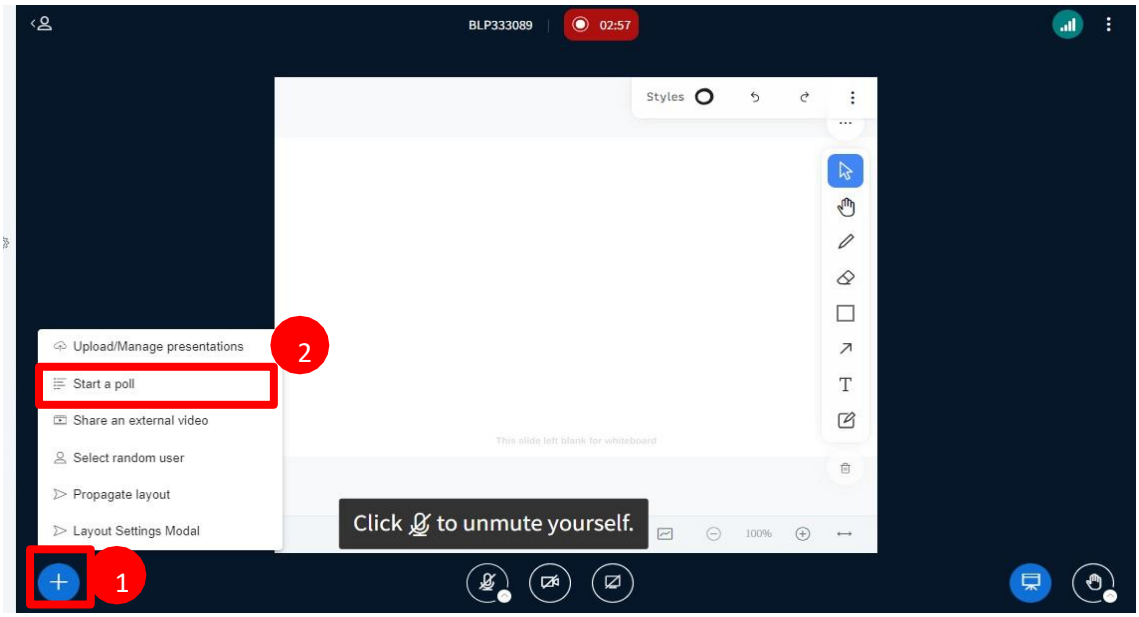
Step-2. Select Upload/Manage presentations.



Step-3. Click on “or browse for files” or drag and drop the file in the given space.

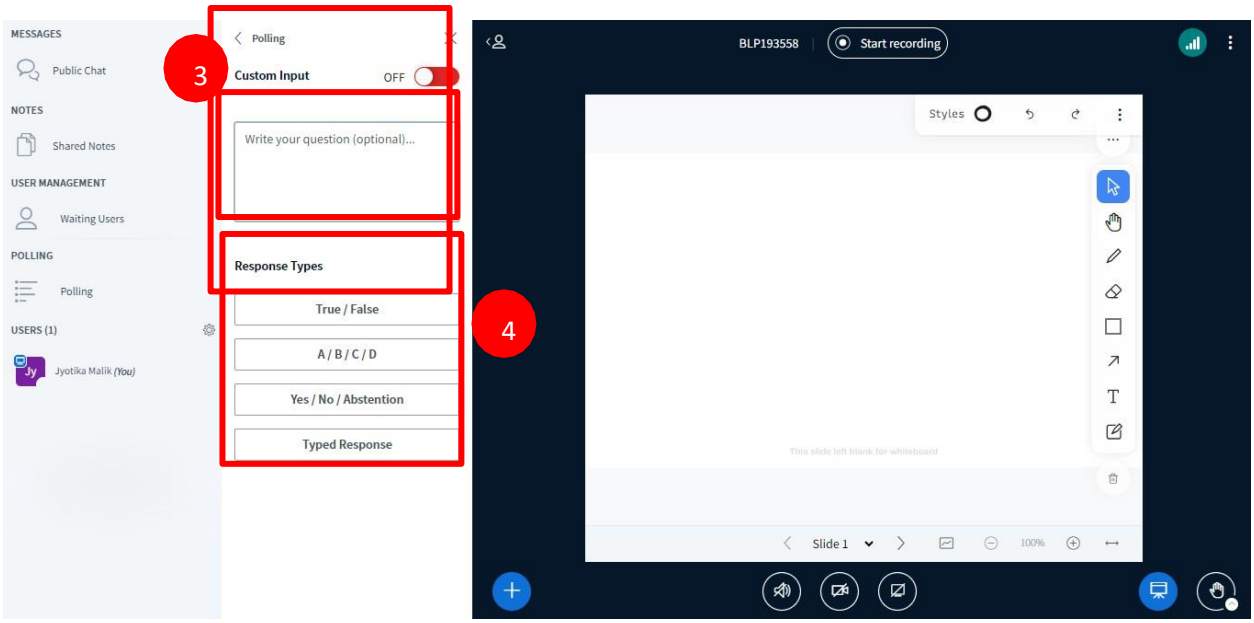
Step-4. Click on “Confirm”.

How to start a poll?



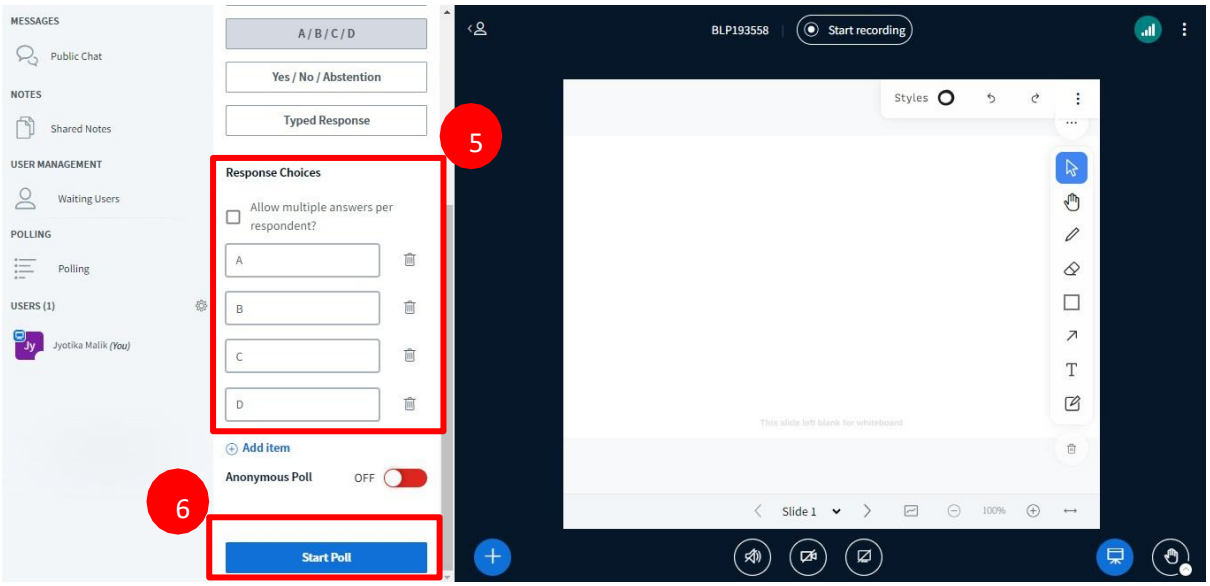
Step-1. Click “+” in blue.

Step-2. Select “Start a poll”.



Step-3. Type a question under “Write your question”.

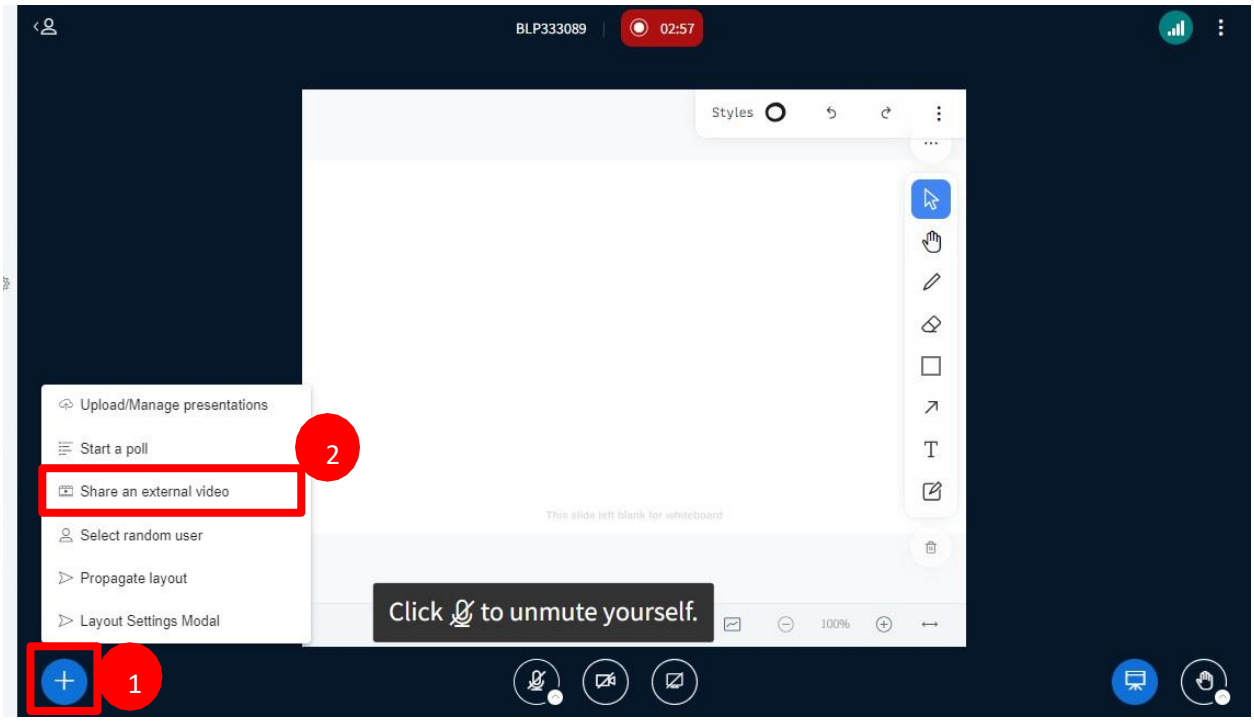
Step-4. Select “Response types”.



Step-5. Enter answers.

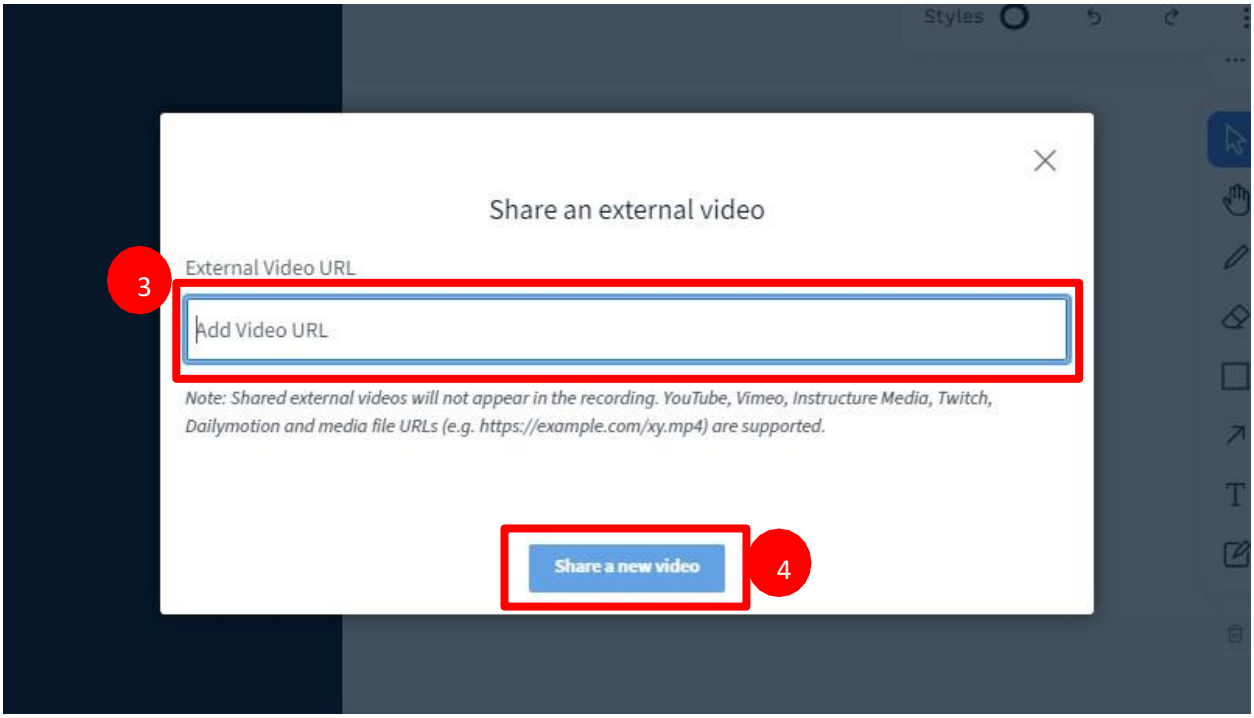
Step-6. Click on “Start Poll”.

How to share an external video?



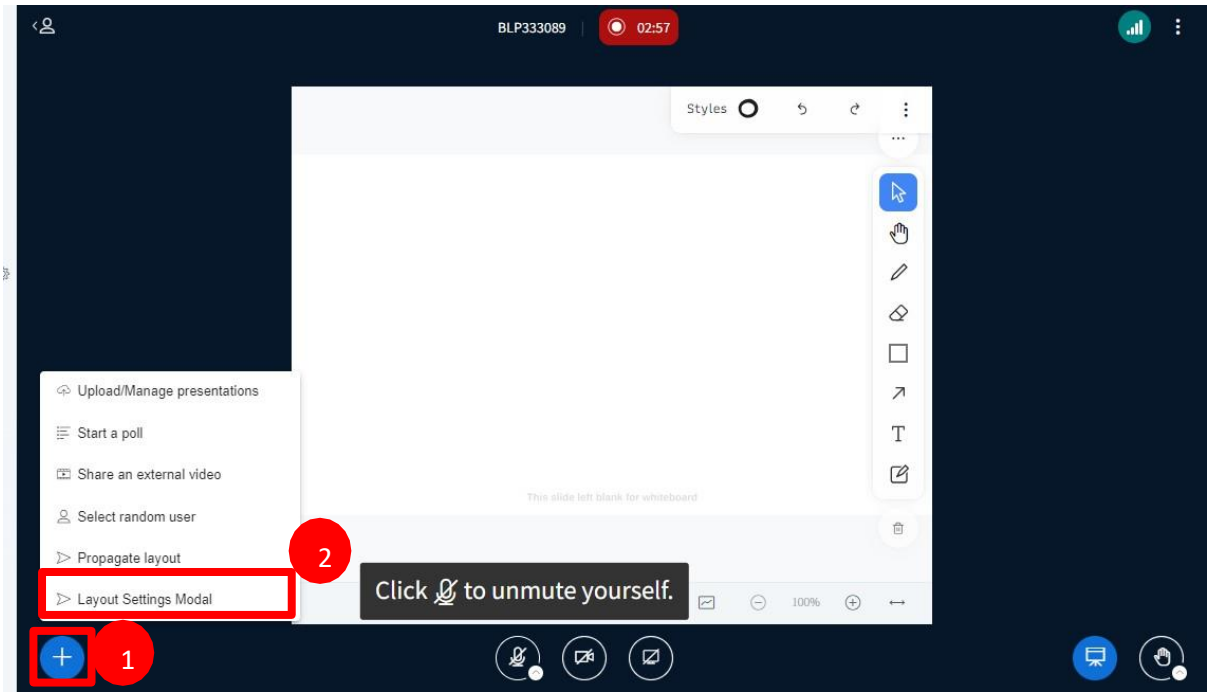
Step-1. Click “+” in blue.

Step-2. Select “Share an external video”.



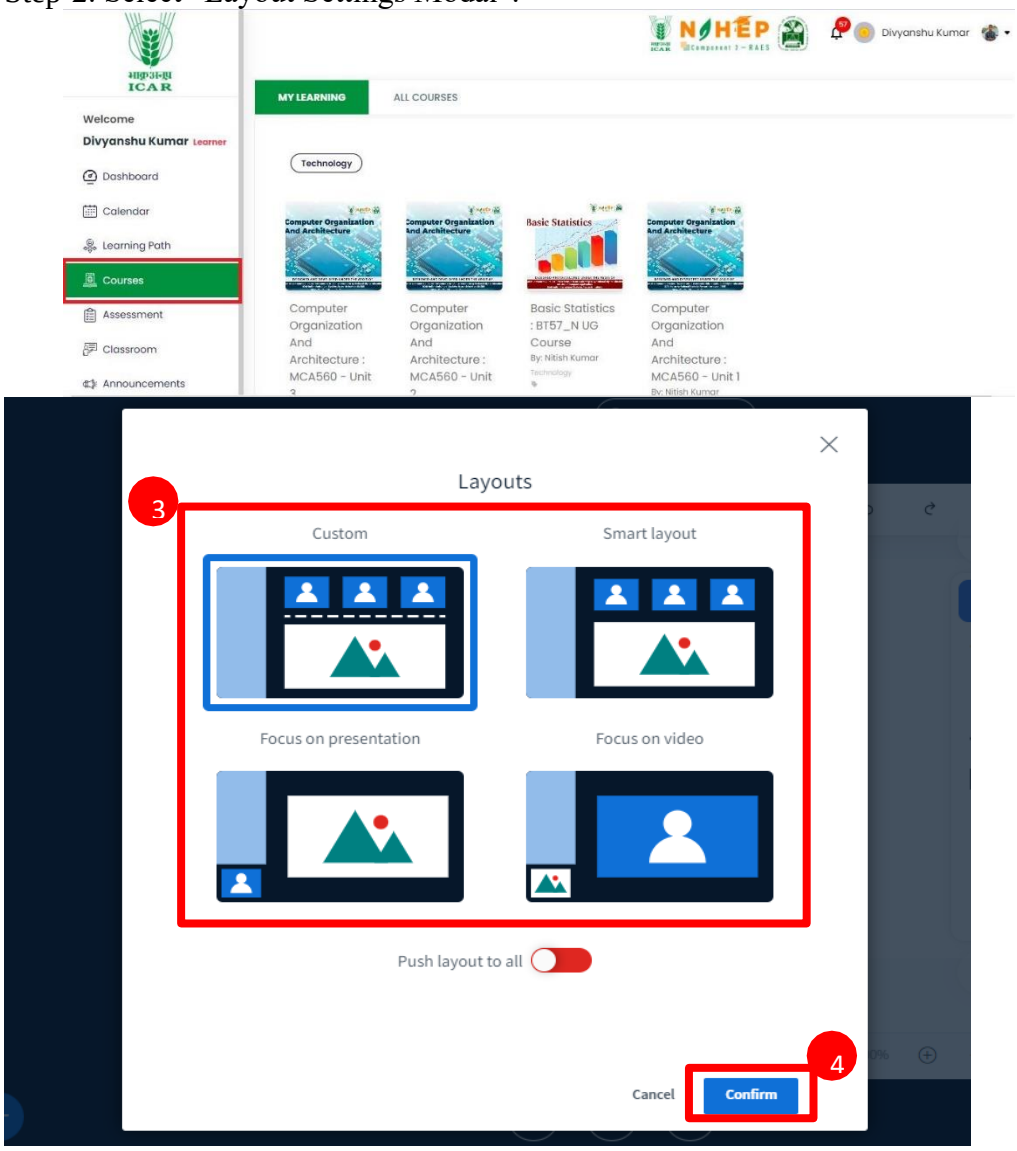
Step-3. Enter the URL, under Add Video URL. Step-4. Click on “Share a new video”

Layout Settings



Step-1. Click “+” in blue.

Step-2. Select “Layout Settings Modal”.



Step-3. Select the layout.

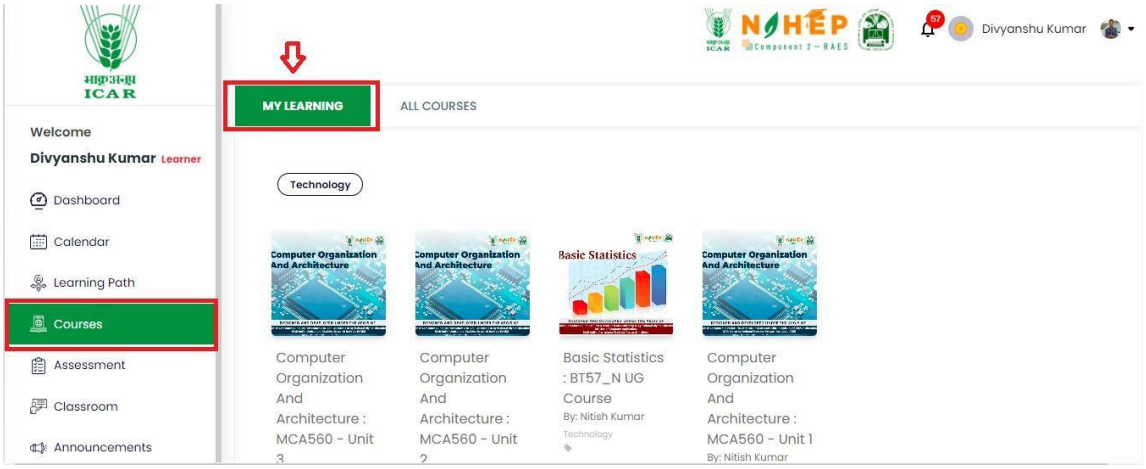
Step-4: Click on “Confirm”

Development of Courses and Assessment in student Module of NARES-Blended Learning Platform

1. **How View Courses?**

   Step 1 – After successful login to the application go to the ‘Courses’ module, the below screen will appear. There will be two tabs: ‘MY LEARNING’ and ‘ALL COURSES’

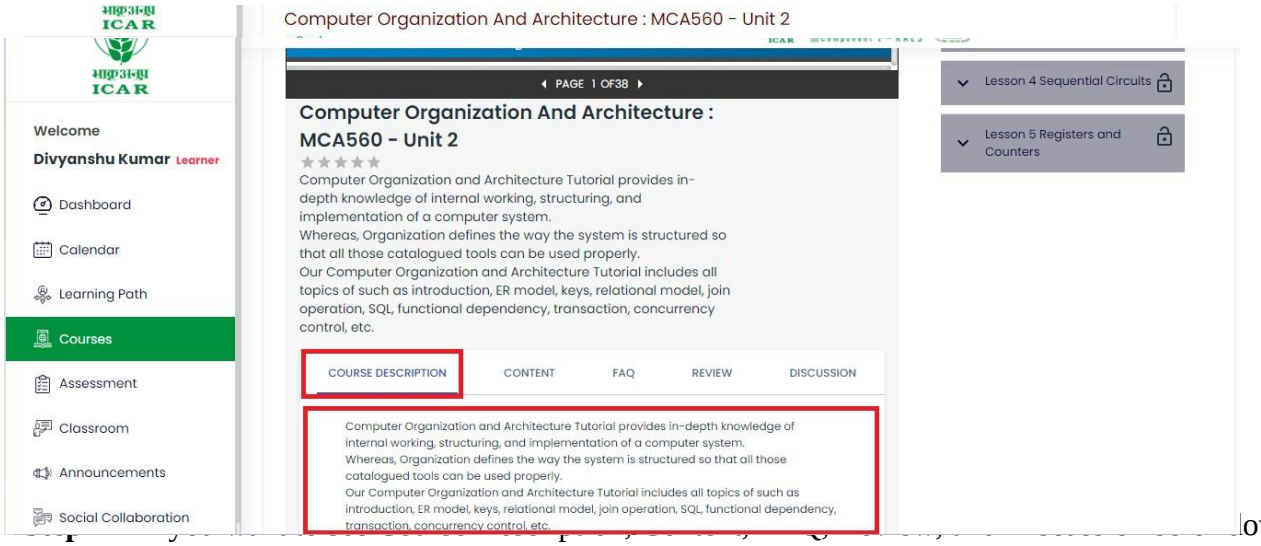
1.1 How to See Assigned Courses?



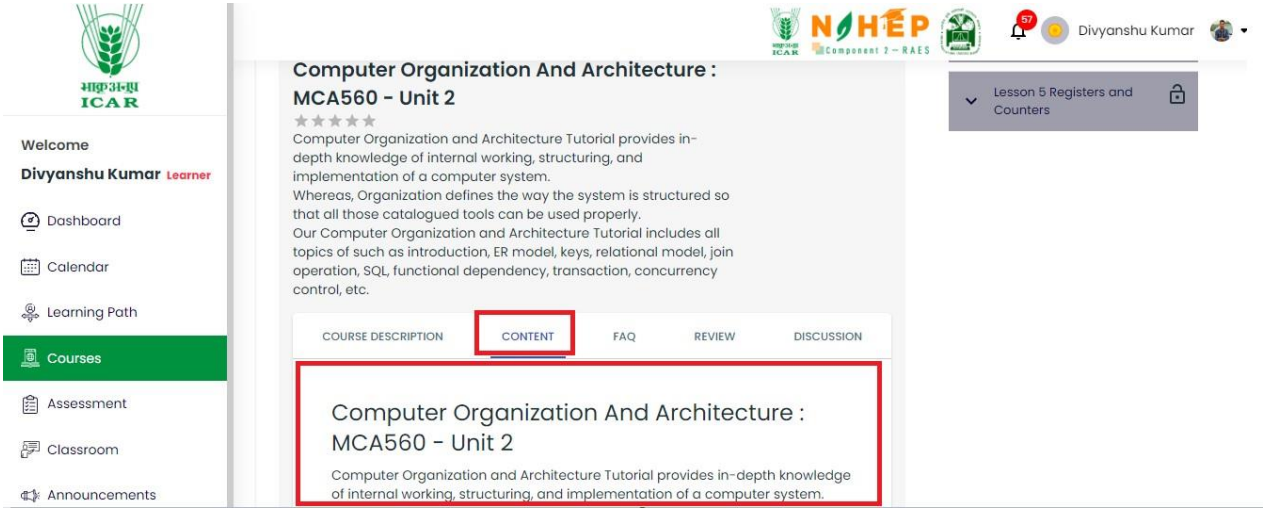
Step 1 – Users can see all the assigned courses to them under MY LEARNING section, as shownbelow.

