INT332

Lecture Zero

Devops Virtualization And Configuration Management

VISION

To be a globally recognized school through excellence in teaching, learning and research for creating Computer Science professionals, leaders and entrepreneurs of future contributing to society and industry for sustainable growth

MISSION

- To build computational skills through hands-on and practice-based learning with measurable outcomes.
- To establish a strong connect with industry for in-demand technology driven curriculum.
- To build the infrastructure for meaningful research around societal problems.
- To nurture future leaders through research-infused education and lifelong learning.
- To create smart and ethical professionals and entrepreneurs who are recognized globally

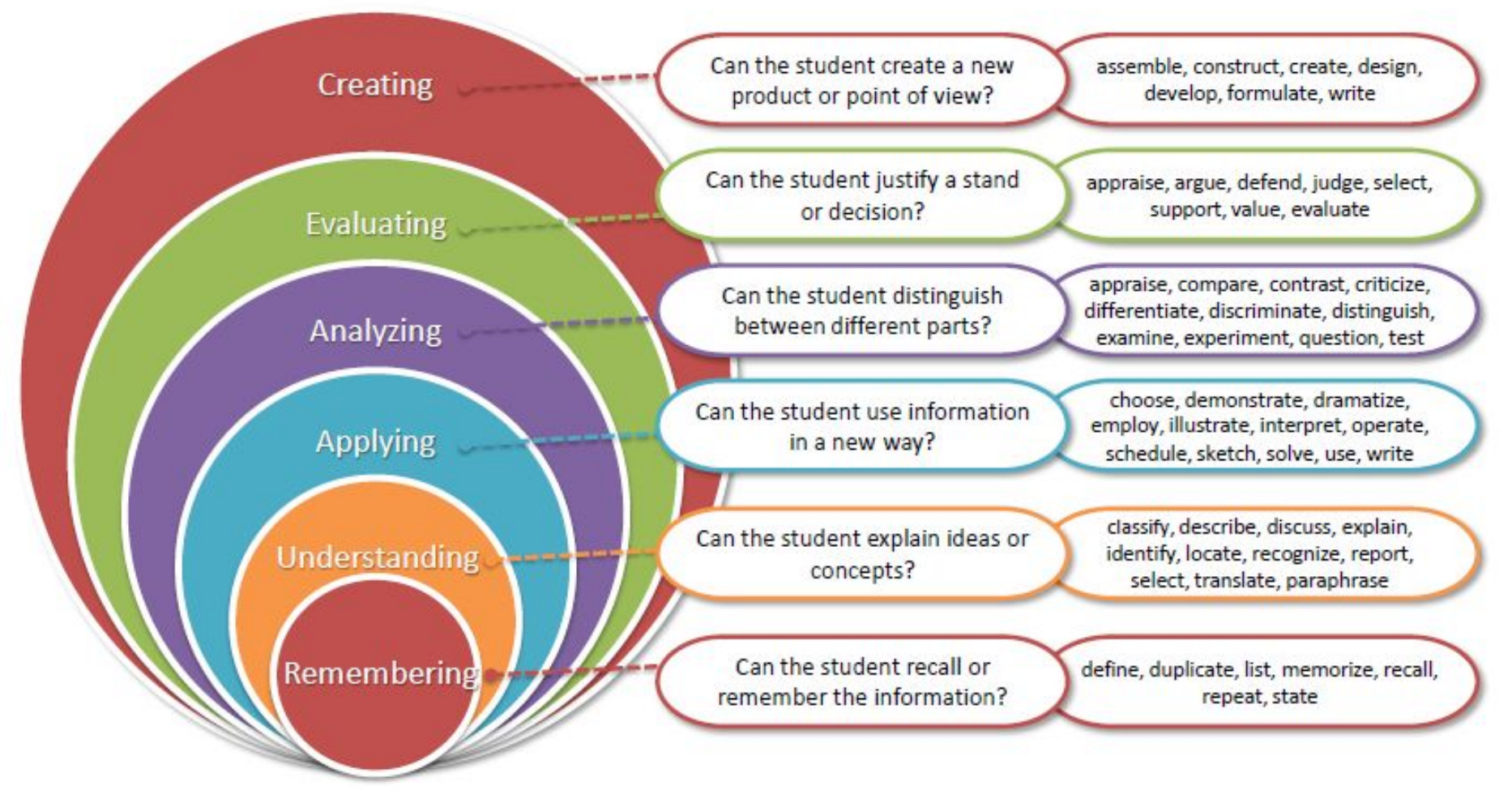
Course Overview

- This course introduces students to DevOps foundations through containerization, virtualization, and automated software delivery pipelines.
- Focus on Docker, microservices, and CI/CD workflows using tools such as Docker Compose, Maven, GitHub Actions, and Jenkins for building and deploying modern applications.

L T P

- Lecture - 2
- Tutorial - 0
- Practical - 2
- Credits – 3

Revised Bloom's Taxonomy



Unit:-1 Basics of DevOps Infrastructure

Introduction to Containers (Origin of containers, Emergence of Modern Containerization and Integration into DevOps), Container runtime, Process isolation & namespaces, Control Groups (cgroups) for resource limits, Container images & layers, Image registries & distribution, Introduction to Docker, Docker Architecture, Docker daemon, Docker CLI, Docker registry & Hub, Object types: container, image, network, volume, Docker layering & filesystem

Unit:-2 Image Building & Container Management

Docker file Core Concepts: Image layering, Build context & .dockerignore, Dockerfile writing, Basic instructions (FROM, RUN, COPY, ADD, CMD, ENTRYPOINT, WORKDIR, ENV, EXPOSE, VOLUME etc),, Image Creation in Detail: docker build process, Images Tagging /versioning, Inspecting images (history, layers), Docker Networking: Bridge network, Host & overlay networks, DNS inside Docker, Linking containers, Port mapping between host and container, Docker Storage: Volumes vs bind mounts, backing data on host, , Copy-on-write mechanism, Registries: Docker Hub, GitHub Container Registry (GHCR),, Private registries, Authentication & access tokens.

Unit:-3 Microservices with Docker Compose

Microservices Architecture: Need for microservices, Monolithic vs microservices, Advantages: scalability, isolation, agility, API Gateway., Docker Compose: YAML Structure, Writing docker-compose.yml, version, services, volumes, networks, Environment variables, Secrets and configs, Build vs image fields in YAML, Service dependency ordering., Use Case Deployments: Multi-container apps (database + backend + frontend), WordPress + MySQL, Node.js + MongoDB, Java Spring Boot + PostgreSQL.

Unit:-4 Maven Build Automation

Why build tools exist, Problems solved by automated builds, Project Object Model (POM), Directory structure, Build lifecycle phases, validate, compile, test, package, verify, install, deploy, Parent POM, Dependency scope, Transitive dependencies, Version conflicts & resolution, Using dependency Management, Maven Plugins & Execution: Compiler plugin, Surefire plugin (unit testing), Shade plugin (uber jar), Maven wrapper (mvnw), Maven and Docker Integration: dockerfile-maven-plugin, Dockerizing Maven-based applications, Pushing artifacts to registries.

Unit:-5 Continuous Integration (CI) with GitHub Actions

Understanding workflow automation, Events, triggers, workflow directory structure, Key components: workflows, Key components: jobs, steps, actions, runners, Workflow triggers: push, pull request, schedule, , manual workflow, Jobs & matrix strategies, Steps & shell commands, Using marketplace actions, Language-specific actions, Using caching for faster builds, Multi-job workflows, Deploying to servers/cloud using Actions, Runners: GitHub-hosted runners, Runners:Self-hosted runners ,Runner security & management, Docker &GitHub Actions: Building Docker images in CI, Pushing to Docker Hub, Pushing to GitHub Container Registry (GHCR), Deployments to servers/clouds

Unit:-6 CI/CD with Jenkins

Jenkins architecture (Master/Agent model), Installation & UI overview, Plugins management, Security, users, roles, Jenkins Pipelines: Freestyle vs Pipeline jobs, Declarative pipeline syntax, Scripted pipeline syntax, Jenkinsfile structure, Parameters,environment variables, Jenkins multi-branch pipelines, Pipeline Stages: Checkout code from Git,Build, Test, Package, Post actions, Managing artifacts, Docker and Jenkins Integration: Building Docker images using Jenkins, Docker inside Jenkins agents, Using Docker plugins, Publishing images to Docker Hub/GHCR, Jenkins and GitHub integration, Backup & restore, Pipeline best practices., Jenkins and Maven: Maven installation in Jenkins, Global tool configuration, Running Maven builds in pipelines, Code coverage & test reports, Jenkins CI/CD Deployment Flows: Triggering builds (pollSCM, webhook), Pipeline libraries, Jenkins agents (SSH/SFTP/Container-based), Deployments to servers/clouds

List of Practicals

- Installing Docker on Windows/Linux, Concept of Hyper-V, Basic Docker commands.
- Create a Docker-hub account, Pushing the Docker to container-hub, Saving changes to a Docker container.
- Creating a Docker file
- Types of Docker storage, Linking docker container
- Introduction to micro services
- Docker compose installation, Introduction to YAML files, Writing a Docker Compose file, deploying WordPress using docker compose.
- Introduction to container orchestration
- Creating a Docker swarm cluster, Initializing Docker Swarm create a service in a Docker Swarm.
- Introduction to Docker networks, Creating a service.
- Introduction to continuous integration.
- Jenkins Architecture, Installing Jenkins.
- Managing nodes on Jenkins.
- Understanding CI / CD Pipelines, Creating CI / CD Pipelines.

