Full Stack Engineering

Project Report

Semester-V (Batch-2023)

QuestEd

Supervised By:

Rahul Sir

Submitted By:

Aarushi Sharma - 2310990258 (G-4)
Geetanshi - 2310990265 (G-4)
Aditya Goyal - 2310990266 (G-4)
Chaitanya Rana - 230990316 (G-4)

Department of Computer Science and Engineering
Chitkara University Institute of Engineering & Technology,
Chitkara University, Punjab

Table of Contents

1. Abstract

QuestEd is a modern Learning Management System (LMS) designed to provide an accessible, secure, and engaging platform for both students and educators. The system integrates essential academic features such as student and educator registration/login, personalized dashboards, course creation and enrollment, video-based lectures, category-wise course browsing, and graphical analytics. It aims to create a digital ecosystem where students can enhance their learning journey while educators efficiently manage and deliver their courses.

The platform ensures interactive engagement by allowing students to manage their profiles, explore courses by categories like AI/ML and Data Analytics, enroll in courses, watch lectures, and write reviews. Educators can create and edit courses, upload lecture videos, track student engagement, and view detailed graphical analytics of their courses. QuestEd also includes secure authentication and logout features, ensuring data privacy and a smooth user experience.

QuestEd is built on a scalable **MERN stack architecture (MongoDB, Express.js, React, Node.js)** to handle concurrent users while maintaining performance and reliability. The use of MongoDB ensures efficient data storage and retrieval, while React provides a dynamic and responsive user interface. Express and Node.js enable secure, server-side logic and API management.

This report presents the project's background, objectives, problem definition, methodology, results, and future improvements. It demonstrates how QuestEd addresses the challenges of digital learning platforms by providing a comprehensive, user-friendly, and interactive LMS that empowers both students and educators in their academic journey.

2. Introduction

The rapid growth of digital technologies has transformed the way education is delivered and accessed worldwide. Traditional classroom-based learning is increasingly being complemented—or even replaced—by online platforms that offer flexibility, accessibility, and personalized learning experiences. However, many existing Learning Management Systems (LMS) are either costly, overly complex, or lack the interactivity and user-friendliness required by modern learners and educators.

QuestEd is a web-based Learning Management System designed to overcome these challenges by providing a secure, engaging, and scalable platform for both students and educators. Students can register, log in, manage their profiles, browse courses by categories such as AI/ML and Data Analytics, enroll in courses, watch video lectures, and leave reviews. Educators, in turn, can create and edit courses, upload lecture videos, and track student engagement through interactive graphical dashboards.

QuestEd is developed using the **MERN** (MongoDB, Express.js, React.js, Node.js) stack, ensuring responsiveness, efficiency, and scalability. Its design emphasizes simplicity, interactivity, and usability, making it a comprehensive solution that bridges the gap between traditional education and modern e-learning needs.

.

2.1 Background

The adoption of e-learning platforms has accelerated globally, driven by the demand for flexible, accessible, and cost-effective education. Learning Management Systems (LMS) play a crucial role in this shift by enabling structured content delivery, online collaboration, and progress tracking. Despite their importance, many existing LMS platforms are either too generic, lack personalization, or provide limited interactivity between learners and educators.

QuestEd has been developed to address these limitations by offering a feature-rich platform tailored to the needs of both students and educators. It allows students to manage their learning journey through profile management, course exploration by categories such as AI/ML and Data Analytics, enrollment, video-based learning, and peer reviews. At the same time, educators can create, update, and organize courses, upload lecture videos, and analyze learner engagement using graphical analytics.
By combining robust technology with user-centered design, QuestEd bridges the gap between conventional education and modern e-learning, ensuring a more interactive, personalized, and scalable academic environment.

2.2 Problem Statement

While online education has become increasingly popular, many existing Learning Management Systems (LMS) still fail to deliver a personalized and engaging experience for both students and educators. Students often face difficulties in discovering relevant courses by category, managing their learning progress, and providing meaningful feedback. Educators, on the other hand, encounter challenges in creating structured courses, uploading and managing lecture content, and effectively tracking learner engagement.

Moreover, most platforms lack interactive features such as **course reviews, video-based learning integration, and graphical analytics** that are essential for enhancing the teaching and learning process. The challenge, therefore, lies in designing a secure, user-friendly, and scalable LMS that enables students to explore and enroll in categorized courses, manage their profiles, and review content—while allowing educators to create, update, and analyze courses with video lectures and performance insights, all within one seamless platform.

3. System Analysis and Requirements

3.1 Objectives

The primary goal of **QuestEd** is to bridge the gap between students seeking quality education and educators who wish to deliver structured courses through a flexible and accessible digital platform. The specific objectives of the project include:

- To provide **secure student and educator authentication** through a streamlined registration and login system.
- To enable students to **explore, filter, and enroll in courses** across categories such as AI/ML, Data Analytics, and Web Development.
- To allow students to **manage their profiles, watch lectures, and write reviews**, thereby creating an interactive learning environment.
- To empower educators to **create, edit, and manage courses**, including uploading video lectures and course details.
- To provide educators with **graphical analytics** for monitoring course engagement and performance.
- To design an **intuitive and user-friendly interface** that ensures smooth navigation and enhances the overall learning experience.
- To ensure **data privacy and secure logout functionality**, building trust among both students and educators.

3.2 Features and Functionality

a. **User Registration and Authentication**

- Students and educators can create accounts by providing their basic details.
- Role-based login ensures that students and educators access features specific to their needs.
- Secure authentication with encrypted passwords and session management.

b. **Student Dashboard and Profile Management**

- Students can view and edit their profile details such as name, email, and bio.
- Dashboard displays available courses, enrolled courses, and personalized recommendations.
- Logout option ensures secure exit from the system.

c. **Course Browsing and Enrollment**

- Students can browse courses by category (AI/ML, Data Analytics, Web Development, etc.).
- Each course provides details such as title, description, cost, and duration.
- Students can enroll in courses and access all available lectures.

d. **Learning and Reviews**

- Enrolled students can access video-based lectures uploaded by educators.
- Option to leave reviews and ratings for courses to share feedback.
- Students can read reviews from peers to make informed enrollment decisions.

e. **Educator Dashboard and Course Management**

- Educators can create, edit, and publish courses by entering course name, type, description, and cost.
- Option to manage multiple courses and keep track of enrolled students.
- Courses can be updated or deleted when necessary.

f. **Lecture Management**

- Educators can add lectures with titles and upload videos for each course.
- Ability to edit or replace lecture videos as needed.
- Students access these lectures seamlessly through the course page.

g. **Graphical Analytics**

- Educators have access to visual analytics that display student enrollment trends and course engagement.
- Helps educators improve content and track their success over time.

3.3 Technology Stack

Frontend

- **React.js (v19)** → main framework for building the UI.
- **React Router DOM** → for navigation and routing between pages.
- **Redux Toolkit + React-Redux** → state management.
- **Tailwind CSS** → utility-first CSS framework for styling.
- **Axios** → for making HTTP requests to the backend.
- **Recharts** → for displaying graphical analytics.
- **React Toastify** → for showing notifications.
- **React Icons, React Spinners, React Simple Star Rating** → for UI components and loading indicators.
- **Vite** → as the build tool for faster development.

Backend

- **Node.js + Express.js (v5)** → server and REST API handling.
- **MongoDB + Mongoose** → database and ODM for schema models.
- **JWT (jsonwebtoken)** → authentication and secure sessions.
- **bcryptjs** → password hashing for security.

- **Multer** → handling file uploads (video lectures).
- **Cloudinary** → for video storage/hosting.
- **Nodemailer** → for sending emails (possibly confirmation, notifications).
- **Validator** → input validation.
- **CORS, Cookie-Parser, Dotenv** → request handling, cookies, and environment config.
- **Nodemon** → development tool for hot reload.

