EDUCATION LEARNING PLATFORM

A

MAJOR PROJECT-II REPORT

Submitted in partial fulfillment of the requirements

for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

GROUP NO. 10

Ritik Sharma	**0187CS211136**
Shubh Dengre	**0187CS211162**
Satyam Sen	**0187CS211152**
Rohit Chourasiya	**0187CS211138**

Under the guidance of

Prof. Amit Swami

(Assistant Professor)

Department of Computer Science & Engineering

Sagar Institute of Science &Technology (SISTec), Bhopal (M.P)

Approved by AICTE, New Delhi & Govt. of M.P.

Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.)

June -2025

Sagar Institute of Science & Technology (SISTec), Bhopal (M.P.)

Department of Computer Science & Engineering

CERTIFICATE

We hereby certify that the work which is being presented in the B.Tech. Major Project-II Report entitled **EDUCATION LEARNING PLATFORM,** in partial fulfillment of the requirements for the award of the degree of ***Bachelor of Technology,*** submitted to the Department of **Computer Science & Engineering**, Sagar Institute of Science & Technology (SISTec)**,** Bhopal (M.P.) is an authentic record of our own work carried out during the period from Jan-2025 to Jun-2025 under the supervision of **Prof. Amit Swami.**

The content presented in this project has not been submitted by me for the award of any other degree elsewhere.

Ritik Sharma	**Shubh Dengre**	**Satyam Sen**	**Rohit Chourasiya**
0187CS211136	**0187CS211162**	**0187CS211152**	**0187CS211138**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Prof. Amit Swami	**Dr. Amit Kumar Mishra**	**Dr. D.K. Rajoriya**
Project Guide	**HOD, CSE**	**Principal**

ACKNOWLEDGEMENT

We would like to express our sincere thanks to **Dr. D. K. Rajoriya**, **Principal, SISTec** and **Dr. Swati Saxena, Vice Principal SISTec** Gandhi Nagar, Bhopal for giving us an opportunity to undertake this project.

We also take this opportunity to express a deep sense of gratitude to **Dr. Amit Kumar Mishra, HOD, Department of Computer Science & Engineering** for his kindhearted support

We extend our sincere and heartfelt thanks to our guide, **Prof. Amit Swami,** for providing us with the right guidance and advice at crucial junctures and for showing us the right way.

I am thankful to the **Project Coordinator, Prof. Deepti Jain,** who devoted her precious time in giving us the information about various aspects and gave support and guidance at every point of time.

I would like to thank all those people who helped me directly or indirectly to complete my project whenever I found myself in any issue.

TABLE OF CONTENTS

ABSTRACT

In the digital era, e-learning has emerged as a vital tool for providing flexible and accessible education across geographies. This project, EduVantage, is a comprehensive web-based e-learning platform developed using the MERN stack (MongoDB, Express.js, React.js, Node.js) to bridge the gap between students and educators. The platform facilitates seamless course creation by teachers, efficient content delivery, and course purchasing options for students through secure integration with Razorpay. It also features Cloudinary for optimized media storage and a built-in SQL IDE to support hands-on learning.

EduVantage follows a three-tier architecture and adopts the MVC2 design pattern, ensuring a clean separation between interface, logic, and data layers. It supports role-based access control for students, teachers, and admins, where the admin oversees all CRUD operations and manages course categories. Core functionalities are implemented using object-oriented programming concepts, enhancing modularity and scalability.

The documentation covers key software engineering aspects including problem definition, system analysis, software requirements, system design, database schema, user interface screens, and deployment strategy. The project also highlights the application of agile methodology, internal testing processes, and integration of modern development tools.

This platform not only aims to democratize learning but also provides a scalable, secure, and user-centric solution for online education, with future scope for enhancements like mobile app integration and real-time collaborative tools.

LIST OF ABBREVIATIONS

ACRONYM	FULL FORM
API	Application Programming Interface
CRUD	Create, Read, Update, Delete
DBMS	Database Management System
ERD	Entity-Relationship Diagram
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
MVC	Model-View-Controller
ODM	Object Data Modeling
OOP	Object-Oriented Programming
RAM	Random Access Memory
RBAC	Role-Based Access Control
REST	Representational State Transfer
SaaS	Software as a Service
SQL	Structured Query Language
SSL	Secure Sockets Layer
UI	User Interface
UX	User Experience
VM	Virtual Machine
VS Code	Visual Studio Code
VPN	Virtual Private Network

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 ABOUT PROJECT

EduVantage is a web-based E-learning platform crafted to offer a smooth and interactive remote learning environment for students, while providing an efficient course management system for teachers and administrators.

The platform enables the creation, enrollment, and administration of online courses, allowing educators to deliver content effectively and students to access resources anytime, anywhere.

Developed using Django (Python), HTML, CSS, Jinja templates, Tailwind CSS, and SQLite3, EduVantage delivers a scalable and responsive system for modern online education.

Key features include course creation and purchase, payment integration using Razorpay, media handling via Cloudinary, and a built-in interactive IDE for real-time database learning and query execution.

EduVantage addresses common challenges of existing E-learning systems by embedding practical, hands-on environments alongside theoretical lessons. The platform's IDE allows learners to practice SQL queries instantly, closing the gap between reading and doing.

An intuitive admin dashboard further simplifies the management of users, courses, and categories, maintaining smooth operations across the platform.

With the use of up-to-date web frameworks, secure transactions, and reliable cloud storage, EduVantage enhances the accessibility and usability of digital education.

It simplifies content delivery for educators and makes learning more engaging for students through a clean interface and interactive components. Robust security measures, such as session-based authentication and data protection, ensure a safe and seamless user experience.

1.2 PROJECT OBJECTIVES

The primary objective of EduVantage is to provide a comprehensive and user-friendly E-learning platform that enhances the teaching and learning experience by leveraging modern web technologies. The key objectives of the project include:

1.2.1 FACILITATE REMOTE LEARNING: Enable students to access high-quality educational content anytime and from anywhere, promoting flexibility in learning.

1.2.2 SIMPLIFY COURSE MANAGEMENT: Provide teachers with an intuitive dashboard to create, upload, and manage course content effectively using Django's admin and views.

1.2.3 SEAMLESS COURSE PURCHASE & PAYMENT PROCESSING: Integrate Razorpay within the Django backend for secure and efficient course transactions, ensuring smooth financial operations.

1.2.4 INTERACTIVE SQL LEARNING ENVIRONMENT: Incorporate a custom-built SQL IDE using Django forms and SQLite3, allowing students to write and run SQL queries in real-time.

1.2.5 EFFICIENT CONTENT STORAGE & DELIVERY: Use Cloudinary to handle media storage, enabling fast and secure access to videos, documents, and images within the platform.

1.2.6 ROLE-BASED ACCESS & ADMINISTRATION: Implement role-based access for admins, teachers, and students using Django’s authentication system and permissions.

1.2.7 SECURE & SCALABLE ARCHITECTURE: Develop a robust system using Django (Python) and SQLite3, ensuring high scalability, maintainability, and secure user interactions.

1.2.8 USER-FRIENDLY INTERFACE: Design a clean, responsive UI using Jinja templates and Tailwind CSS for smooth navigation and enhanced accessibility.

1.3 FUNCTIONALITY

EduVantage provides a range of functionalities tailored for students, teachers, and administrators to ensure a seamless and interactive online learning experience. The key features and functionalities of the platform are as follows:

1.3.1 STUDENT FUNCTIONALITY

- **Browse & Enroll in Courses:** Students can explore available courses and enrol in the ones they find suitable**.**
- **Secure Course Purchase:** Integrated Razorpay payment gateway enables secure transactions for purchasing paid courses.
- **Access Course Content:** Once enrolled, students can access video lectures, documents, and other learning materials.
- **Inbuilt IDE:** Students can practice SQL queries in a real-time coding environment without needing external software.
- **User Dashboard:** View purchased courses, track progress, and access learning resources easily.

1.3.2 TEACHER FUNCTIONALITY

- **Create & Manage Courses:** Teachers can upload new courses, including videos, PDFs, and other materials.
- **Organize Course Content:** Structure courses into modules or lessons for better learning flow.
- **Monitor Student Enrollment:** Track the number of students enrolled in each course.
- **Update & Delete Courses:** Modify course content as needed or remove outdated courses.

1.3.3 ADMIN FUNCTIONALITY

- **Manage Users:** Add, remove, or update student and teacher accounts.
- **CRUD Operations on Courses:** Perform Create, Read, Update, and Delete (CRUD) operations on all courses.

- **Category Management:** Organize courses into relevant categories for easier navigation.
- **Monitor Transactions:** Track payments and financial transactions securely.

1.3.4 PLATFORM-WIDE FUNCTIONALITIES

- **User Authentication & Authorization:** Secure login and access control for students, teachers, and admins.
- **Cloud-based Storage:** Store course materials securely using Cloudinary for optimized delivery.
- **Responsive UI:** Ensure smooth usability across different devices (desktop, tablet, mobile).
- **Search & Filter:** Allow users to quickly find courses based on categories, keywords, or instructors.

1.4 INTERFACE

EduVantage is designed with a user-friendly, intuitive, and responsive interface to ensure seamless navigation and interaction for students, teachers, and administrators. The interface is built using React.js, providing a smooth and dynamic user experience across different devices. The platform consists of multiple role-based interfaces to cater to the specific needs of each user type.

1.4.1 STUDENT INTERFACE

- **Home Page:** Displays featured courses, search options, and category filters.
- **Course Catalog**: Allows students to browse courses with details like title, instructor, price, and ratings.
- **Course Details Page:** Provides an overview of the course, including a syllabus, instructor profile, and preview content.
- **Dashboard:** Shows enrolled courses, progress tracking, and access to learning materials.
- **IDE:** A built-in interface where students can write and execute SQL queries in real time.

- **Payment Page:** Integrated Razorpay gateway for secure course purchases.

1.4.2 TEACHER INTERFACE

- **Course Management Panel:** Allows teachers to create, update, and delete courses.
- **Content Upload Section:** Upload course videos, PDFs, and other materials using Cloudinary for storage.
- **Student Enrollment Tracking:** View the number of students enrolled in each course.
- **Edit Course Details:** Modify descriptions, pricing, or content structure.

1.5 DESIGN AND IMPLEMENTATION CONSTRAINTS

EduVantage is designed to be a scalable, secure, and efficient E-learning platform. However, during development, certain constraints influence design choices, technology stack selection, and implementation strategies. These constraints are categorized as follows:

1.5.1 HARDWARE CONSTRAINTS

- **Server Requirements:** The platform requires a server with sufficient RAM (minimum 4GB), CPU power (Quad-core or higher), and storage (SSD preferred) to handle user requests efficiently.
- **Hosting Infrastructure:** A reliable cloud-based hosting solution (such as AWS, Vercel, Render) is needed to ensure high availability.
- **Client-side Requirements:** The platform is web-based, so it requires users to have a device (PC, laptop, or mobile) with a modern web browser.

1.5.2 SOFTWARE CONSTRAINTS

- **Technology Stack:** EduVantage is built using the MERN stack (MongoDB, Express.js, React.js, Node.js), restricting the use of other technology stacks.
- **Database Management**: Uses MongoDB, meaning relational database features like SQL joins are unavailable, which requires careful schema design and indexing for performance optimization.

- **Payment Gateway:** Razorpay is used for payment processing, meaning the platform depends on Razorpay's API availability and transaction policies.
- **Cloud Storage:** Cloudinary is used for media storage, limiting content storage to the policies and pricing of Cloudinary services.

1.5.3 FUNCTIONAL CONSTRAINTS

- **Role-Based Access:** Users must be classified as students, teachers, or admins, limiting access to certain features based on roles.
- **Course Content Restrictions:** Only teachers can upload courses, and only enrolled students can access premium content.
- **SQL IDE Limitations:** The built-in SQL IDE is restricted to executing only SELECT queries, preventing potentially harmful operations like DROP or TRUNCATE for security reasons.

1.5.4 PERFORMANCE CONSTRAINTS

- **Concurrent Users:** The platform must efficiently handle multiple students accessing courses simultaneously, requiring load balancing and caching strategies.
- **API Rate Limits:** API calls for payment processing (Razorpay) and media storage (Cloudinary) are subject to rate limits, which must be managed effectively.
- **Rendering Performance:** React.js ensures a smooth user experience, but excessive API calls or large datasets could slow down the interface.

1.5.5 SECURITY CONSTRAINTS

- **User Authentication:** The platform must use JWT (JSON Web Token) authentication for secure login and API access.
- **Data Encryption:** Payment transactions and sensitive user data must be encrypted using HTTPS and secure hashing algorithms.
- **Prevention of Unauthorized Access:** Admin and teacher panels must have proper authorization checks to prevent misuse.

1.6 ASSUMPTIONS AND DEPENDENCIES

The development and functionality of EduVantage rely on several assumptions and external dependencies. These factors influence the platform's performance, scalability, and overall effectiveness.