COURSE OUTCOMES

CO1 :: Explain DevOps infrastructure concepts including containerization, runtimes, process isolation, namespaces, and resource management mechanisms.

CO2 :: Apply Docker concepts to build, manage, and distribute container images using networking, storage, and registries.

CO3 :: Construct microservices applications using Docker Compose with multi-container services and dependency configuration.

CO4 :: Analyze and implement automated builds using Maven through lifecycle management, dependency control, plugins, and Docker integration.

CO5 :: Apply Continuous Integration workflows using GitHub Actions for automated builds, testing, image creation, and deployment.

CO6 :: Develop end-to-end CI/CD pipelines using Jenkins integrating source control, build tools, containers, and deployments

PROGRAM OUTCOMES

PO-1 Engineering knowledge::Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO-2 Problem analysis::Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO-3 Design/development of solutions::Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO-4 Conduct investigations of complex problems::Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO-5 Modern tool usage::Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO-6 The engineer and society::Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO-7 Environment and sustainability::Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO-8 Ethics::Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO-9 Individual and team work::Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO-10 Communication::Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO-11 Project management and finance::Demonstrate knowledge and understanding of the engineering, management principles and apply the same to one's own work, as a member or a leader in a team, manage projects efficiently in respective disciplines and multidisciplinary environments after consideration of economic and financial factors.

PO-12 Life-long learning::Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES

PSO1: Apply basic knowledge in areas such as Software Engineering, Networking and Security, Database Management Systems, Intelligent Systems, Operating Systems, and System Architecture for building Software products.

PSO2: Provide effective and efficient real-time solutions using attained knowledge in interdisciplinary domains for societal benefits through projects.

Detail of Academic Tasks

AT1 : BYOD Practical (Before Midterm)
AT2 : Project (After MidTerm)
End Term Practical

**** Both ATs are Compulsory****

Weightage Division

- Attendance - 5 marks
- CA - 45 marks
- ETP - 50 marks

Resources

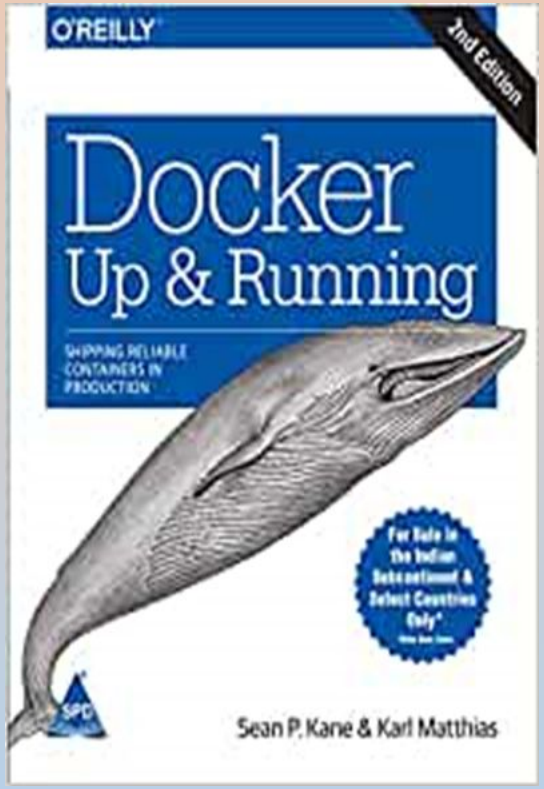
- Text Book:

  DOCKER DEEP DIVE by NIGEL POULTON, SHROFF/O'REILLY

- References:

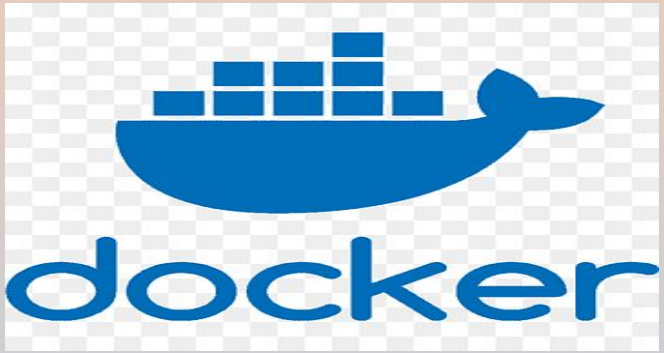
  1. DOCKER: UP AND RUNNING by KARL MATTHIAS, SEAN P. KANE, O'REILLY

  2. JENKINS CONTINUOUS INTEGRATION COOKBOOK by ALAN MARK BERG, O'REILLY 3. JENKINS 2 UP AND RUNNING by BRENT LASTER, O'REILLY



Tools & Platforms Used

- Docker Desktop/AWS platform



- Jenkins for CI/CD

- Maven

Innovative Pedagogy

- Discussion-based learning
- Hands-on Practice
- Case studies on real-world deployments with docker compose
- CI/CD Pipeline through Github actions and Jenkins
- BYOD practical approach

Practical Applications of the Course

Unit 1 – Practical Applications

Containers & DevOps Infrastructure

- Deploying isolated applications using Docker on Linux/Windows
- Used in industry by companies like Netflix, Uber, Walmart
- Real-world use: Package once and run anywhere without conflicts
- Docker Hub enables global sharing & versioning of application images

Practical Applications of the course

Unit 2– Practical Applications

Image Building, Networking & Storage

- Building portable application containers using Docker file
- Managing real projects with versioned container image repositories
- Real-world use: Secure, repeatable deployments across teams and machines
- Multi-host connectivity using Docker networks for backend + database

Practical Applications of the course

Unit 3– Practical Applications

Microservices & Docker Compose

- Running multi-service apps (frontend/API/database) in one command
- Used by modern platforms like Amazon, Swiggy, Meesho
- Real-world use: Independent scaling, zero downtime updates
- WordPress + MySQL deployment for blogging and e-commerce solutions

Practical Applications of the course

Unit 4– Practical Applications

Maven Build Automation

- Packaging complete Java applications for production release
- Used by enterprise software companies (TCS, Infosys, Capgemini)
- Real-world use: Automatic dependency management & conflict resolution
- CI-ready builds using Maven plugins, JUnit tests & packaged JAR/WAR files

Practical Applications of the course

Unit 5– Practical Applications

GitHub Actions – Continuous Integration

- Automating build, test & deploy pipelines from GitHub repositories
- Used by startups and open-source communities for fast releases
- Real-world use: Automated testing prevents bugs from reaching production
- Deploy containers to Docker Hub, AWS, Azure or Kubernetes clusters

Practical Applications of the course

Unit 6– Practical Applications

Jenkins CI/CD Pipelines

- Orchestrating enterprise-level build → test → deploy pipelines
- Used in large organizations like banks, telecom & SaaS platforms
- Real-world use: Automatic deployments after every code commit
- Jenkins plugins integrate with Docker, Maven & cloud servers for end-to-end delivery

Competitions & Recruitment Exams

- Hackathons: Smart India Hackathon, DevOps Hackathons
- Recruitment Roles: DevOps Engineer, Cloud Engineer, SRE, Automation Engineer
- Eligibility Exams: GATE (CS/IT), Company-specific coding rounds
- Certifications: AWS DevOps Engineer, Docker Certified Associate (DCA), GitHub Actions Certification.

Skill Attainment Through INT332

- Docker and Containerization
- Image building & registry management
- Microservices deployment using Compose
- Maven lifecycle automation
- GitHub Actions CI workflows
- Jenkins CI/CD pipelines
- DevOps pipelines from build to deploy
- Practical cloud-ready DevOps skills

Outcome of Zero Lecture

- Students understand course roadmap, units, and learning objectives
- Introduction to core DevOps tools: Docker, Maven, GitHub Actions, Jenkins
- Awareness of evaluation pattern, labs, assignments, and project requirements
- Real-world relevance of containerization and CI/CD in modern software delivery
- Students recognize industry demand and career opportunities in DevOps & Cloud

Need to Learn Docker

Virtualization and containerization

- With the changing times, businesses started looking for solutions to reduce overhead costs, enhance scalability, and standardize application deployment process.

- They started considering the following two approaches to reduce costs −

1. Virtualization
2. Containerization

What is Virtualization?