Step 2 – If you wish to watch videos or read the course, select the course topic, a new screen will appear as shown below.

Step 3 – Further lessons can be learnt by clicking on Course Content as shown below.

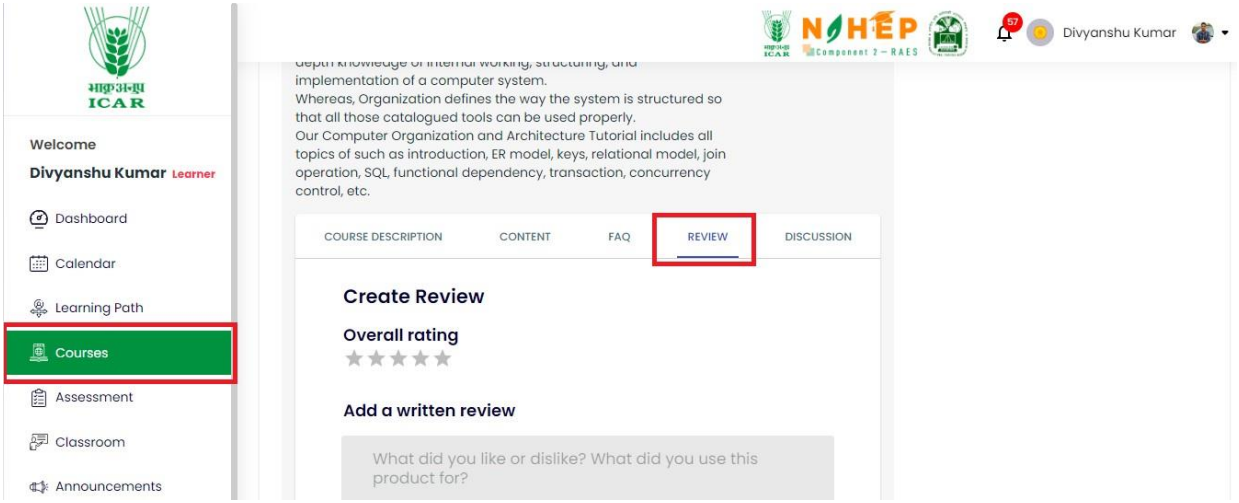


Step 2 – If you want to see the course description, scroll down the page a little and the screen will appear as shown below.

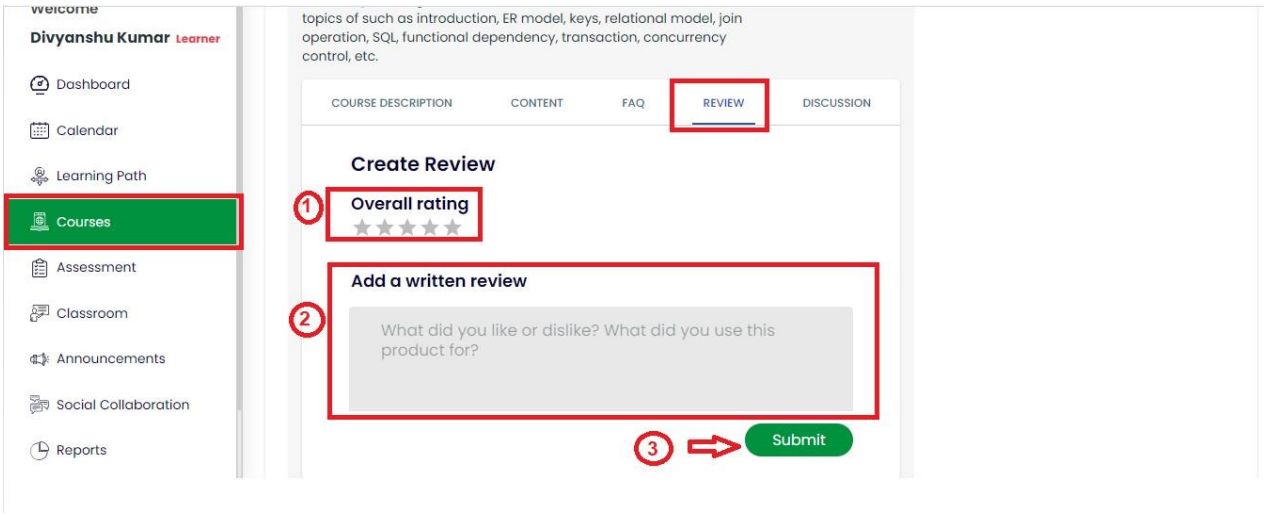


1.1.1 How to Rate a Course?

Step 1 – If you want to rate/review the course, click the Review option, and a page below will appear.

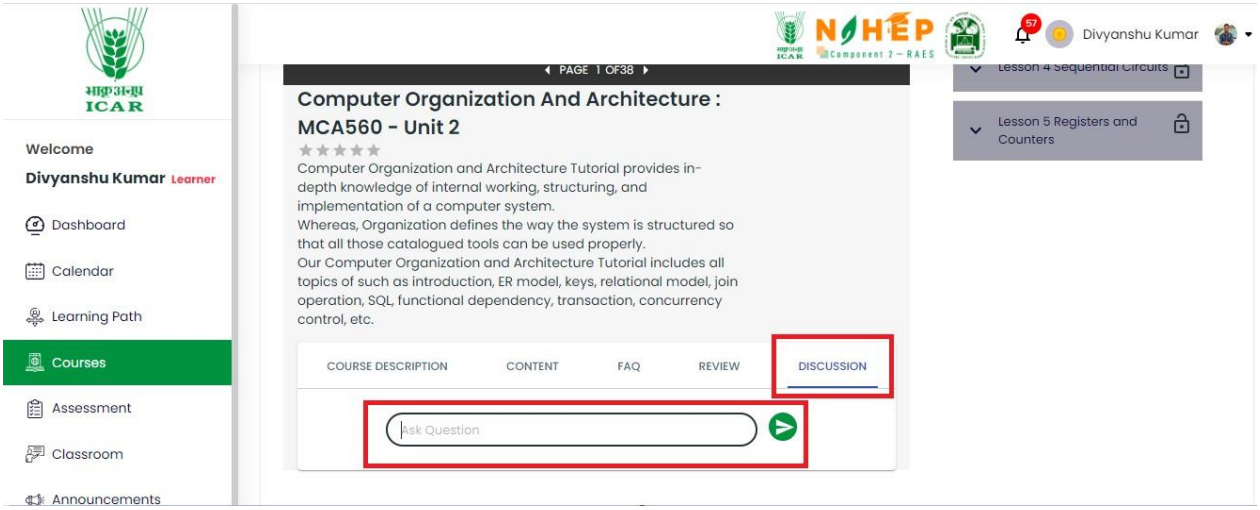


Step 2 – Put a star rating, write a comment, and click on submit button as shown in the below screen.



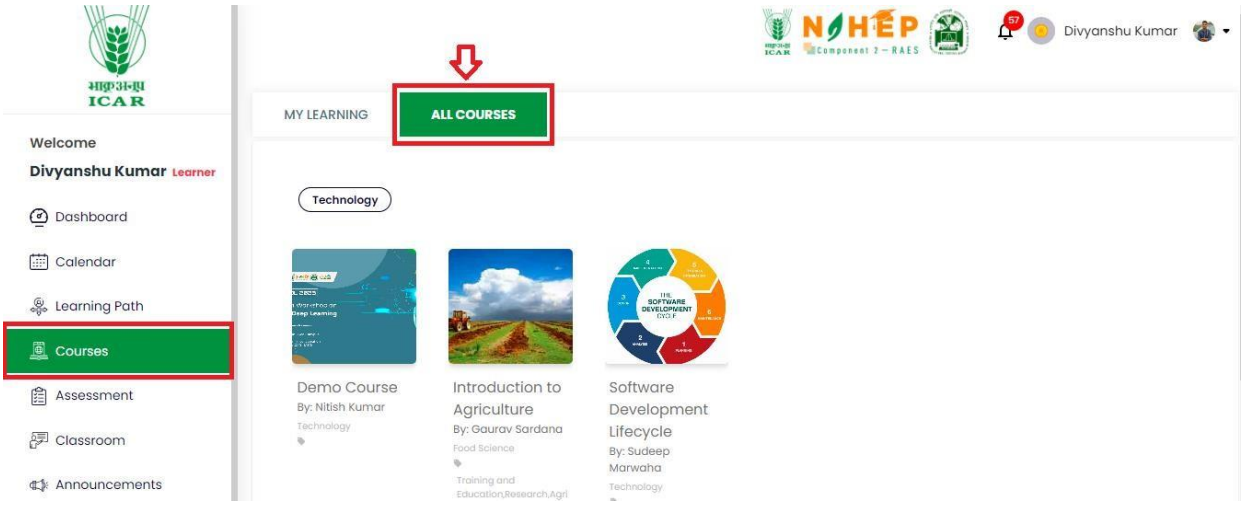
1.1.2 How to Start Discussion/Ask a Question?

Step 1 – If you wish to start a discussion or ask questions on the course with other students and faculties, click on the Discussion option, a below screen will appear and you can ask a question/start discussion.



1.1.3 Want to explore more Courses?

Step 1 – If you wish to explore more courses then they are available under the 'ALL COURSES' section as shown in the below screen.



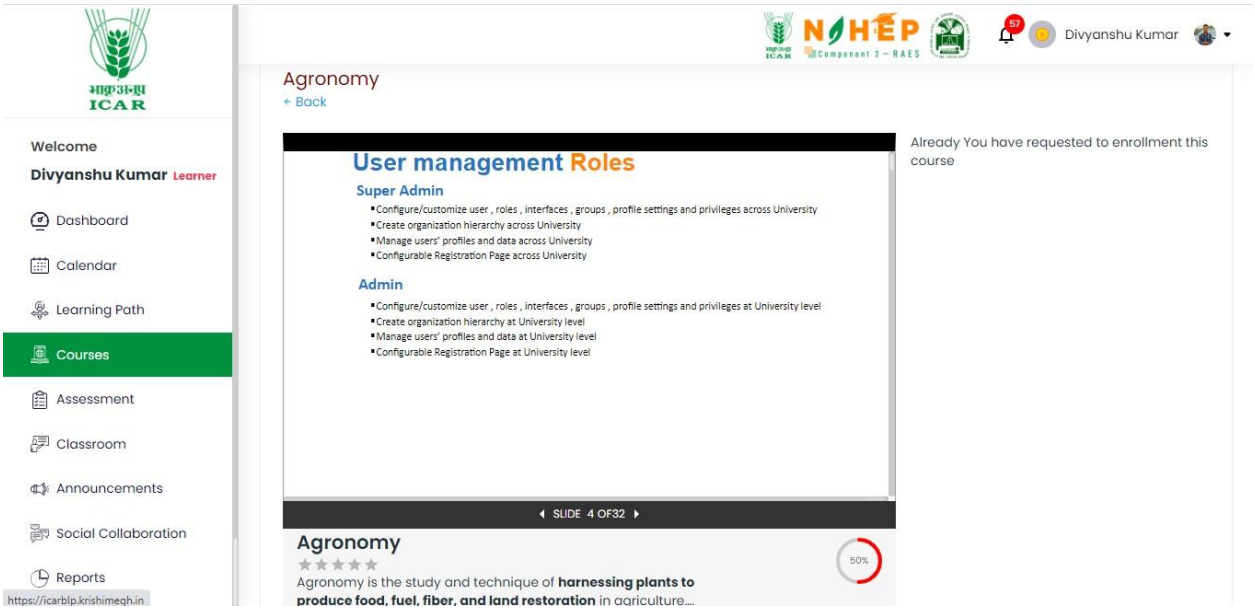
Step 2 – Click on any course of your choice, a below screen will appear with a short introduction about the course and an 'Enroll' button on the right-hand side as shown in the below screen.

Step 3 – Click on Enroll button, a below page will appear.

Ste w will appear as sho

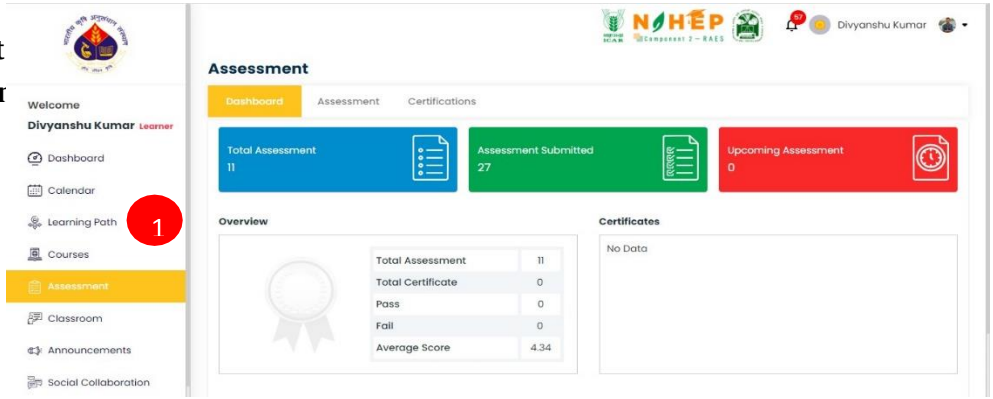
1.1.4. Want to see Course Progress?

Step 1 – Click on any assigned course and see the course percentage below. Suppose you have completed only 4 lessons out of 8 lessons then the course percentage will be seen as 50% as shown below.



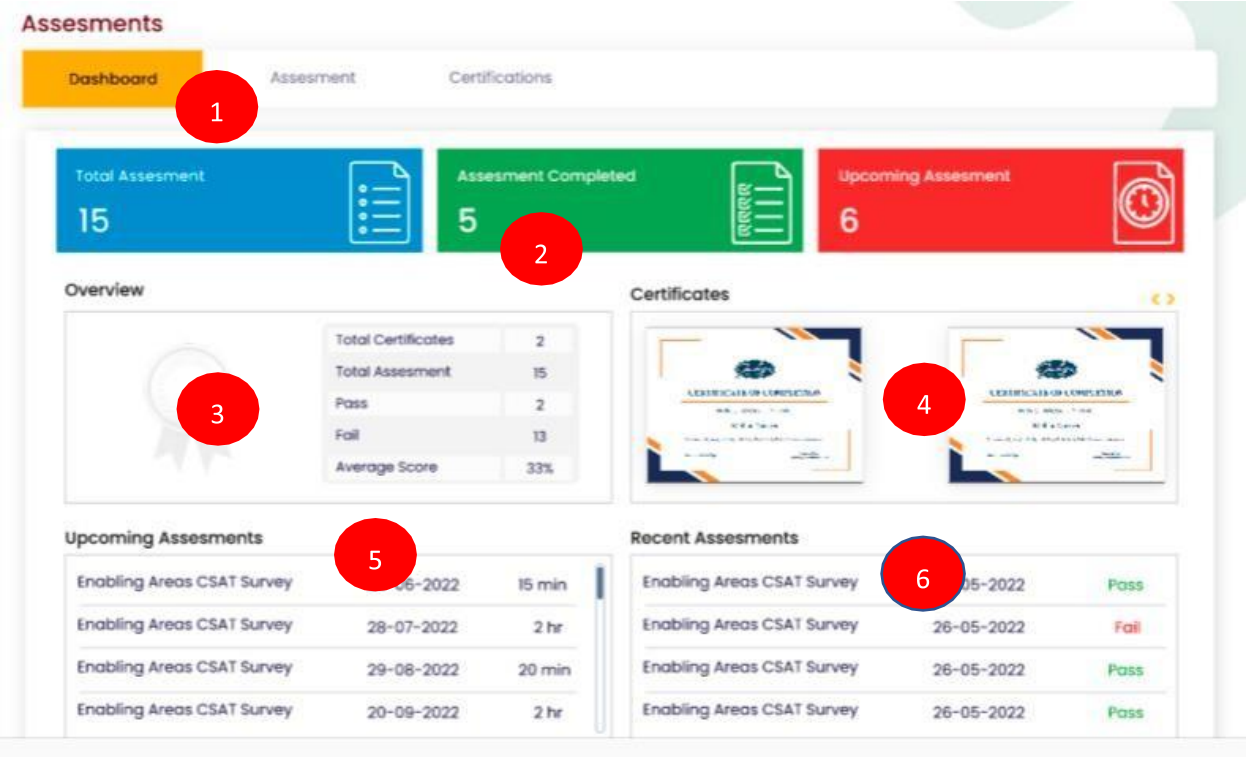
2. How to start

2.1 Assessr



Step1. The User scrolls down the left menu bar to click on the Assessment module.

2.2 Dashboard



Step1. Users can see the assessment dashboard.

Step2. Users can see the Total Assessment count, Total Assessment Completed count, andUpcoming Assessment count.

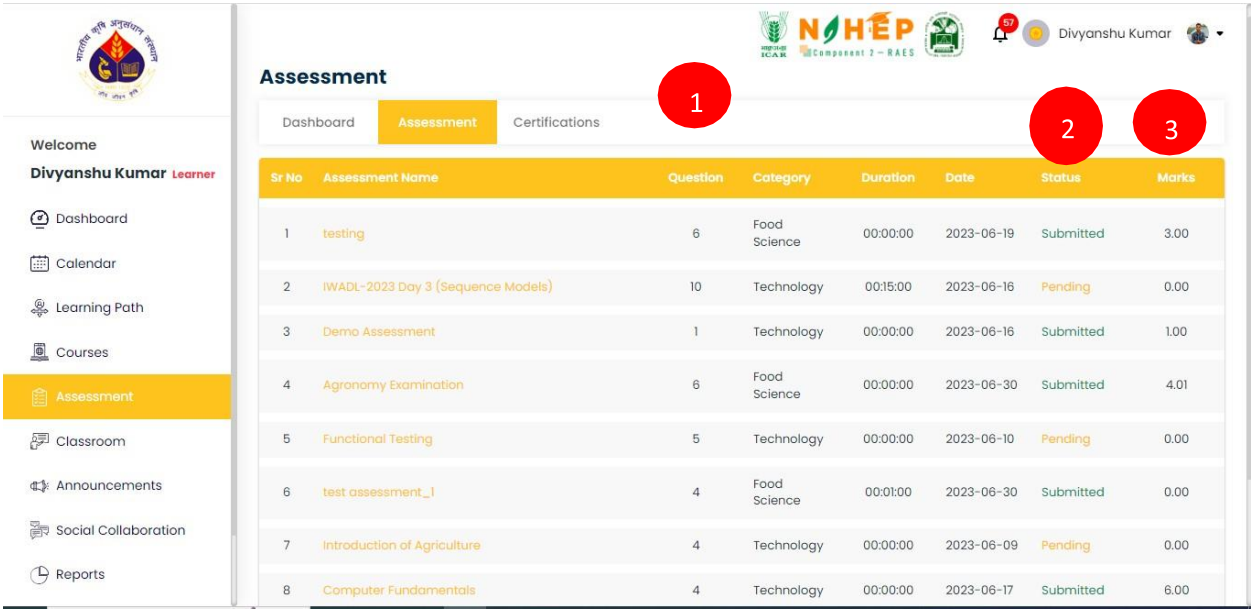
Step3. Users can see the Overview of the assessments completed like Total Certificates, TotalAssessments, the Pass-Fail, and Average score

Step4. Users can see the Certificates received.

Step5. Users can see the details of the Upcoming Assessments

Step6. Users can see Assessments that are recently completed.

2.3 Assessment

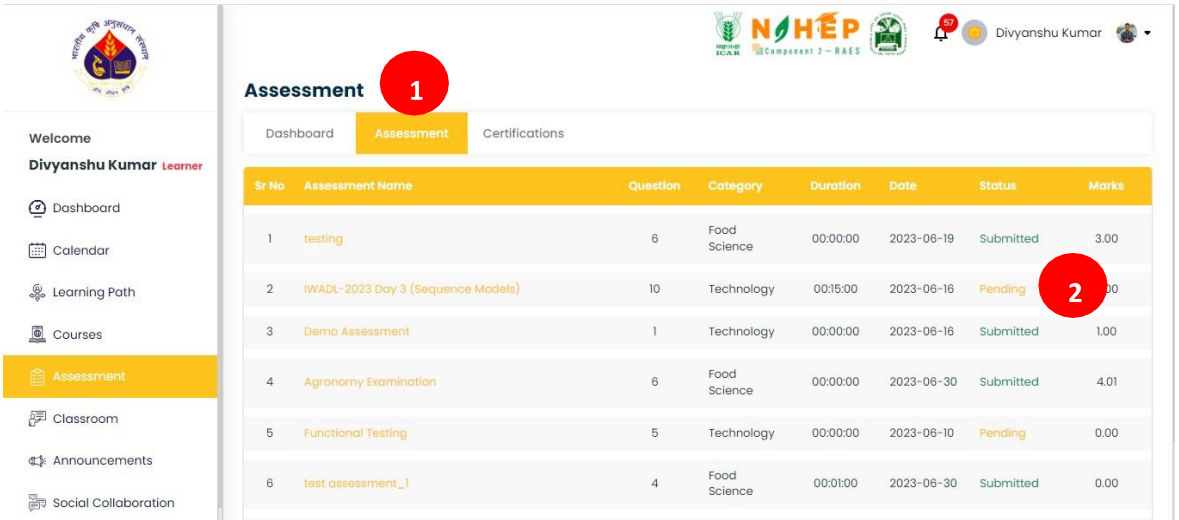


Step1. The users can see the list of all the assessments assigned to them.

Step2. Can check the status of the assessments, i.e., submitted or pending.

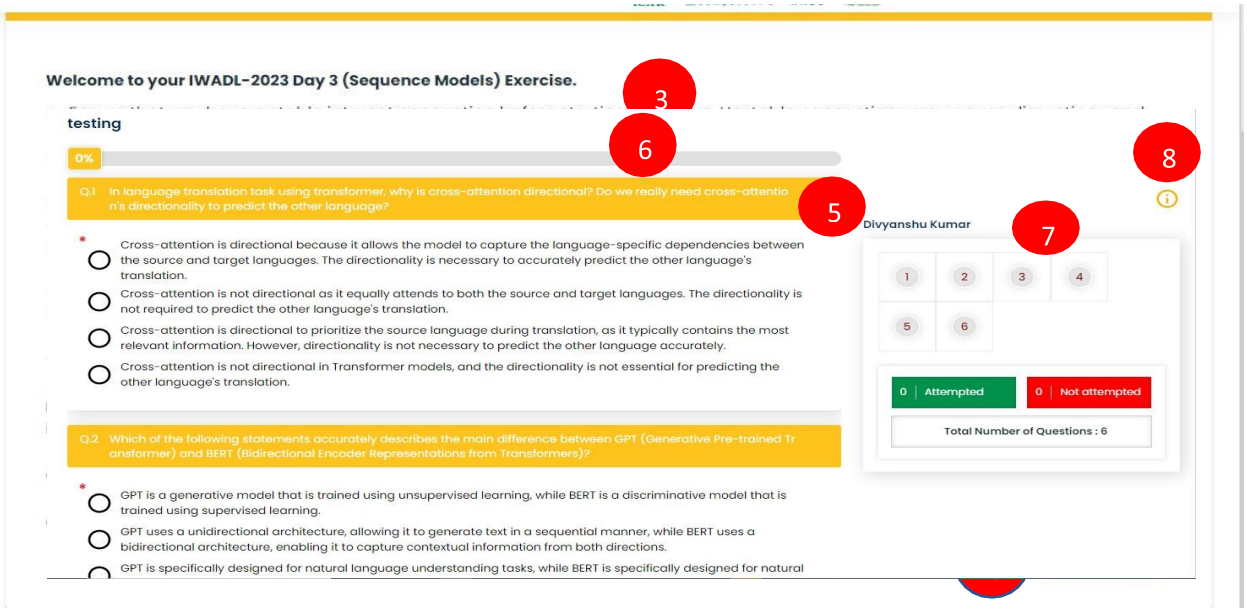
Step3. Click on the 'Marks' option to see the assessment marks received

2.4 Attempting Assessment.



Step1. The users click on the assessment tab to view the assessments assigned

Step2. The users can see the list of all the assessments assigned to them and clicks on theassessment name with the status as pending.



Step3. The users can now be able to see the instructions on the screen.

Step4. The users click on next to attempt the assessment.