1.6.1 ASSUMPTIONS

- **Users Have a Stable Internet Connection:** The platform requires an active internet connection to access courses, make transactions, and interact with the IDE.
- **Users Have Basic Technical Knowledge:** Students and teachers are assumed to have basic knowledge of using web applications, including logging in, navigating the platform, and making online payments.
- **Teachers Upload High-Quality Content:** The effectiveness of courses depends on the quality of materials uploaded by teachers, including videos, documents, and exercises.
- **Users Provide Accurate Information:** Students and teachers are expected to enter valid details during registration and payment to avoid authentication or transaction failures.
- **IDE is Used for Educational Purposes Only:** It is assumed that students will use the inbuilt IDE only for learning and not attempt malicious activities like injecting harmful queries.
- **Admin Will Manage Platform Effectively:** The admin is expected to monitor and manage courses, users, and transactions responsibly to maintain system integrity.

1.6.2 DEPENDENCIES

- **Technology Stack (MERN):** The platform is dependent on MongoDB, Express.js, React.js, and Node.js for its core functionality. Any updates or changes in these technologies may impact system compatibility.

- **Payment Gateway (Razorpay):** Course purchases depend on Razorpay's API availability and transaction policies. Downtime or API failures may affect payment processing.
- **Cloud Storage (Cloudinary):** All course materials are stored using Cloudinary, making the platform reliant on its storage limits, pricing model, and API availability.
- **Hosting Service:** The platform needs a reliable hosting provider (e.g., Vercel, Render, AWS) to ensure continuous availability and performance.
- **Browser Compatibility:** The web application depends on modern web browsers (Chrome, Firefox, Edge, Safari) for optimal performance. Older browsers may not support all features.
- **Third-Party Libraries & APIs:** Various NPM packages, authentication modules, and UI frameworks are used. Any deprecation or updates in these libraries may require modifications in the codebase.

CHAPTER 2
SOFTWARE & HARDWARE REQUIREMENTS

2.1 INTRODUCTION

For the successful development, deployment, and operation of EduVantage, certain software and hardware requirements must be met. The software requirements include development frameworks, databases, cloud services, and third-party APIs essential for building and maintaining the platform. Additionally, testing tools are needed to ensure functionality, security, and performance.

On the hardware side, the platform requires adequate computing resources for development, hosting, and end-user accessibility. Developers need a powerful system for coding and testing, while the server must be robust to handle multiple users simultaneously. End users, including students and teachers, require a device with an internet connection and a modern web browser to access the platform efficiently.

This chapter provides a detailed breakdown of the essential software and hardware components required for EduVantage to function smoothly and efficiently.

2.2 SOFTWARE REQUIREMENTS

2.2.1 DEVELOPMENT TOOLS & TECHNOLOGIES

- **Frontend:** HTML5, CSS3 with Jinja2 (templating engine) and Tailwind CSS (Utility-first CSS framework for responsive UI)
- **Backend**: Django (Python-based web framework for handling server-side logic and APIs)
- **Database**: SQLite3 (Lightweight, file-based relational database integrated with Django)
- **Operating System:** Windows 10/11, macOS, or Linux
- **Internet Connection:** Stable broadband connection for real-time testing and cloud-based media integration

2.2.2 TESTING TOOLS

- **Postman:** API testing and debugging
- **Django Test Client & unittest**: Used for unit and integration testing of views, models, and forms

2.2.3 DEPLOYMENT & HOSTING SERVICES

- Web Hosting: Render / Railway / Heroku for Django app deployment
- Media Hosting: Cloudinary for storing and managing course-related media files
- Domain Management: Namecheap / GoDaddy for domain and SSL setup
- Version Control: Git & GitHub for collaborative source code management

2.3 HARDWARE REQUIREMENTS

2.3.1 DEVELOPER SYSTEM REQUIREMENTS

- **Processor:** Intel i3 or AMD Ryzen 5 (or higher)
- **Storage:** Minimum 250GB SSD (Recommended: 512GB SSD)
- **Operating System:** Windows 10/11, macOS, or Linux
- **Internet Connection:** Stable broadband connection for real-time testing
- **RAM:** Minimum 8GB RAM (Recommended: 16GB)

2.3.2 SERVER REQUIREMENTS (HOSTING)

- **Processor:** Quad-core or higher
- **RAM:** Minimum 8GB RAM (Recommended: 16GB for handling multiple concurrent users)
- **Storage:** Minimum 100GB SSD (Expandable based on media storage needs)
- **Bandwidth:** High-speed internet with unlimited data transfer

2.3.3 END-USER REQUIREMENTS

- **Device:** Desktop, Laptop, Tablet, or Smartphone
- **Processor:** Dual-core or higher (for smooth course playback)
- **RAM:** Minimum 4GB RAM (Recommended: 8GB)

- **Browser:** Latest versions of Chrome, Firefox, Edge, Safari
- **Internet Speed:** Minimum 5 Mbps (Recommended: 10 Mbps or higher for smooth video streaming)

CHAPTER 3
PROBLEM DESCRIPTION

3.1 OVERVIEW

With the increasing demand for online education, traditional learning methods face several limitations. Many students and educators struggle with limited access to quality resources, lack of interactive learning environments, and inefficient course management systems. Additionally, existing E-learning platforms often suffer from high costs, complex interfaces, or restricted functionalities, making it difficult for educators to create and sell courses easily.

EduVantage aims to address these challenges by providing a user-friendly, scalable, and feature-rich E-learning platform that facilitates seamless course creation, student engagement, and efficient administration.

3.2 PROBLEMS IN THE EXISTING SYSTEM

3.2.1 LIMITED ACCESSIBILITY & FLEXIBILITY

- Many students lack access to quality educational content due to geographical and financial constraints.
- Traditional learning methods require physical presence, limiting flexibility for learners.

3.2.2 INEFFICIENT COURSE MANAGEMENT FOR TEACHERS

- Many platforms lack simplified tools for teachers to create and manage courses easily.
- Uploading, structuring, and updating course materials can be time-consuming and complex.

3.2.3 HIGH COSTS & PAYMENT INCONVENIENCE

- Some existing platforms charge high commission fees on course sales, making it less profitable for educators.
- Secure and seamless payment processing is often not integrated efficiently.

3.2.4 LACK OF HANDS-ON LEARNING OPPORTUNITIES

- Many platforms provide only theoretical learning without interactive tools for practical application.
- Students studying databases often struggle without an integrated IDE for hands- on practice.

3.2.5 STORAGE & SCALABILITY ISSUES

- Managing large volumes of videos, documents, and other resources requires efficient cloud storage solutions.
- Many platforms lack a scalable system that can handle increasing users and course content efficiently.

3.2.6 SECURITY & ROLE MANAGEMENT ISSUES

- Some platforms do not have proper role-based access, leading to security risks.
- Admins need better control over course categories, user management, and financial transactions.

3.3 PROJECT IMPACT AND SIGNIFICANCE

EduVantage addresses these challenges by providing a scalable, feature-rich E-learning platform with the following key solutions:

3.3.1 REMOTE LEARNING ACCESSIBILITY: Students can access courses anytime, anywhere, removing geographical barriers.

3.3.2 EFFORTLESS COURSE MANAGEMENT: Teachers can easily upload, structure, and update courses with a user-friendly interface.

3.3.3 SECURE PAYMENT GATEWAY: Integrated Razorpay ensures safe and seamless transactions for course purchases.

3.3.4 HANDS-ON SQL PRACTICE: An inbuilt IDE allows students to write and execute queries in real-time.

3.3.5 CLOUD STORAGE FOR CONTENT: Cloudinary is used for efficient and secure storage of course materials.

3.3.6 ROLE-BASED ACCESS CONTROL: Admins, teachers, and students have dedicated functionalities, ensuring security and organized management.

CHAPTER 4
LITERATURE SURVEY

4.1 LITERATURE SURVEY

The evolution of E-learning platforms has significantly transformed modern education, providing flexible and accessible learning opportunities. This section explores the existing research, technologies, and tools used in E-learning systems, identifying their strengths and limitations to highlight how EduVantage improves upon them.

4.1.1 EVOLUTION OF E-LEARNING SYSTEMS

The concept of E-learning has evolved from simple web-based tutorials to advanced AI-powered, cloud-integrated, and interactive learning platforms. Initially, Learning Management Systems (LMS) such as Moodle, Blackboard, and Canvas provided structured course management features, but they were often complex to set up and required institutional backing [1].

The rise of MOOCs (Massive Open Online Courses) such as Coursera, Udemy, and edX revolutionized online learning, making high-quality education accessible to a global audience [2]. However, these platforms often impose high commission fees on educators, making it difficult for independent instructors to monetize their courses. EduVantage aims to bridge this gap by offering a cost-effective, flexible, and educator-friendly course-selling platform.

4.1.2 ROLE OF TECHNOLOGY IN E-LEARNING

Modern E-learning systems leverage several cutting-edge technologies to enhance learning experiences:

4.1.2.1 MERN Stack for Web-Based E-learning

- MongoDB, Express.js, React.js, and Node.js (MERN) provide a scalable, real- time, and interactive web experience [3].

- EduVantage utilizes MERN for efficient frontend-backend communication, ensuring a smooth user experience.

4.1.2.2 Cloud-Based Storage Solutions

- Platforms such as Google Drive, AWS S3, and Cloudinary offer scalable content storage for educational platforms [4].
- EduVantage integrates Cloudinary for secure, scalable media storage, ensuring seamless course content delivery.

4.1.2.3 Payment Gateway Integration

- Secure online transactions are essential for E-learning platforms [5].
- Platforms like Udemy use PayPal and Stripe, while EduVantage integrates Razorpay for fast and secure payments.

4.1.2.4 Built-in IDE for Practical Learning

- Many traditional platforms lack an interactive coding environment for database learning [6].
- EduVantage provides an integrated IDE, allowing students to run queries in real time, enhancing hands-on learning.

4.1.3 CHALLENGES IN E-LEARNING

4.1.3.1 Student Engagement & Retention

- Research shows that only 10-15% of students complete online courses due to lack of motivation [7].
- Solution: EduVantage provides an interactive learning experience with SQL practice, engaging UI, and structured course progress tracking.

4.1.3.2 Security & Payment Fraud Prevention

- Online platforms face risks such as unauthorized access, data breaches, and payment fraud [8].
- Solution: EduVantage uses JWT authentication, HTTPS encryption, and Razorpay's secure API for fraud prevention.

4.1.3.3 Scalability & Performance Optimization

- Platforms must handle thousands of concurrent users without slowing down.
- Solution: EduVantage utilizes MongoDB indexing, server-side caching, and optimized API calls for better performance.

4.1.4 FUTURE OF E-LEARNING PLATFORMS

The future of E-learning will focus on:

- AI-driven personalized learning paths [10].
- VR/AR-based interactive education [11].

CHAPTER 5
SOFTWARE REQUIREMENTS SPECIFICATION

5.1 FUNCTIONAL REQUIREMENTS

5.1.1 USER ROLES AND FUNCTIONALITIES

1. **Admin**
   - Manage users
   - Perform CRUD operations on courses.
   - Create and manage course categories.
   - View and manage payment transactions.
2. **Teacher**
   - Register/Login using email authentication.
   - Create, edit, and delete courses.
   - Upload course content
   - Set course pricing and manage enrolled students.
   - View earnings and transaction history.
3. **Student**
   - Register/Login using email authentication.
   - Browse available courses and filter by category.
   - Purchase courses securely using Razorpay.
   - Access enrolled courses and study materials.
   - Execute SQL queries using the built-in IDE.

5.1.2 PAYMENT SYSTEM FUNCTIONALITY

- Secure payment processing via Razorpay API.
- Automated invoice generation after payment.

5.1.3 COURSE MANAGEMENT

- Teachers can create, update, and delete courses.
- Admin has the authority to manage all course listings.
- Courses include multiple content types.
- Course progress tracking for students.

5.1.4 IDE FUNCTIONALITY

- Students can write and execute SQL queries.
- Teachers can assign SQL-based assignments.
- Supports basic SELECT operations on a sample database.

5.1.5 AUTHENTICATION & SECURITY

- JWT-based authentication for secure logins.
- Role-based access control for users.
- Encryption for sensitive user data (e.g., passwords).

5.2 NON-FUNCTIONAL REQUIREMENTS

5.2.1 PERFORMANCE REQUIREMENTS

- The system should handle at least 500 concurrent users without lag.
- Course videos should load within 2 seconds on a standard broadband connection.
- SQL IDE should process queries within 1 second for a smooth experience.

5.2.2 SECURITY REQUIREMENTS

- All user data must be encrypted using bcrypt for passwords.
- Secure payment processing with Razorpay’s PCI-DSS compliance.
- HTTPS enforced for secure communication between client and server.

5.2.3 USABILITY REQUIREMENTS

- The platform should have a user-friendly interface for easy navigation.
- Responsive design to work on desktops, tablets, and smartphones.

5.2.4 SCALABILITY REQUIREMENTS

- The backend should support horizontal scaling using load balancers.
- Cloudinary storage should scale dynamically based on content uploads.
- The database (MongoDB) should handle millions of records efficiently.

5.2.5 AVAILABILITY & RELIABILITY

- 99.9% uptime guarantee with cloud-based deployment.
- Automatic failover mechanisms to prevent downtime.