3.4 Software & Hardware Requirements:

Software Requirements

- **Operating System**: Windows / Linux / macOS
- **Frontend**: Any modern browser (Chrome, Edge, Firefox)
- **Backend Runtime**: Node.js (v18 or later)
- **Database**: MongoDB (local or cloud – MongoDB Atlas)
- **Package Managers**: npm / yarn
- **Deployment**: Vercel/Netlify (frontend), Render/Heroku/AWS (backend), Cloudinary (video storage)
- **Other Tools**: Git, VS Code (or any IDE)

Hardware Requirements

- **For Development:**
  - **Processor:** Intel i5 / AMD equivalent or higher
  - **RAM:** Minimum 8GB
  - **Storage:** 256GB SSD
- **For Deployment / Server:**
  - Cloud server with at least 2 vCPUs, 4–8 GB RAM, SSD storage
  - Scalable infrastructure for handling concurrent users (Cloud deployment recommended)

4. Proposed Design And Methodology

4.1 System Structure

QuestEd follows a **MERN (MongoDB, Express.js, React.js, Node.js) stack architecture**, ensuring scalability, security, and smooth user interactions. The system is designed as a **modular client-server architecture**, where the frontend, backend, and database interact seamlessly.

1. Frontend

- Built with **React.js** and **Redux Toolkit** for state management.
- Provides a **responsive and interactive UI** using Tailwind CSS.
- Enables students to register/login, view and edit profiles, browse categorized courses (AI/ML, Data Analytics, etc.), enroll in courses, watch video lectures, and provide reviews.
- Educators can log in, create and edit courses, upload video lectures, and track analytics.
- Visualization features (via **Recharts**) display course engagement and progress.

2. Backend

- Developed using **Node.js** and **Express.js** to handle API requests, authentication, and middleware.
- Implements **JWT authentication** and **bcrypt.js** for secure login and user sessions.
- Integrates **Multer** and **Cloudinary** to manage lecture video uploads and storage.
- Provides REST APIs for user management, course management, reviews, and analytics.
- Ensures **role-based access control**, separating functionalities for students and educators.

3. Database

- **MongoDB with Mongoose ODM** for flexible, schema-based data modeling.
- Stores student and educator profiles, courses, enrolled course details, video links, reviews, and analytics.
- Indexed collections ensure fast retrieval of course and user data.
- Cloud-hosted (MongoDB Atlas) for scalability and availability.

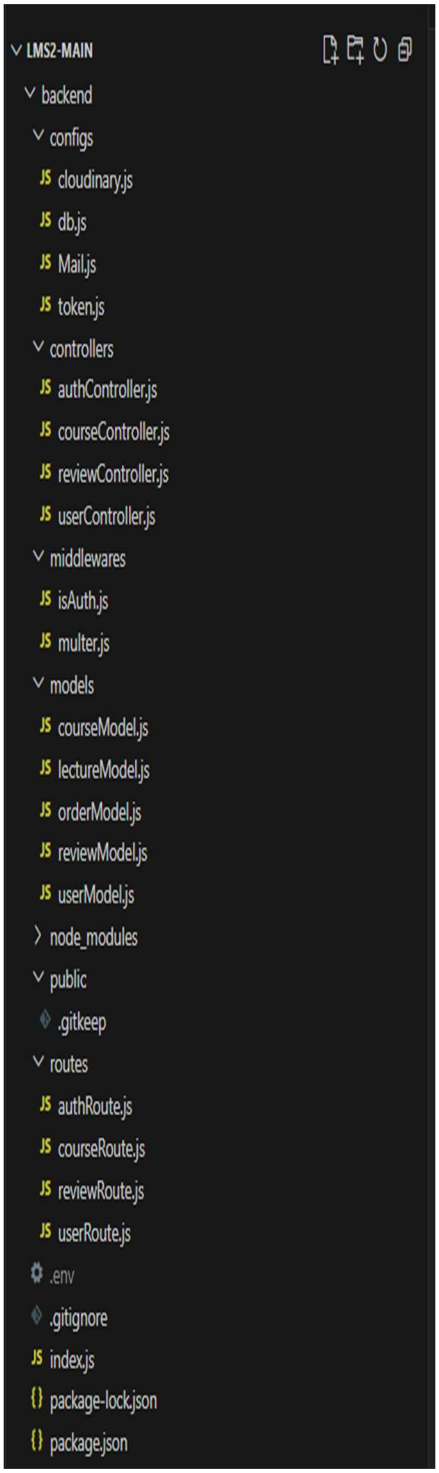
4. Analytics Module

- Educators have access to **graphical analytics** (via Recharts) to monitor course popularity, enrollment trends, and learner engagement.
- Students can track enrolled courses and performance progress.

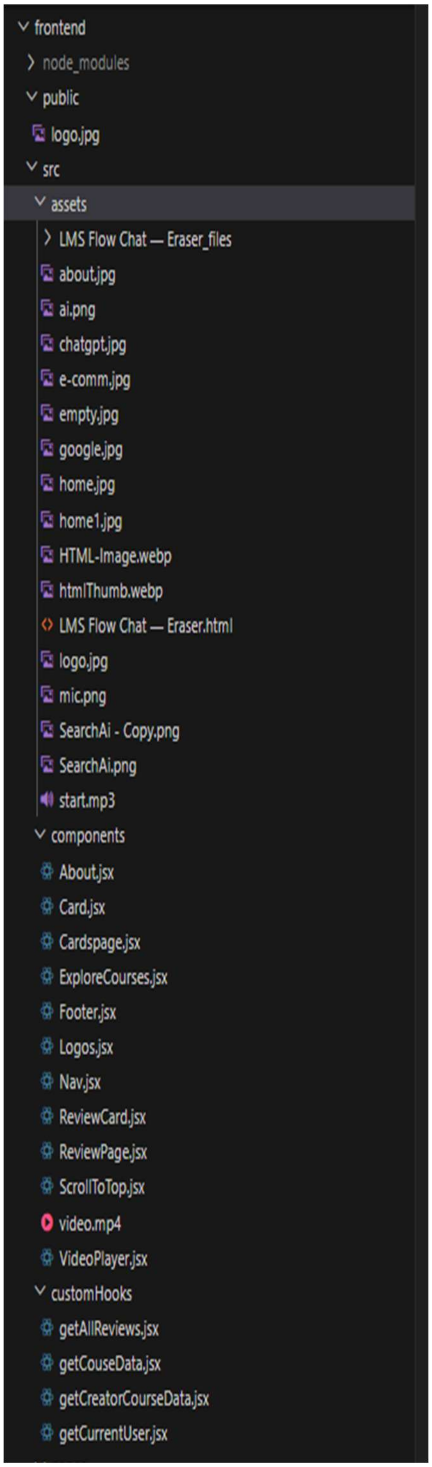
5. Security & Deployment

- **JWT tokens** secure all user sessions.
- Passwords are encrypted using **bcrypt.js**.
- **CORS and cookie-parser** ensure safe communication between client and server.
- **Deployment-ready** with frontend hosted on Vercel/Netlify and backend on Render/Heroku, with Cloudinary integration for media storage.