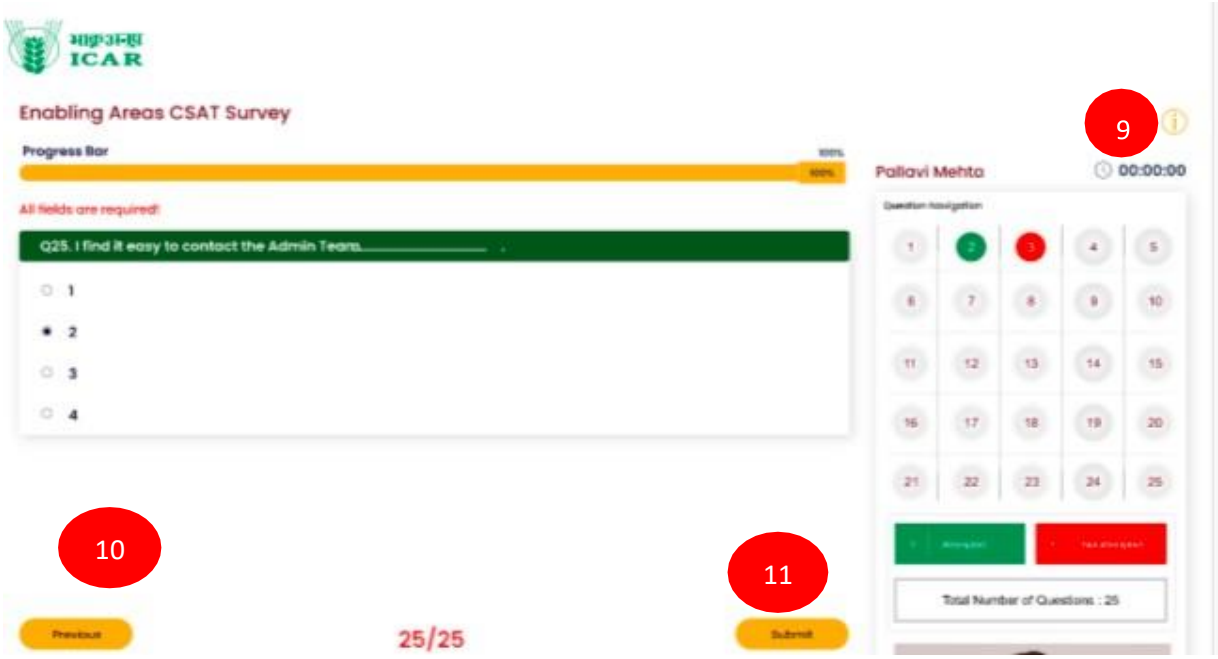
2.5 Assessment Screen

Step5. The users can now able to view the assessment questions.

Step6. The users can track their progress through the progress bar.

Step7. The users can view question view the question palette to track the status of individualquestions. Question numbers marked in green represent attempted, while those in red represent unattempted.

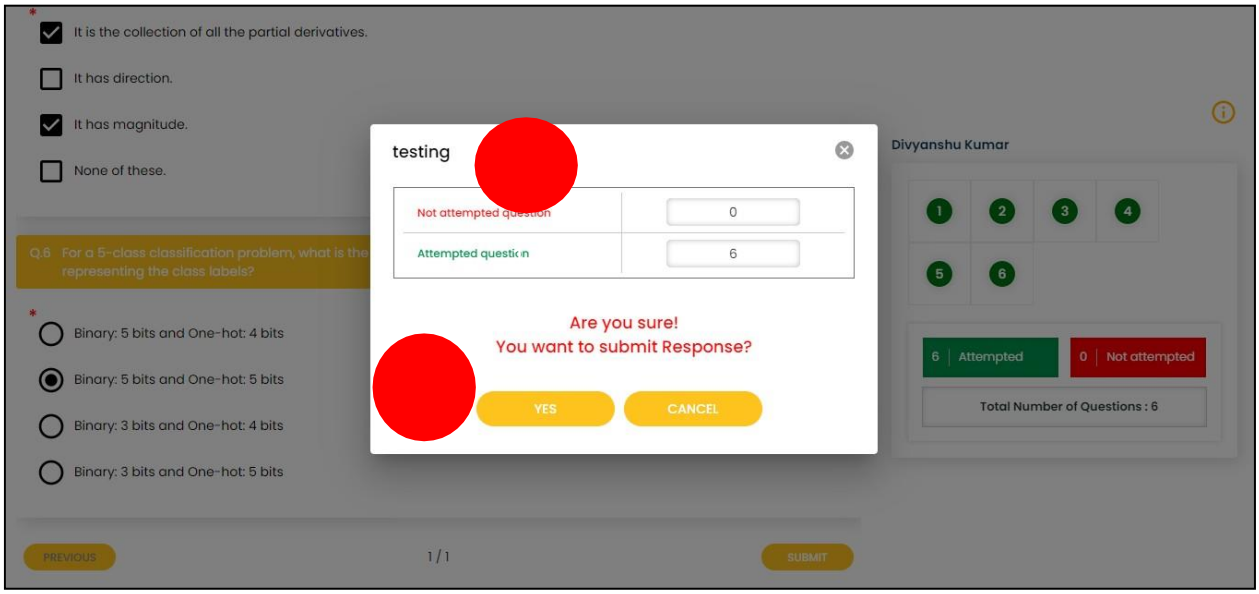
Step8. The user can see the assessment instructions again by clicking on the icon.



Step9. The users can see the timer running if the assessment is time-bound.

Step10. The users can move to previous questions by clicking on the previous button.

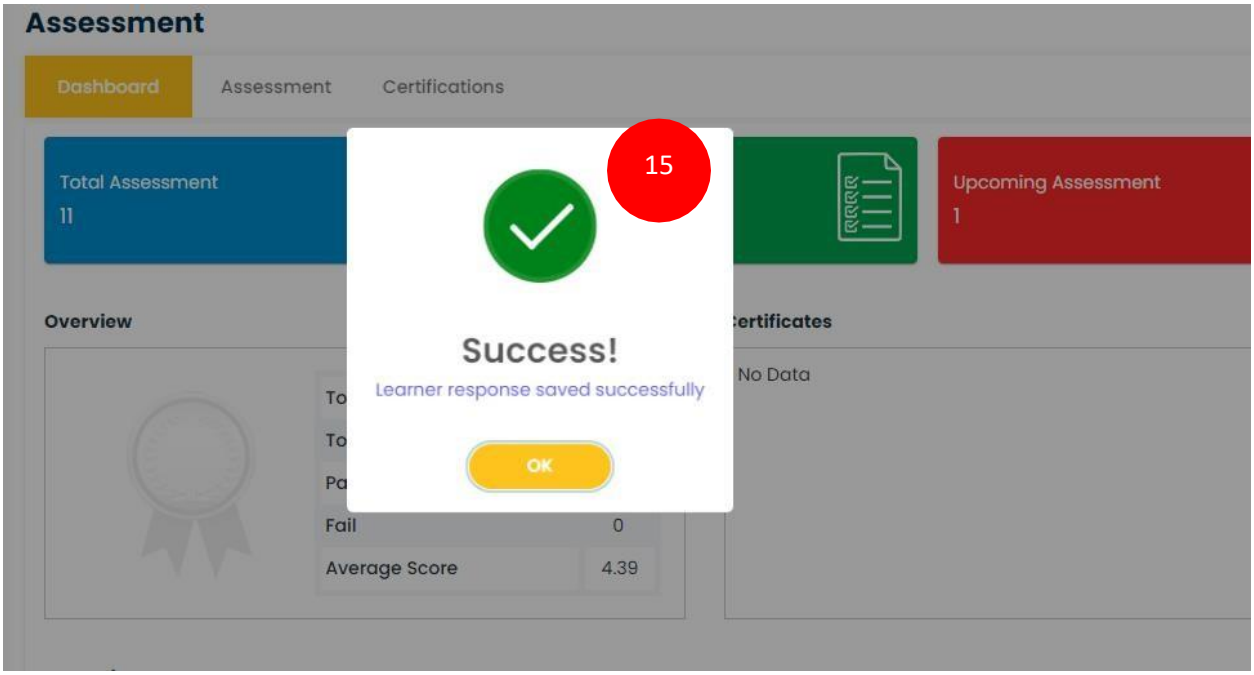
Step11. The user can submit the assessment by clicking on submit button



Step12. The users get the confirmation pop-up for the final submission.

Step13. The users can now click on Yes.

Step14. The user gets Thank you screen.



Step15. The user gets a Success screen after successful submission.

Development of Classroom, Announcement and Reports in Student Module of NARES-Blended Learning Platform

Classroom Module- Blended Learning Platform

The classroom model of BLP aims to combine traditional face-to-face instruction with technology enhanced learning experiences providing an interactive and engaging environment for students.

Key components:

1. **Interactive sessions:**
   a) **Instructor-Led Discussions:** Engage students in live and interactive discussions to deepen their understanding and knowledge.
   b) **Group exercises:** Incorporate collaborative group exercises to encourage critical thinking.

2. **Technology Integration:**
   a) **Supplementary materials:** Utilize digital resources such as videos, slides, or supplementary reading materials to complement in-class discussions.
   b) **Digital Whiteboards or Presentation Tools:** Enhance visual learning using digital boards to illustrate complex concepts and facilitate real-time collaboration.

3. **Assessment and Feedback:**
   a) **Formative Assessments:** Conduct in-class quizzes or short assessments to gauge understanding and provide immediate feedback.
   b) **Class Participation Evaluation:** Encourage active participation and provide feedback to students based on their engagement during discussions and activities.
4. **Collaborative Learning Environment:**

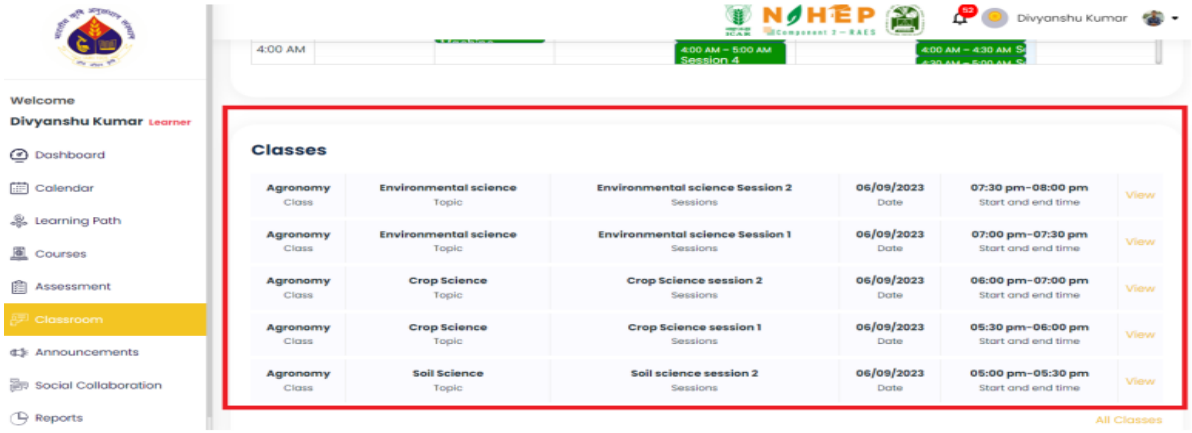
a) Peer-to-Peer Interaction: Encourage peer learning through group discussions, presentations, and peer review sessions.

b) **Facilitate Discussions:** Promote an open forum for students to ask questions, share insights, and explore diverse perspectives.

How to view classes?

Step 1: - Navigate to Classroom.

Step 2: - Click on "view all" to get access to all classes.



How to join a class?

Step 1: - Navigate to Classroom.

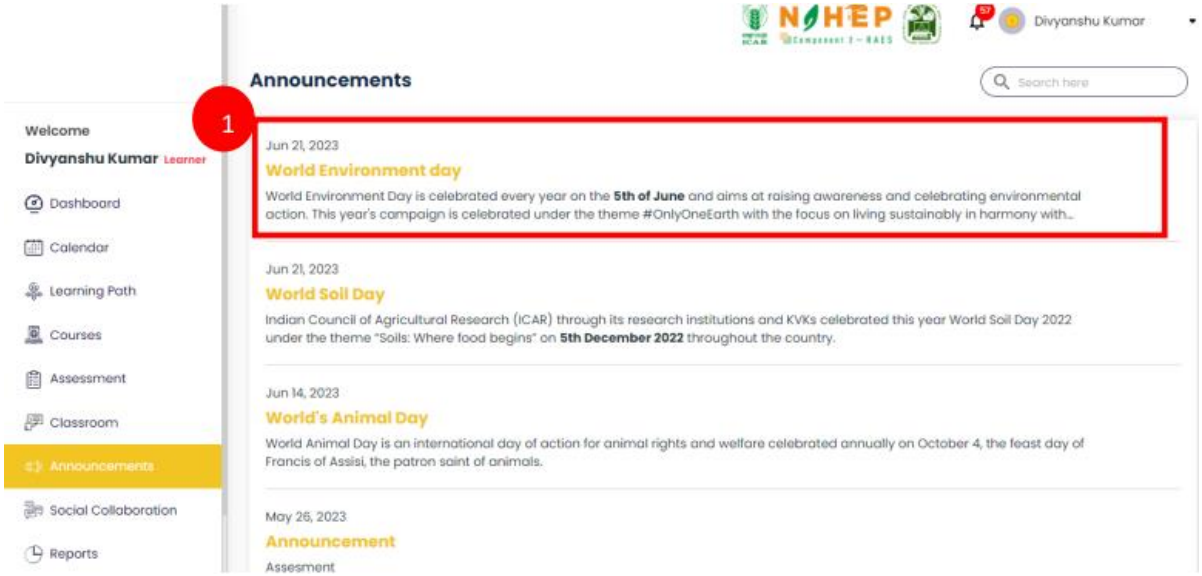
Step 2: - Click on “Play” icon to join the session.

Announcement Module- Blended Learning Platform

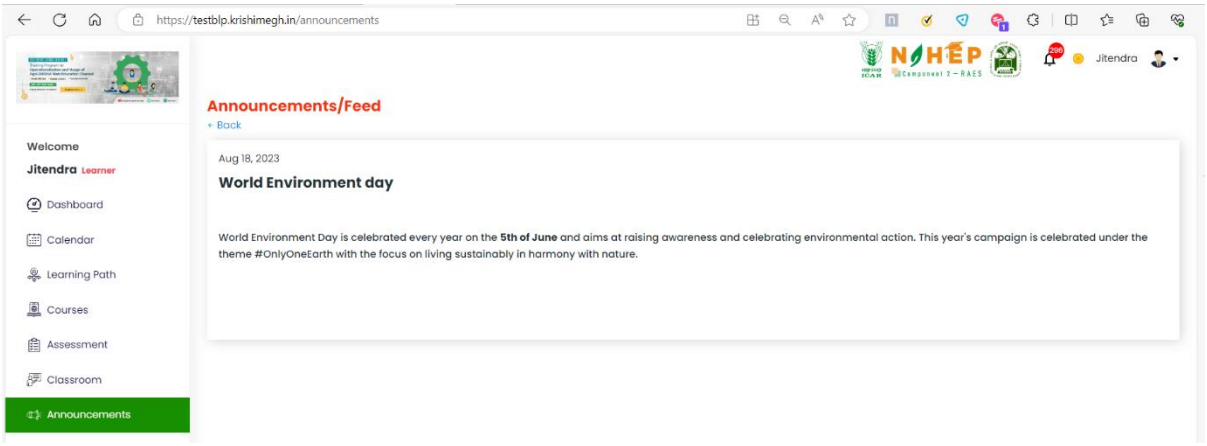
Announcement section with Blended Learning platform plays a crucial role for keeping student informed regarding an important event, course related updates or an important announcement.

How to view announcements?

Step 1: - Navigate to announcements.

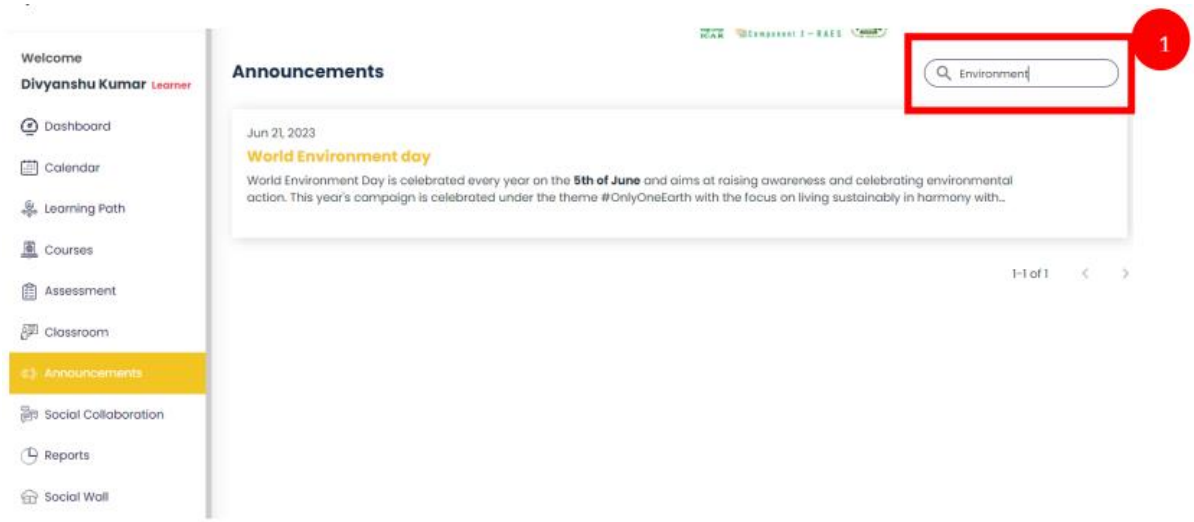


Step 2: - If a student clicks on an announcement, then they would be able to see that particular announcement.



How to search for an announcement?

Student can search for an announcement with the help of search functionality given in announcement section.

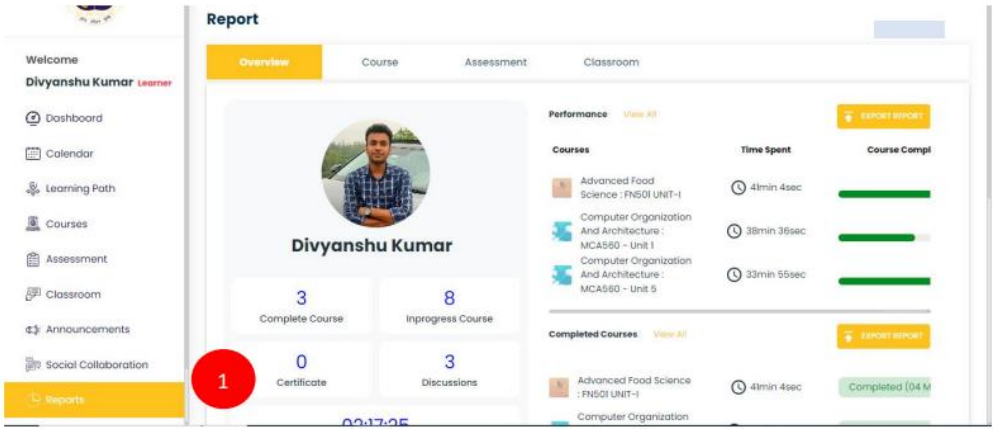


Reports Module- Blended Learning Platform

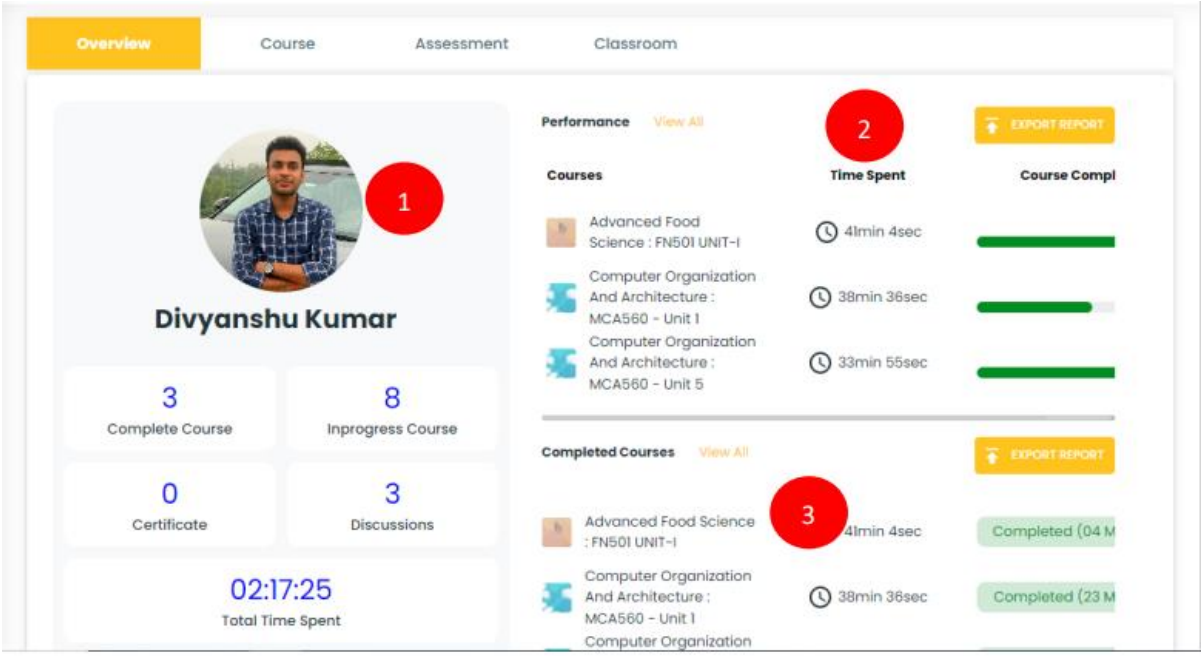
The report module in Blended learning Platform provides student with insights about their academic progress, performance metrics and classroom.

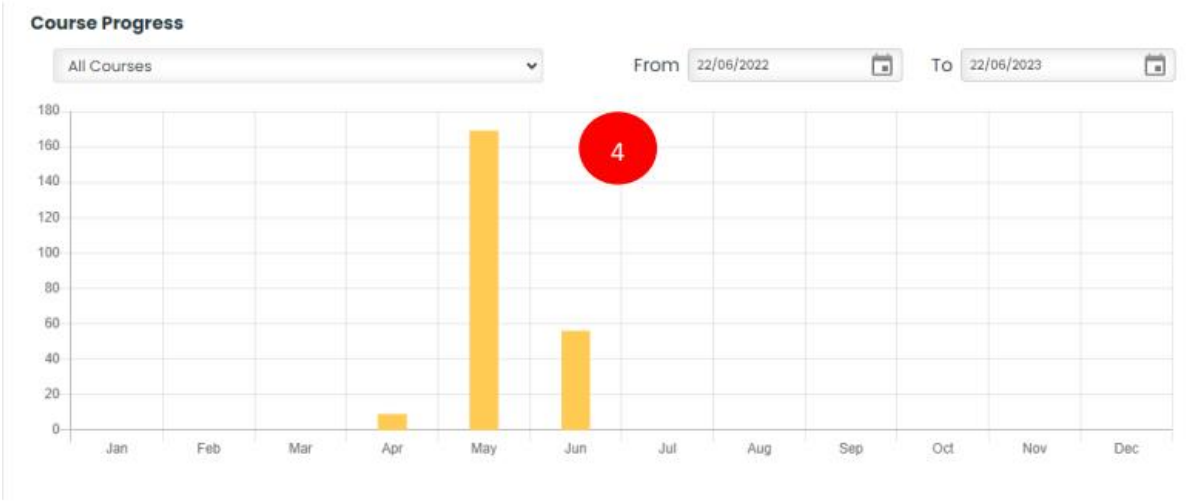
How to access reports?

Step 1: - Navigate to reports.



Step 2: - Click on overview to view overview report.