5.2.6 MAINTAINABILITY & EXTENSIBILITY

- Modular codebase following MVC architecture.
- Microservices-ready design for future scalability.
- Easily extendable to add new features like AI-powered course recommendations.

5.2.7 COMPLIANCE REQUIREMENTS

- Follows GDPR for data protection and privacy.
- Payment system complies with PCI-DSS security standards.
- Content policies align with DMCA copyright regulations.

CHAPTER 6
SOFTWARE DESIGN

6.1 USE CASE DIAGRAM

The Use Case Diagram represents user interactions with the system.

Use Case Diagram for EduVantage:

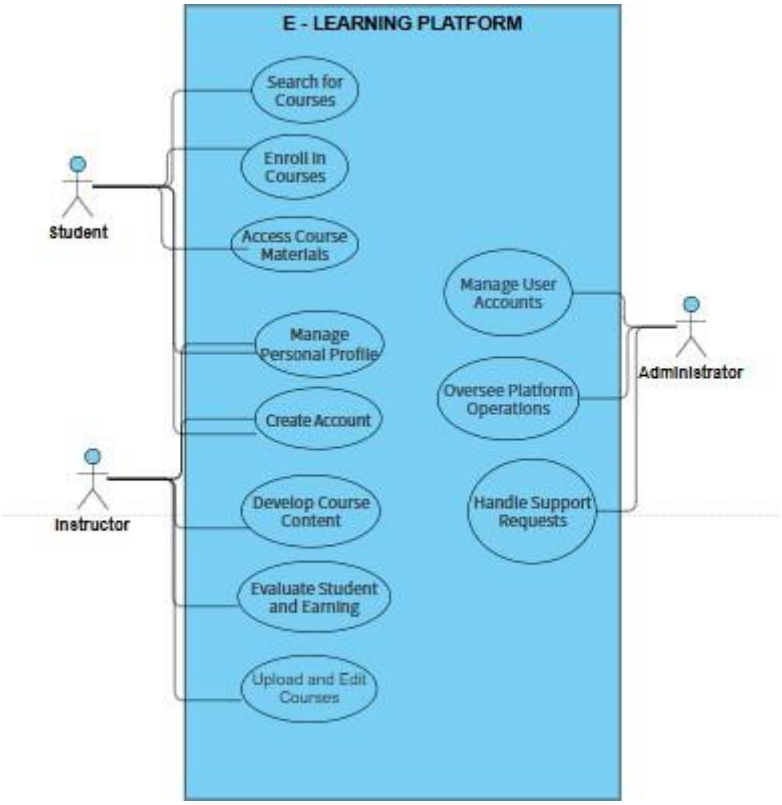

Figure 6.1.1: Use Case Diagram

6.2 ER DIAGRAM

Since MongoDB is a NoSQL database, it follows a document-oriented model, where data is stored in collections instead of relational tables.

Key Collections in the System:

1. **Users** – Stores Admin, Teacher, and Student details.
2. **Profiles** – Stores additional user details.
3. **Courses** – Contains course details, sections, and subsections.
4. **Categories** – Stores course categories.
5. **Course Progress** – Tracks student progress in enrolled courses.
6. **Rating and Reviews** – Stores student feedback for courses.
7. **Sections and Subsections** – Manages course content structure.

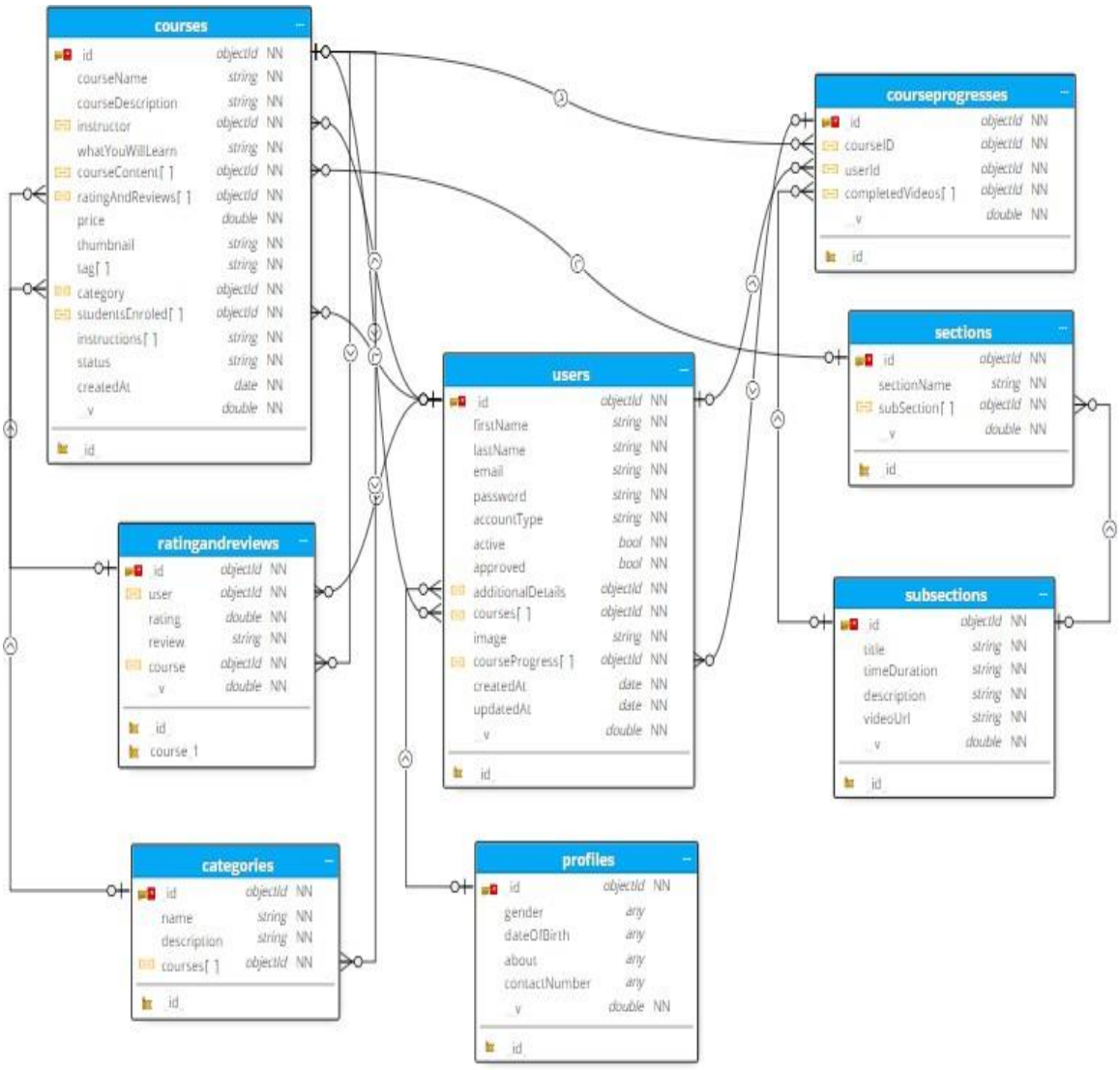

Figure 6.2.1: ER Diagram

6.3 DATA FLOW DIAGRAM

The Data Flow Diagram (DFD) represents the flow of information within the EduVantage system. It visually depicts how data moves between users, processes, and storage components. The DFD is structured into two levels:

- **Level 0:** High-level overview of the system.
- **Level 1:** Detailed flow of data between different entities.

1. Level 0 DFD

The Level 0 DFD shows the EduVantage system as a single entity interacting with Users (Admin, Teachers, Students).

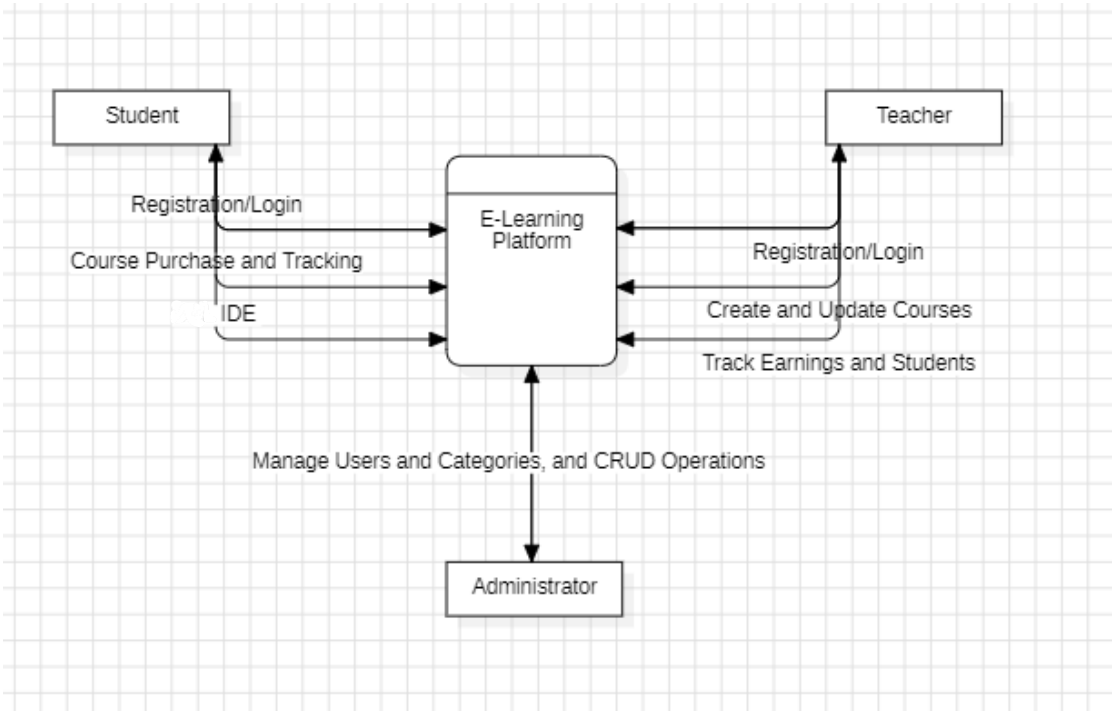

Figure 6.3.1: Level 0 Data Flow Diagram

2. Level 1 DFD

The Level 1 DFD breaks down the system into multiple processes, showing how data is transferred between different components.

Processes in Level 1 DFD:

- Users (Admin, Teachers, Students) register/login.
- Data is stored in the Users collection.

- Teachers create/update courses.
- Data is stored in Courses collections.
- Course media (videos) is uploaded to Cloudinary.
- Students browse and purchase courses.
- Payment details are processed via Razorpay.