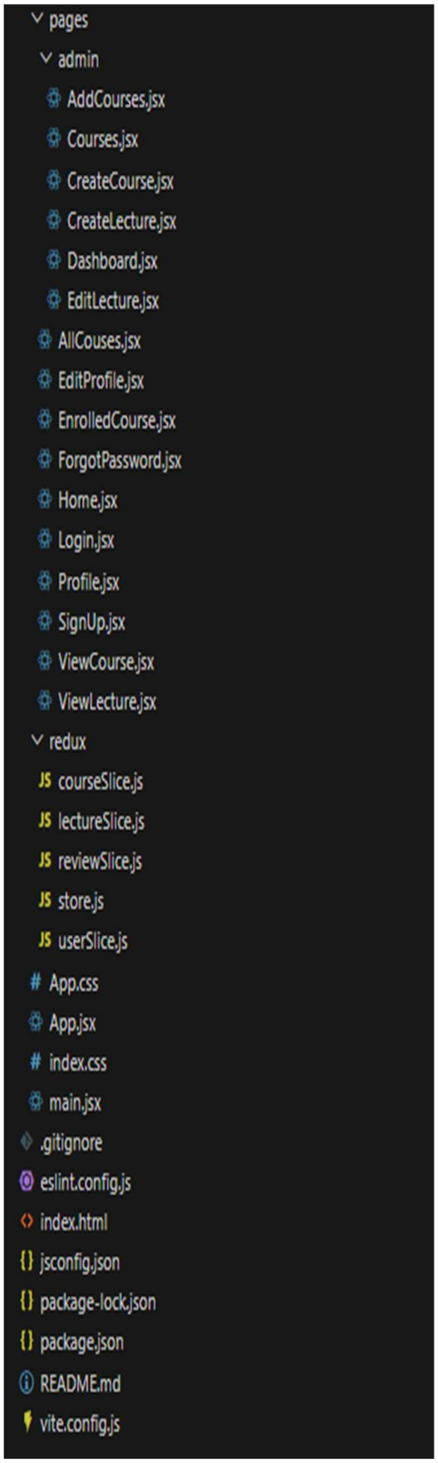
4.2 File Structure



```
LMS2-MAIN
  backend
    configs
      cloudinary.js
      db.js
      Mail.js
      token.js
    controllers
      authController.js
      courseController.js
      reviewController.js
      userController.js
    middlewares
      isAuth.js
      multer.js
    models
      courseModel.js
      lectureModel.js
      orderModel.js
      reviewModel.js
      userModel.js
    node_modules
    public
      .gitkeep
    routes
      authRoute.js
      courseRoute.js
      reviewRoute.js
      userRoute.js
    .env
    .gitignore
    index.js
    package-lock.json
    package.json
```



```
frontend
  node_modules
  public
    logo.jpg
  src
    assets
      LMS Flow Chat — Eraser_files
      about.jpg
      ai.png
      chatgpt.jpg
      e-comm.jpg
      empty.jpg
      google.jpg
      home.jpg
      home1.jpg
      HTML-Image.webp
      htmlThumb.webp
      LMS Flow Chat — Eraser.html
      logo.jpg
      mic.png
      SearchAi - Copy.png
      SearchAi.png
      start.mp3
    components
      About.jsx
      Card.jsx
      Cardspage.jsx
      ExploreCourses.jsx
      Footer.jsx
      Logos.jsx
      Nav.jsx
      ReviewCard.jsx
      ReviewPage.jsx
      ScrollToTop.jsx
      video.mp4
      VideoPlayer.jsx
    customHooks
      getAllReviews.jsx
      getCouseData.jsx
      getCreatorCourseData.jsx
      getCurrentUser.jsx
```

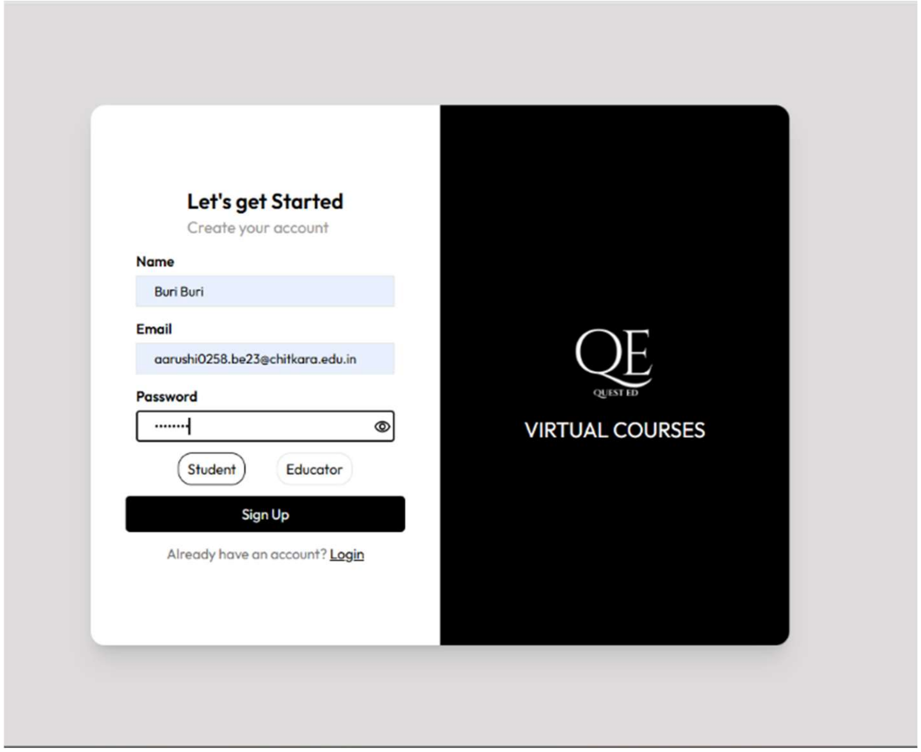


```
    pages
      admin
        AddCourses.jsx
        Courses.jsx
        CreateCourse.jsx
        CreateLecture.jsx
        Dashboard.jsx
        EditLecture.jsx
      AllCouses.jsx
      EditProfile.jsx
      EnrolledCourse.jsx
      ForgotPassword.jsx
      Home.jsx
      Login.jsx
      Profile.jsx
      SignUp.jsx
      ViewCourse.jsx
      ViewLecture.jsx
    redux
      courseSlice.js
      lectureSlice.js
      reviewSlice.js
      store.js
      userSlice.js
    App.css
    App.jsx
    index.css
    main.jsx
  .gitignore
  eslint.config.js
  index.html
  jsconfig.json
  package-lock.json
  package.json
  README.md
  vite.config.js
```

4.3 Algorithms / Logic Used

1. **Authentication & Authorization**
   - Secure login and registration using **JWT (JSON Web Tokens)**.
   - **bcrypt.js** ensures password encryption.
   - Role-based access control differentiates **student** and **educator** functionalities.
2. **Course Management Logic**
   - Educators can **create, update, and delete courses**.
   - Courses are organized by **categories** (AI/ML, Data Analytics, etc.).
   - Students can **browse, filter, and enroll** in available courses.
3. **Video Upload & Storage**
   - Lecture videos are uploaded via **Multer** and stored securely on **Cloudinary**.
   - Each course can contain multiple lectures with titles and media links.
4. **Profile Management**
   - Students and educators can **view and edit their profiles**.
   - Database updates propagate dynamically to reflect changes.
5. **Review & Rating System**
   - Students can **submit reviews and ratings** for enrolled courses.
   - Average ratings are dynamically updated for each course.
6. **Graphical Analytics**
   - Educators can view **visual insights** of their courses using **Recharts**.
   - Metrics include enrollment trends, learner engagement, and performance data.
7. **Search & Filtering Logic**
   - Students can search for courses by **name or category**.
   - Results are retrieved from MongoDB using efficient query filters.
8. **Logout & Session Handling**
   - Secure logout clears **JWT tokens** and ends active sessions.
   - Prevents unauthorized access after session termination.