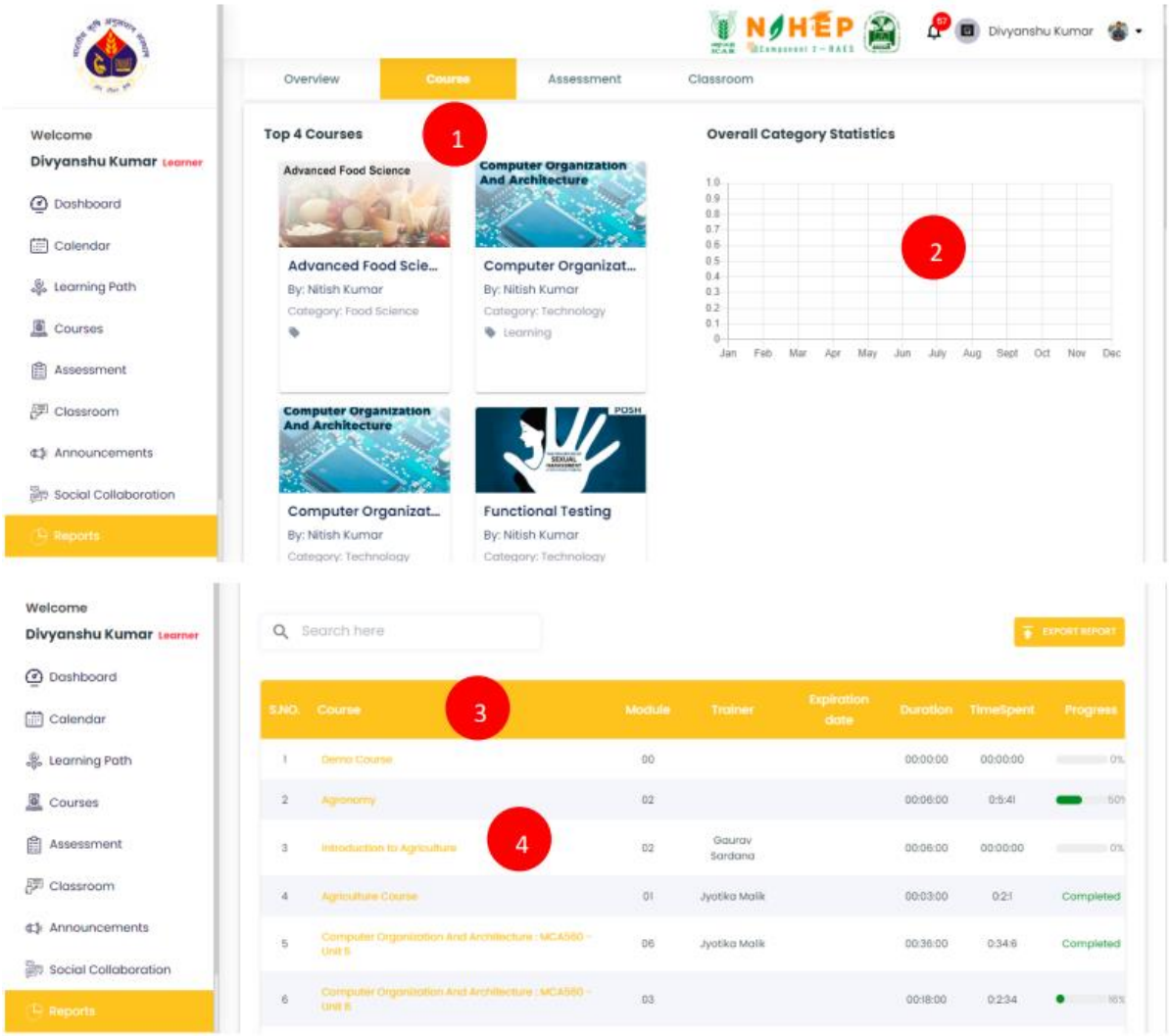
1: - BLP dashboard/report card

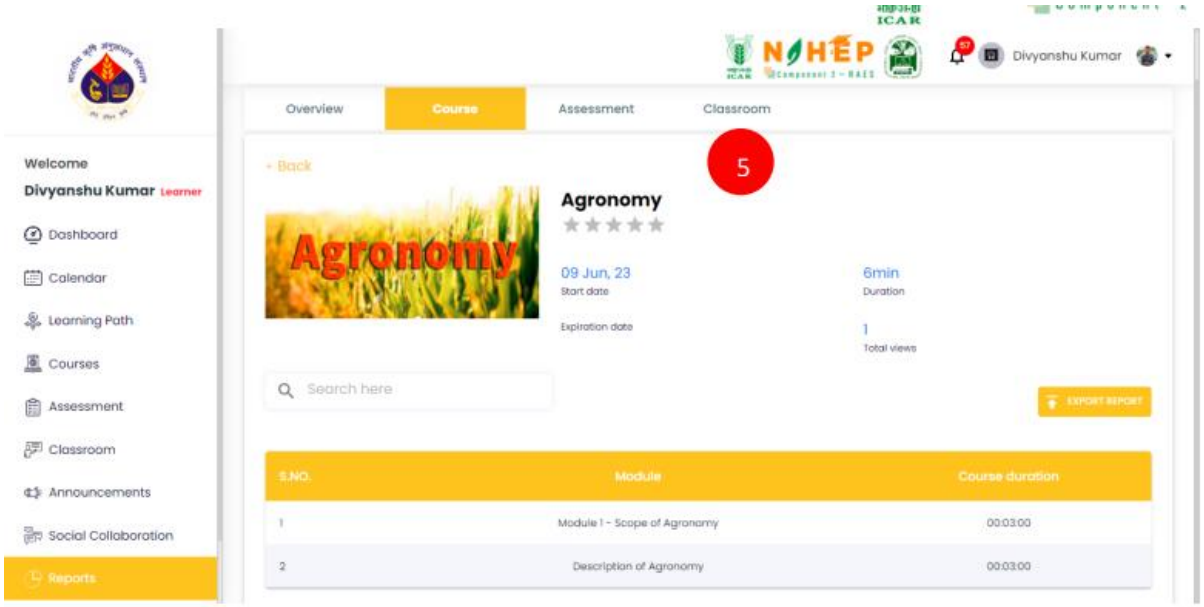
2: - Performance report

3: - Completed course report

4: - Course progress graph

Step 3: - Click on courses tab to view course report.





1: - Top courses

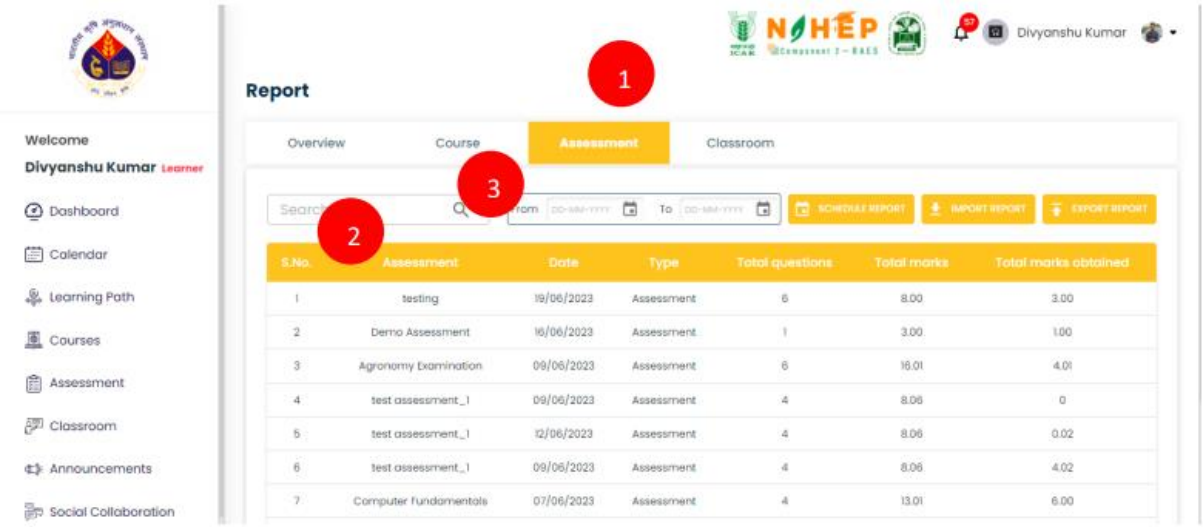
2: - Overall category statistics

3: - Course report

4: - User can click on individual course to view individual course report.

5: - Individual course report with module and duration

Step 4: - Click on assessment tab to view assessment report.

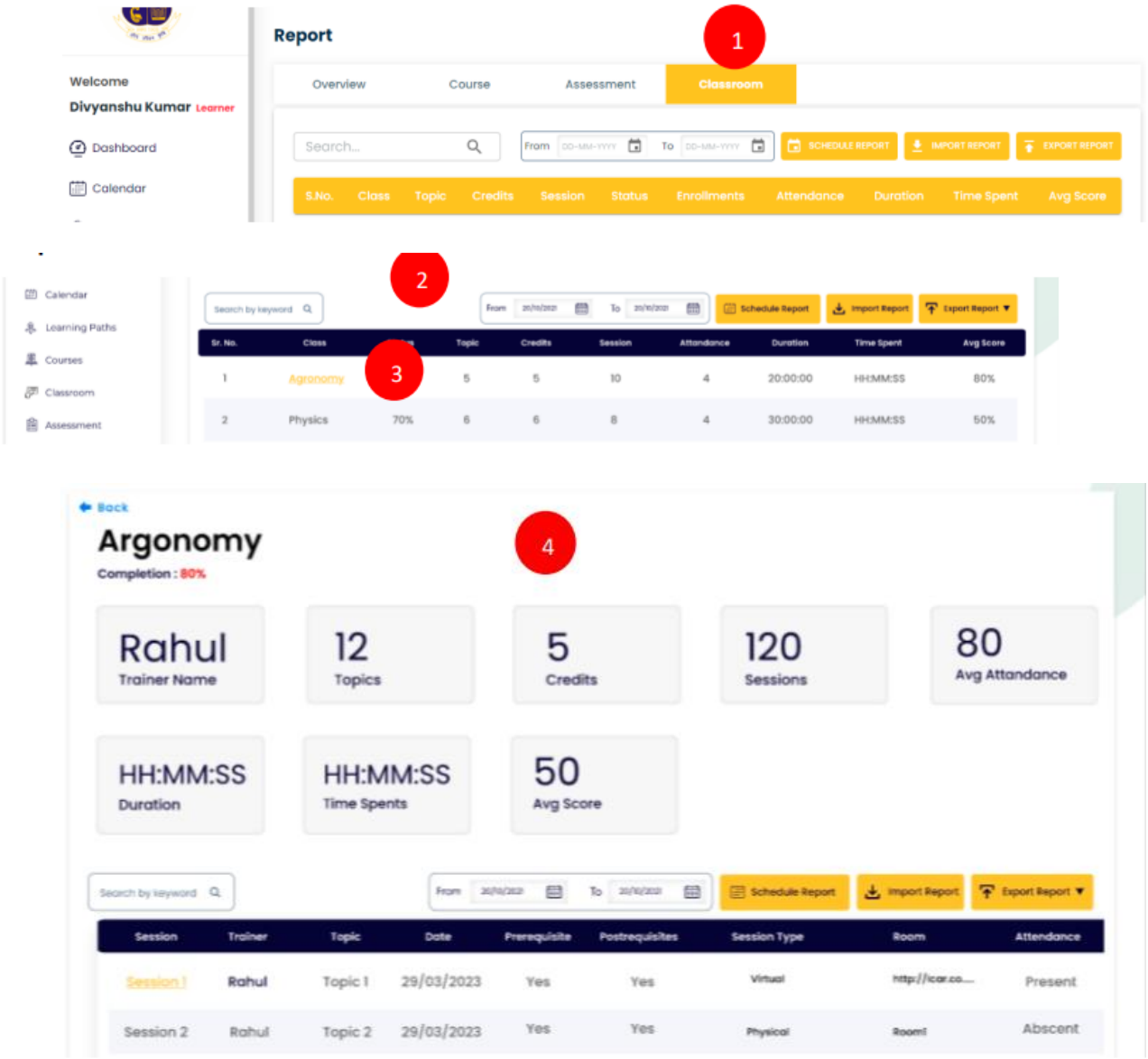


1:- Assessment tab

2:- Student can schedule, export or import report.

3: - Date filter with the help of which student can view assessment report for a particular period.

Step 5:- Click on classroom tab to view classroom report.



1: - Classroom tab

2: - Date filter with the help of which student can view classroom report for a particular period.

3: - Student can click on class name to view detailed report.

4: - Detailed report of class along with session details.

Development of Assessment and Announcement in Faculty Module of NARES-Blended Learning Platform

Assessment

The assessment module is the functionality that enables the faculty to create assessments in theform of

surveys, polls, assessments, and quizzes. Student's responses will be reviewed post completing and submitting the assessments. The assessment created can comprises various types of questions like

- Multiple Choice
- Multiple Response
- Large Open Answer
- Small Answer
- Matching

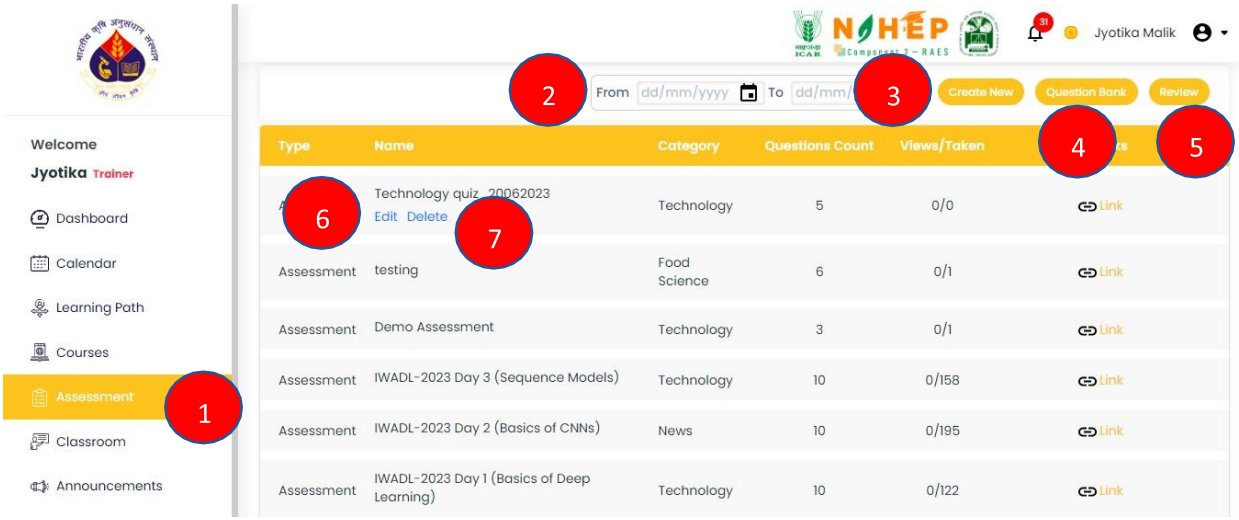
Questions can also be tagged through advanced options, and durations for the questions can also be set. Different types of messages can be set for the assessment, like messages to be displayed before quiz comment box messages, messages displayed at the end of the quiz, etc. Some other options like the number of questions per page, the time limit for the assessment, force submission after time expiry, the number of times the assessment can be attempted, and the start time and end time of the assessment.

Once assessment framing is complete from the faculty end, the same can be assigned to the students. Students will be notified through notifications about the assessments.

Other features of the Assessment Module are as follows:

- Conduct online tests, assignments, and examinations with efficient mechanisms of evaluation and feedback.
- Ability to weigh and grade individual questions within an assessment.
- Get notifications for upcoming assessments.
- View assessment feedback.
- Ability to Create polls/surveys.
- Ability to report on question-level data from surveys and assessments/tests.
- Assign to Individual Learner or a Group in a fixed or Random Mode Bulk Upload.

Step1. The users can navigate to the assessment module by scrolling down the menu bar on the left side.
Step2. Users can set the date range to filter assessments.
Step3. Users can click on 'Create New' button to create new assessments.
Step4. Users can click on 'Question Bank' to Import, Export, or Add new questions to the question bank.
Step5. Users can click on 'Review' button to review the assessments sent for review.
Step6. Users can click on 'Edit' to edit the assessment.
Step7. Users can click on 'Delete' to delete the assessments.



Create Assessment

Step1. The user Clicks on 'Create New' button.

Step2. A Create New pop-up will appear.

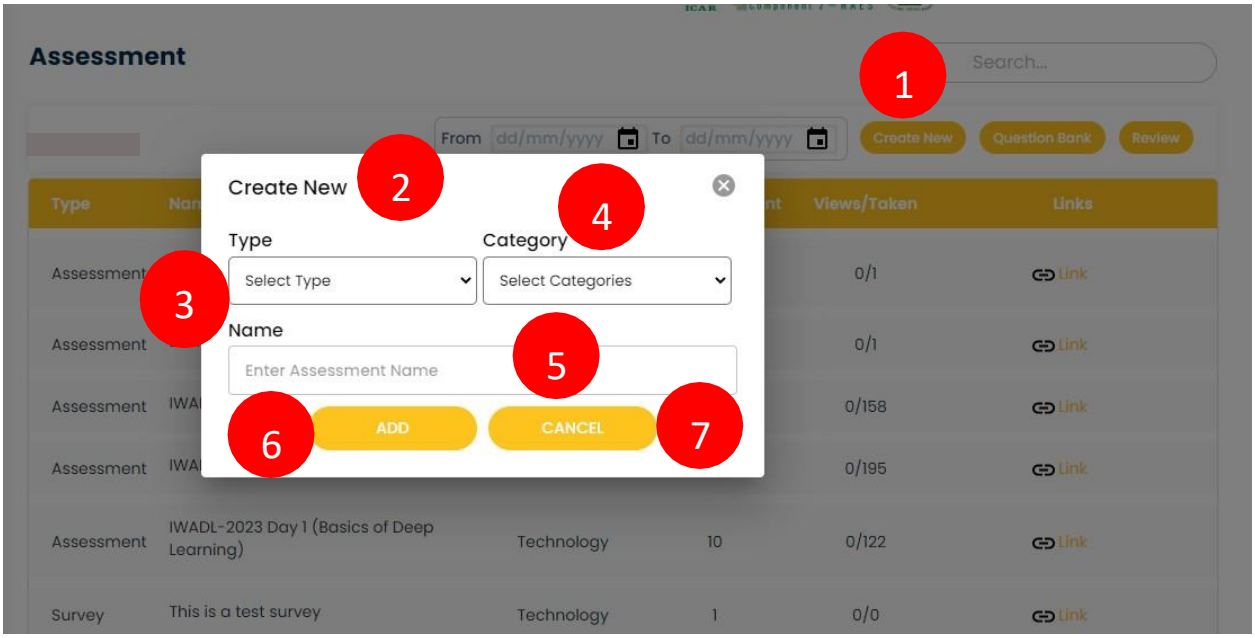
Step3. The users select the assessment types.

Step4. The users select the assessment category.

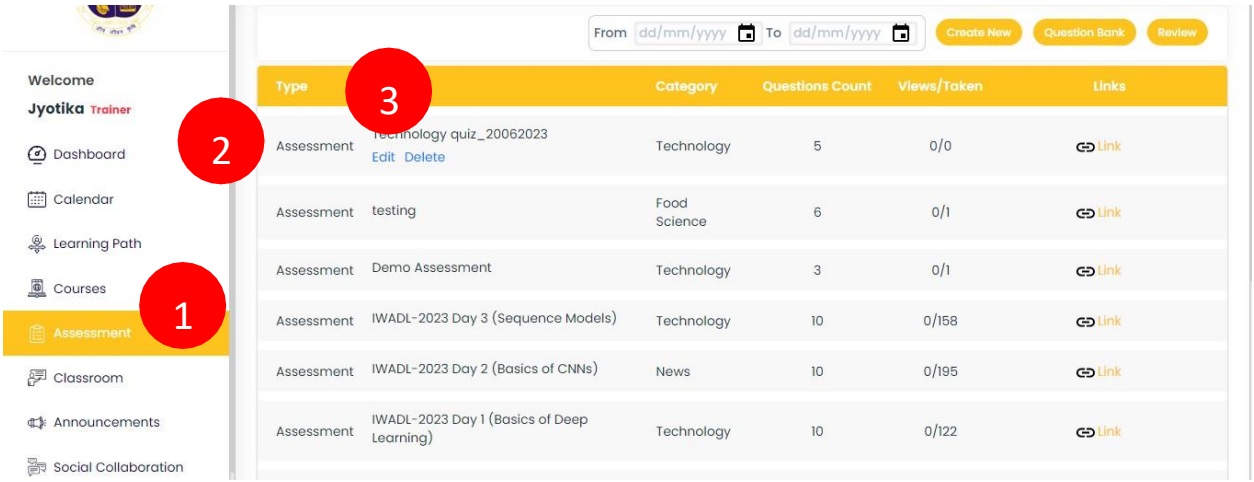
Step5. The users enter the assessment name.

Step6. The users click on 'Add' button to add the assessment framework.

Step7. The users can click on cancel if he/she wants to cancel the assessment.

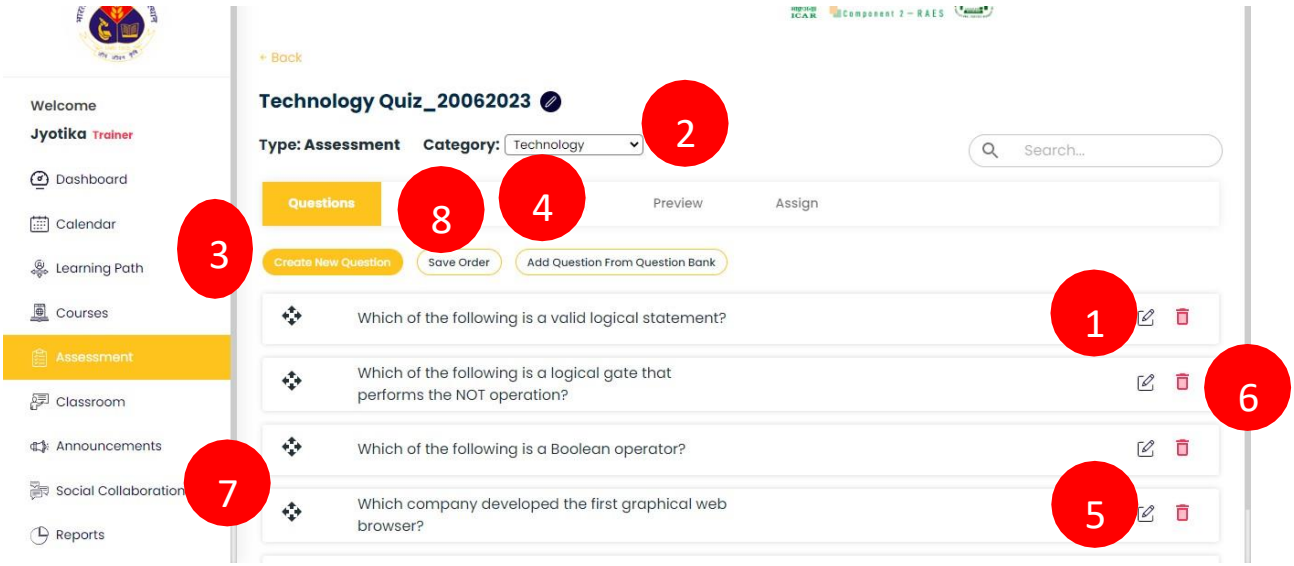


Add Questions



Step1. The users click on the assessment module.
Step2. The users find the assessment name.
Step3. The users click on edit.

Add Question Screen



Step1. The users click on the edit icon to edit the assessment name.
Step2. The users can change categories through the drop-down.
Step3. The users can click on create new questions to add new questions.
Step4. The users click on Add question from Question Bank to add questions from questionbank.
Step5. The user clicks on the edit icon to edit the question.
Step6. The user clicks on the delete icon questions to delete the question.
Step7. The user clicks on the drag icon to drag the question to change the order.

Step8. The user clicks on the save order icon to save the order.

Steps to Create Questions

Multi-Choice

Step1. The users can set the question as a multi-choice through the drop-down

Step2. The users can type the question.

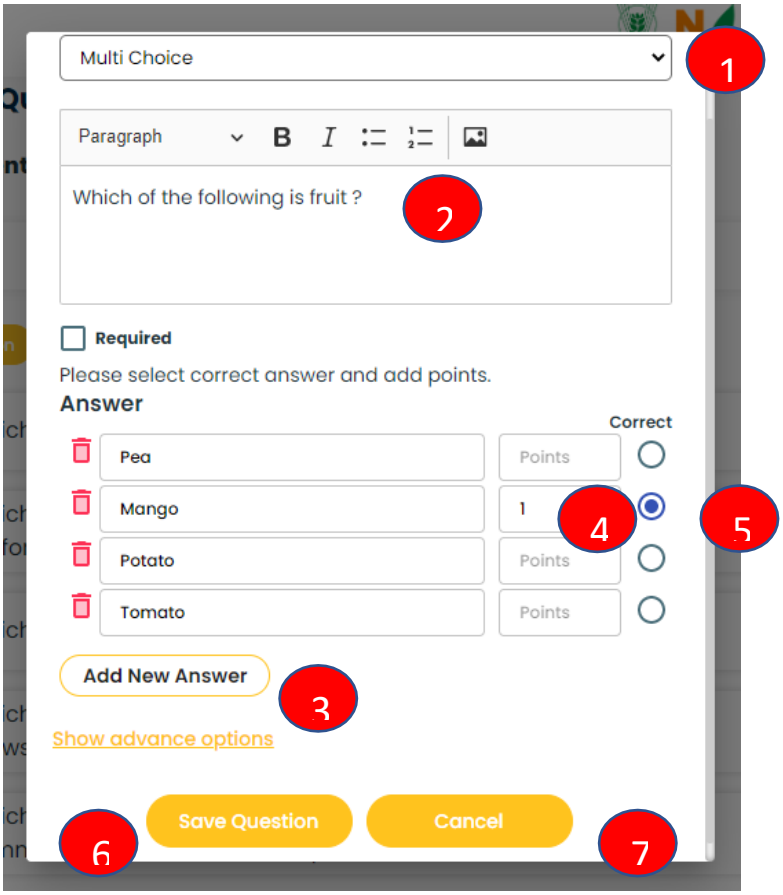
Step3. The users clicks on Add New Answer to add the answers.

Step4. The users enter the point for the correct answer.

Step5. The users can check the correct radio button.

Step6. The users click on ‘Save Question’ to save the question.

Step7. The users click on the ‘Cancel’ button to cancel the question.