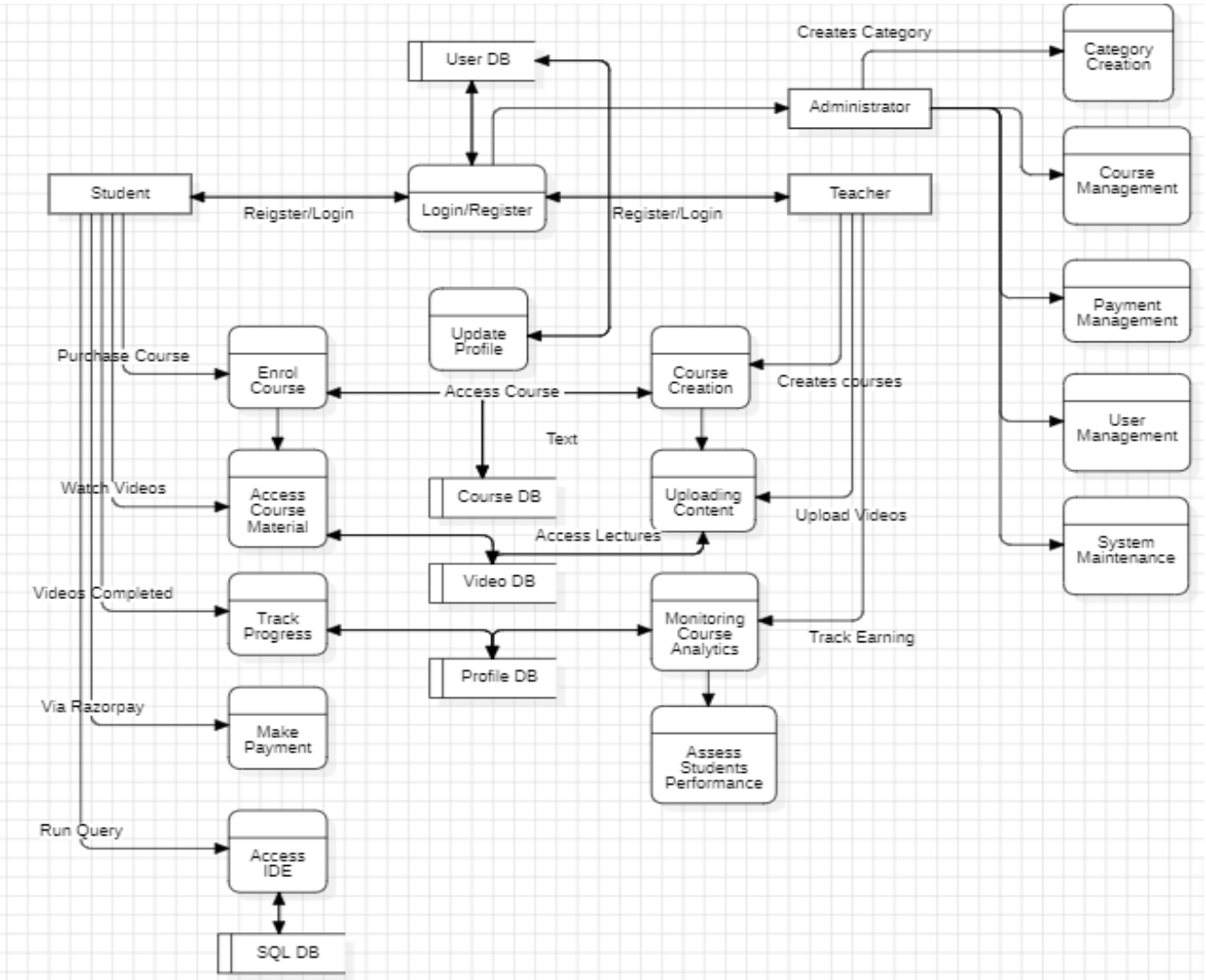

Figure 6.3.2: Level 1 Data Flow Diagram

CHAPTER 7
OUTPUT SCREENS

7.1 OUTPUT SCREENS

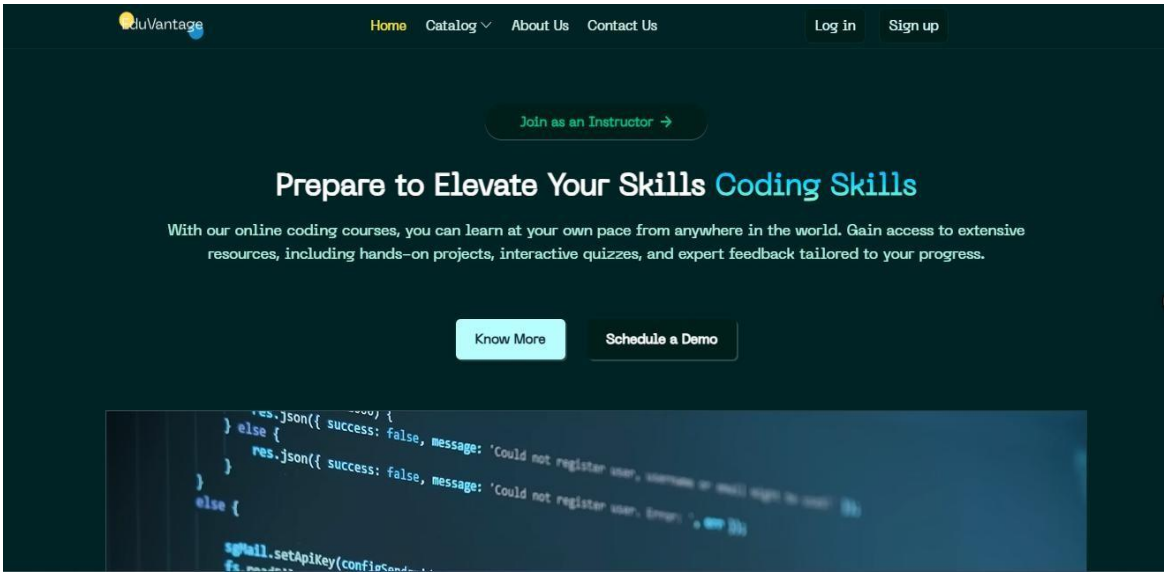

Figure 7.1.1: Home Page

Figure 7.1.2: About Us Page

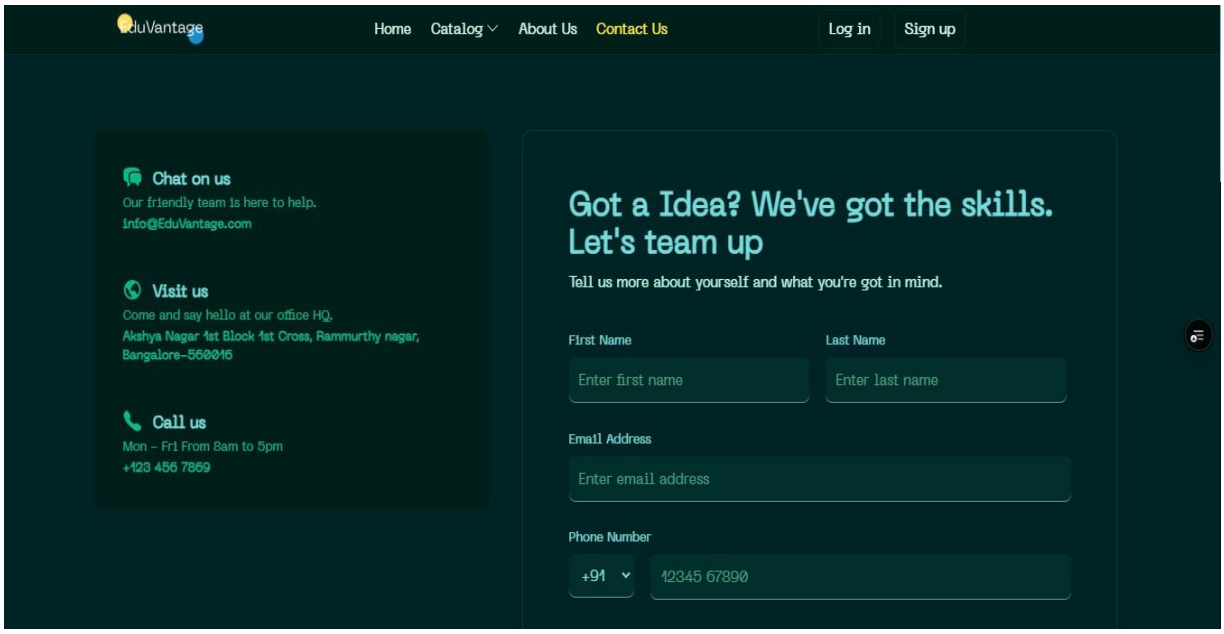

Figure 7.1.3: Contact Us Page

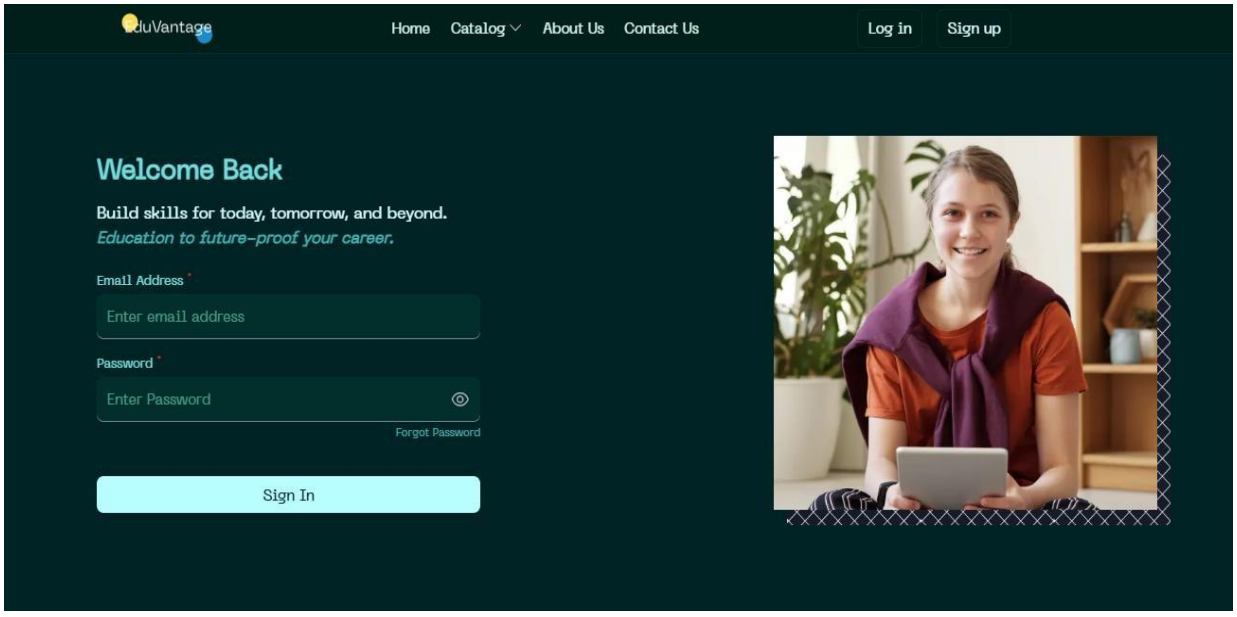

Figure 7.1.4: Login Page

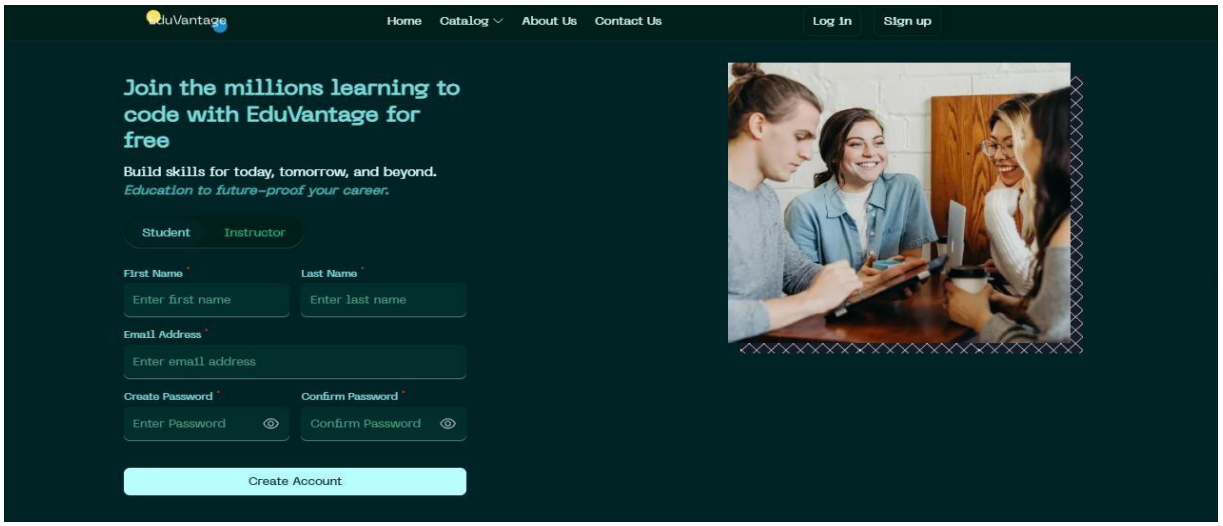

Figure 7.1.5: Sign Up Page

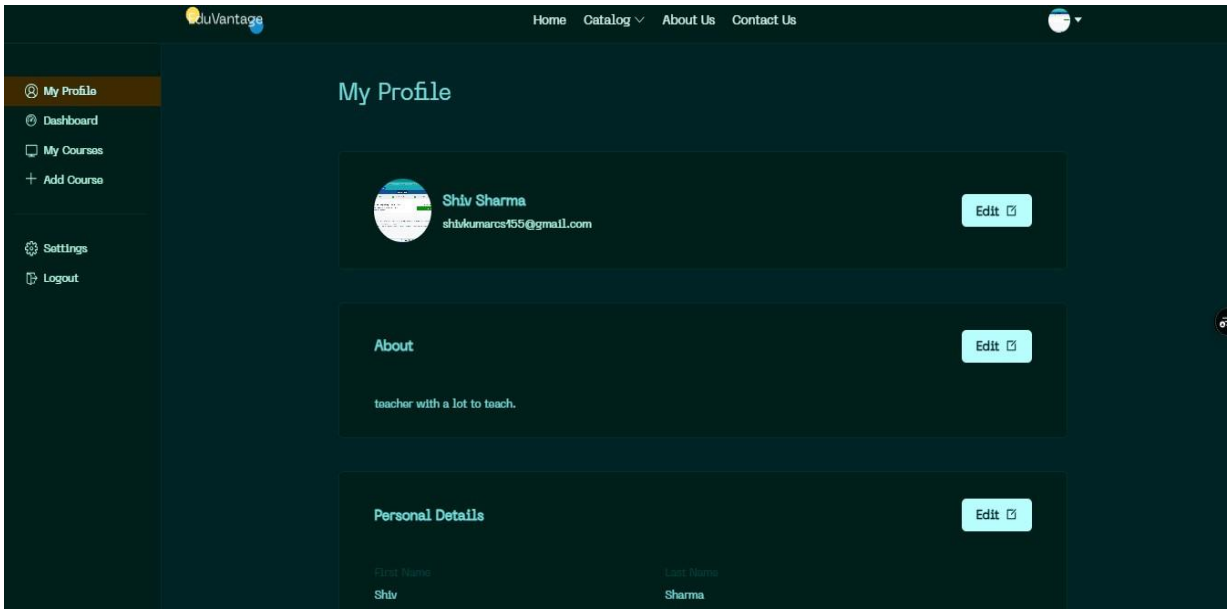

Figure 7.1.6: Teacher's Dashboard

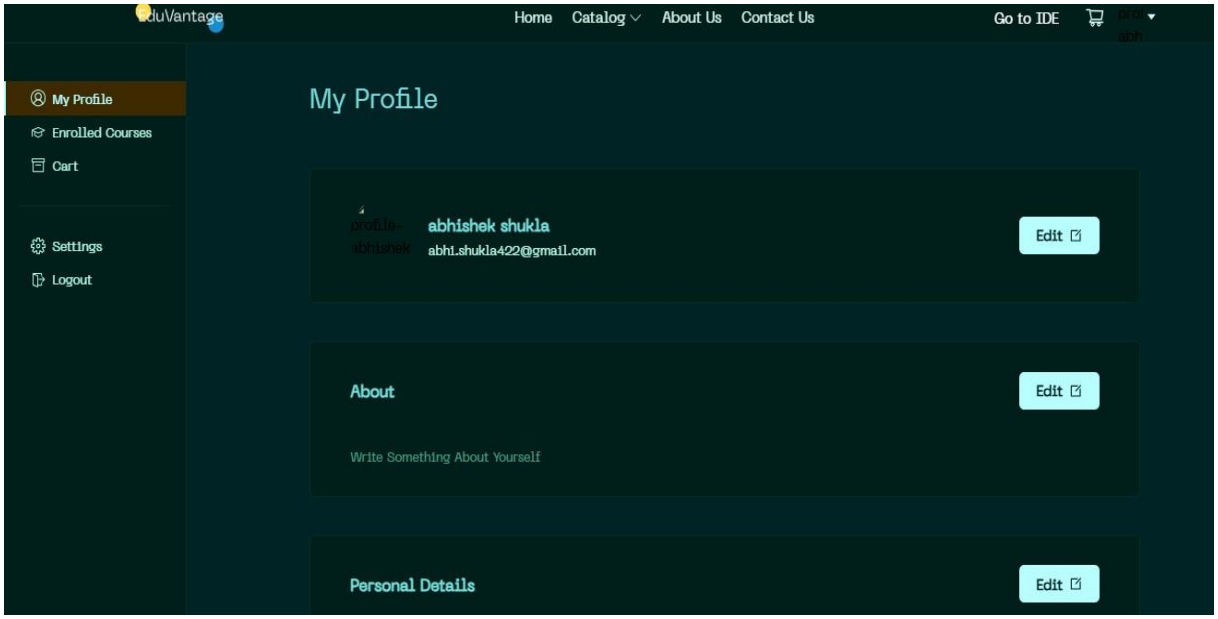

Figure 7.1.7: Student's Dashboard