5. Results



Let's get Started

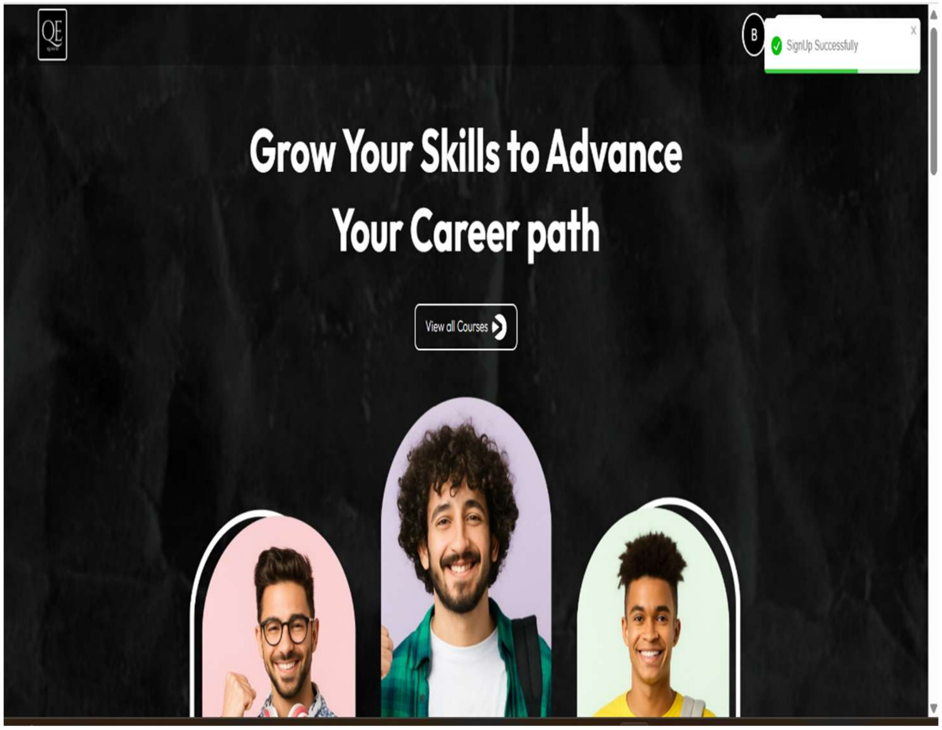
Create your account

Name: Buri Buri

Email: aarushi0258.be23@chitkara.edu.in

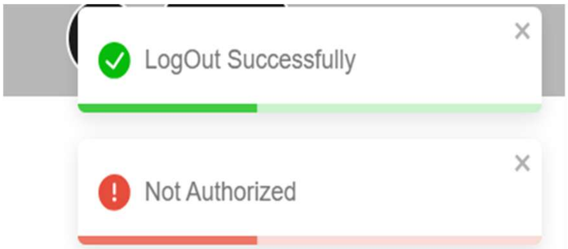
Password: •••••••

Already have an account? Login

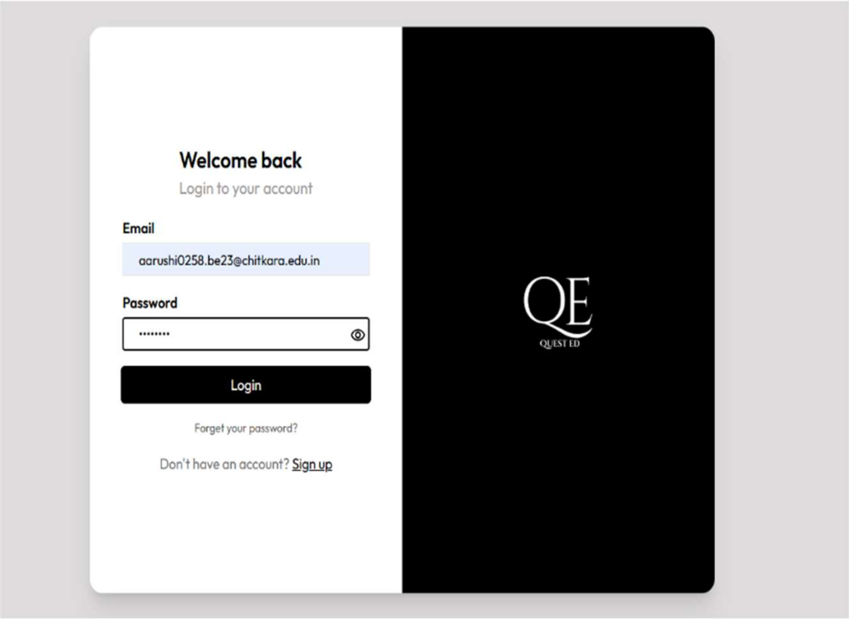


SignUp Successfully

Grow Your Skills to Advance Your Career path


LogOut Successfully

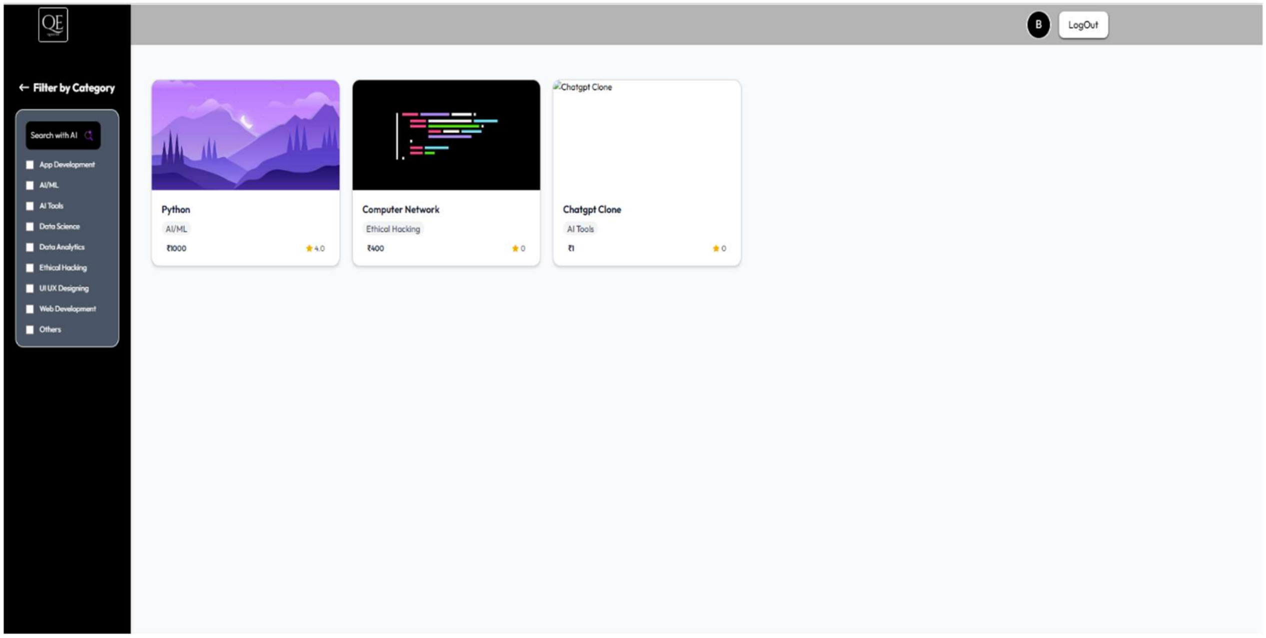
Not Authorized


Welcome back

Login to your account

Email: aarushi0258.be23@chitkara.edu.in

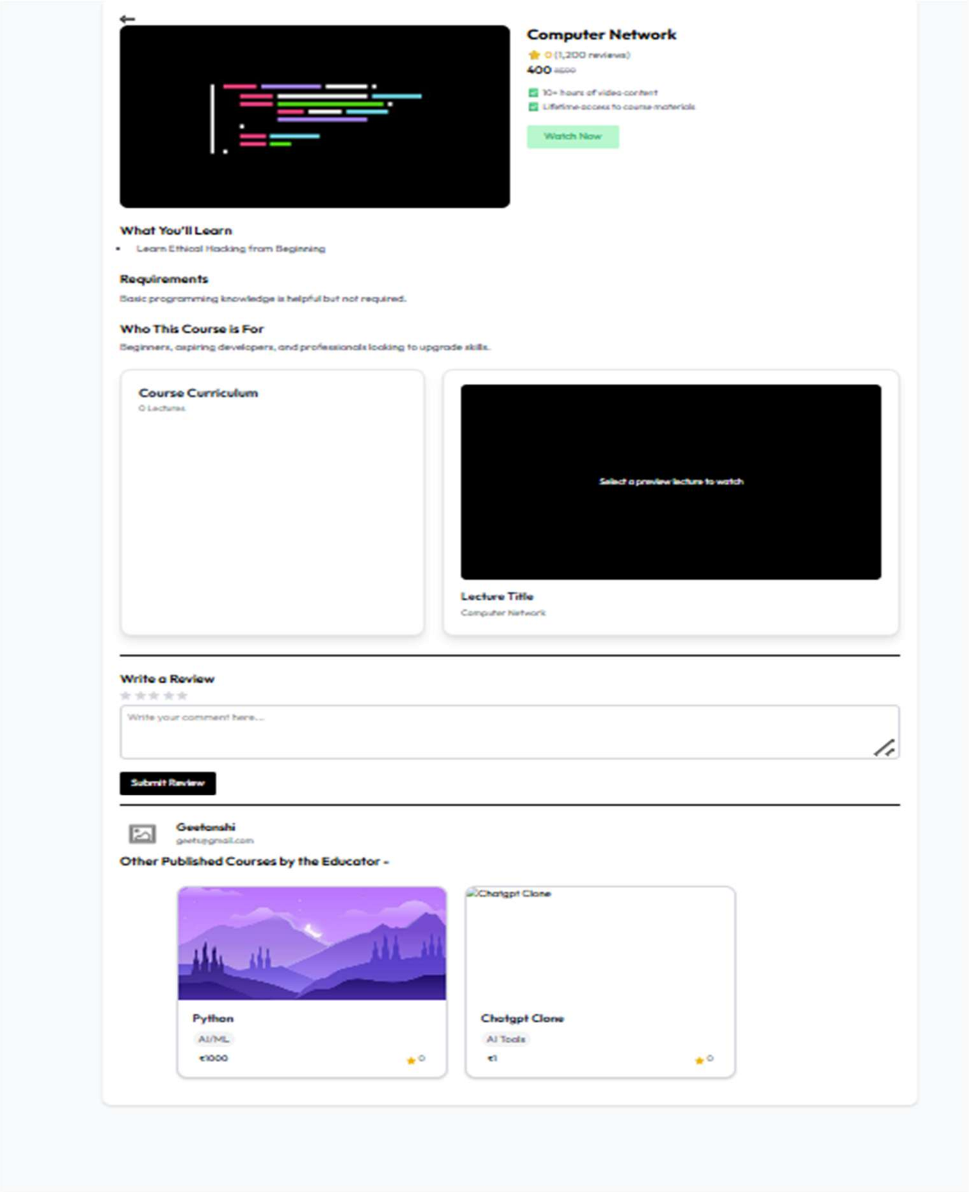
Password: •••••••••



Python
AI/ML
₹1000 ★ 4.0

Computer Network
Ethical Hacking
₹400 ★ 0

Chatgpt Clone
Chatgpt Clone
AI Tools
₹1 ★ 0



Computer Network

★ 0 (1,200 reviews)

400 ~~[illegible]~~

☑ 10+ hours of video content