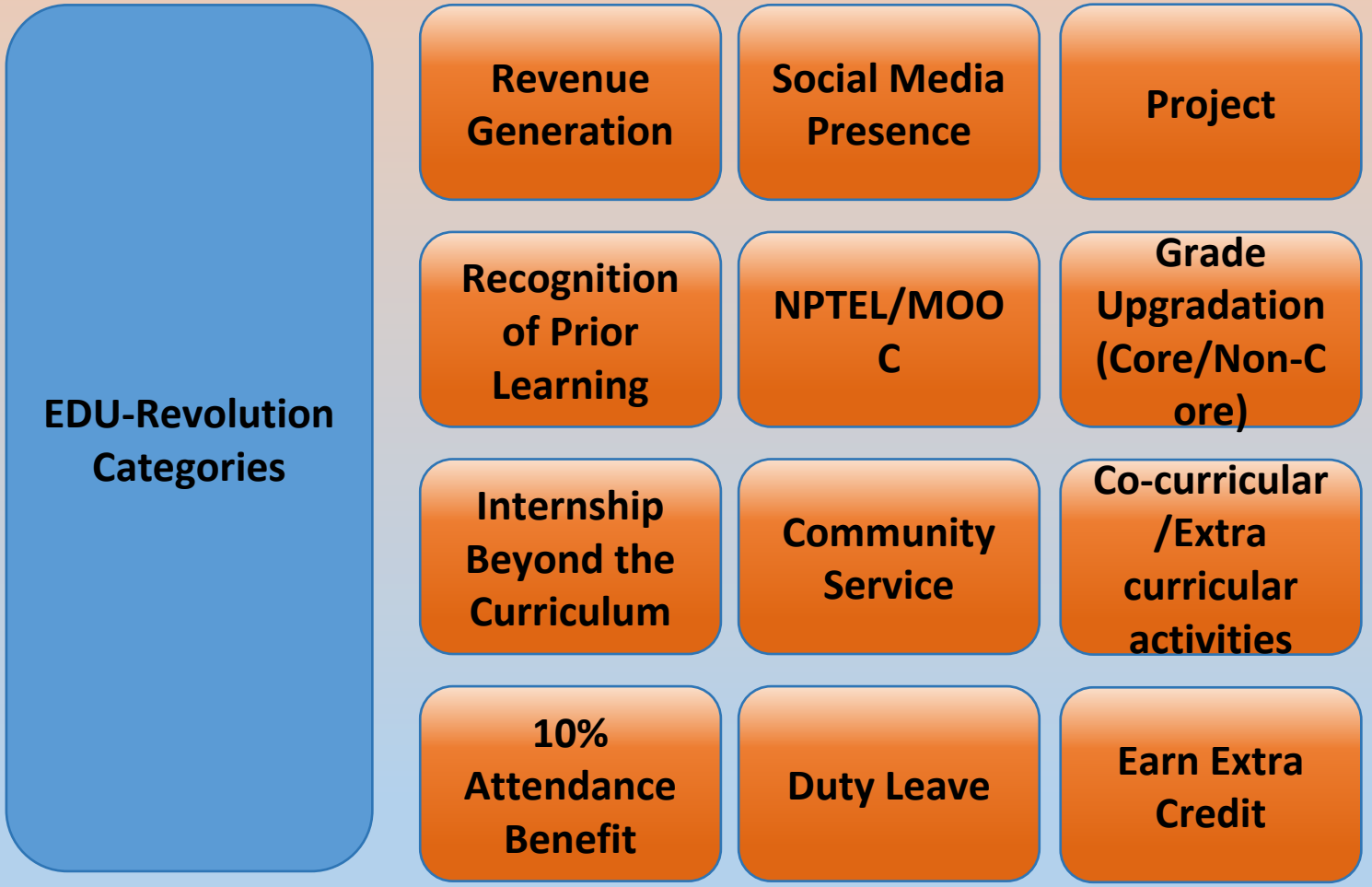
- Virtualization is the process of partitioning a physical server into multiple virtual servers.
- The process of partitioning is carried out using a software called 'hypervisor'.
- After partitioning, the virtual servers act and perform just as a physical server.
- Essentially, it means using the same hardware set-up more efficiently, thereby freeing the resources for other tasks or retiring the resources altogether.

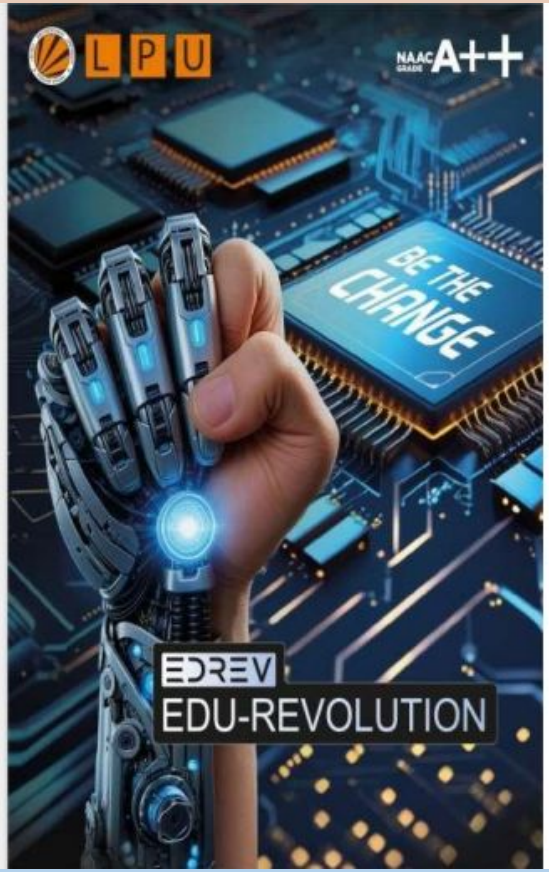
Advantages of Virtualization

- Enhanced performance
- Promotes agile IT infrastructure
- Promotes use of resources in optimum manner
- Better disaster management if any physical hazards take place to resources
- Better security as the infected VM can be isolated from other VMs and the host server
- Space saving
- The capital investment cost on hardware is saved, and thereby maintenance cost is saved, hence overall cost savings for a business.

For example, instead of dedicating 1 server and 1 operating system to execute 1 dedicated task, the same server is partitioned such as 1 server and 2 operating systems, which can allow independent tasks running on each of them.

What is Containerization?


EDU-Revolution Categories
Revenue Generation
Social Media Presence
Project
Recognition of Prior Learning
NPTEL/MOOC
Grade Upgradation (Core/Non-Core)
Internship Beyond the Curriculum
Community Service
Co-curricular /Extra curricular activities
10% Attendance Benefit
Duty Leave
Earn Extra Credit



EDUCATION REVOLUTION @ LPU

"...Gamified Learning"

UCATION REVOLUTION

GAMIFIED LEARNING

The Beginning: BE THE CHANGE

Personalized, Student-Centric Learning: Tailored academic pathways to meet individual learning needs.

Beyond the Classroom: Enabling students to explore, engage, and excel through flexible learning platforms.

Academic Agility: Freedom to choose and customize key academic areas to match students' aspirations.

Real-World Integration: Moving beyond traditional pen-and-paper methods to hands-on, practical learning.

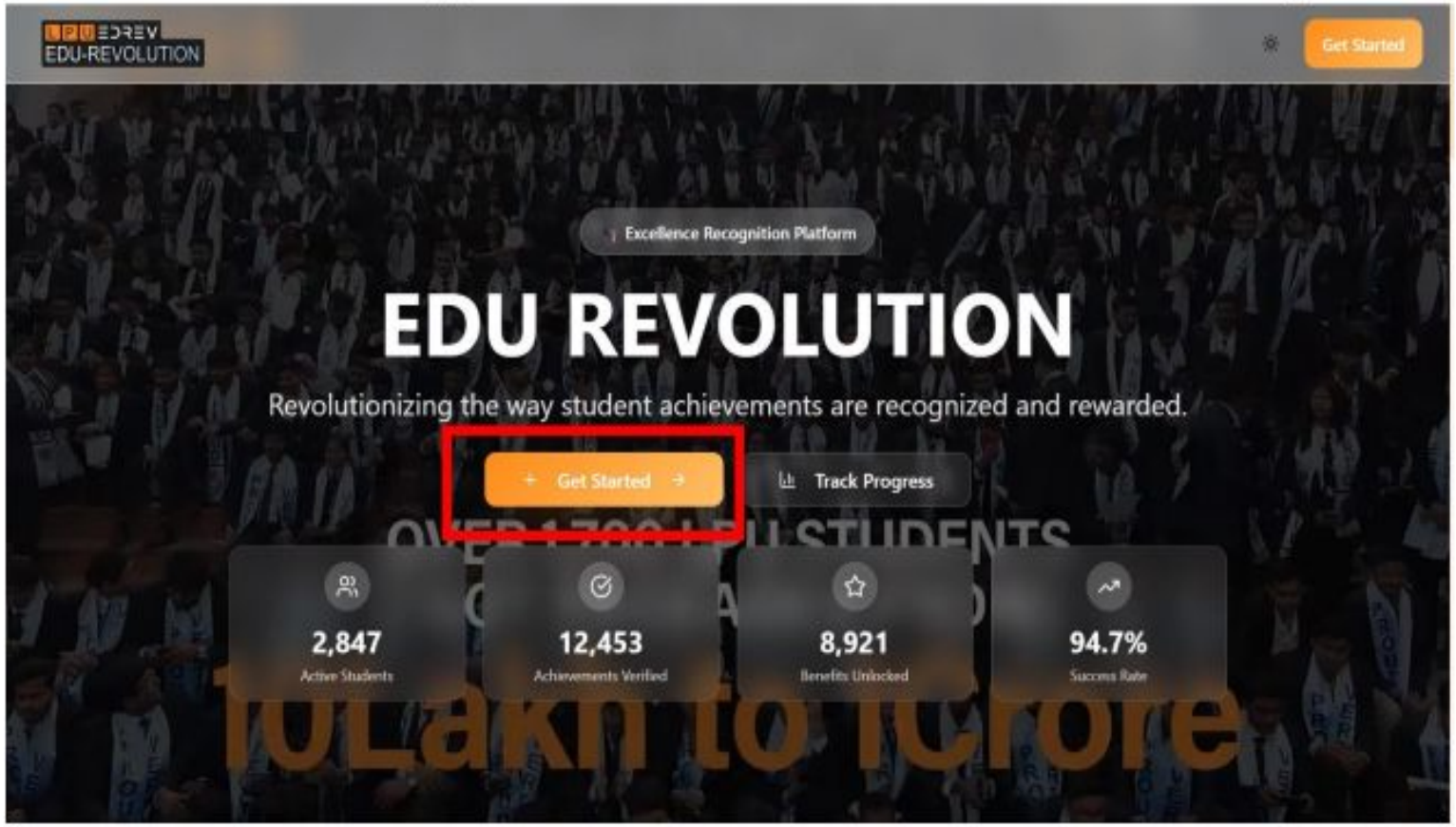
Experiential Learning Focus: Bridging theory with real-world practice to enhance skills and competencies.