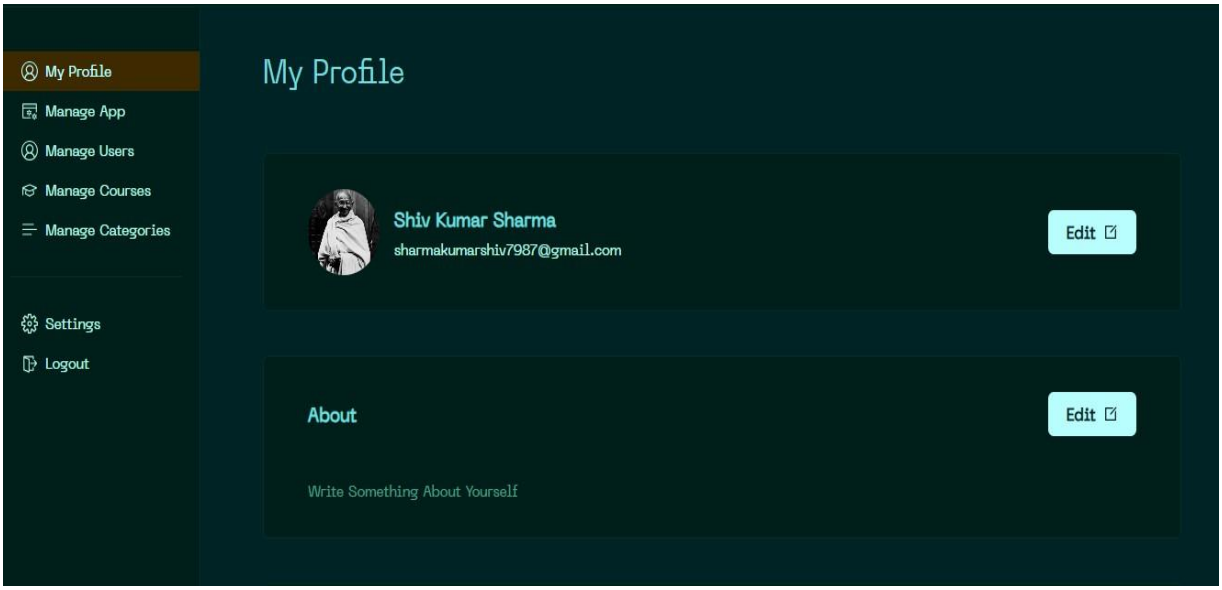

Figure 7.1.8: Admin's Dashboard

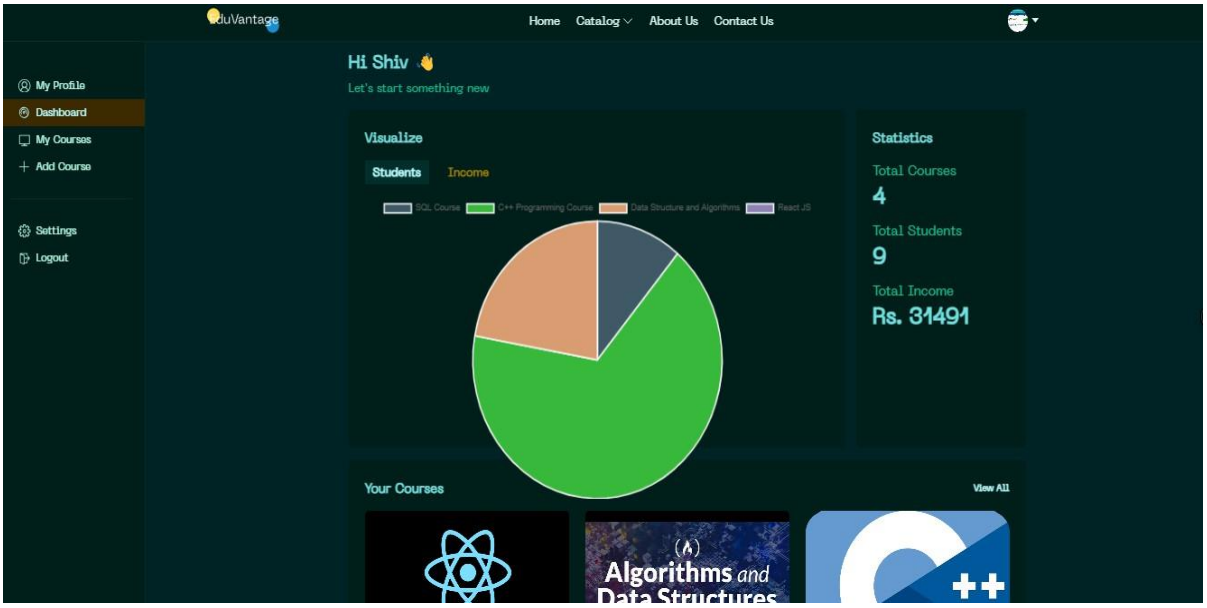

Figure 7.1.9: Teacher's Stats Page

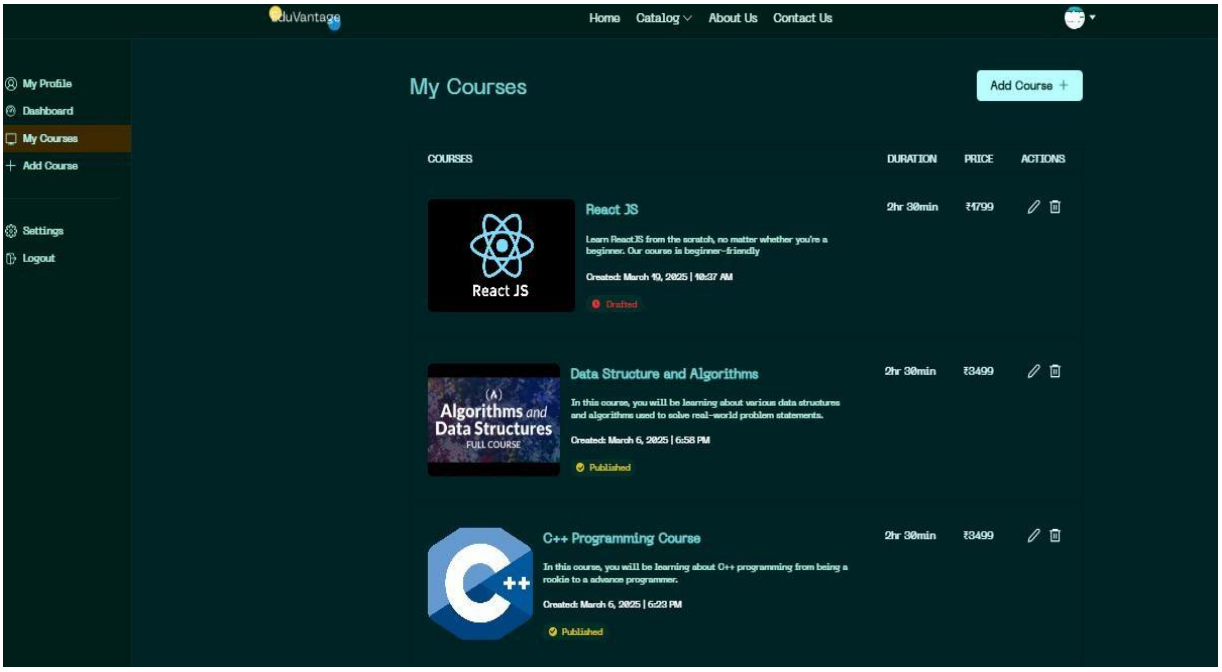

Figure 7.1.10: Teacher's My Course Page

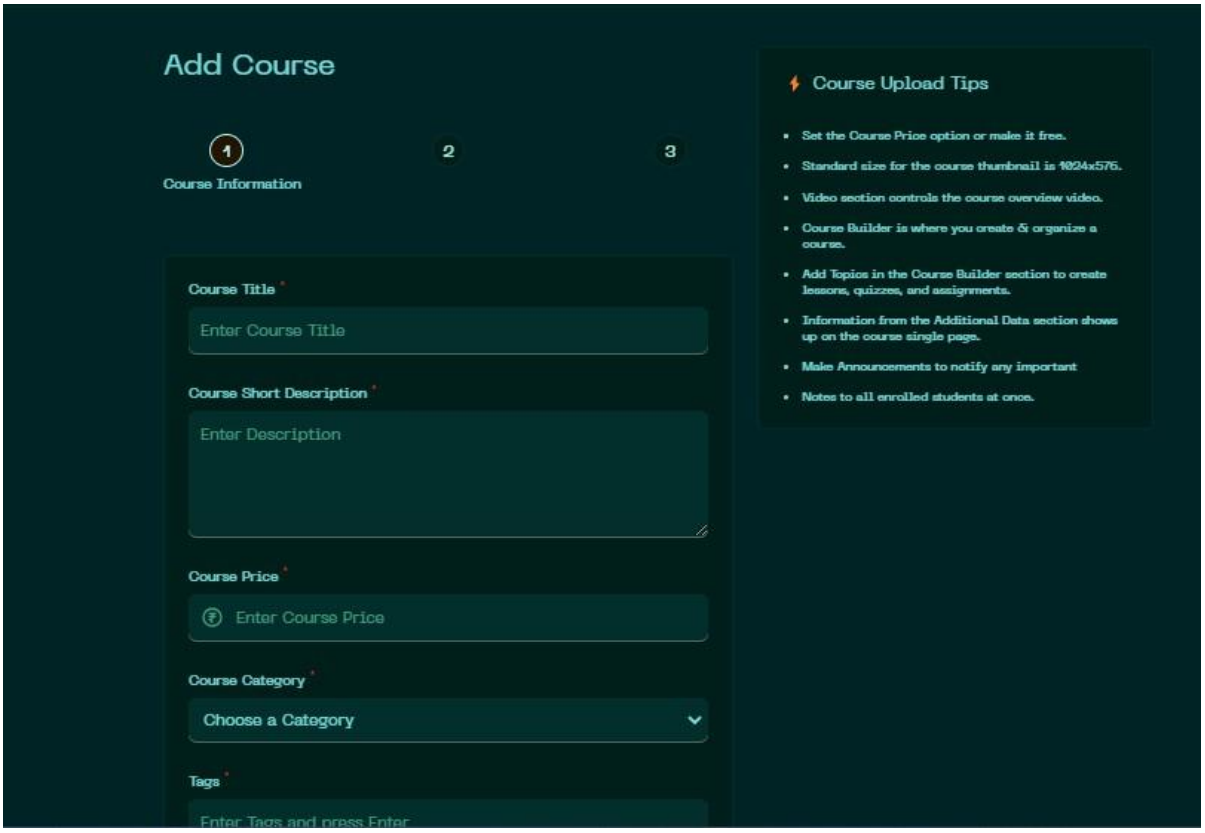

Figure 7.1.11: Add Course Page

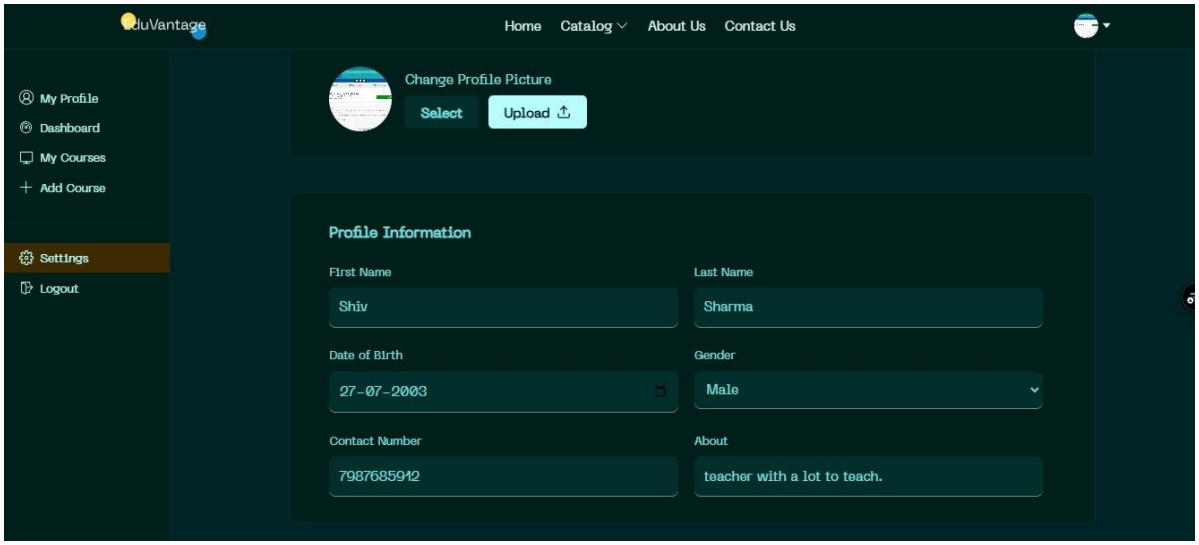

Figure 7.1.12: Update Profile Page

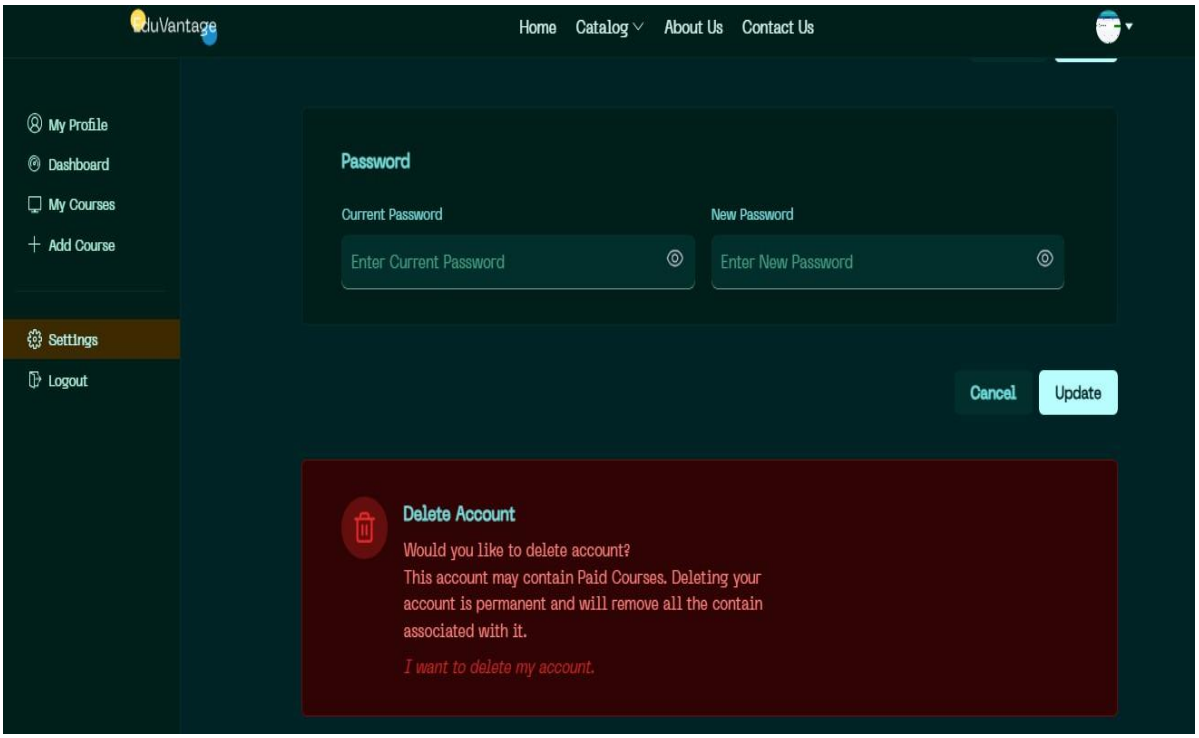

Figure 7.1.13: Update Password and Delete Account Page

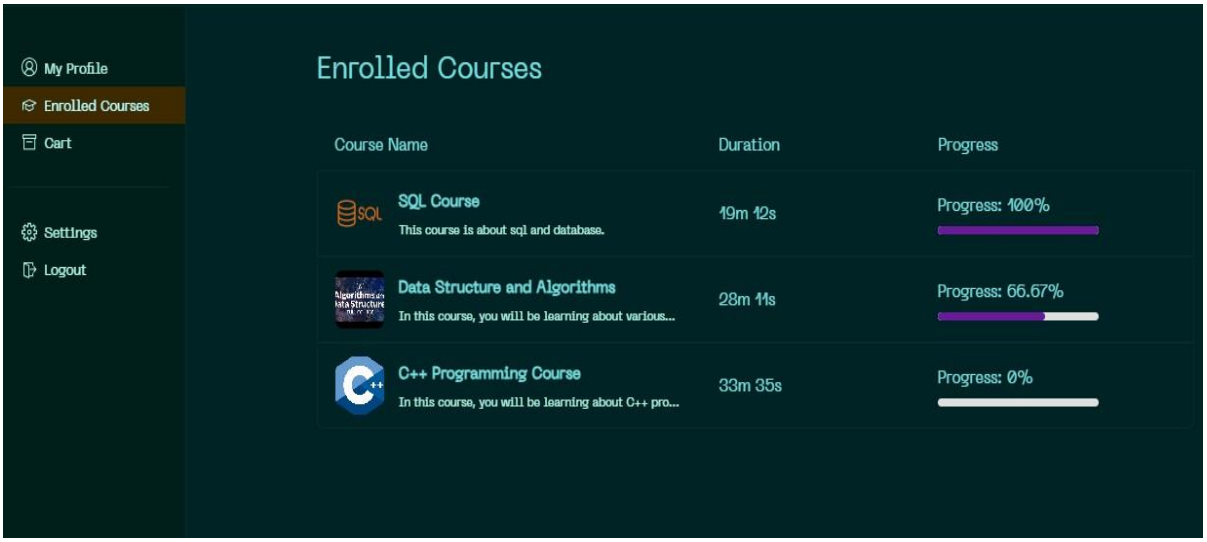

Figure 7.1.14: Enrolled Course Page

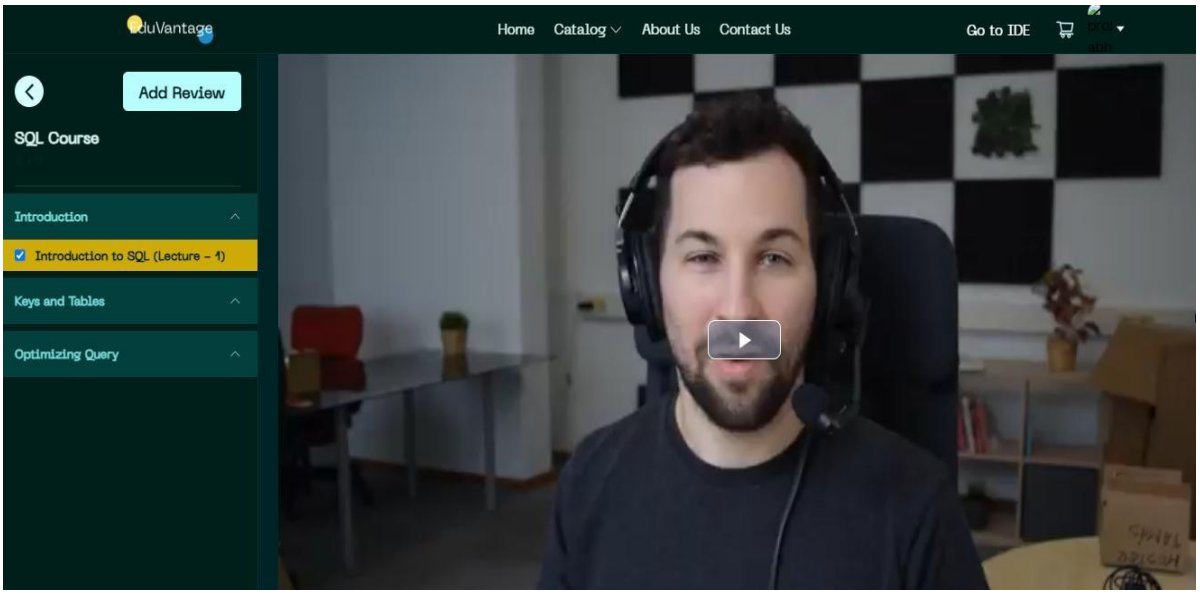