☑ Lifetime access to course materials

What You'll Learn

- Learn Ethical Hacking from Beginning

Requirements

Basic programming knowledge is helpful but not required.

Who This Course is For

Beginners, aspiring developers, and professionals looking to upgrade skills.

Course Curriculum

0 Lectures

Select a preview lecture to watch

Lecture Title

Computer Network

Write a Review

☆☆☆☆☆

Write your comment here...

Geetanshi

geetu@gmail.com

Other Published Courses by the Educator -

Python

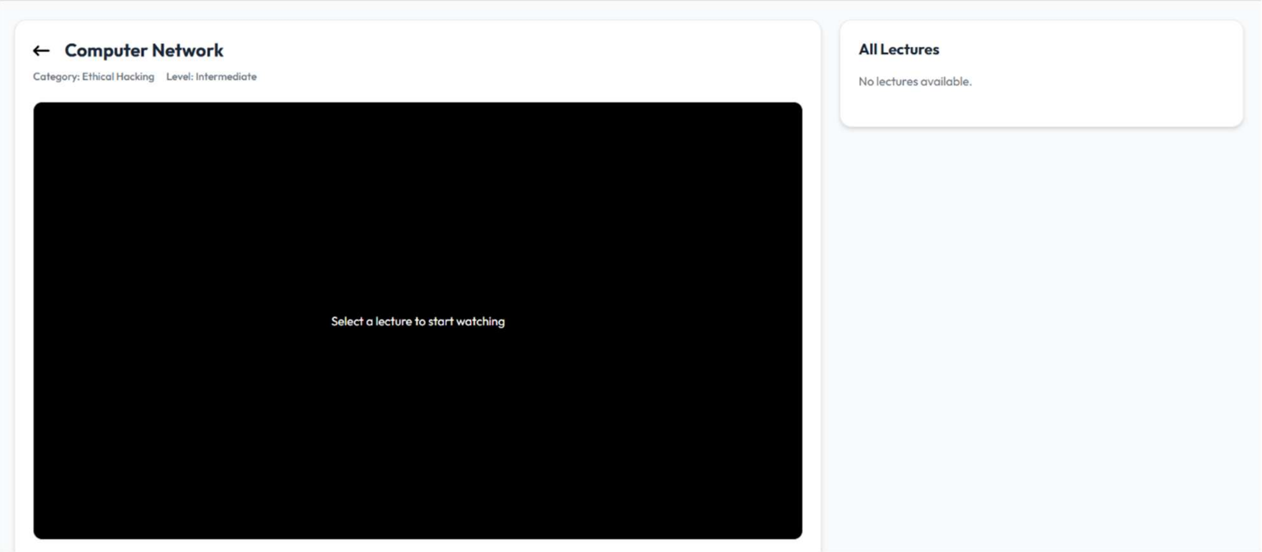
AI/ML

₹1000 ★ 0

Chatgpt Clone

AI Tools

₹1 ★ 0



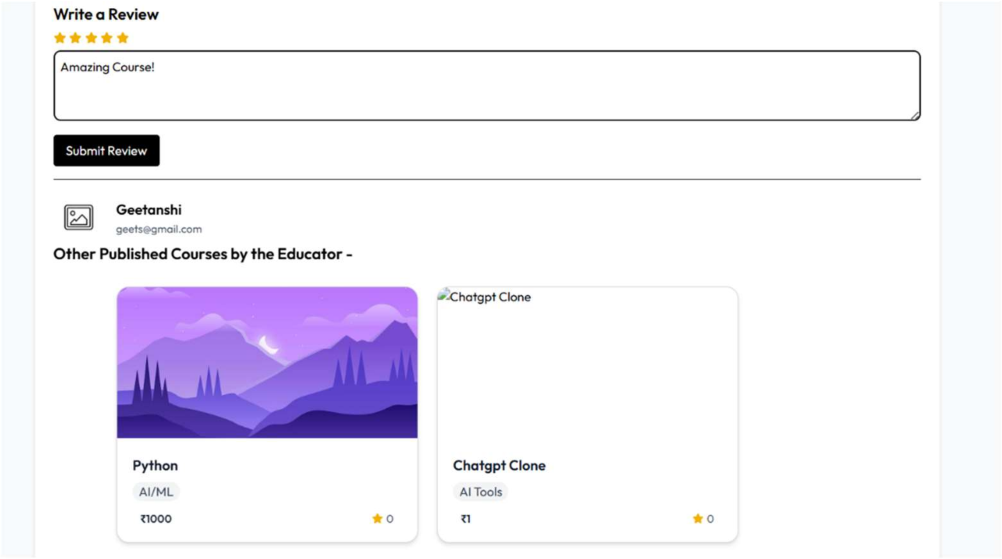
Computer Network

Category: Ethical Hacking Level: Intermediate

Select a lecture to start watching

All Lectures

No lectures available.



Write a Review

★★★★★

Amazing Course!