Large Open Answer

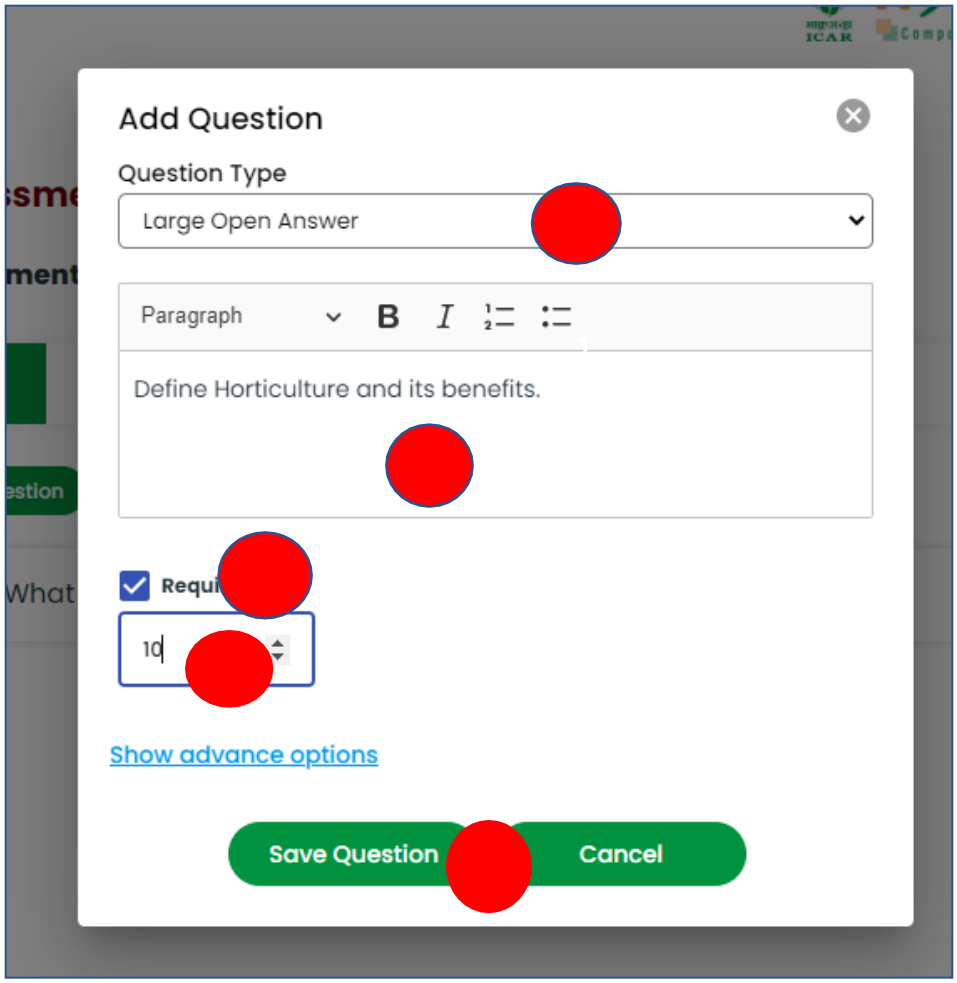
Step1. The user can set the question as Large Open Answer through the drop-down

Step2. The users can type the question.

Step3. The users check the required option.

Step4. The users enter the point details.

Step5. The users click on Save Question to save the question.



Small Answer

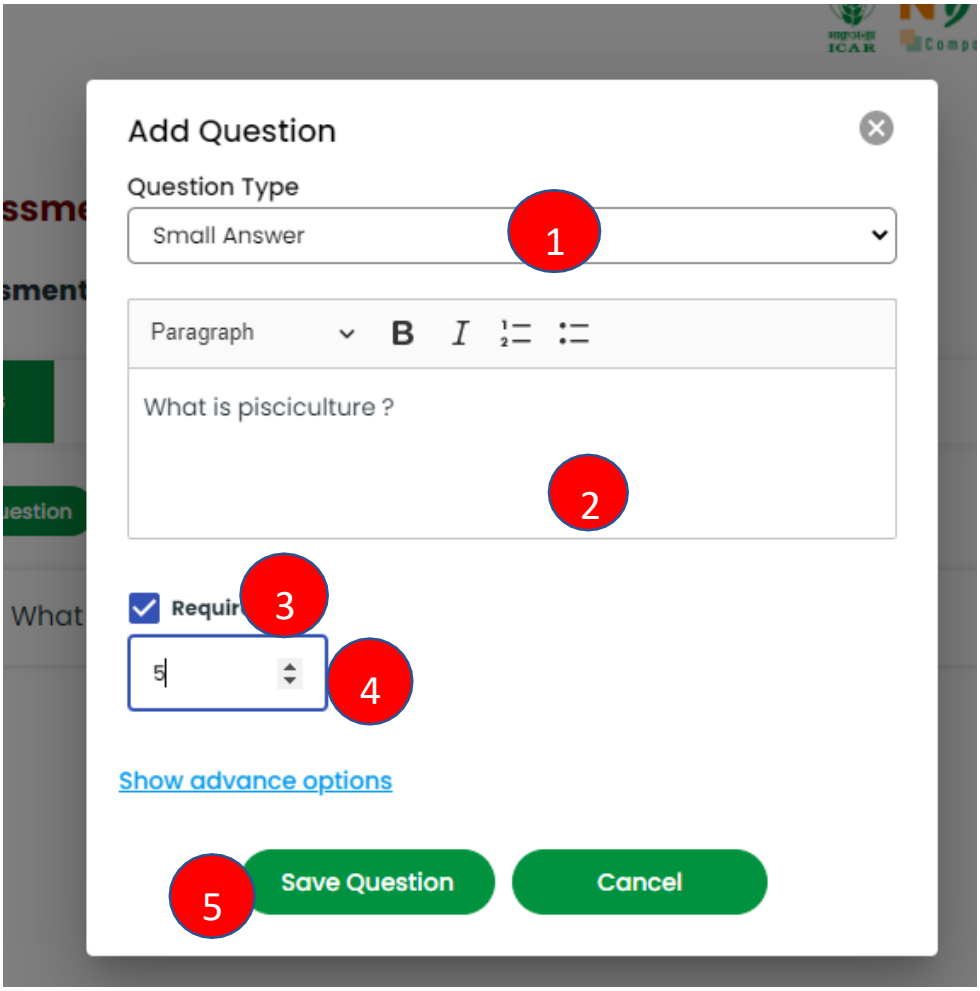
Step1. The users can set the question as Small Answer through the drop-down

Step2. The users can type the question.

Step3. The users check the required option.

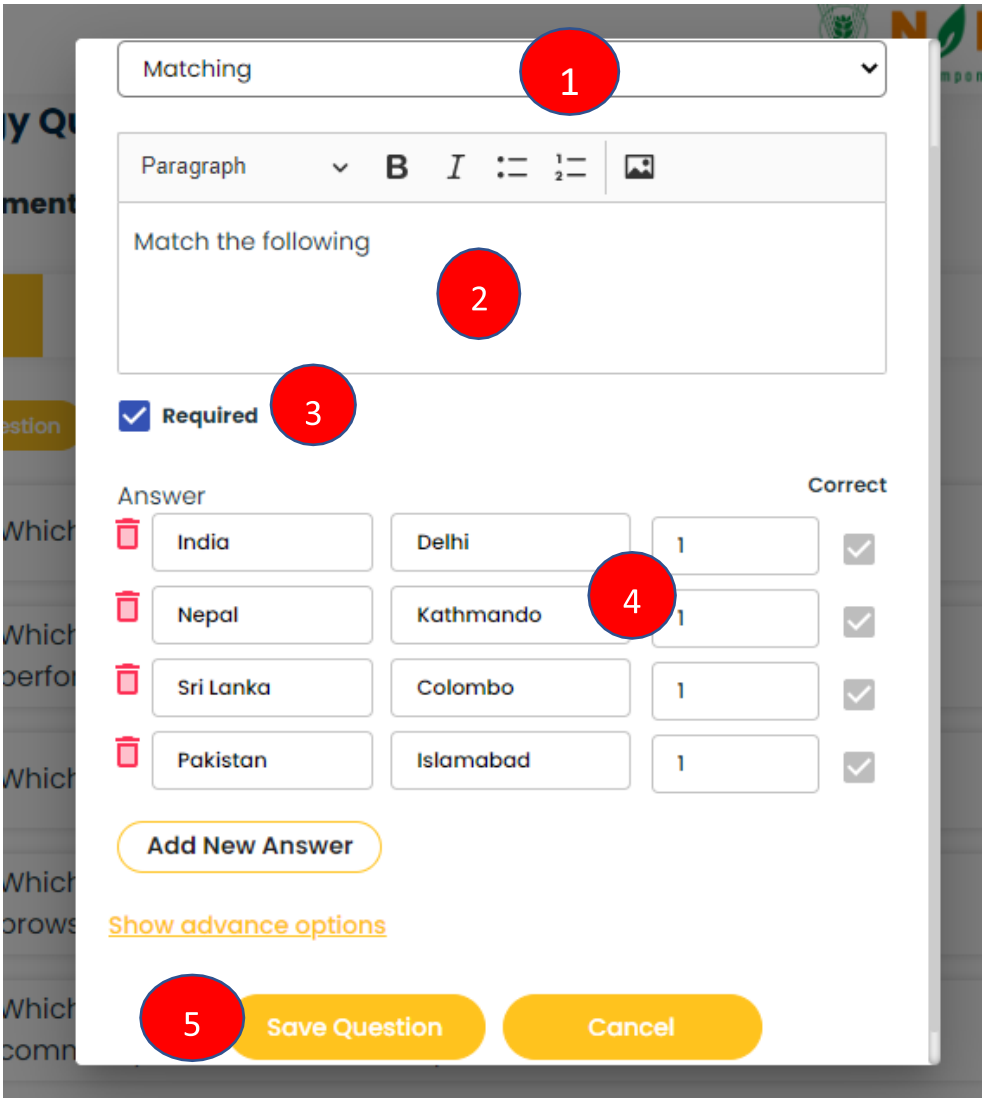
Step4. The users enter the point details.

Step5. The users click on Save Question to save the question.



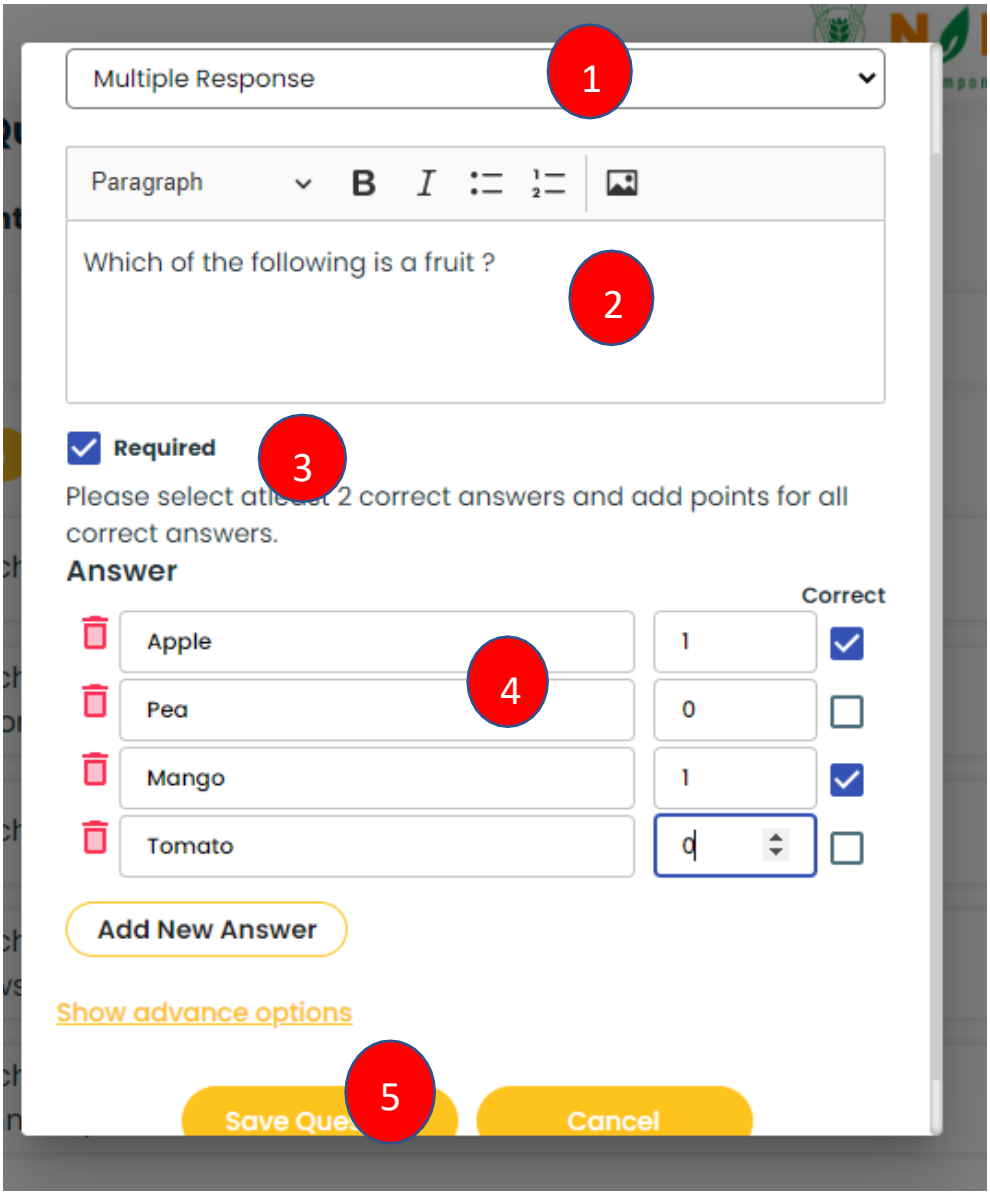
Match the following

Step1. The users can set the question as Matching through the drop-down
Step2. The users can type the question.
Step3. The users can check the required option.
Step4. The users add the answer and assigns the points and checks the correct option.
Step5. The users click on the save question.

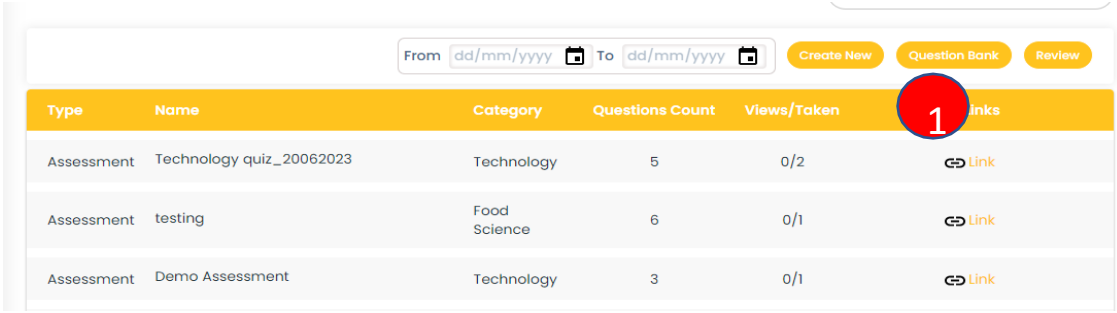


Multi Response

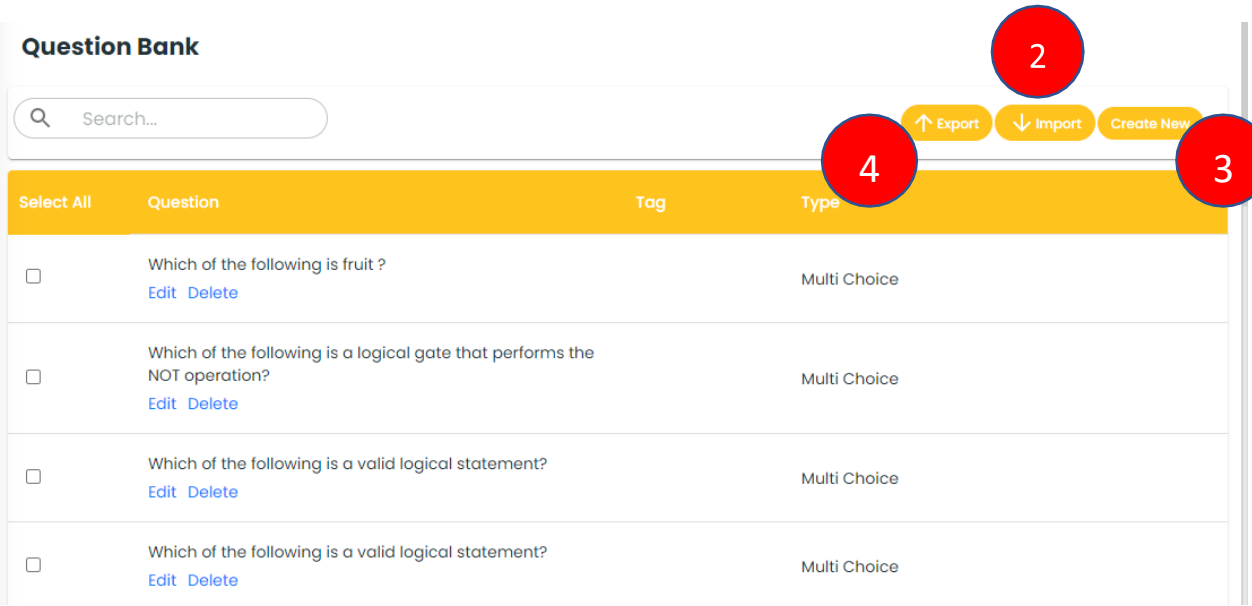
Step1. The users can set the question as Multiple Responses through the drop-down
Step2. The users can type the question.
Step3. The users check the required option.
Step4. The users add the answers, provides the points, and checks the correct option.
Step5. The users click on Save Question to save the question.



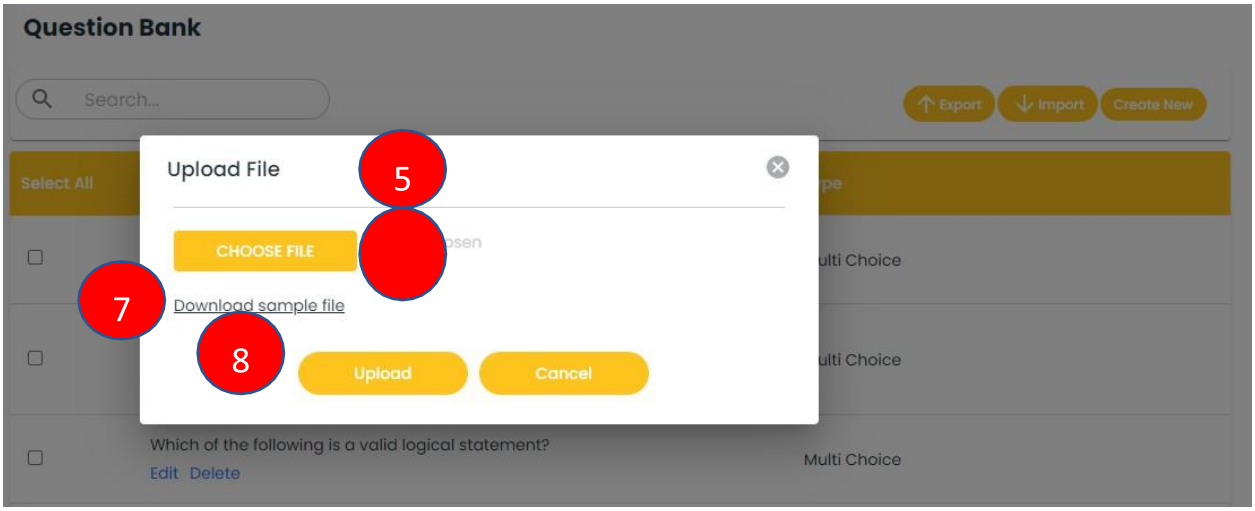
Steps to Create/Export/Import Question Bank



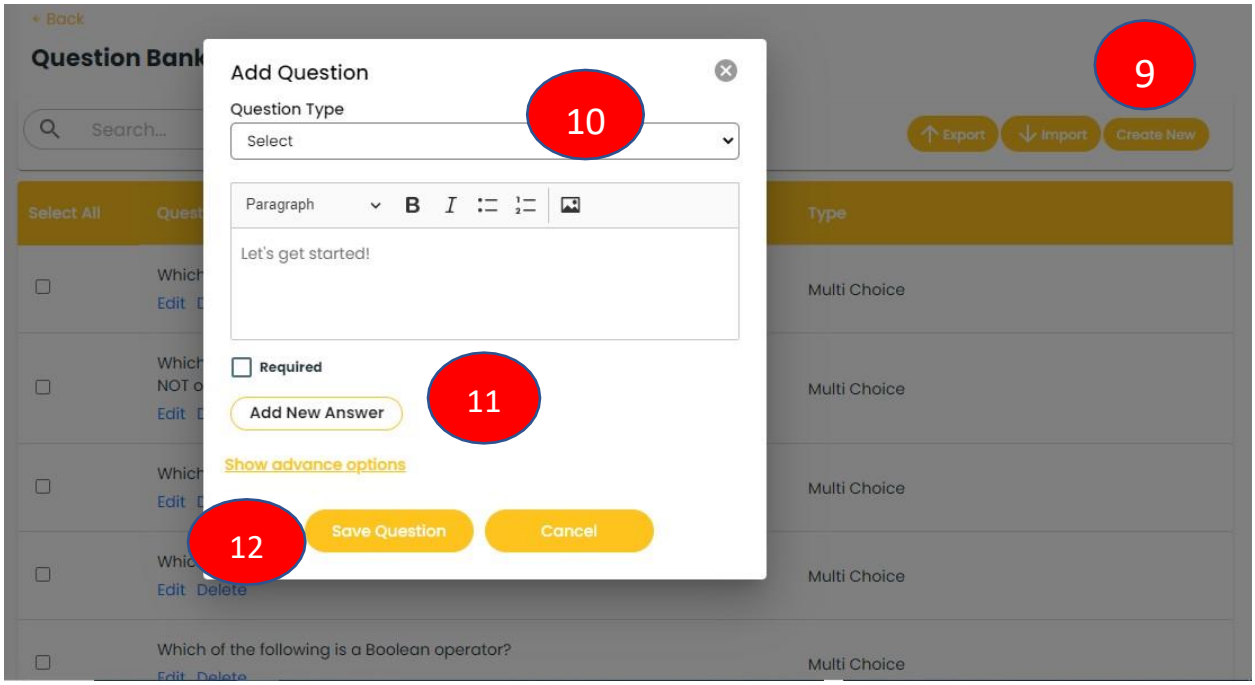
Step1. The users can create a question by clicking on the question bank.



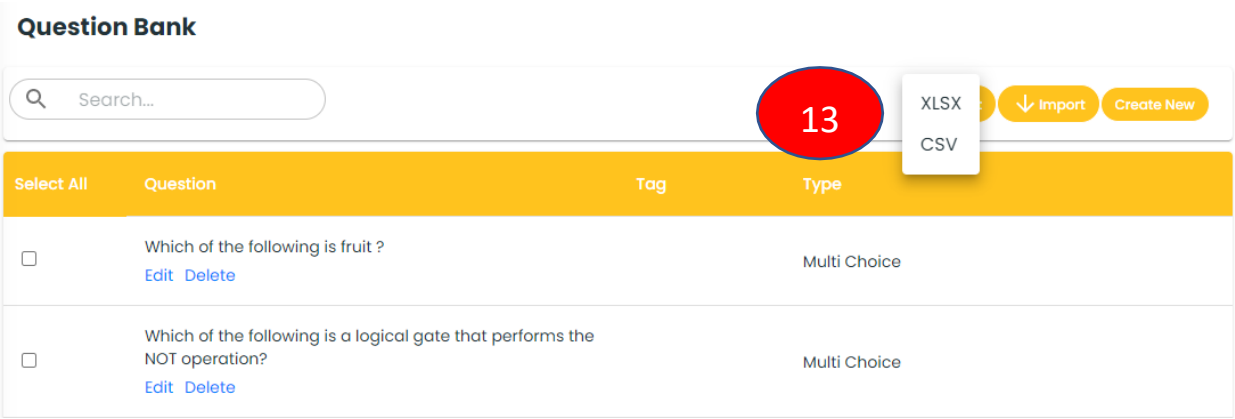
Step2. Users can import the questions by clicking on the import button.
Step3. Users can add new questions from 'Create New' button.
Step4. Users can export questions from the Export button.



Step5. The users get the upload file screen when he/she clicks on the import button.
Step6. Users can now choose the file to upload by clicking on choose file.
Step7. Users can download the sample file to prepare the question sheet to be uploaded.
Step8. The users click on the upload button once have browsed and selected the question file.