Holistic Development: Fostering intellectual, emotional, and social growth for well-rounded graduates.

EDU-REVOLUTION INTERFACE

Step 1: Access the EDU-Revolution Portal

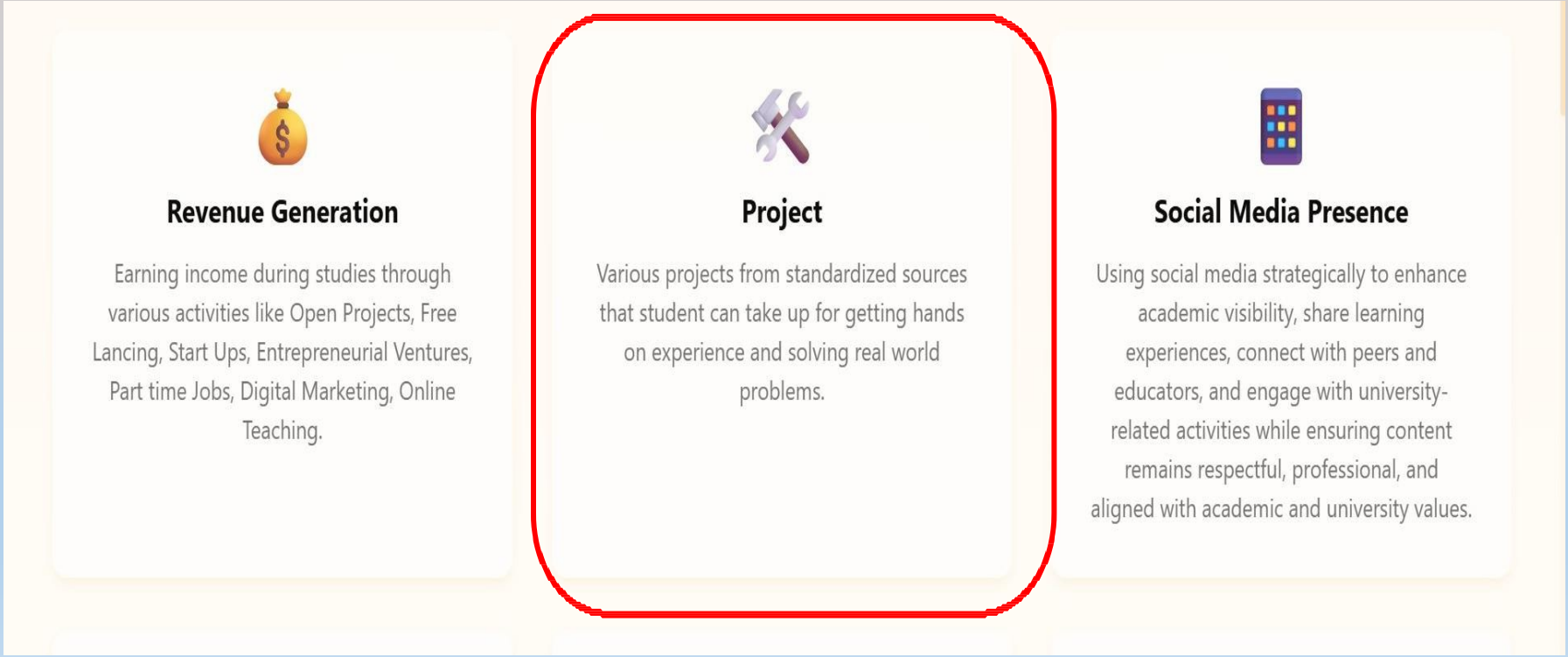
- Navigate through UMS → LMS → EDU-Revolution: Be the Change,
- or click the dashboard shortcut "Apply for EDU-Revolution." and click on "Get Started" to see the categories



EDU-REVOLUTION INTERFACE

Step 2: Choose a Category

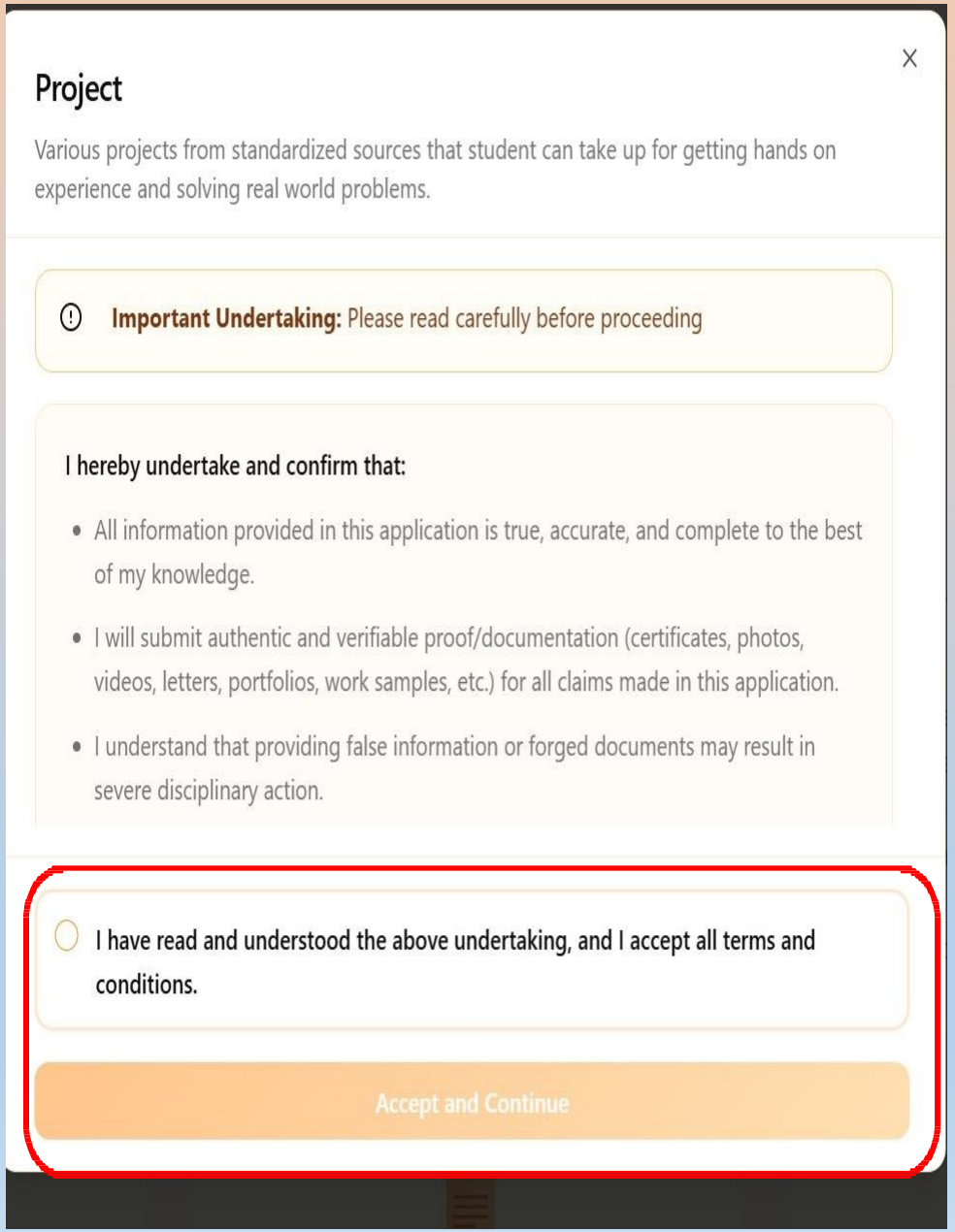
Select the category for which you want to submit your nomination.