Figure 7.1.15: Video Lecture Page

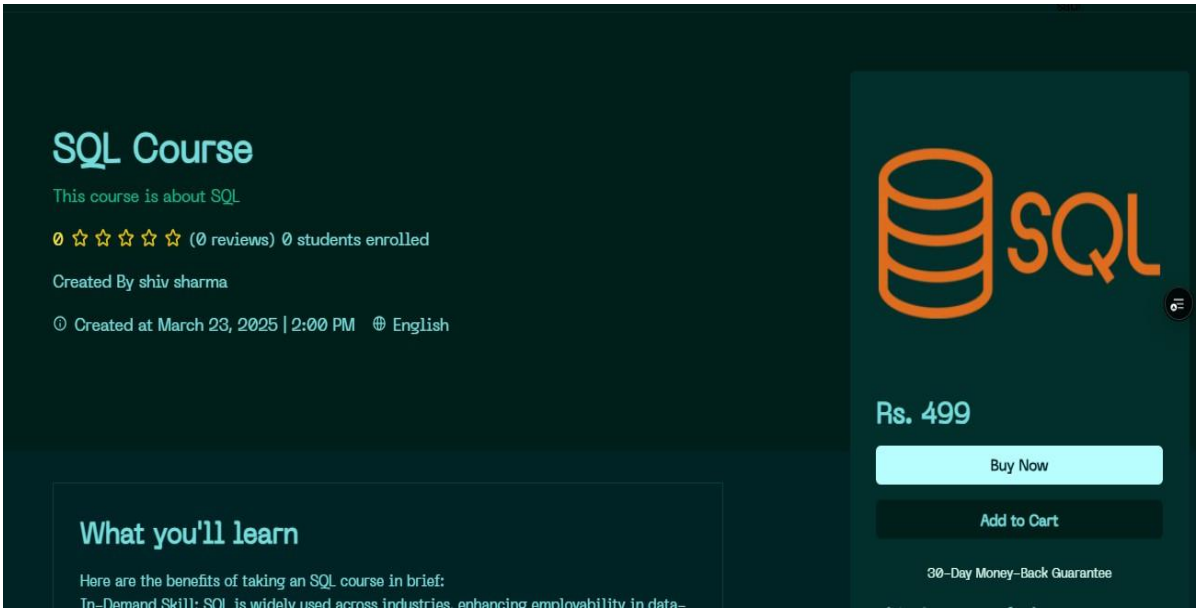

Figure 7.1.16: Buy Course Page

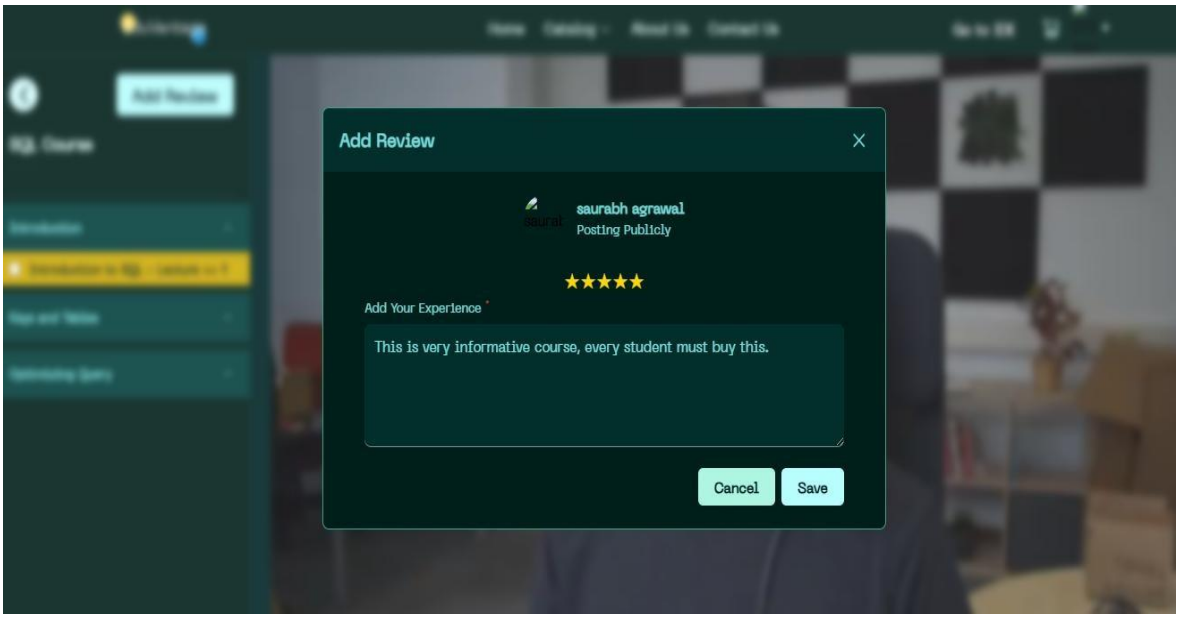

Figure 7.1.17: Give Review Page

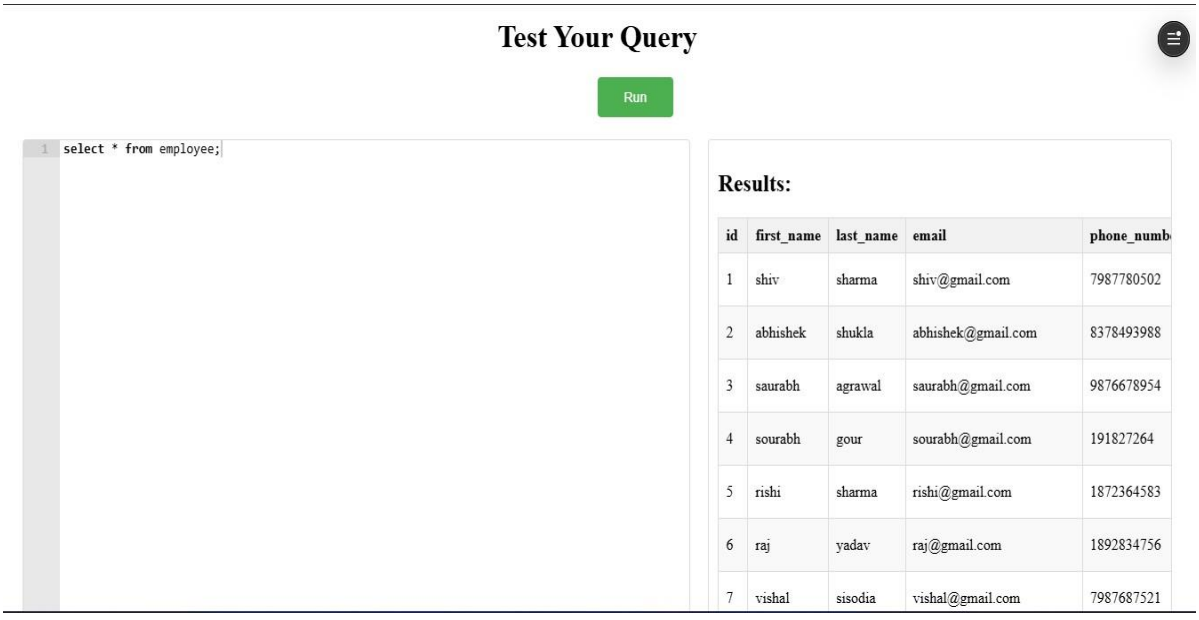

Figure 7.1.18: IDE Page

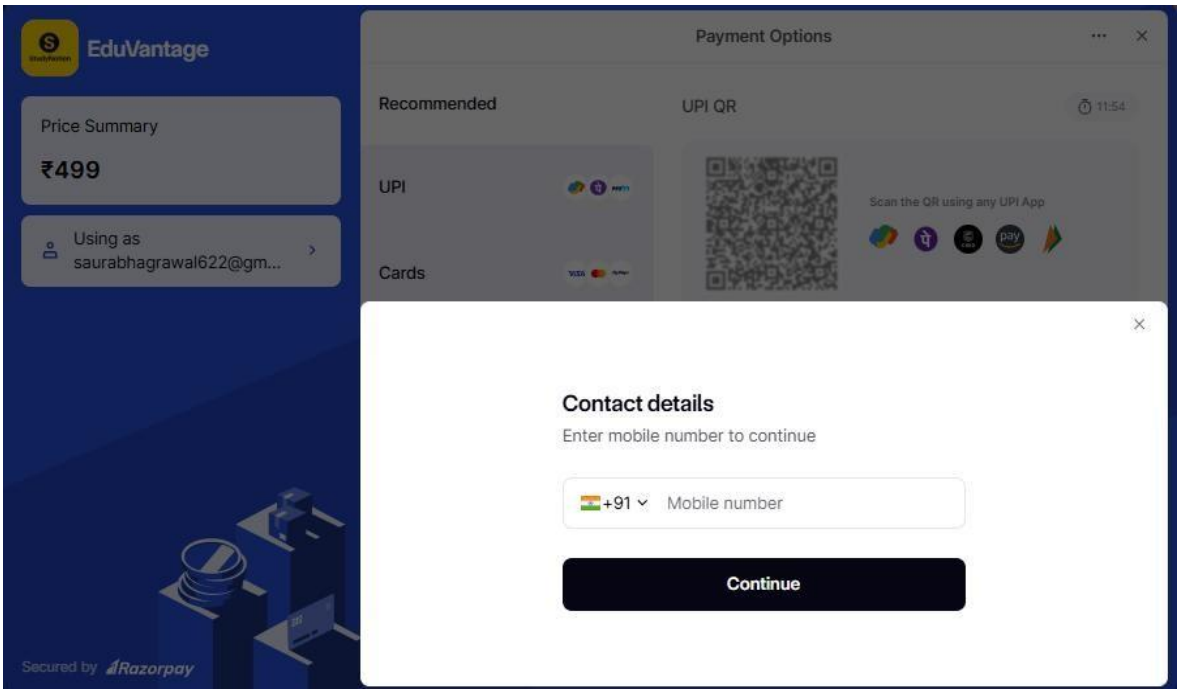

Figure 7.1.19: Razorpay Interface Page

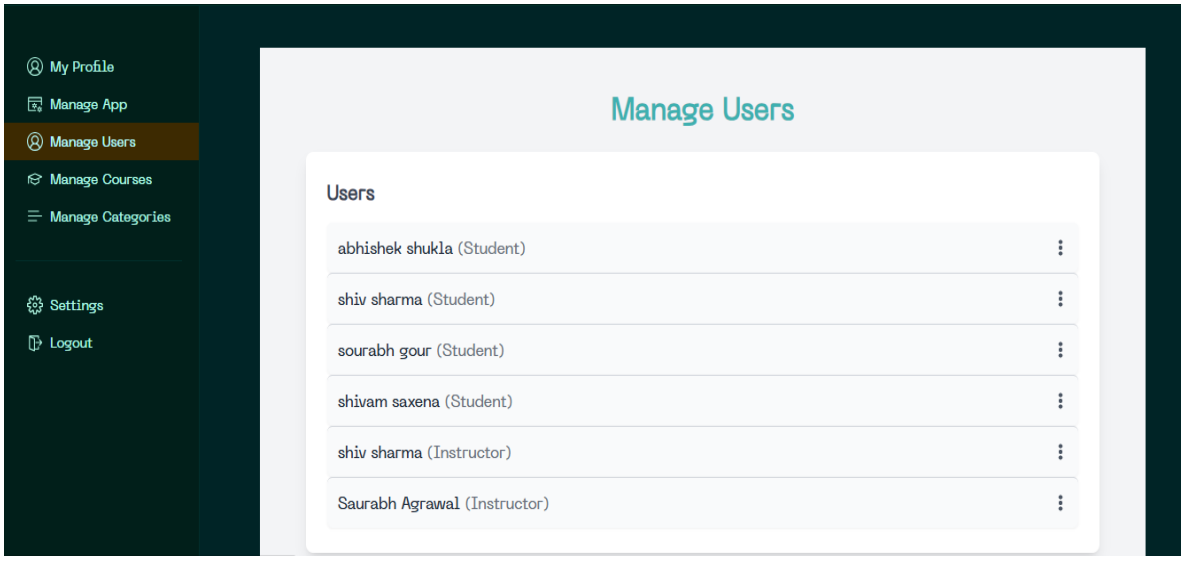

Figure 7.1.20: Manage Users Page

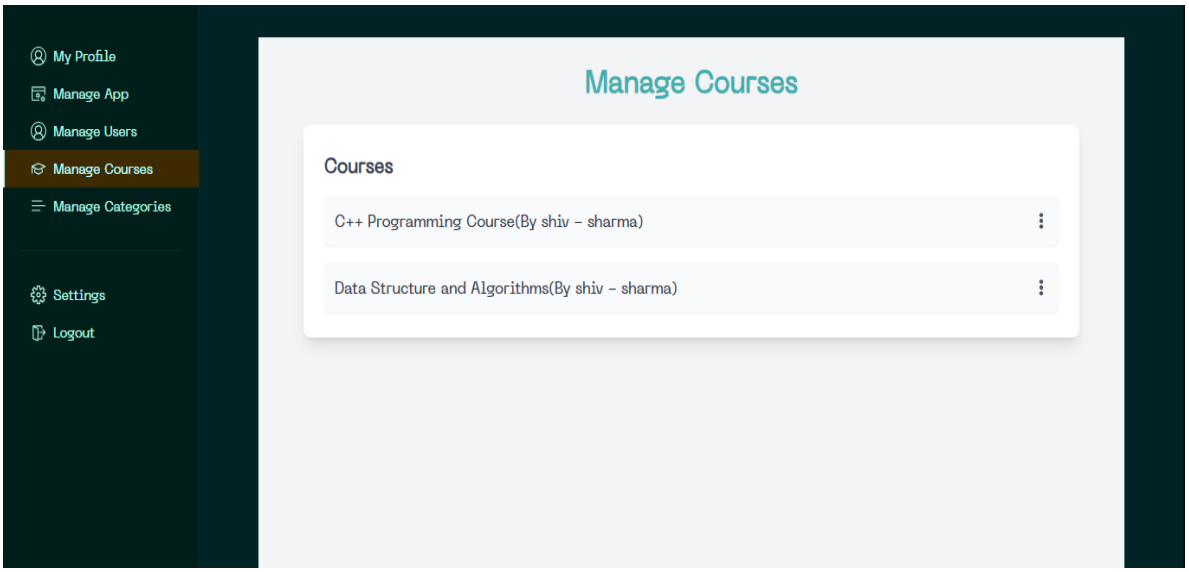

Figure 7.1.21: Manage Courses Page

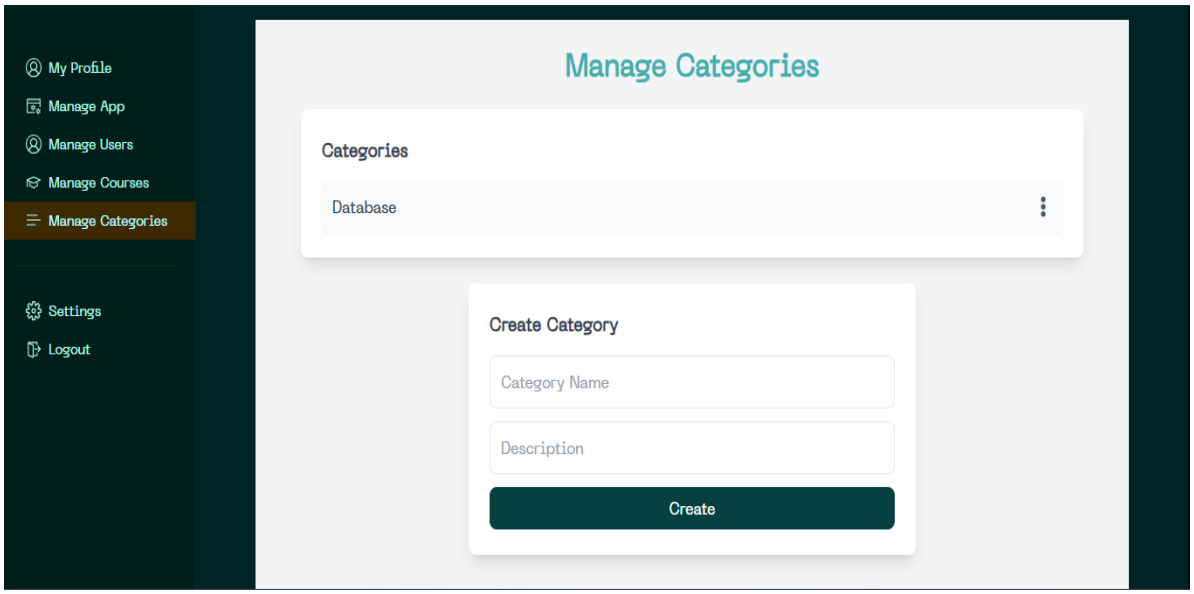

Figure 7.1.22: Manage Categories Page

CHAPTER 8
DEPLOYMENT

8.1 OVERVIEW

Deployment is the final phase of software development, where the EduVantage e-learning platform is made accessible to users over the internet

In this project, the backend is deployed on Render, a cloud platform that supports Django applications and provides automated deployment with scalability. The frontend is rendered dynamically using Django's templating engine (Jinja2) integrated with HTML, CSS, and Tailwind CSS to serve user-facing content efficiently. SQLite3 is used as the database during deployment due to its lightweight and file-based architecture, suitable for small to medium-scale applications.