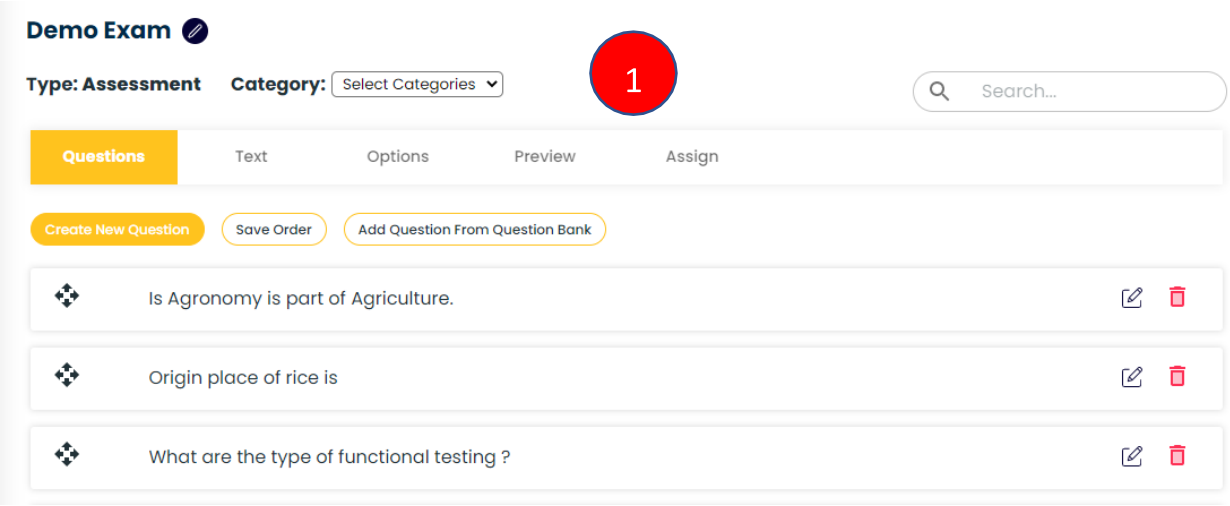


Step9. The users click on create new button to add question screen.
Step10. The users select the question type and types of the question.
Step11. Users click on the Add New Answer button to add answer options.
Step12. The users can click on Save Question to save the questions.

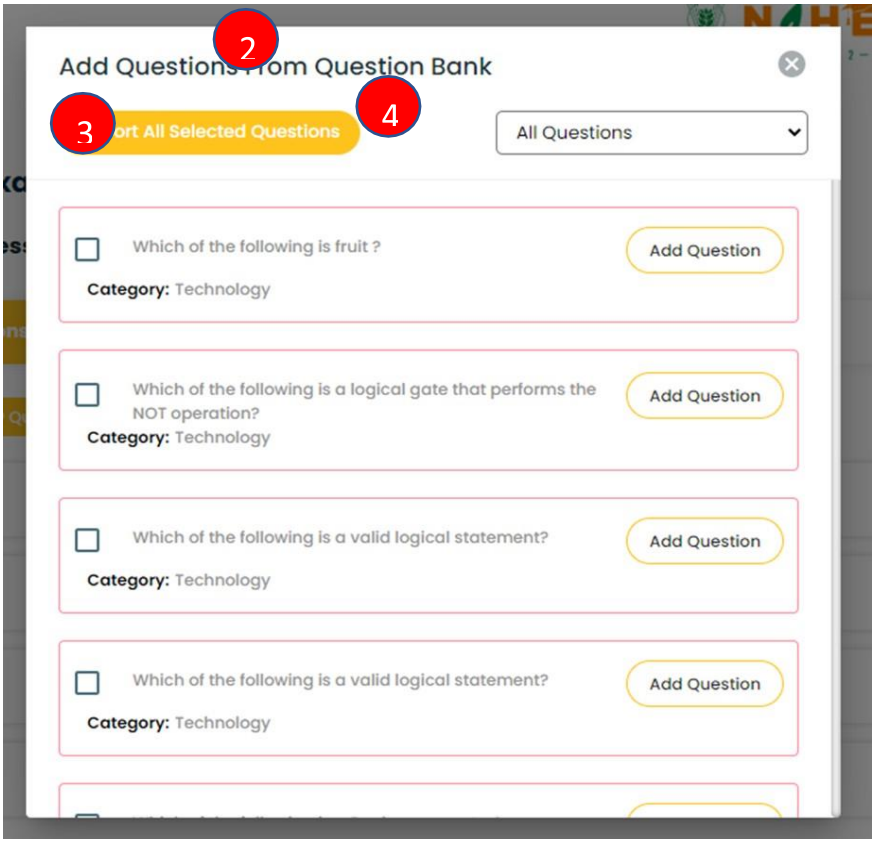


Steps13. The users click on the export button to export the questions in .xlsx and .csv.

Steps to Create Assessment from Question Bank

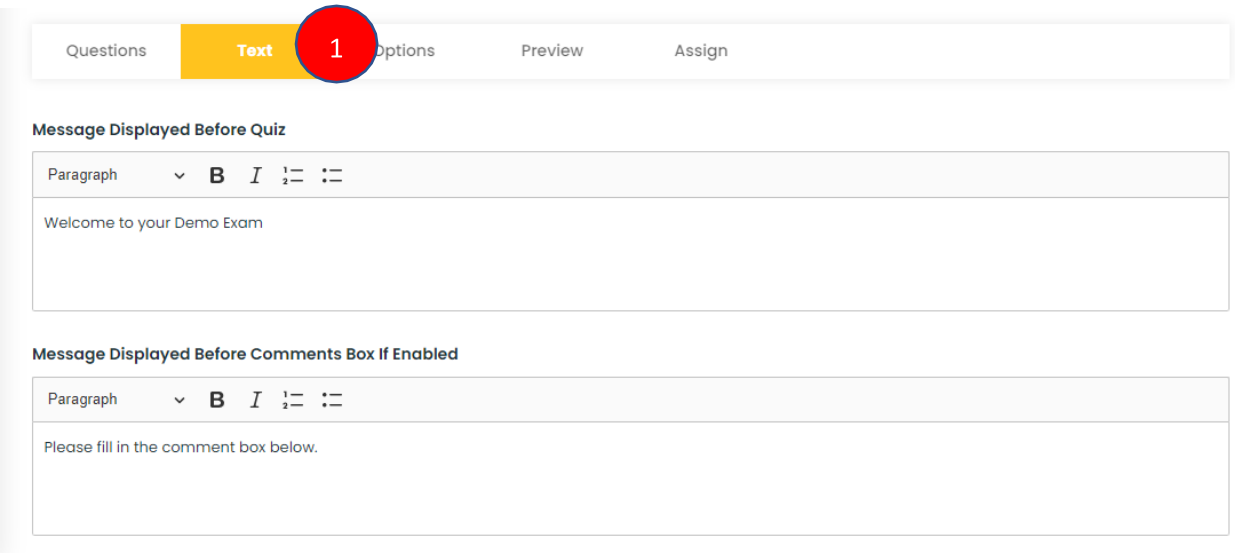


Step1. The users click on Add Questions from the question bank.

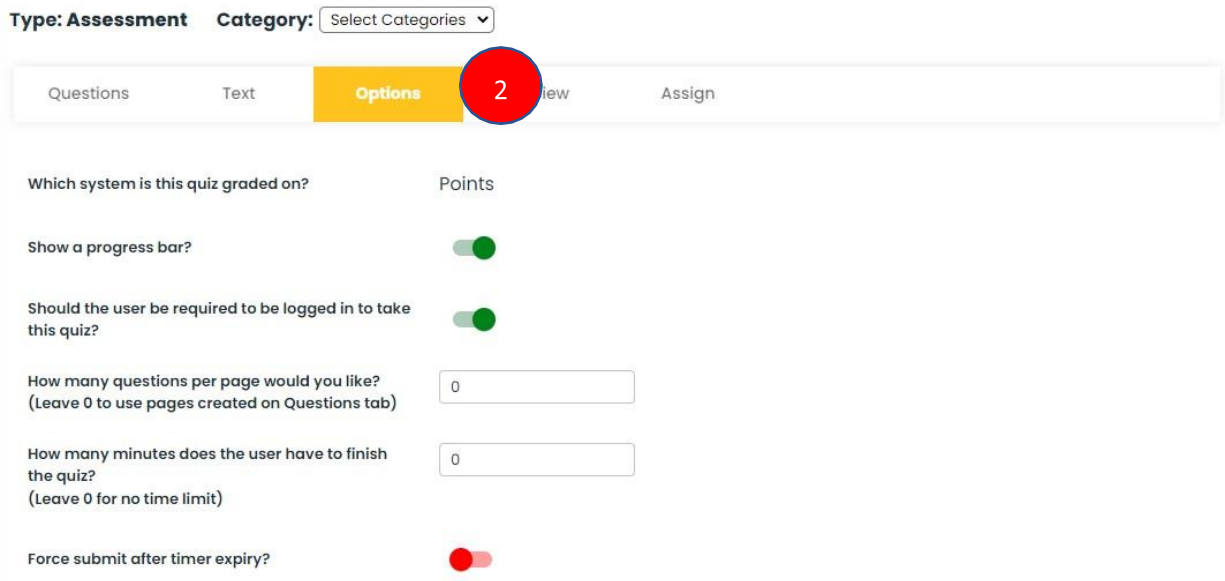


Step2. Add Questions from Question Bank Screen appears.
Step3. Select a question to be imported and click on Import All Selected Questions.
Step4. Users can use all question filters to filter questions.

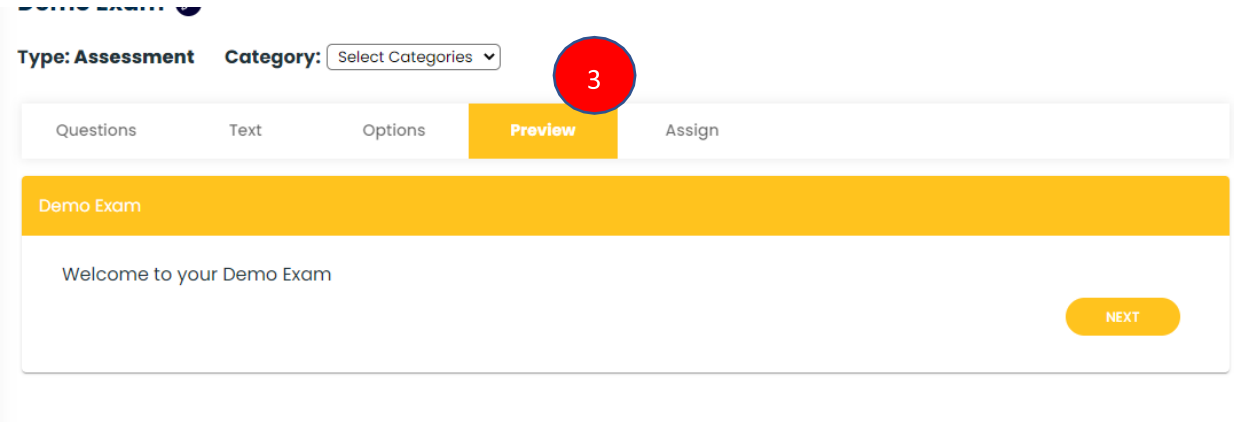
Managing Assessment Configuration



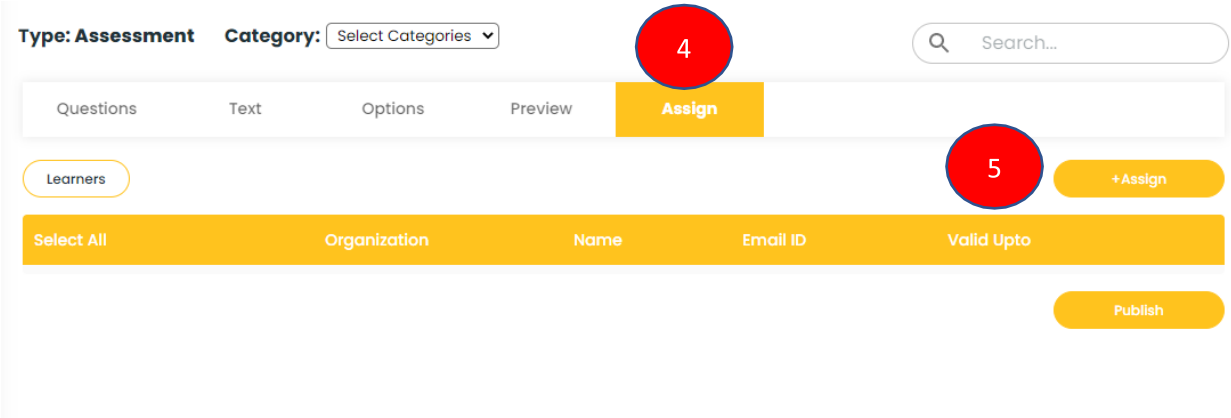
Step1. The users click on the ‘Text’ tab and can now edit the pre-configured messages from thissection.



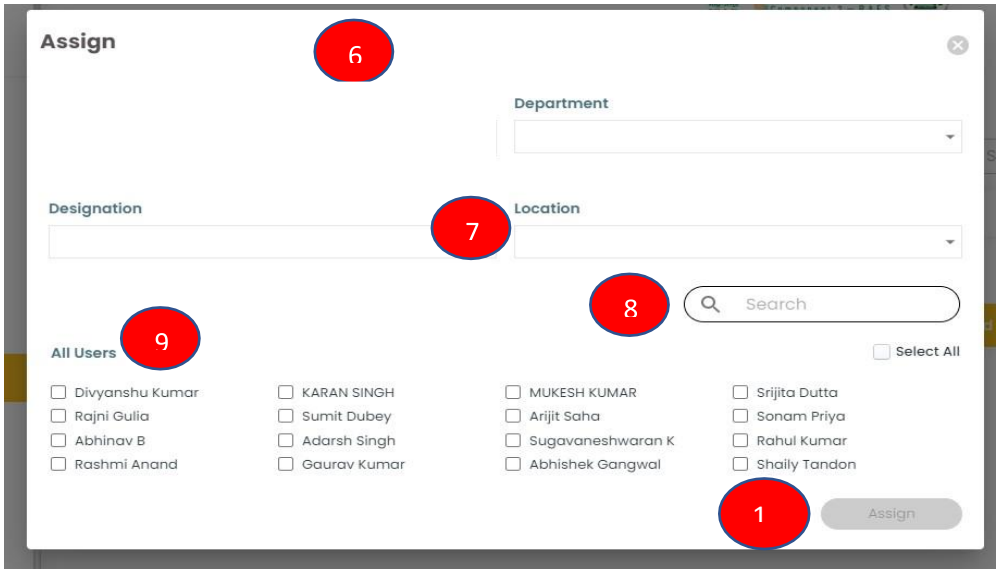
Step2. The users click on the ‘Options’ tab to edit the assessment option from this section.

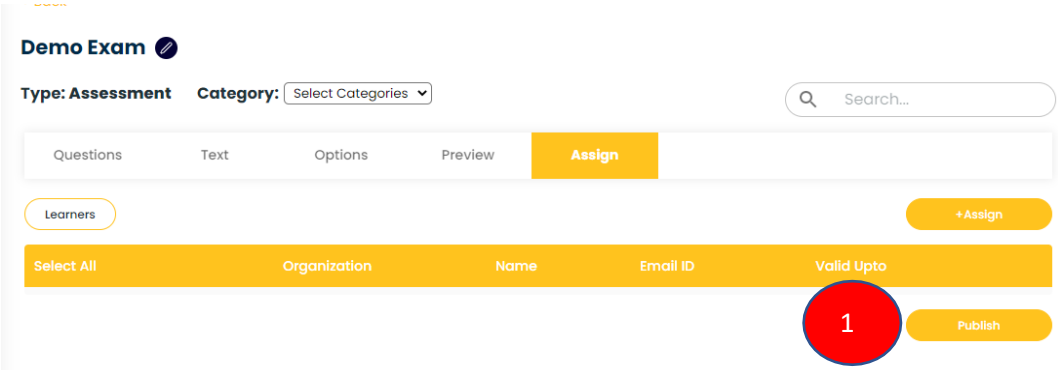


Step3. Users can click on the Preview tab to get a preview of the assessment.



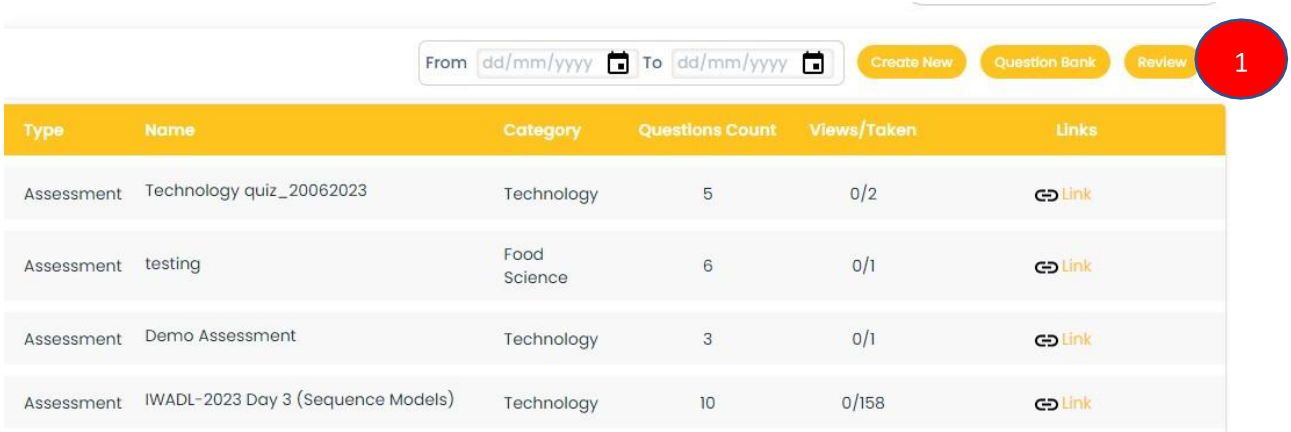
Step4. Users can click on Assign to assign the assessment to the students.
Step5. The users click on +Assign Button to assign the assessment to the students.
Step6. The users get the assigned screen.
Step7. The users select department, designation, and location from the drop-down.
Step8. Users can search for specific users from the search bar.
Step9. Users can select specific users from the list of users.
Step10. The users click on assign once the students have been selected.



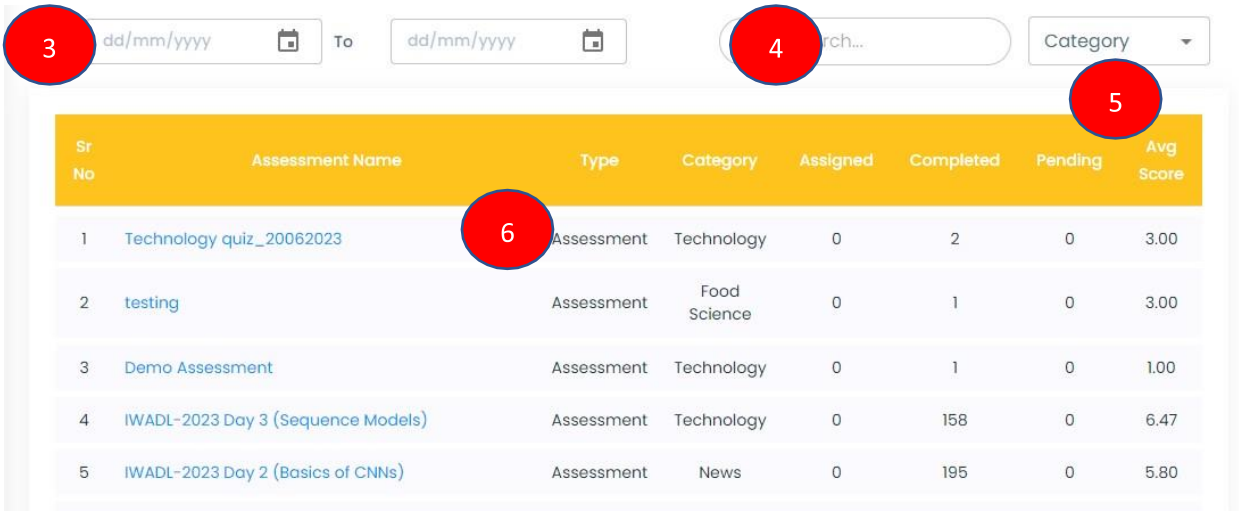


Step11. The users click on 'Publish' button to publish the assessment.

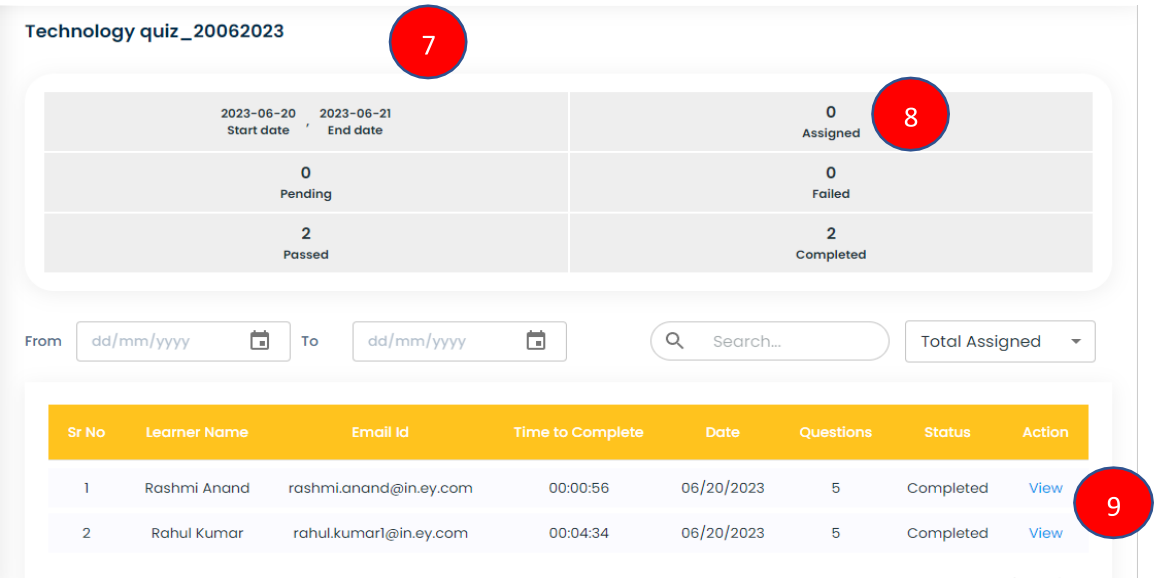
Assessment Review



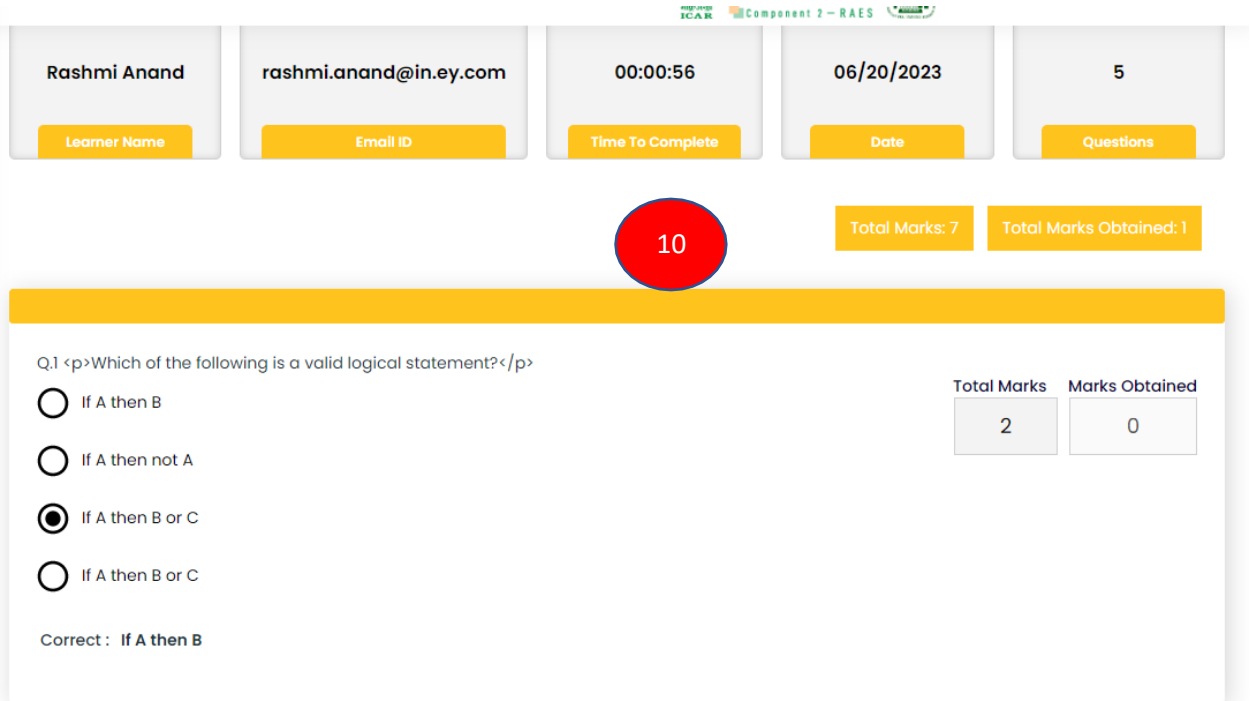
Step1. The users click on the 'Review' button to view the assessments.