EDU-REVOLUTION INTERFACE

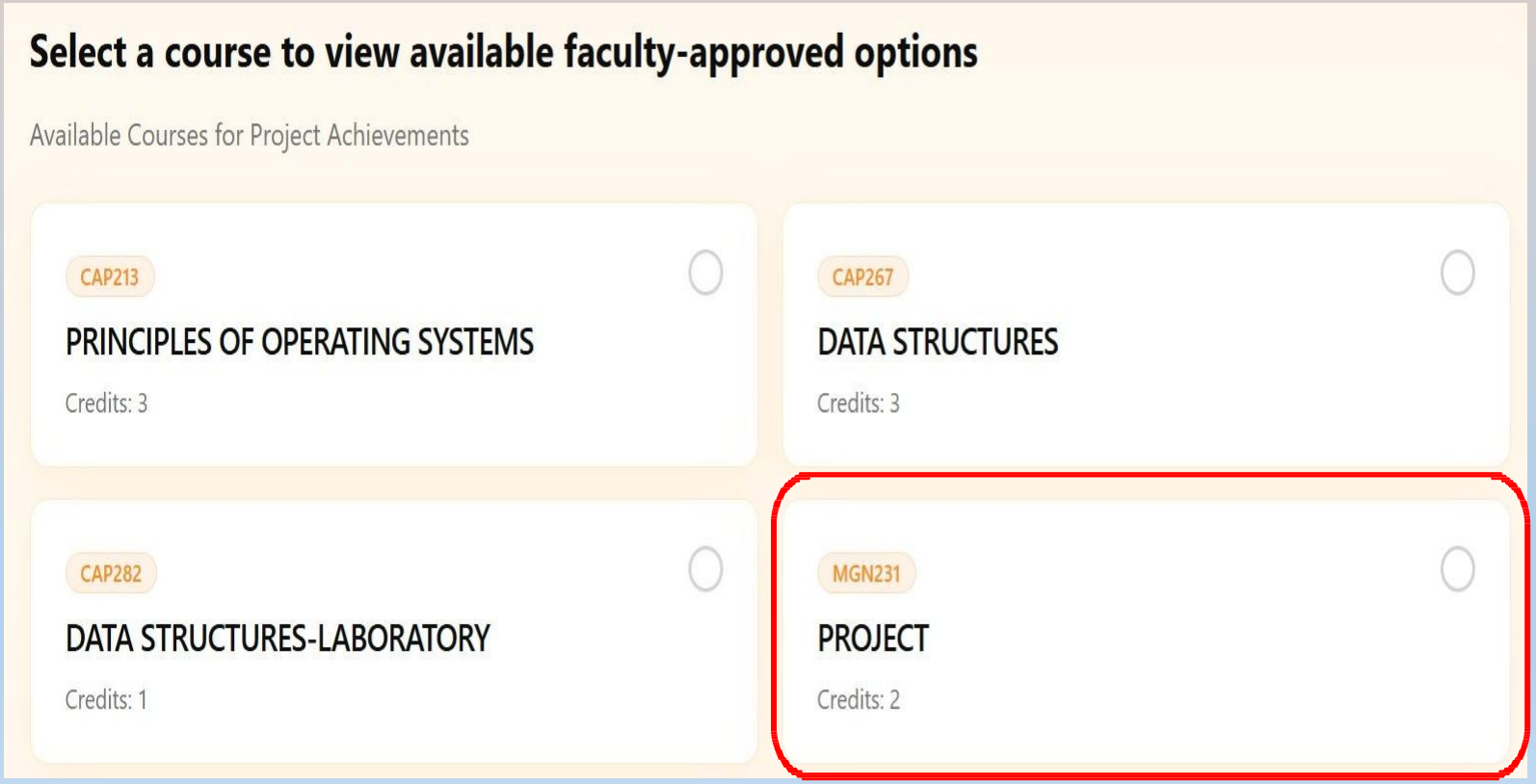
Step 3 : Read the undertaking carefully

Click on Accept and continue to see the course list.



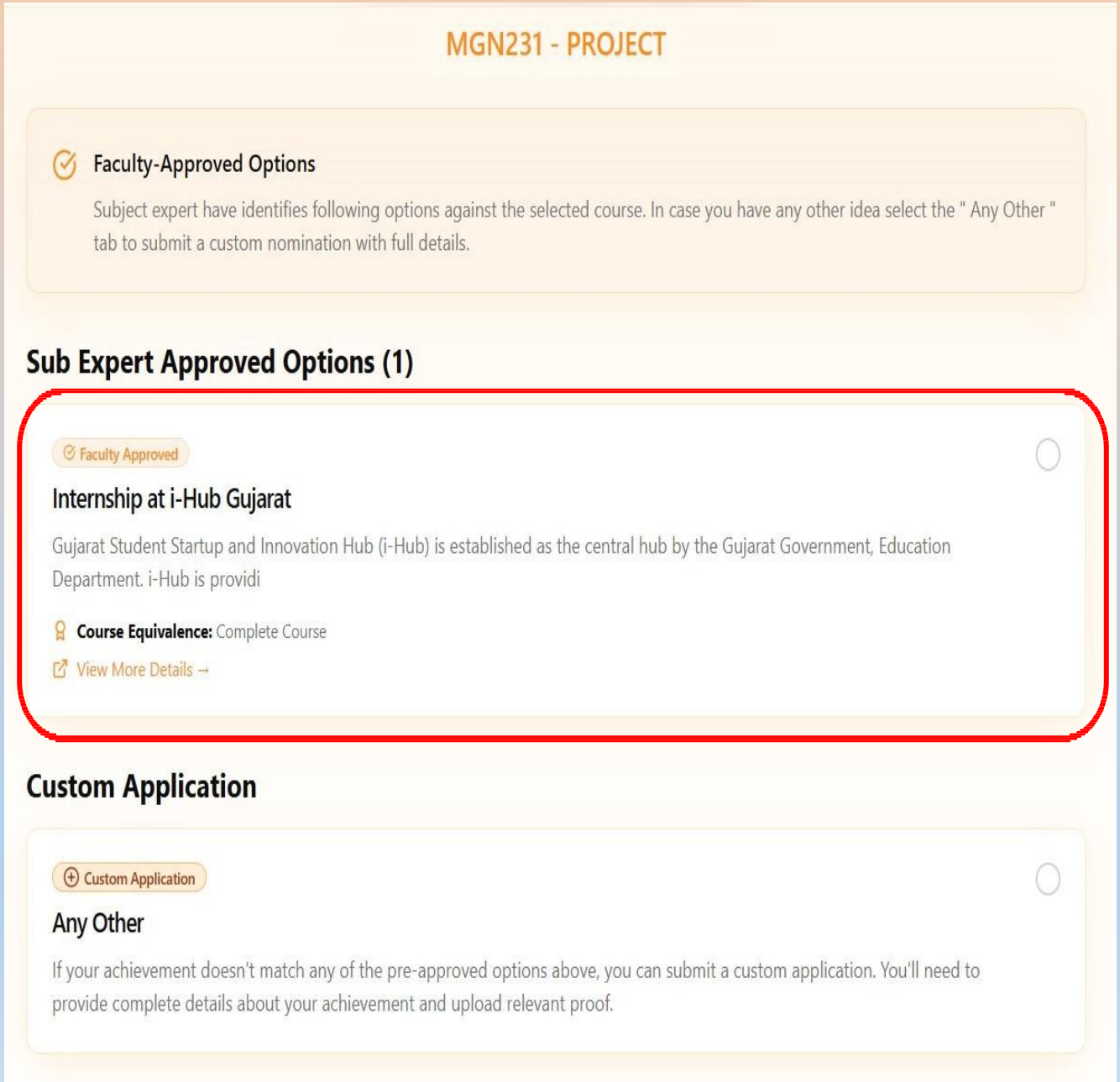
Step 4: Review Expert-Defined Options

Explore the list of courses in which you can express your interest.