Additionally, Cloudinary is used for handling media storage, allowing efficient management of course materials such as images and videos. The Razorpay payment gateway is integrated with Django views to process secure transactions for course purchases. This ensures reliability in online payments.

This chapter details the deployment process, server setup, environment variables configuration, and Git-based continuous deployment strategy to maintain a secure, scalable, and user-ready production environment.

8.2 DEPLOYMENT ARCHITECTURE

- **Frontend:** HTML, CSS, Jinja, Tailwind CSS (Hosted on Render)
- **Backend:** Django (Python-based, Hosted on Render)
- **Database:** SQLite3 (Lightweight, file-based relational database)
- **Cloud Storage:** Cloudinary (For media files like images and videos)
- **Payment Gateway:** Razorpay (For secure online transactions)

8.3 DEPLOYMENT OF FRONTEND

The frontend is built using Django templates (HTML, Jinja) and styled with Tailwind CSS. It is hosted along with the backend on Render. Render offers automatic deployments and static file handling using White Noise for Django projects.

Steps to Deploy Django Frontend on Render

- Push the entire Django project to GitHub.
- Create a new Web Service on Render.
- Connect the GitHub repository.
- Add Environment Variables (Razorpay keys, Cloudinary credentials).
- Set build and start commands (pip install -r requirements.txt, gunicorn projectname.wsgi). Once deployed, Render provides a public URL accessible for the web application.

8.4 DEPLOYMENT OF BACKEND

The backend uses Django and is deployed on Render alongside the frontend. Render supports Python applications with gunicorn for serving the Django app.

Steps to Deploy Django Backend on Render

- Push code to GitHub.
- Login to Render.com and create a Web Service.
- Select the repository and set the Python runtime.
- Configure necessary environment variables (Razorpay, Cloudinary keys).
- Define build and start commands.
- Enable Auto Deploy for continuous updates.

Once done, the backend is available via a public Render URL.

8.5 DATABASE SETUP

SQLite3 is used as the primary database for this project. It requires no additional setup on Render.

Steps for SQLite3 Setup

- Include the db.sqlite3 file in.gitignore for deployment.

- Ensure all migrations are completed using python manage.py migrate.
- Configure static and media files in settings.py. SQLite3 operates seamlessly with Django and works well for small to mid-size applications.

8.6 DOMAIN CONFIGURATION

For a professional interface, a custom domain can be pointed to the Render-deployed Django app.

Steps to Connect a Custom Domain

- Purchase a domain from providers like Namecheap or GoDaddy.
- Add the domain in Render's settings panel.
- Configure DNS records: set the CNAME or A record as instructed by Render. The site is now accessible via the custom domain name.

8.7 CONTINUOUS DEPLOYMENT

Render supports continuous deployment directly from GitHub for Django projects.

Steps to Enable Continuous Deployment

- Connect GitHub repository to Render.
- On each push to the main branch, a new deployment is triggered.
- Monitor build logs and status on Render dashboard. This ensures all changes are deployed automatically without manual steps.

REFERENCE

JOURNAL / RESEARCH PAPERS

1. Aljawarneh, S. (2020). Reviewing and exploring innovative ubiquitous learning tools in higher education. Education and Information Technologies.
2. Pappano, L. (2012). The Year of the MOOC. The New York Times.
3. Kumar, A., & Tiwari, R. (2022). MERN Stack Development for Scalable Web Applications. International Journal of Computer Science.
4. Liu, C., Tang, M., & Zhu, Q. (2019). Cloud storage solutions for educational institutions. Journal of Cloud Computing.
5. Chatterjee, S., & Sood, M. (2020). Secure Payment Gateway Integration in Online Learning Platforms. International Journal of Information Security.
6. Aggarwal, A. (2018). Database Learning and Practical SQL Application. Journal of Computing Education.
7. Reich, J. (2020). MOOC Completion Rates: Why Do Students Drop Out? HarvardX Research.
8. Singh, P., & Sharma, R. (2019). Cybersecurity Threats in E-learning Platforms. Journal of Digital Security.

WEBSITES

9. https://hyperspace.mv/ai-powered-learning-paths/
10. https://zspace.com/
11. https://docs.mongodb.com/manual
12. https://expressjs.com/
13. https://react.dev/
14. https://developer.mozilla.org/

PROJECT SUMMARY

About Project

Title of the project	E-Learning Platform
Semester	8th
Members	4
Team Leader	Ritik Sharma
Describe role of every member in the project	Rohit Choursiya: Work on Software Api Testing. Satyam Sen: Work on Backend Development. Ritik Sharma: Work on Backend Development and Database Management. Shubh Dengre: Work on Frontend Development.
What is the motivation for selecting this project?	The motivation behind EduVantage is to bridge this gap by providing a cost-effective, user-friendly, and feature-rich e-learning solution.
Project Type (Desktop Application, Web Application, Mobile App, Web)	Web Application

Tools & Technologies

Programming language used	Python
IDE used (with version)	Visual Studio Code v1.98
Front End Technologies (With version, wherever Applicable)	HTML5 CSS3 Jinja2 Template Engine Tailwind CSS v3.4.1
Back End Technologies (With version, wherever applicable)	Django v5.0.2
Database used (with version)	SQLite3 v3.45.1

Software/Hardware Design & Coding

Is prototype of the software/hardware developed?	Yes
SDLC model followed (Waterfall, Agile, Spiral etc.)	Agile
Why above SDLC model is followed?	Agile is an SDLC model that defines how software development should be carried out through an iterative and collaborative approach. It is not a single method but a collection of best practices and methodologies that emphasize customer collaboration, continuous delivery, and adaptability to change
Justify that the SDLC model mentioned above is followed in the project.	We used Agile model, so that we could make desired changes whenever needed.
Software Design approach followed (Functional or Object Oriented)	Object Oriented
Name the diagrams developed (According to the Design approach followed)	Use Case Diagram, Data Flow Diagram, ER Diagram
In case Object Oriented approach is followed, which of the OOPS principles are covered in design?	The project follows OOP concepts for better structure and maintainability. Encapsulation secures data using Mongoose models, while inheritance allows Student and Teacher to extend the User class. Polymorphism enables role-based behaviors, like different dashboards for users. Abstraction hides complex logic (e.g., payments), ensuring a clean and scalable codebase.
No. of Tiers (Example 3-tier)	3-Tier
Total no. of Frontend pages	13
Front end validations applied (Yes / No)	Yes
Session management done (in case of web applications)	Yes
Is application browser compatible (In case of web applications)	Yes

What difficulties faced during deployment of project?	During development, we faced challenges like handling real- time payments securely with Razorpay, ensuring scalable cloud storage with Cloudinary, and managing efficient database structuring in MongoDB. Debugging cross-origin issues (CORS) between frontend and backend and implementing role-based authentication were also complex tasks. However, these challenges were overcome through proper API integration, middleware handling, and database optimization.
Total no. of Use-case	8
Give titles of Use-cases	User Registration & Login, Course Creation by Teacher, Course Purchase by Student, Course Category Management by Admin, Course Progress Tracking, Payment Processing via Razorpay, Review & Rating System, Profile Management (User, Teacher, Admin)

Project Requirements

MVC architecture followed (Yes / No)	Yes
If yes, write the name of MVC architecture followed (MVC-1, MVC-2)	MVC-2
Design Pattern used (Yes / No)	Yes
If yes, write the name of Design Pattern used	MVC Pattern
Interface type (CLI / GUI)	GUI
No. of Actors	3
Name of Actors	Student, Teacher, Administrator
List few important non-Functional Requirements	Scalability, Security, Performance, Availability

Testing

Which testing is performed? (Manual or Automation)	Manual
Is Beta testing done for this project?	No

Write project narrative covering above mentioned points

EduVantage is an E-learning platform designed to provide a seamless online education experience for students and teachers. The platform enables teachers to create and manage courses, while students can browse, purchase, and complete them at their own pace. An admin panel is available for managing courses, users, and categories. Built using the MERN stack, the platform integrates Razorpay for payments, Cloudinary for media storage, and an inbuilt SQL IDE for practical learning.

Our team is organized as follows:

- **Ritik Sharma** and **Satyam Sen** are responsible for Backend Development and Database Management
- **Shubh Dengre** and **Rohit Choursiya** are responsible for Software API Testing and Frontend Development.

The project follows an **Agile methodology** to enable iterative development, ensuring flexibility and continuous feedback during the build process. EduVantage follows the MVC2 (Model-View-Controller 2) architecture, ensuring separation of concerns

We have two main user roles:

- **Admin:** Perform CRUD operations on users, courses, and categories.
- **Students:** Browse, purchase, and track course progress.
- **Teachers:** Upload and manage courses, track student engagement.

The aim of EduVantage is to provide a scalable and interactive e-learning platform where teachers can upload courses, students can purchase and learn at their own pace, and admins can efficiently manage the system. It ensures secure payments, real-time progress tracking, and seamless cloud storage, enhancing the digital education experience.

Shubh Dengre	0187CS211162
Satyam Sen	0187CS211152
Ritik Sharma	0187CS211136
Rohit Choursiya	0187CS211138

Guide Signature
(Prof. Amit Swami)

APPENDIX-1 GLOSSARY OF TERMS

A

Admin — A user with the highest privileges who manages courses, categories, and user accounts.

API (Application Programming Interface) — A set of rules that allow the frontend and backend to communicate.

Authentication — The process of verifying user identity using JWT (JSON Web Token).

C

Cloudinary — A cloud-based service used for storing and managing course-related media files.

Course — A structured set of learning materials uploaded by teachers for students.

CRUD Operations — Create, Read, Update, and Delete functionalities, used for managing courses, users, and categories.

D

Database — SQLite3, a lightweight, file-based relational database used to store users, courses, transactions, etc.

Deployment — The process of hosting the Django backend and frontend using platforms like Render, Railway, or Heroku for public access

F

Frontend — The user interface developed using HTML, CSS, Jinja templating, and styled with Tailwind CSS.

I

IDE (Integrated Development Environment) — Software like Visual Studio Code used for developing, editing, and debugging code.

M

Middleware — Functions in Django that process HTTP requests and responses before reaching views.

MVC2 (Model-View-Controller 2) — A design pattern separating Model – Django Models (business logic, data structure) View – Python functions/classes (handle logic) Template – Jinja2 (HTML frontend)

P

Payment Gateway — Razorpay, used for secure online transactions between students and the platform.

R

Razorpay — A secure payment gateway integrated for processing student course purchases.

RESTful API — Django views or Django REST Framework endpoints for communication between frontend and backend using HTTP methods.

Role-Based Access Control (RBAC) — Access permissions set using Django's Group and Permission system, controlling views and functionalities for students, teachers, and admins.

S

Session Management — Managed by Django's built-in session framework, storing session data in cookies or the database.

SQL (Structured Query Language) — Used to interact with SQLite3 database via Django ORM or raw SQL.

State Management — Handled server-side with session variables or context passed in views/templates.

Student — A user role that registers, purchases, and accesses courses.

T

Teacher — A user who creates and manages courses for students.

Testing — The process of debugging, unit testing, and manual testing to ensure system functionality.

U

UI/UX The platform is designed for a clean and intuitive learning experience using Jinja templates and Tailwind CSS.

V

Vercel Can be used to host the static frontend (if separated), while Django backend is deployed elsewhere like Render or Railway.

W

Webhooks Razorpay webhooks used to track payment status updates.

WebSocket Not used in this project, but Django Channels can be integrated in future for real-time features like progress tracking or messaging.