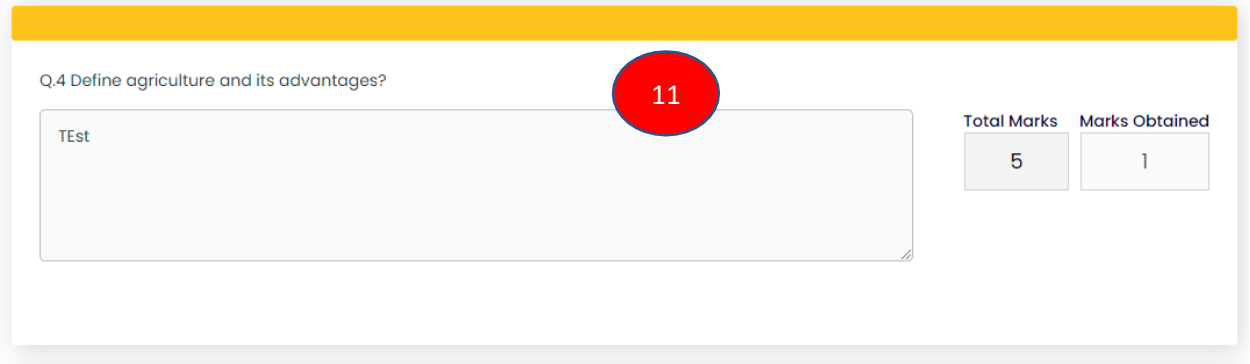
Step2. The users get the assessment review screen.
Step3. Users can set the date range.
Step4. Users can search for the assessment name from here.
Step5. The users can set the filter of category from here.
Step6. The users click the assessment name hyperlink to open the review screen.



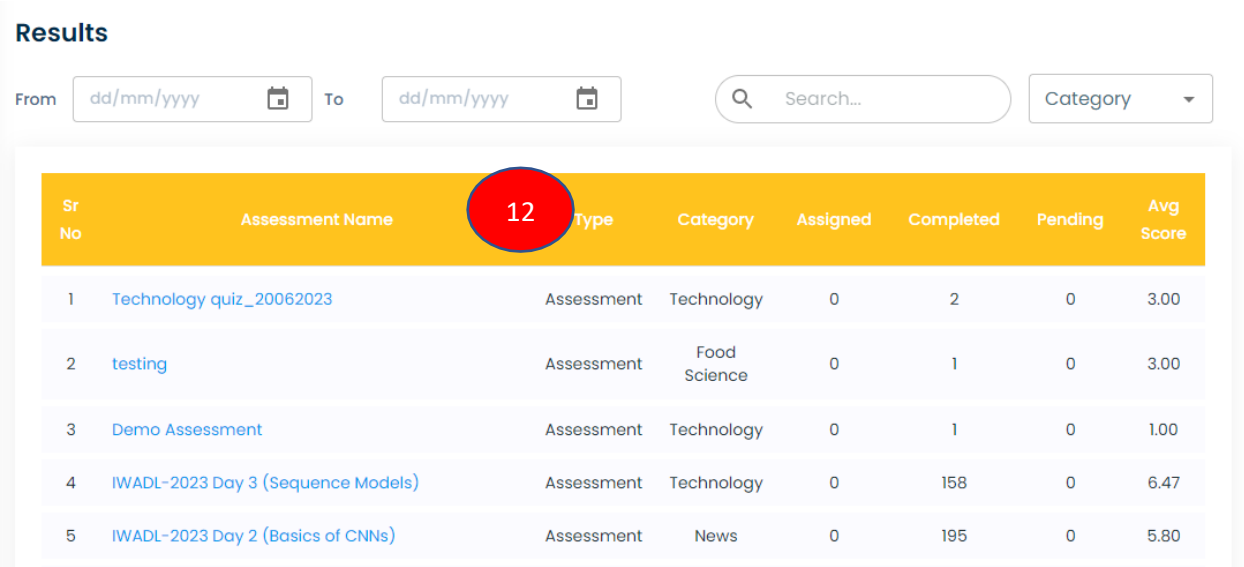
Step7. The users get the assessment review screen.
Step8. The users get the details of the assigned, pending, passed, and failed completed assessments.
Step9. The users click on view to review the assessment of the student.



Step10. The users get the review screen with all the student details.



Step11. The users can only review the long or short answer type question and therefore allocate marks in the marks obtained section and finally clicks on save.



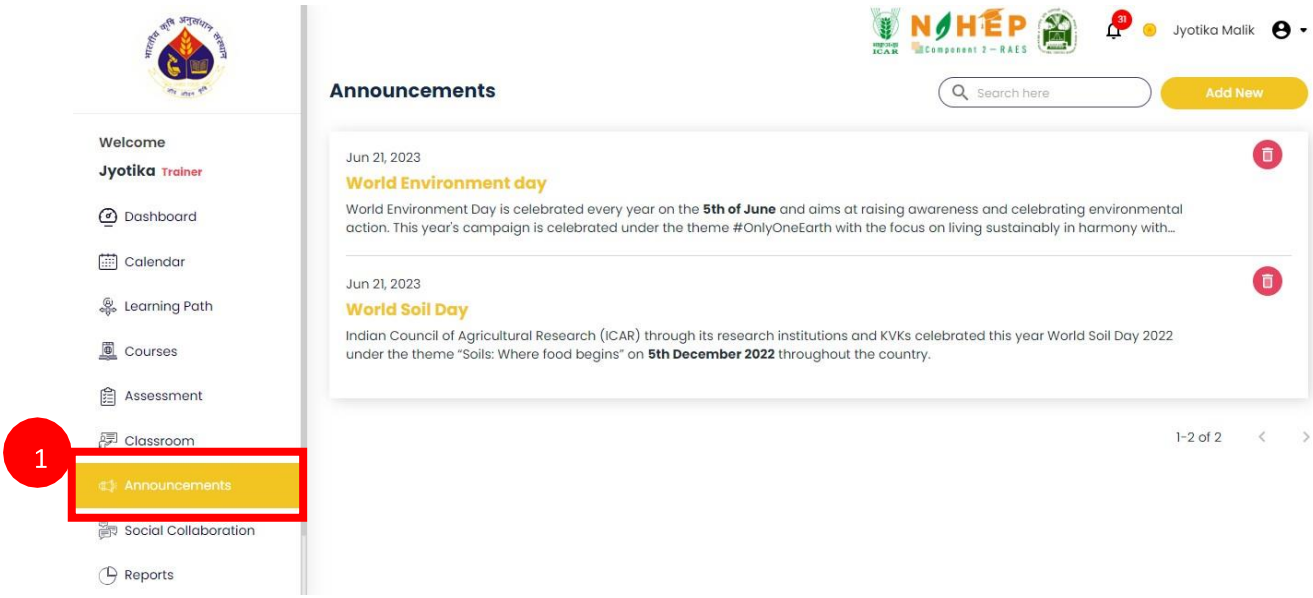
Sr No	Assessment Name	Type	Category	Assigned	Completed	Pending	Avg Score
1	Technology quiz_20062023	Assessment	Technology	0	2	0	3.00
2	testing	Assessment	Food Science	0	1	0	3.00
3	Demo Assessment	Assessment	Technology	0	1	0	1.00
4	IWADL-2023 Day 3 (Sequence Models)	Assessment	Technology	0	158	0	6.47
5	IWADL-2023 Day 2 (Basics of CNNs)	Assessment	News	0	195	0	5.80

Step12. Once the user has completed the review of the assessment, its name will be hyperlinked, and all the corresponding columns of the assessment will be filled accordingly.

Announcement

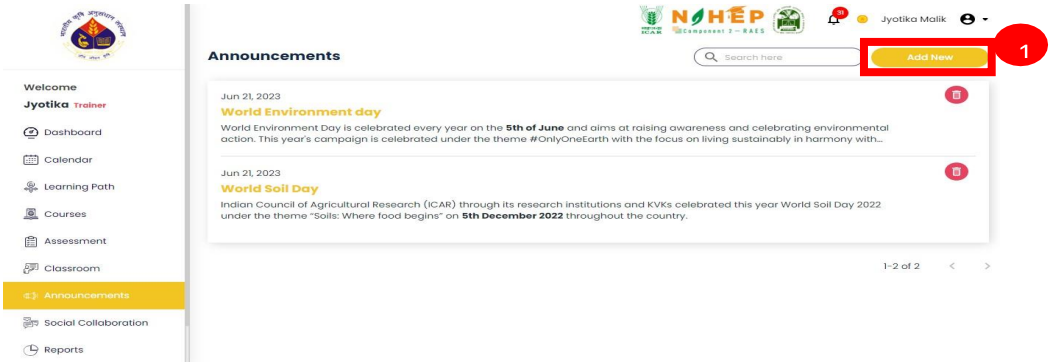
Announcement modules comprise the functionality to make global announcements. Announcements made by faculty can be viewed by the students through the announcement module. These announcements can also be scheduled to be published later.

How to view announcements?
Users can click on the announcement from the left navigation. Once the user selects announcements, they will be able to see the announcements with the date of Publish.

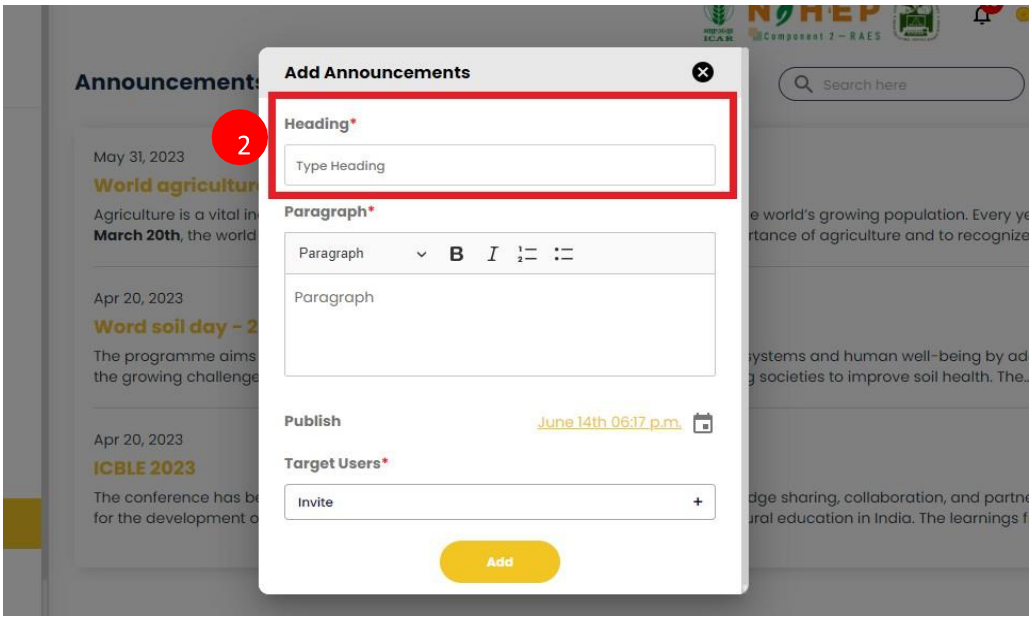


Step-1. Click on the "Announcement Name". Users can read the complete announcement.

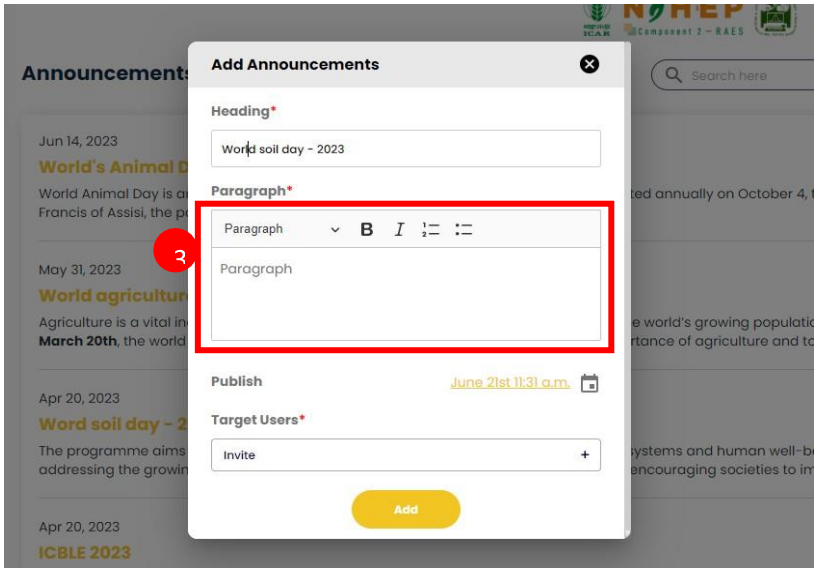
How to add a new announcement?

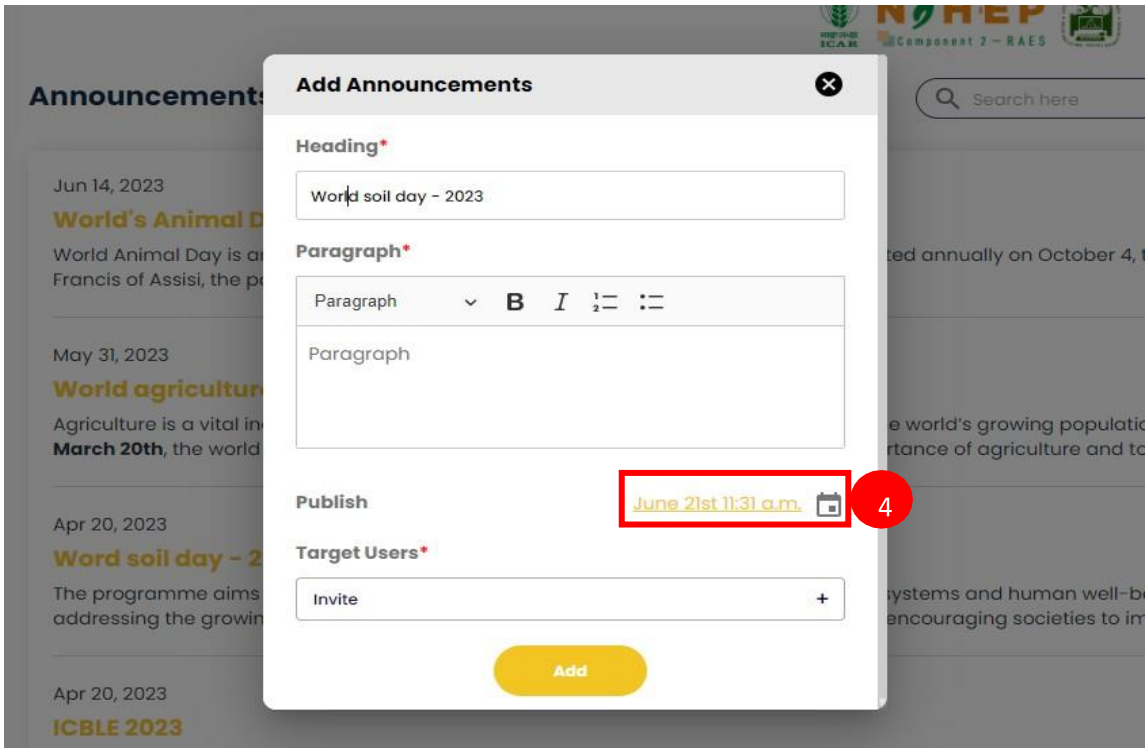


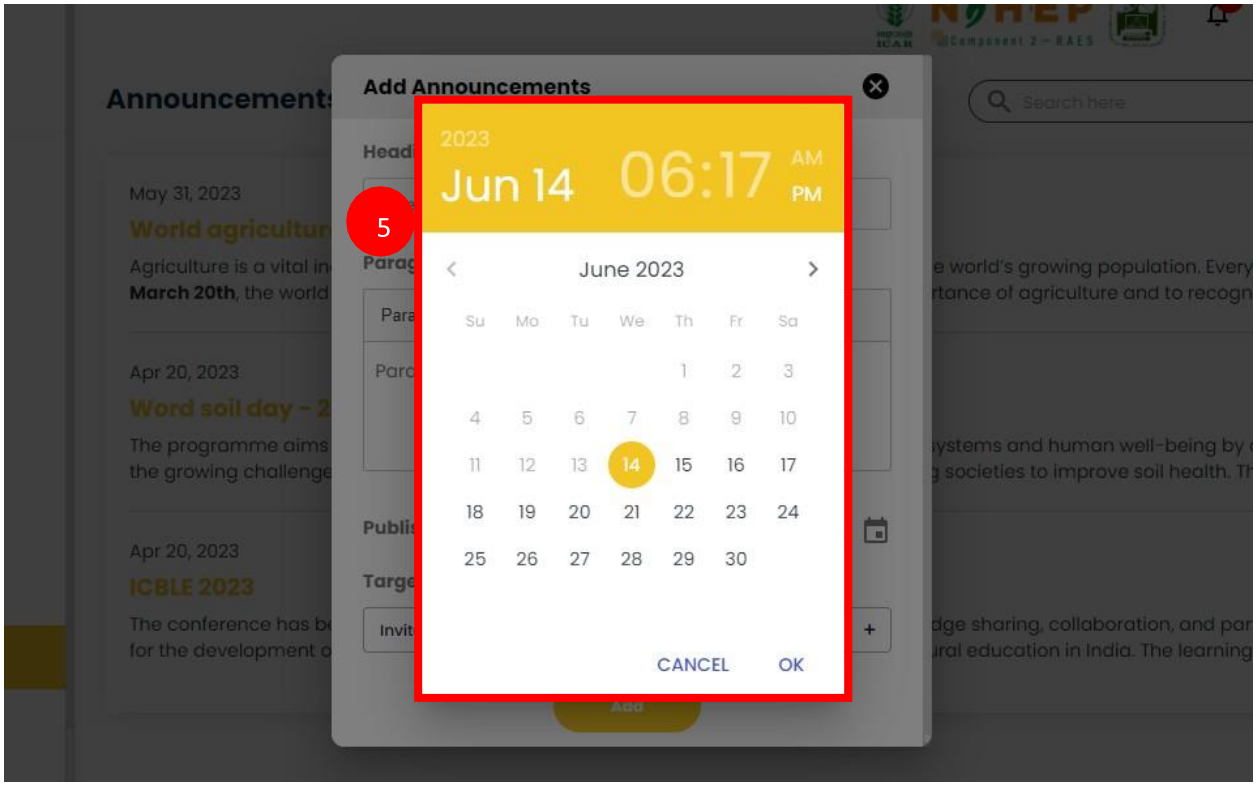
Step- 1. Click on "Add new" button to create a new announcement.
A pop-up will appear, "Add Announcements".



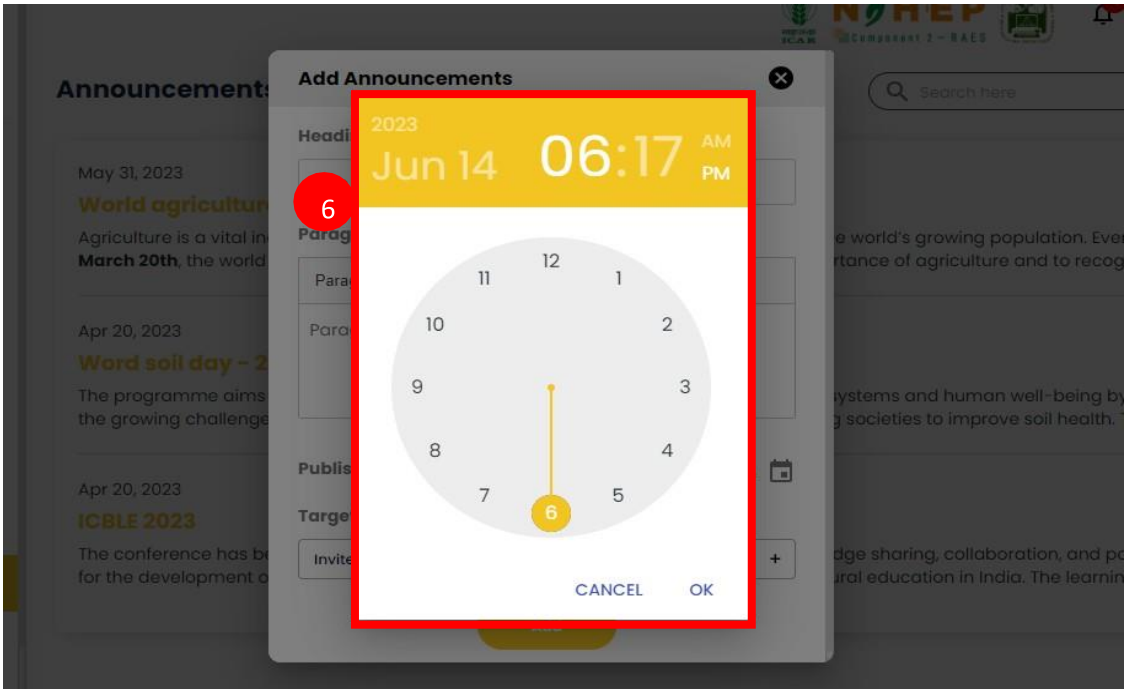
Step-2. Add Heading of the announcement under "Heading" option.
Step-3. Add a description for an announcement under "Paragraph" option.
Step-4. Select Publish date and time of an announcement by clicking on the calendar associated with Publish.


Announcements
Add Announcements
Heading*
World soil day - 2023
Paragraph*
Paragraph
Paragraph
3
Publish
June 21st 11:31 a.m.
Target Users*
Invite
Add
Search here
Jun 14, 2023
World's Animal D
World Animal Day is a
Francis of Assisi, the p
ted annually on October 4,
May 31, 2023
World agricultur
Agriculture is a vital in
March 20th, the world
e world's growing populatio
rtance of agriculture and to
Apr 20, 2023
Word soil day - 2
The programme aims
addressing the growin
ystems and human well-b
encouraging societies to im
Apr 20, 2023
ICBLE 2023

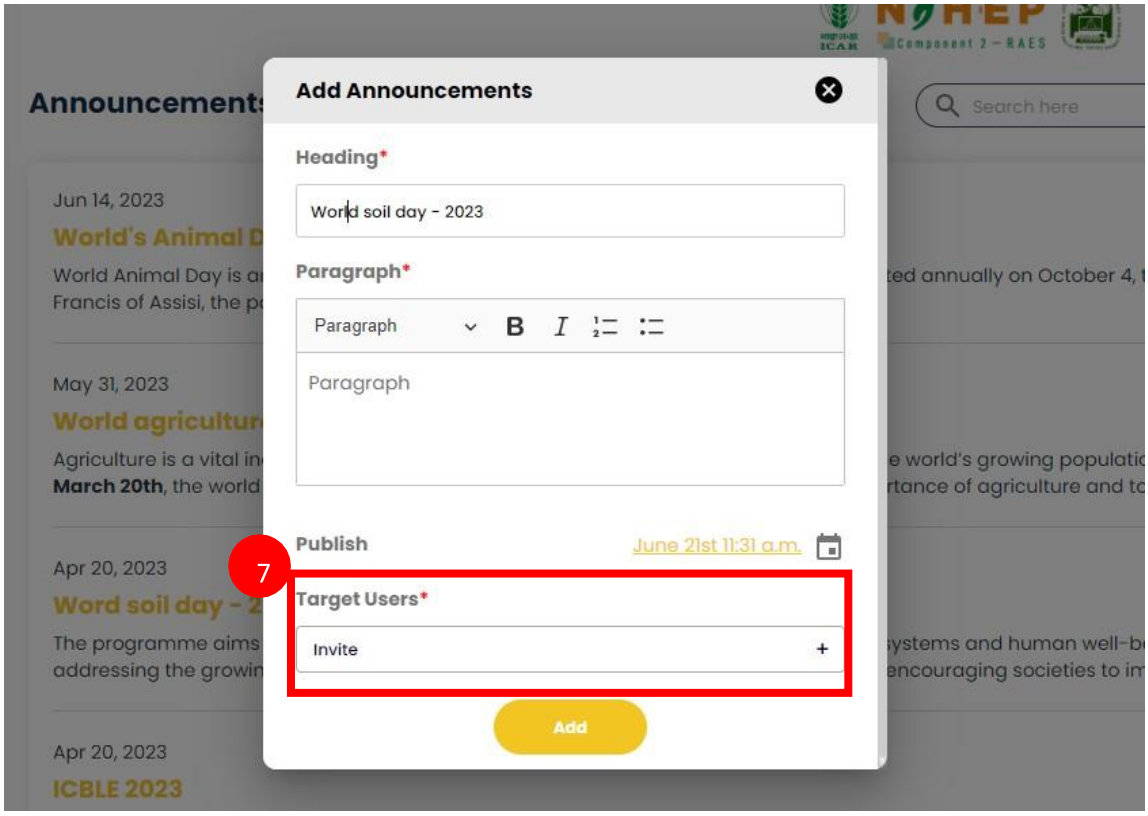

Announcements
Add Announcements
Heading*
World soil day - 2023
Paragraph*
Paragraph
Paragraph
Publish
June 21st 11:31 a.m.
4
Target Users*
Invite
Add
Search here
Jun 14, 2023
World's Animal D
World Animal Day is a
Francis of Assisi, the p
ted annually on October 4,
May 31, 2023
World agricultur
Agriculture is a vital in
March 20th, the world
e world's growing populatio
rtance of agriculture and to
Apr 20, 2023
Word soil day - 2
The programme aims
addressing the growin
ystems and human well-b
encouraging societies to im
Apr 20, 2023
ICBLE 2023



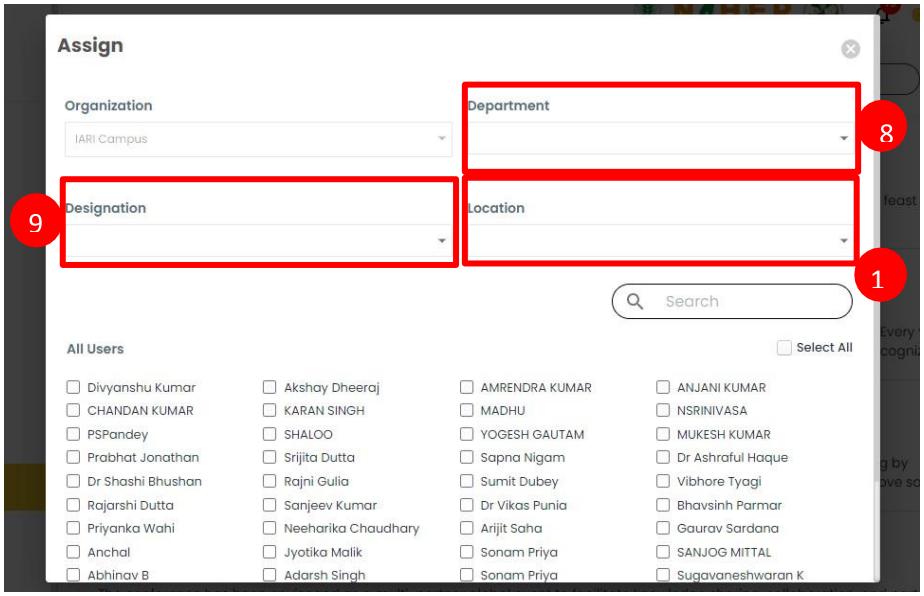
Step-5. Select Date.