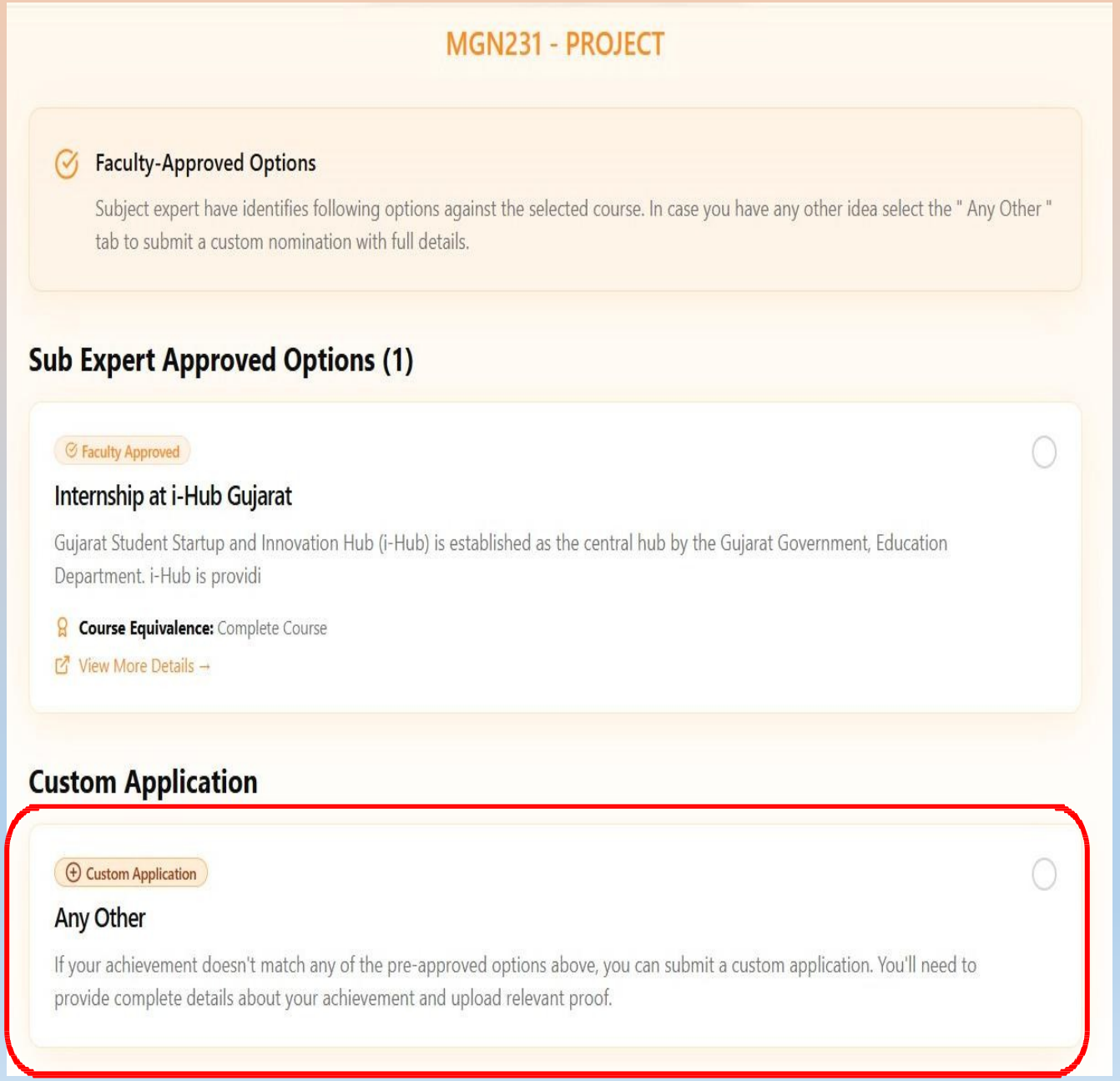
Step :: 5 Review Expert-Defined Options

- Explore the list of course-specific options curated by subject experts, where you can express your interest.

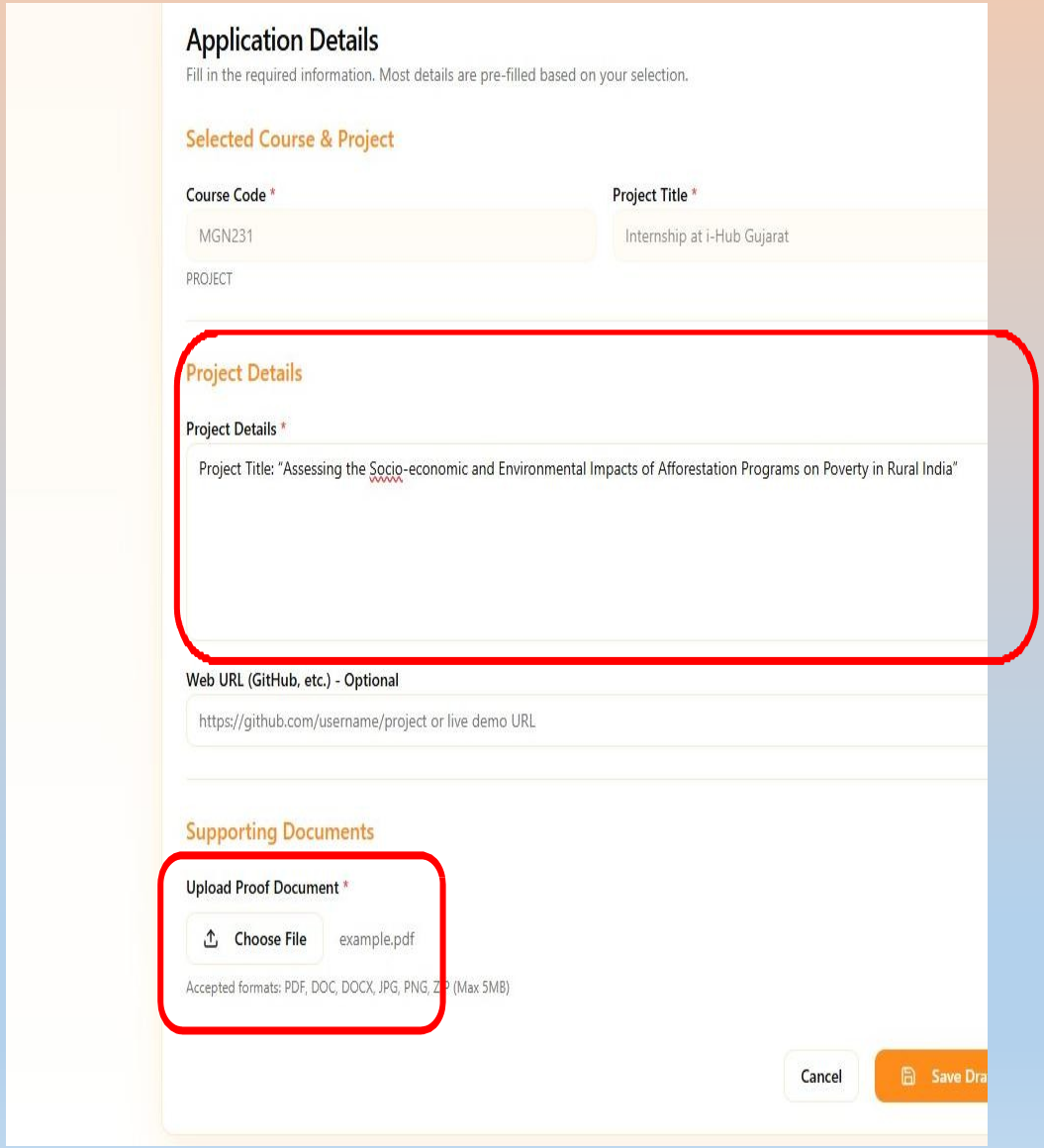


Step 5 (a) ::Use the “Any Other” Option if Needed

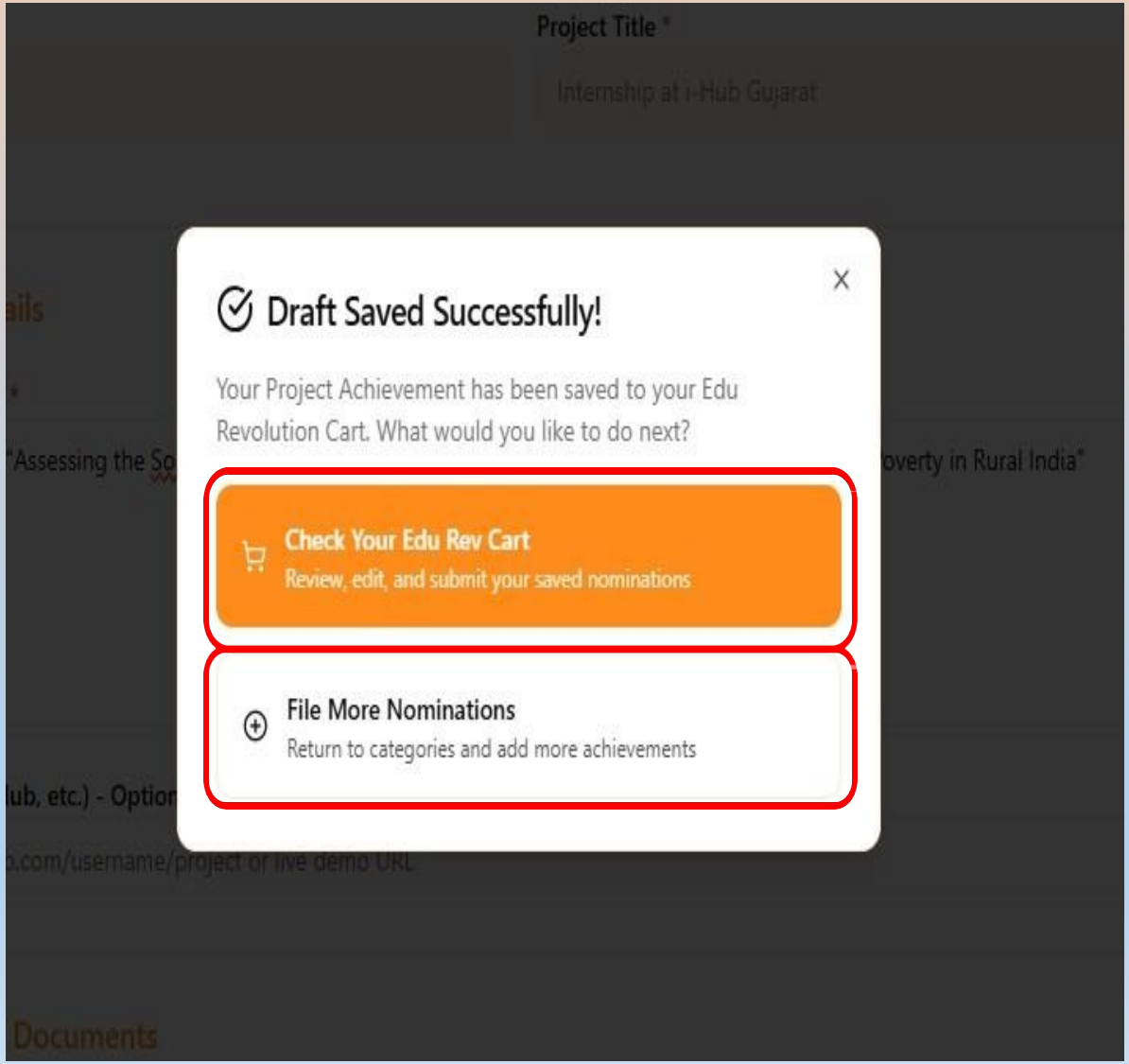
If your preferred option is not listed, choose the “Any Other” option shown under the course.