Submit Review

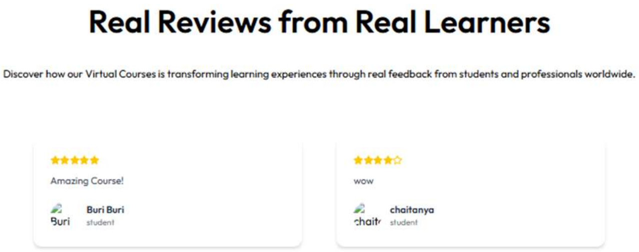
Geetanshi

geets@gmail.com

Other Published Courses by the Educator -

Chatgpt Clone

Python	Chatgpt Clone
AI/ML	AI Tools
₹1000 ★ 0	₹ ★ 0



Real Reviews from Real Learners

Discover how our Virtual Courses is transforming learning experiences through real feedback from students and professionals worldwide.

★★★★★

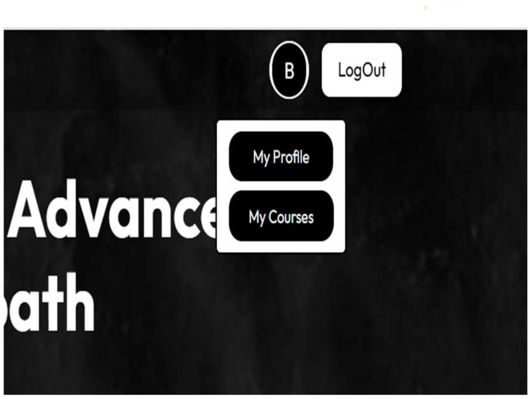
Amazing Course!

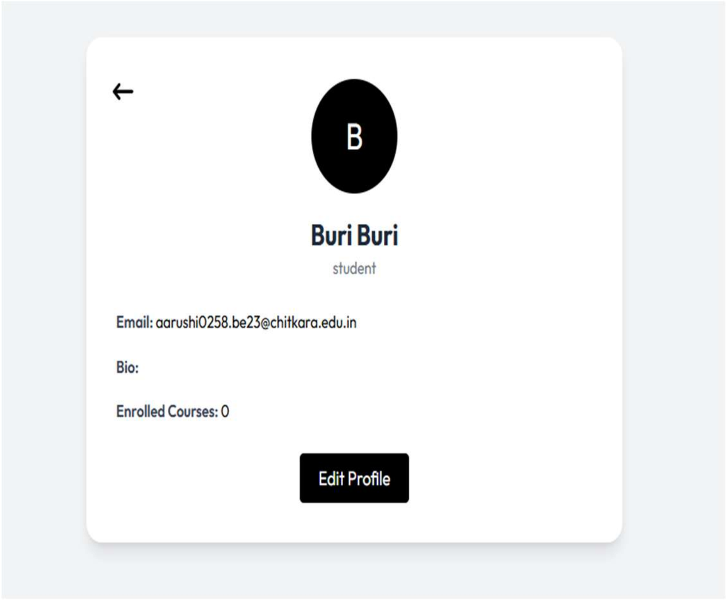
Buri
Buri Buri
student

★★★★☆

wow

chaitı
chaitanya
student





Buri Buri

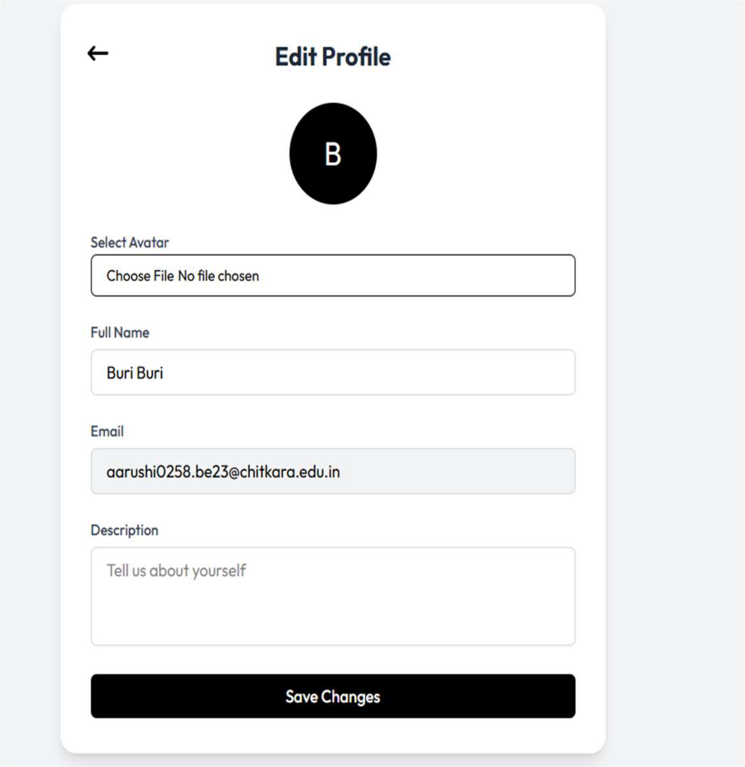
student

Email: aarushi0258.be23@chitkara.edu.in

Bio:

Enrolled Courses: 0

Edit Profile



Edit Profile

Select Avatar

Choose File No file chosen

Full Name

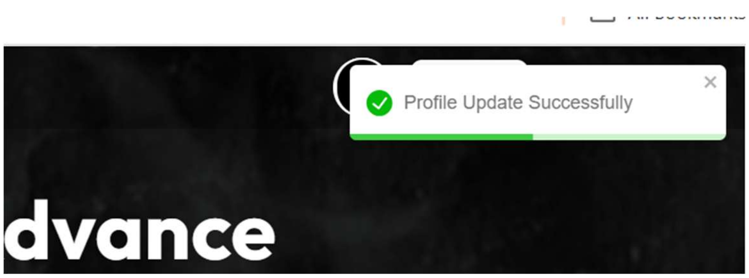
Buri Buri

Email

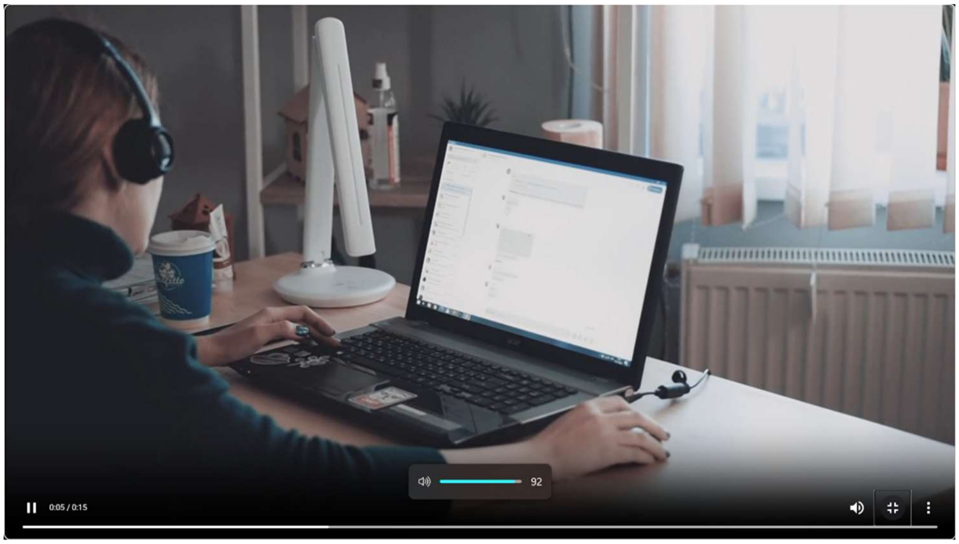
aarushi0258.be23@chitkara.edu.in

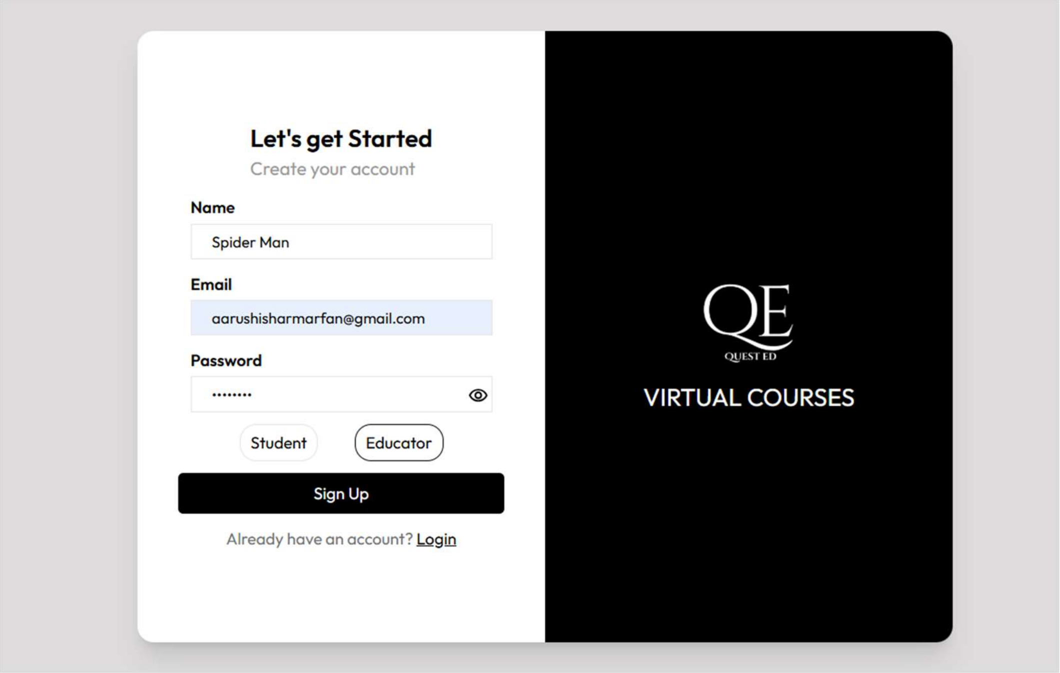
Description

Tell us about yourself



Profile Update Successfully





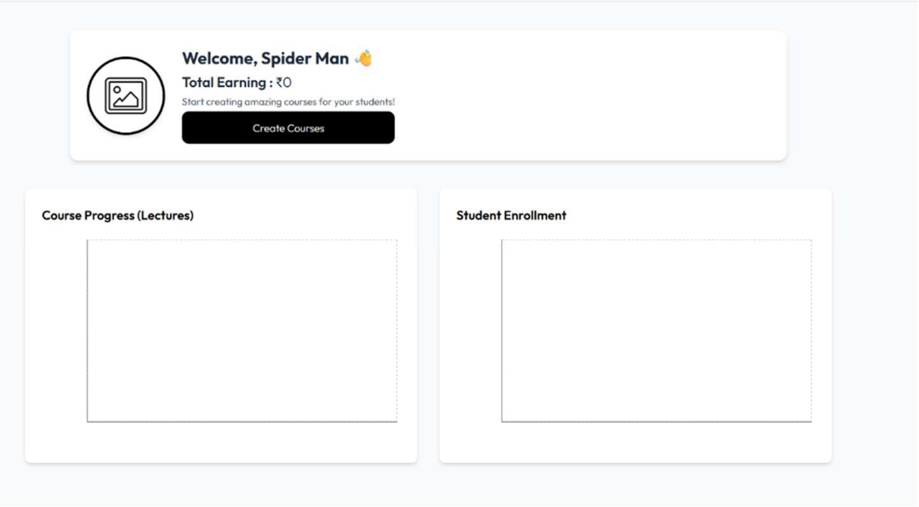
Let's get Started

Create your account

Name: Spider Man

Email: aarushisharmarfan@gmail.com

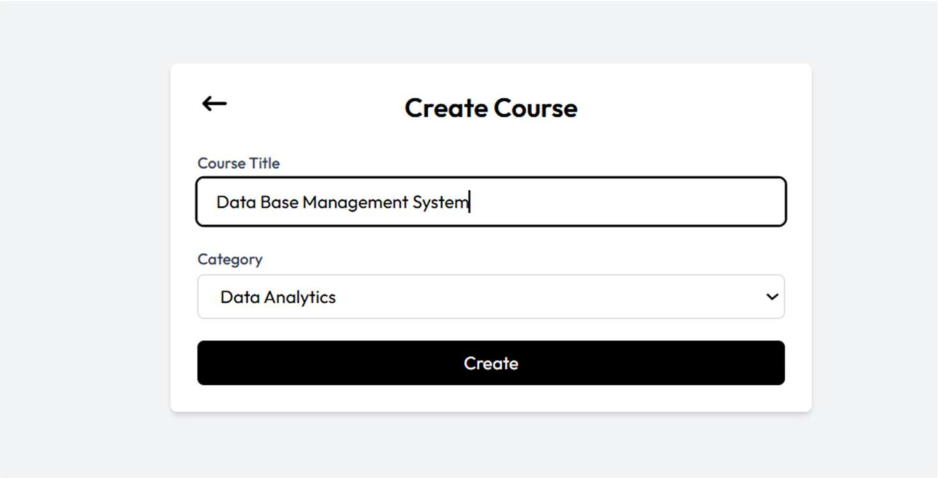
Password: ••••••••



Welcome, Spider Man 👋

Total Earning : ₹0

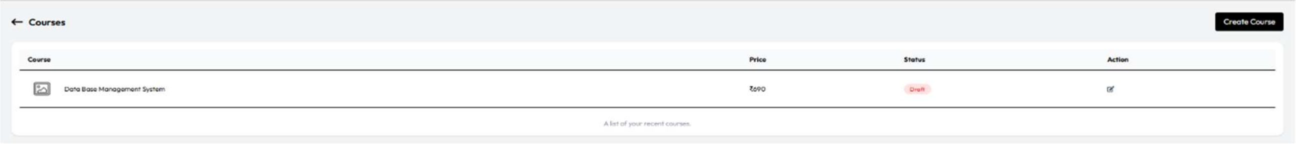
Start creating amazing courses for your students!



Create Course

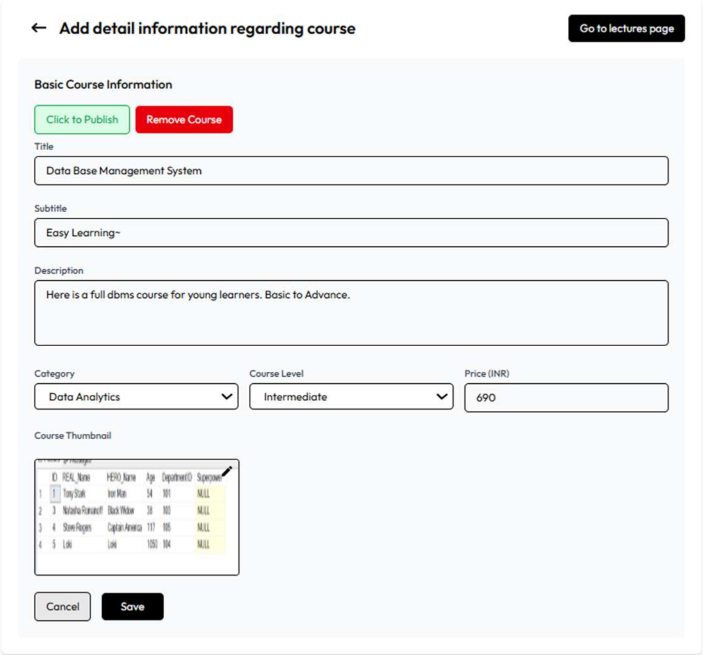
Course Title: Data Base Management System

Category: Data Analytics



Course	Price	Status	Action
Data Base Management System	₹690	Draft	

A list of your recent courses.