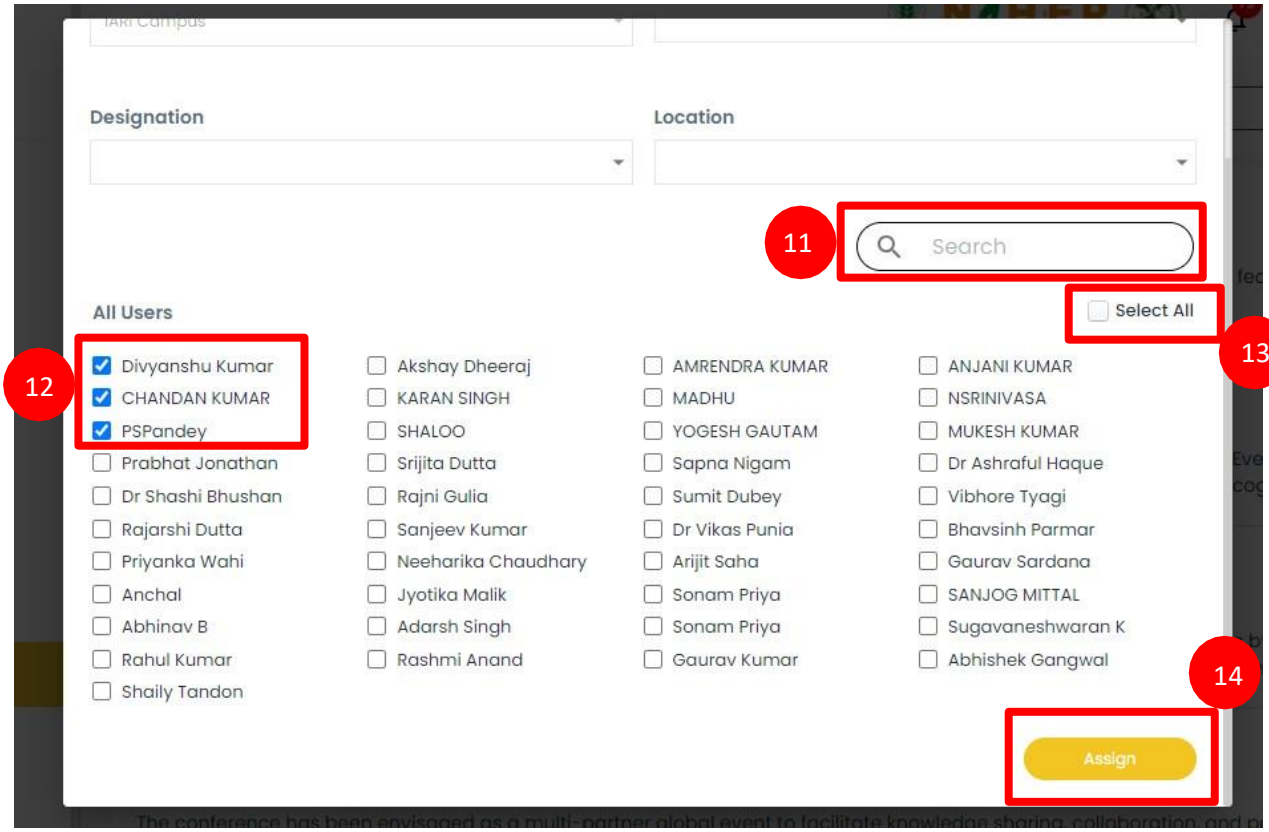
Step-6. Select time.



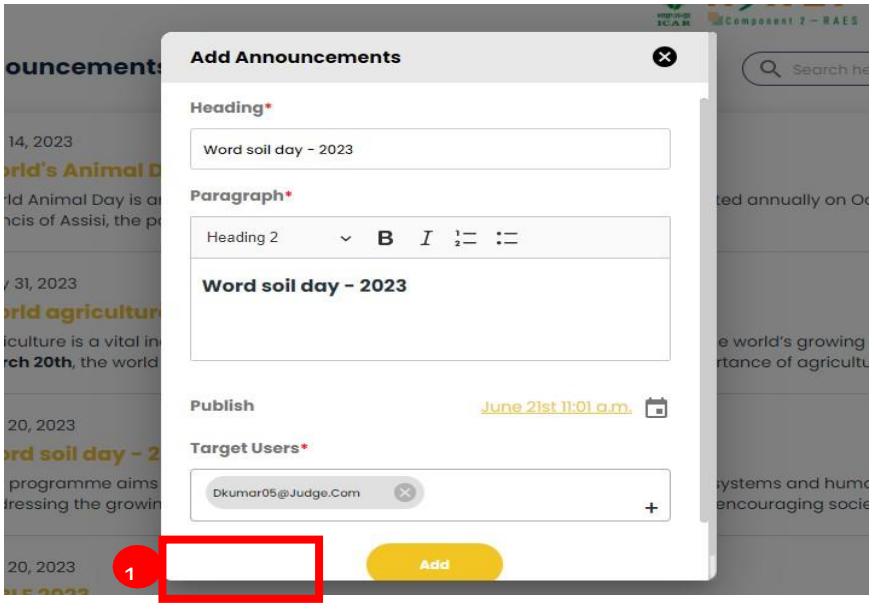
Step-7. Click on the "+" associated with an invite.A pop will display to assign participants



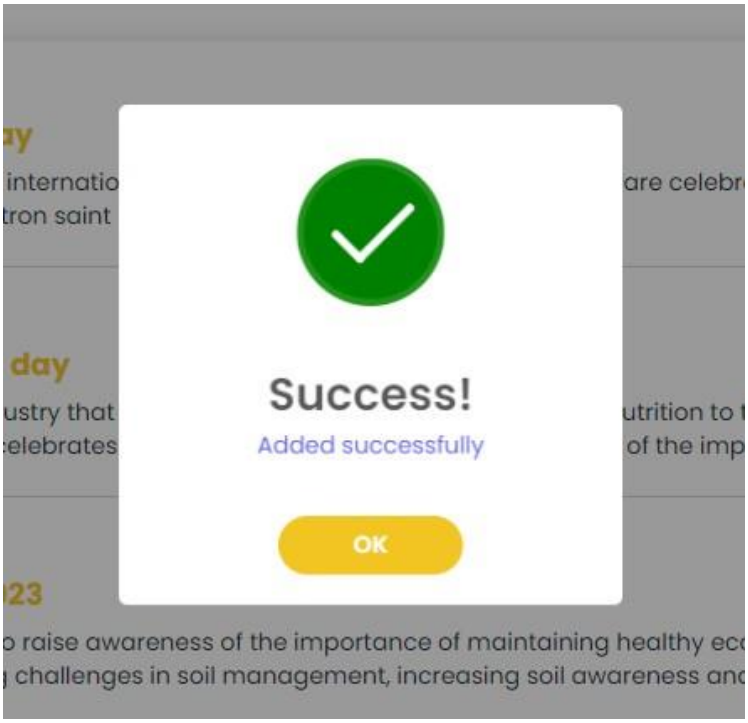
Step-8. Select "Department" from the drop-down menu.
Step-9. Select "Designation" from the drop-down menu.
pStep-10. Select "Location" from the drop-down menu.



Step-11. Search the student's name in the local 'Search' given.
Step-12. Click on the check boxes associated with the names of the students.
Step-13. Click on the check box associated with "Select All" if you wish to select all the students.
Step-14. Click on **"**Assign".



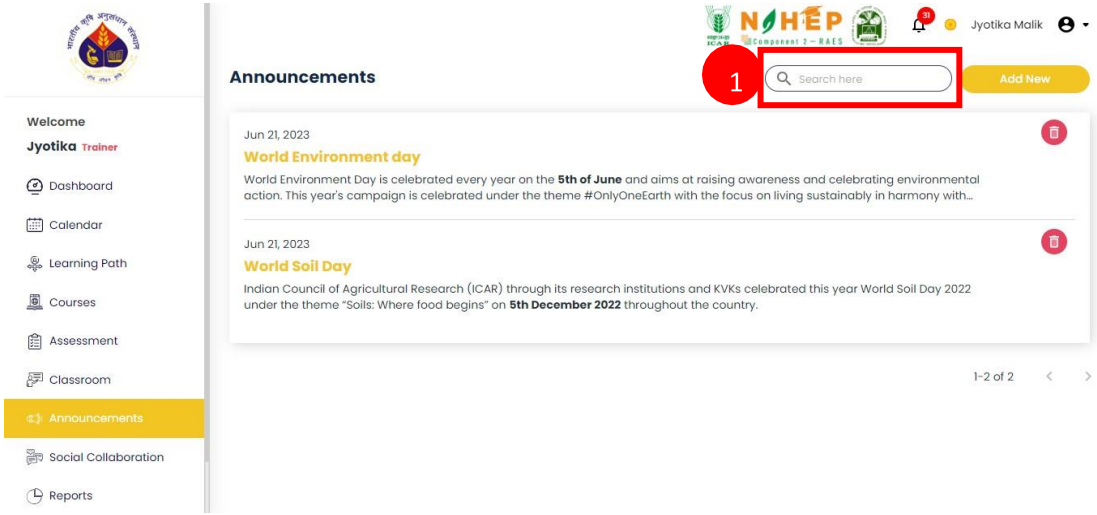
Step-15. Click on "Add" button to save the announcement.



A success message will appear, “Added successfully”.

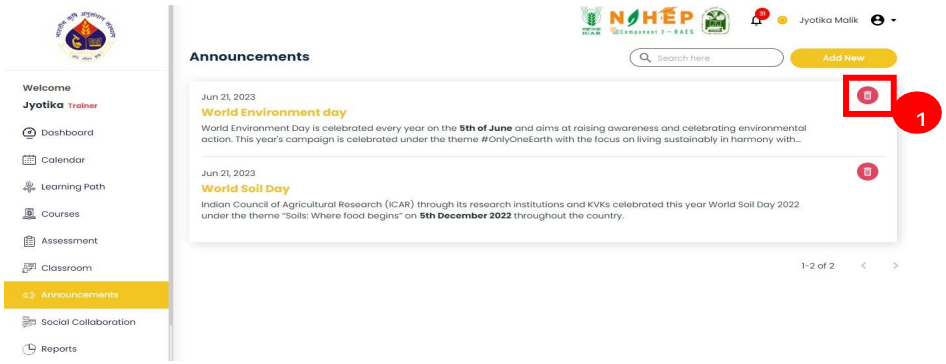
How to search for an announcement?

When the user select announcement from the left menu navigation, they can see an option for search.

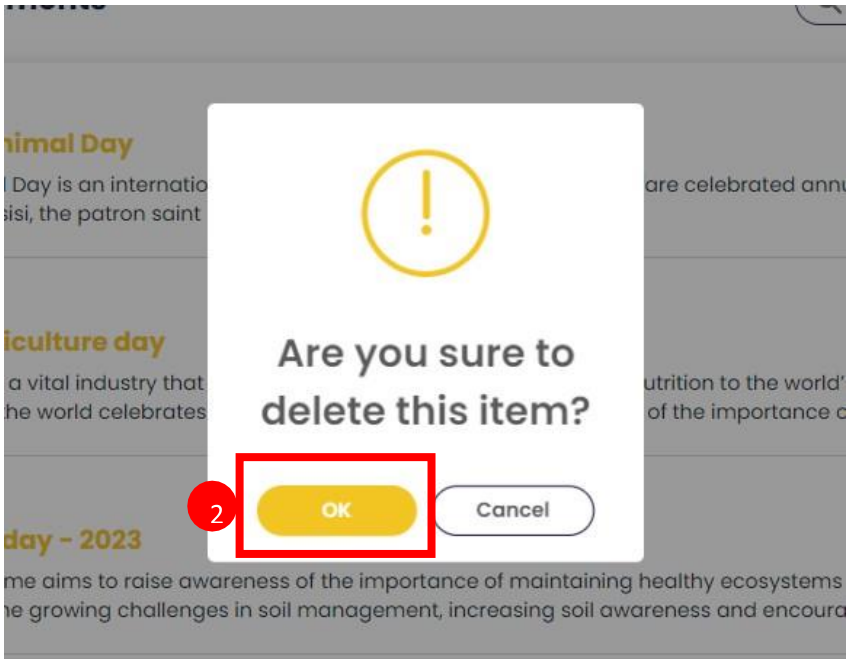


Step-1. Type the heading or keywords to search for any announcement.

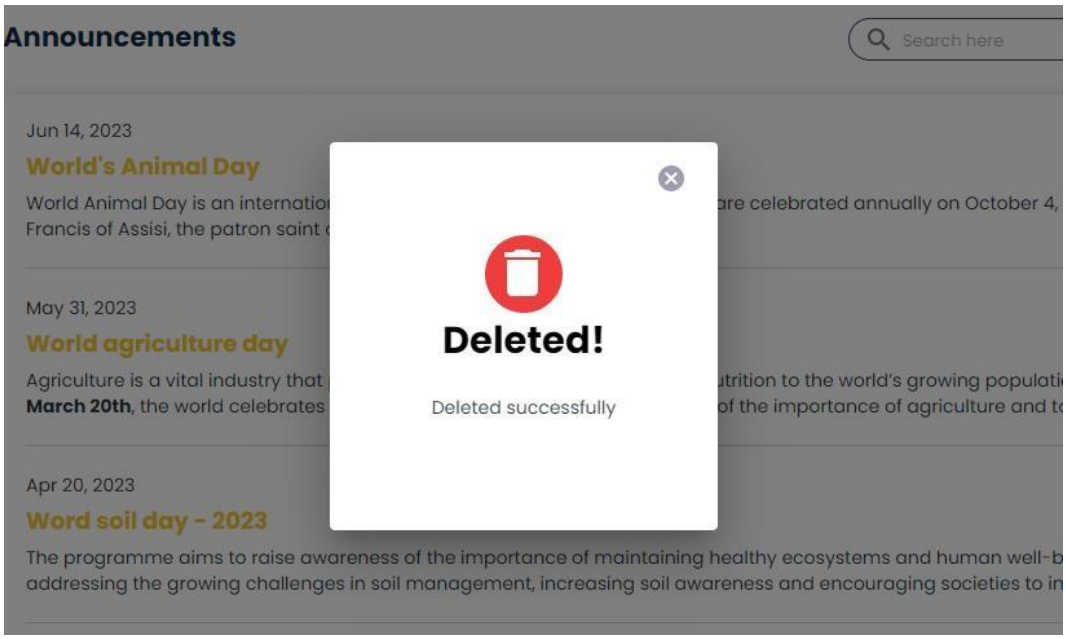
How to Delete an Announcement?



Step-1. Click on the delete button associated with every announcement published.
Once the users click on delete, a pop-up will appear with the message "Are you sure to delete this item"?



Step-2. Click on “OK” to delete this item. Click on “Cancel” to cancel the selection



Development of Learning Paths, Reports & Social Walls under Faculty Modules of NARES-Blended Learning Platform

A. Learning Paths: Creating a learning path in a blended learning platform involves combining online and offline resources to deliver a comprehensive educational experience.

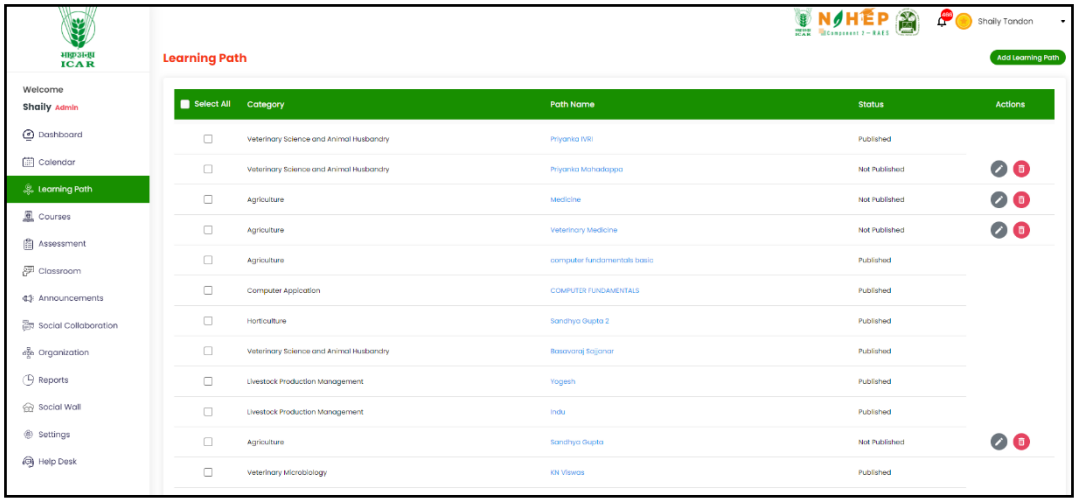


Figure: Learning path dashboard and learning path creation

Learning path can be created by the admin account of the university. Different topics from different courses can be taken and design a learning path for any student. Student can see the assigned learning path in their respective dashboard.

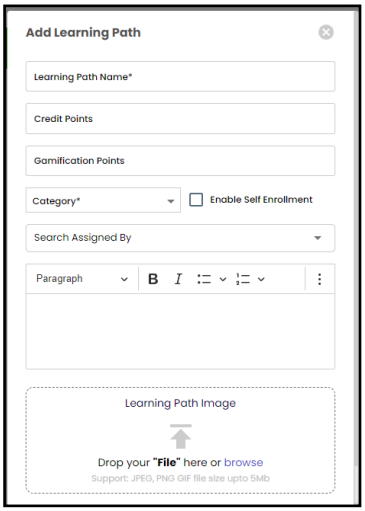


Figure: Adding learning path wizard

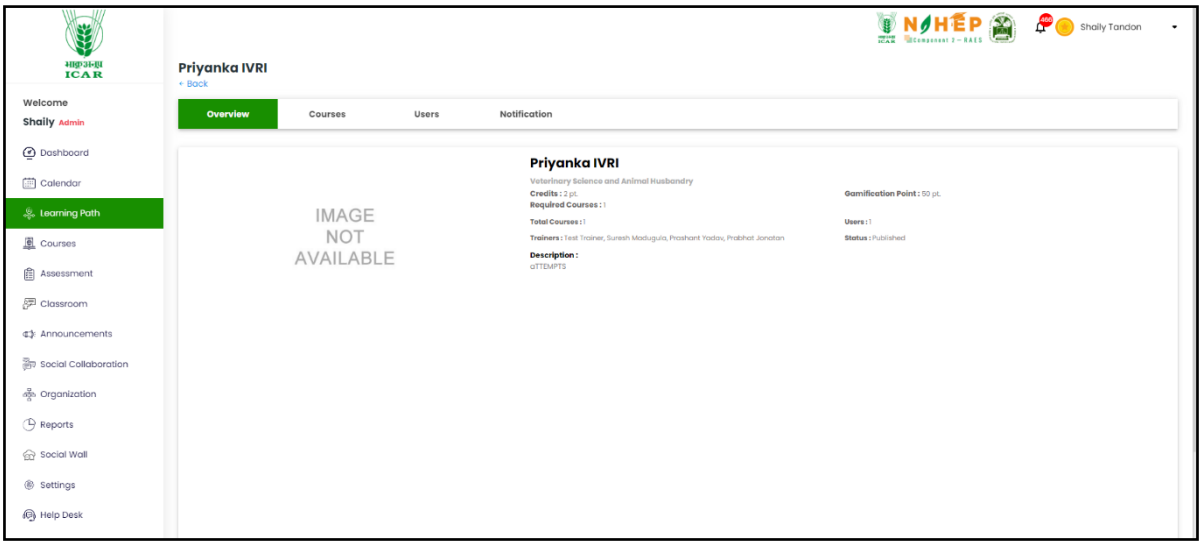


Figure: Dashboard of learning path

Learning path has four tap namely overview, courses, users and notification. In overview the basic description of the learning path can be seen. Courses tab will help to select the courses and the topics from the course list. Users tab will be used for selecting the students and the faculty for any learning path. Notification tab will be used for the publication of the learning path and send notification to the users.

B. Report

The course reporting system is designed to empower users with a comprehensive overview of their educational journey. By offering features such as the ability to view top courses, overall category statistics, and detailed course reports, learners gain valuable insights into their academic progress. The platform further enhances user experience with specific functionalities like assessment reports, accessible through the Assessment Tab, enabling users to export, import, and schedule assessments while applying search filters and date ranges. Additionally, the Classroom Tab facilitates the exploration of classroom reports, with the option to delve into detailed class reports by clicking on specific class names. The system's versatility is highlighted through its importing, exporting, and scheduling functionalities, providing users with a flexible

and convenient way to manage and analyze their educational data efficiently. Overall, this integrated approach empowers users to make informed decisions about their learning journey.

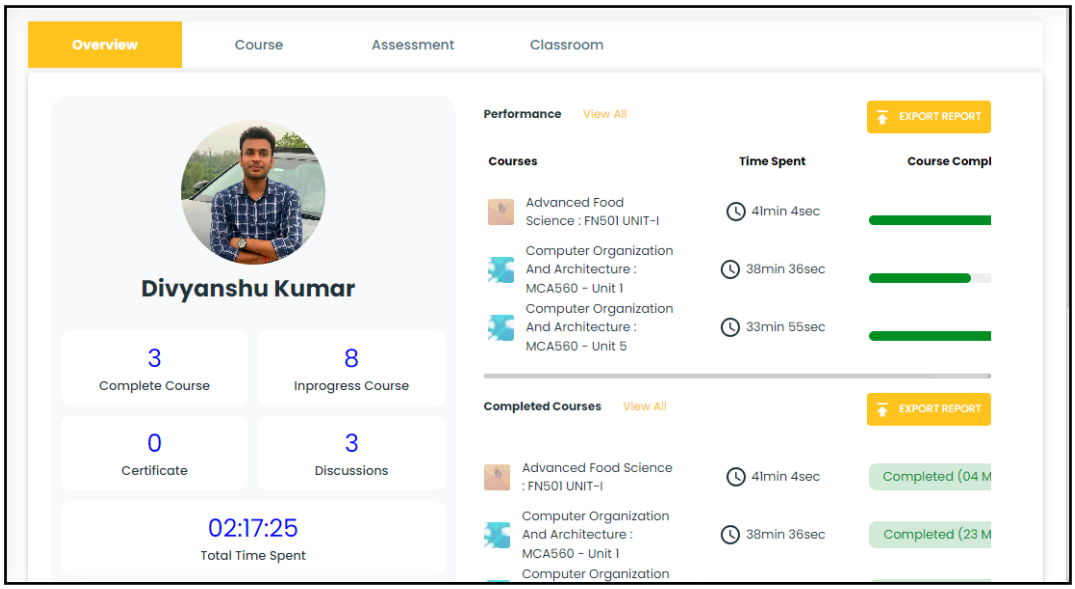
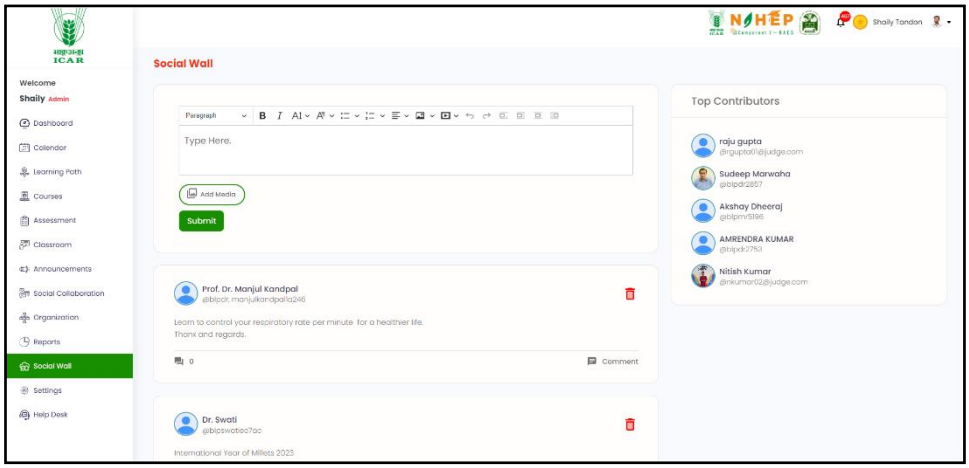


Figure: Report dashboard of the student

C. Social Wall



Social wall is the social media platform for NARES BLP users. In this option user can post the photos and also post comments for any of other post and also reply to own posts.