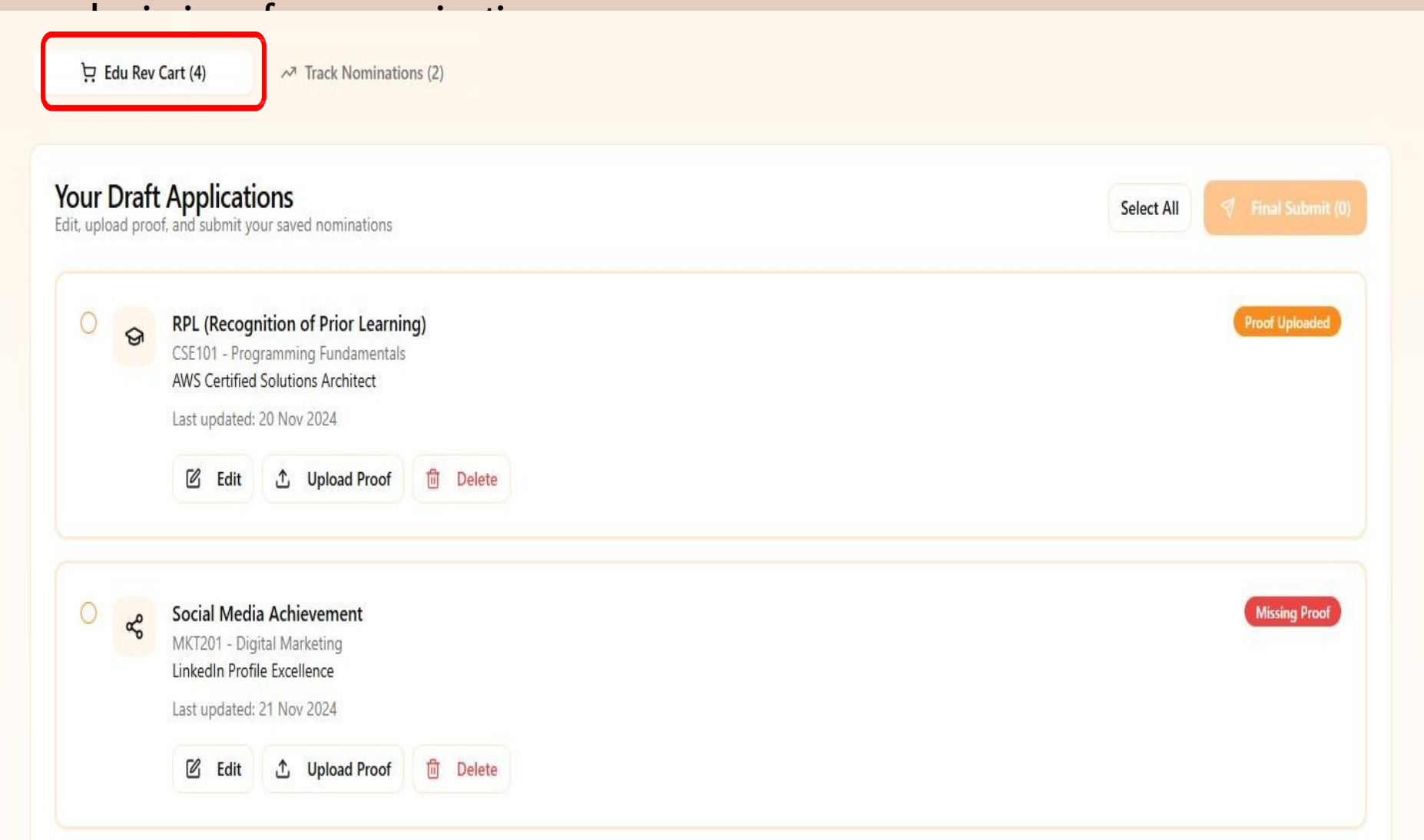
Step 6:: Fill all the relevant details and upload the required proof and click on save draft



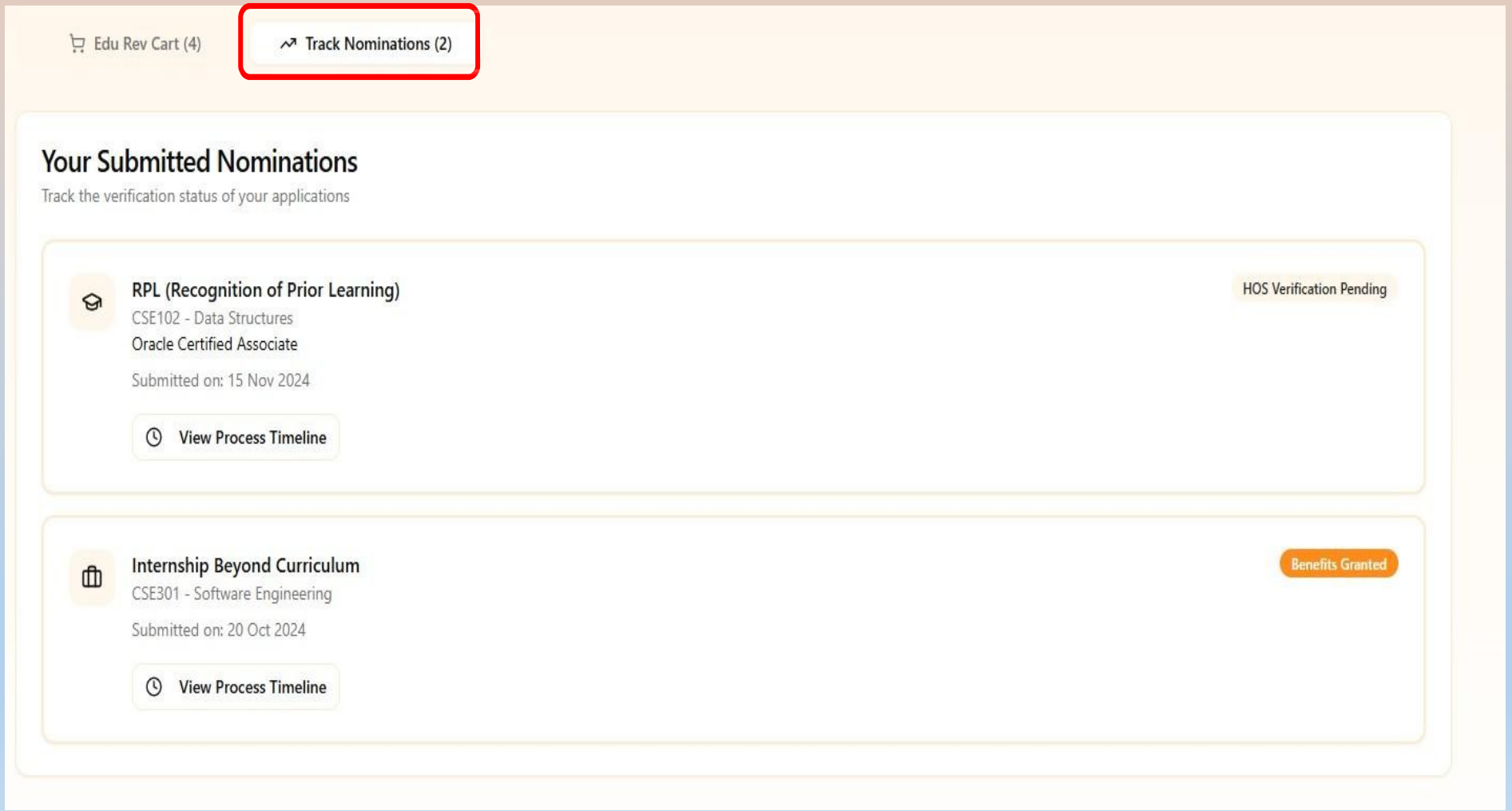
Step 7 :: Your nomination shall be added in the EDU-Revolution Cart: Which you can Review, Modify, and Remove