Basic Course Information

Title: Data Base Management System

Subtitle: Easy Learning~

Description: Here is a full dbms course for young learners. Basic to Advance.

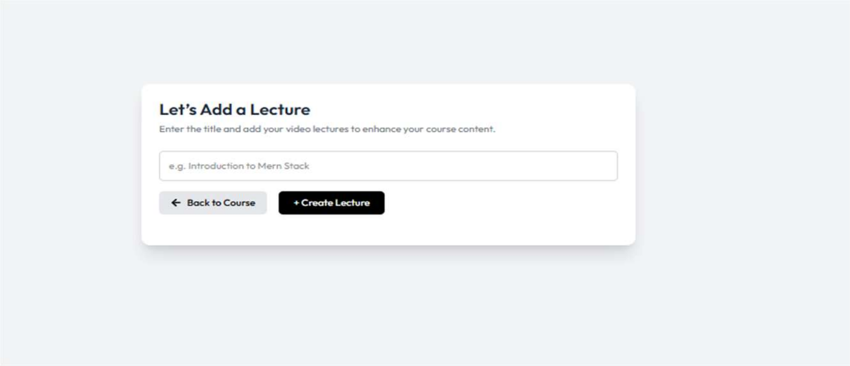
Category: Data Analytics

Course Level: Intermediate

Price (INR): 690

Course Thumbnail

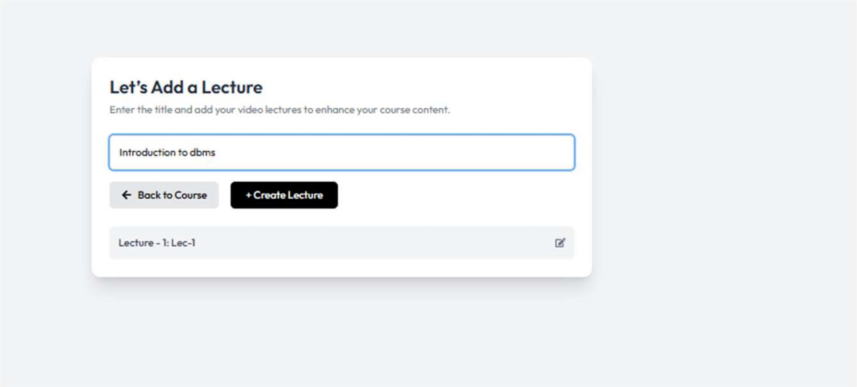
	ID	REAL_Name	HERO_Name	Age	DepartmentID	Superpower
1	1	Tony Stark	Iron Man	54	101	NULL
2	3	Natasha Romanoff	Black Widow	38	103	NULL
3	4	Steve Rogers	Captain America	117	105	NULL
4	5	Loki	Loki	1050	104	NULL



Let's Add a Lecture

Enter the title and add your video lectures to enhance your course content.

e.g. Introduction to Mern Stack

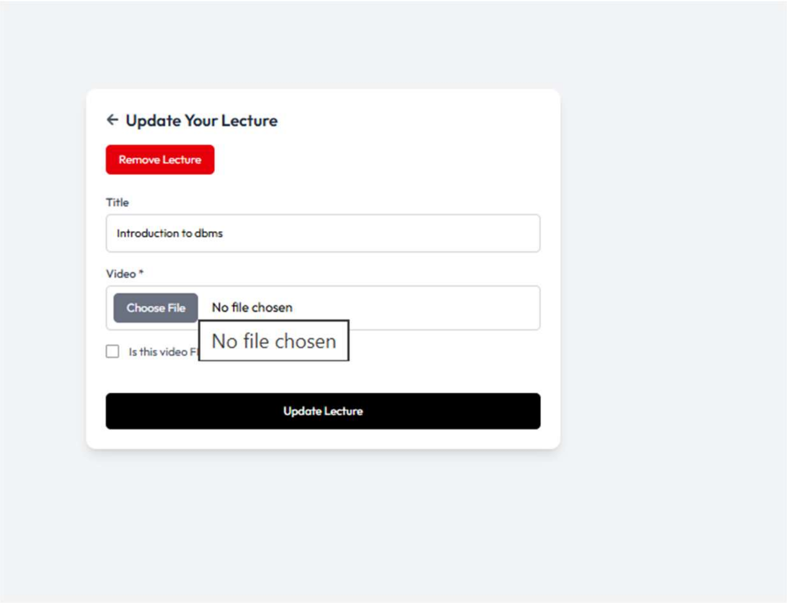


Let's Add a Lecture

Enter the title and add your video lectures to enhance your course content.

Introduction to dbms

Lecture - 1: Lec-1



← Update Your Lecture

Remove Lecture

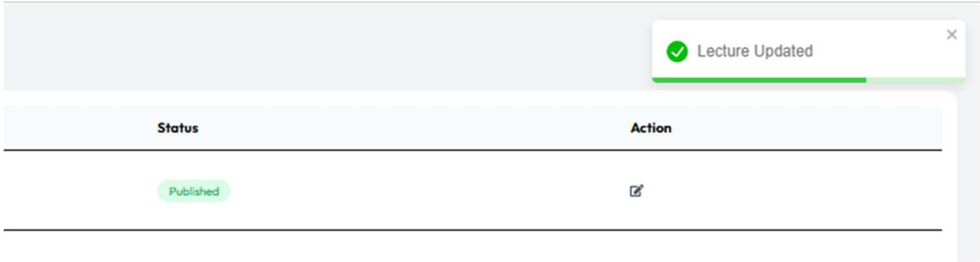
Title: Introduction to dbms

Video *: Choose File No file chosen

No file chosen

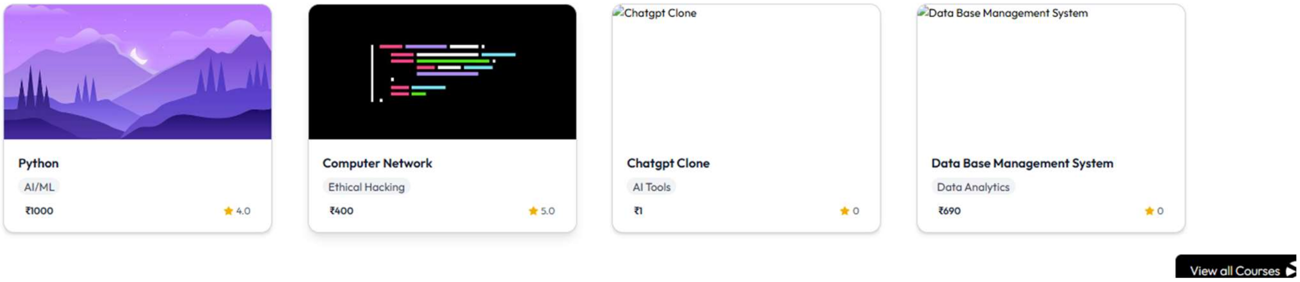
□ Is this video F

Update Lecture



Lecture Updated ×

Status	Action
Published	✎



Python AI/ML ₹1000 ★ 4.0	Computer Network Ethical Hacking ₹400 ★ 5.0	Chatgpt Clone AI Tools ₹1 ★ 0	Data Base Management System Data Analytics ₹690 ★ 0

6. References

1. **Pressman, R. S., & Maxim, B. R. (2015).** ***Software Engineering: A Practitioner's Approach*****. McGraw-Hill Education.**
2. **Bass, L., Clements, P., & Kazman, R. (2012).** ***Software Architecture in Practice*****. Addison-Wesley.**
3. **React.js – https://react.dev**
4. **Node.js – https://nodejs.org**
5. **MongoDB – https://www.mongodb.com**
6. **Express.js – https://expressjs.com**
7. **Firebase Documentation – https://firebase.google.com**