Step 8:: Your nomination shall be added in the EDU-Revolution Cart: Which you can Review, Modify, and Remove. You have to click on “Final Submit” for

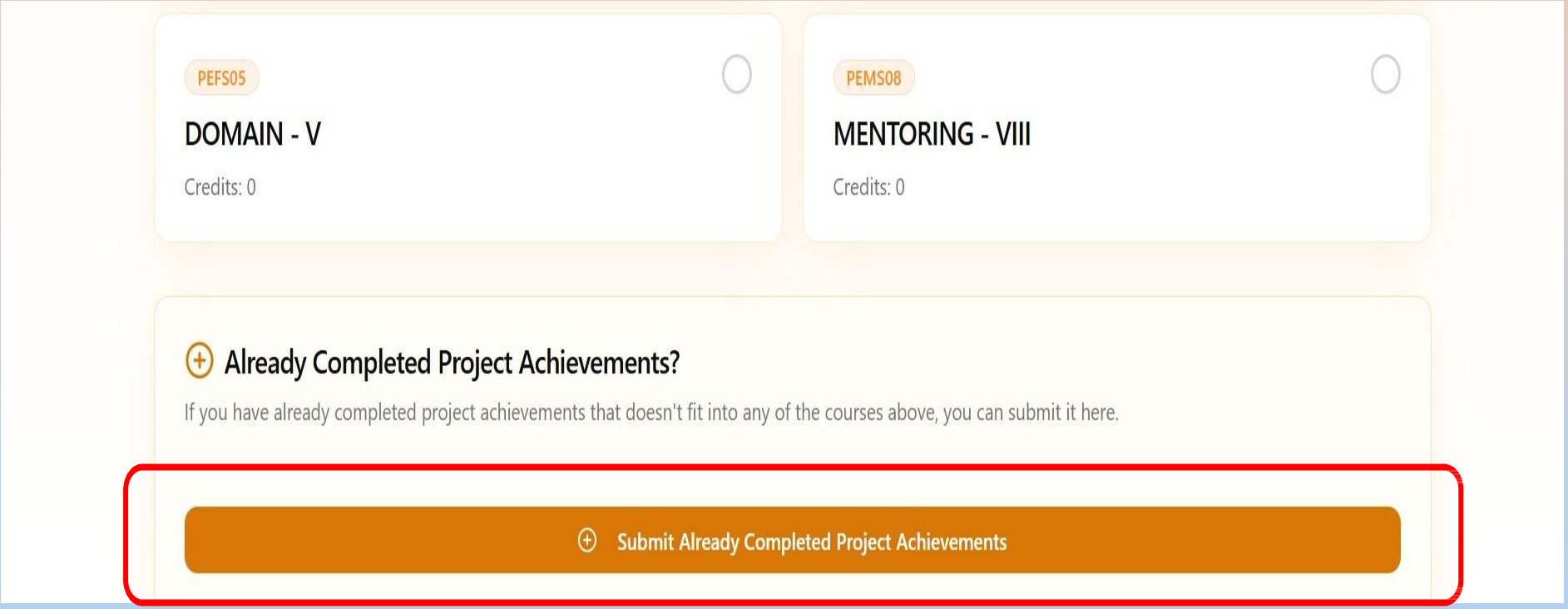


Your can track the progress from the Track Progress tab provided

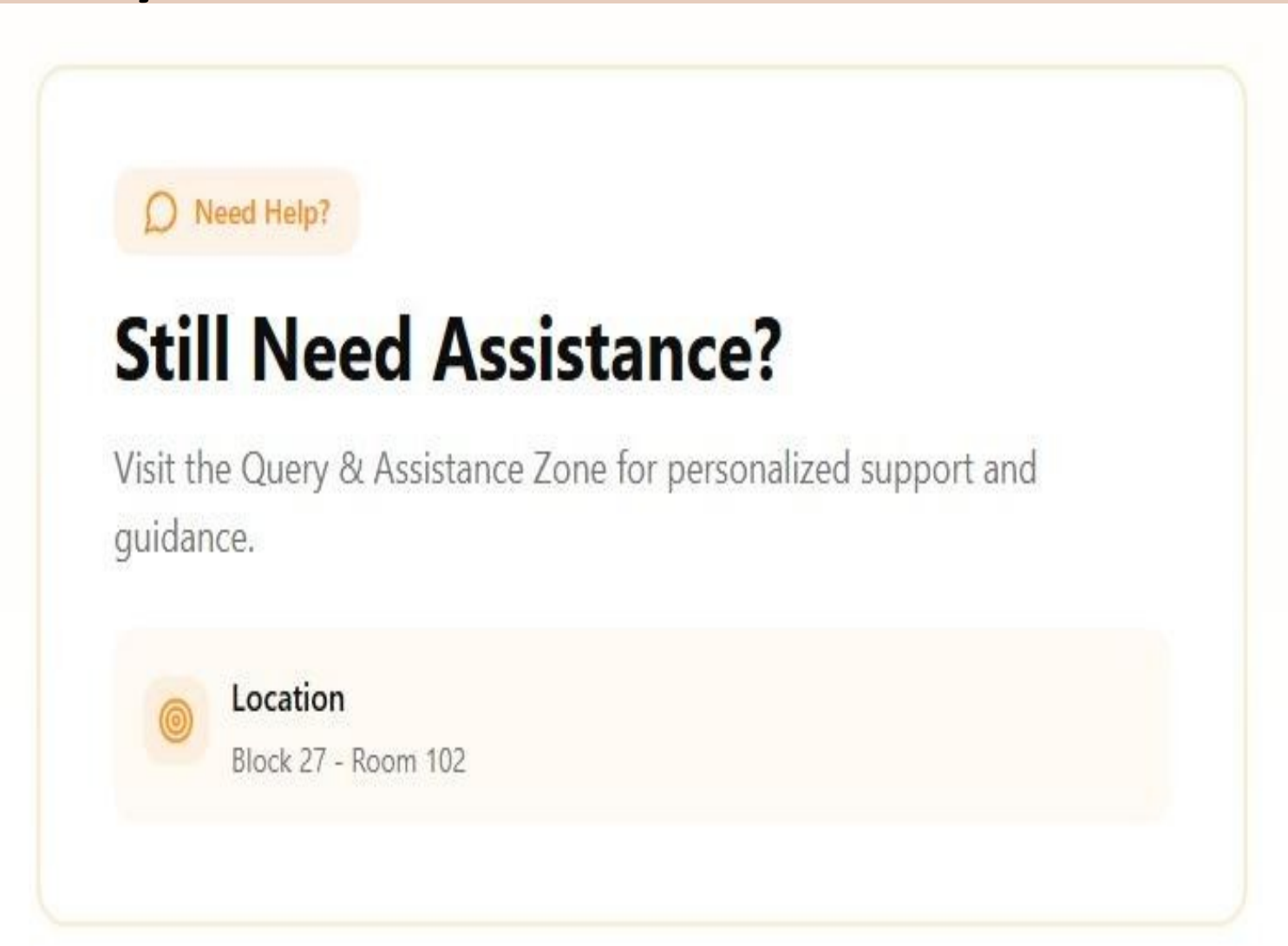


Submit Completed Initiatives

If you have already completed an initiative, upload your achievement using the “Already Completed” button and repeat the steps from Step-6 to Step-8



Query and Assistance Zone



Edu-Revolution

Consider applying on UMS > LMS > Edu-Revolution benefits OR in LPU touch

Consider applying for Edu-revolution in this term, the options are:

RPL- Recognised prior learning Gain benefits on the basis of previously learnt subject and get exempted on current subject. Before the start of each term, students who believe they already possess the required knowledge or skills in a course registered to him/her in upcoming term, can request a relaxation from the course and attendance via an online interface.	**CARE- Continuous Assessment Reduction through Experience** Gain benefits towards your alternative approaches may apply for CA modifications by opting for real-life problem-solving assignments, social service, blogs, vlogs on social media etc . The course teacher will provide details about the various CA options available for the course during the Zero Lecture.
10 % Attendance waiver for students in Pre-final/Final year of the program. Students with a CGPA of 8 or higher can apply for an attendance relaxation in their pre-final/final year. Eligible students will be granted a 10% attendance relaxation to support their career development for pursuing the following aspirations	**GROW- Grade Revision and Overall Welfare** Gain benefits by participating in internationally or nationally recognized competitions or hackathons to request for grade upgradation in an equivalent course studied in current/previous terms.

Feedback of Zero Lecture

Or

visit URL: https://forms.gle/ZgMJ9oJtrYyQoor46

Next Class: Basics of DevOps Infrastructure

THANKS

Any Queries!