# CHAPTER 1

# INTRODUCTION

## 1.1 ABOUT PROJECT

EduVantage is a web-based E-learning platform designed to provide a seamless remote learning experience for students while offering an intuitive course management system for teachers and administrators. The platform facilitates the creation, purchase, and management of online courses, enabling educators to share knowledge efficiently and students to access learning materials anytime, anywhere.

Built using the MERN stack (MongoDB, Express.js, React.js, Node.js), EduVantage ensures a scalable and dynamic environment for online education. The platform incorporates several key functionalities, including course creation and purchase, payment processing via Razorpay, cloud-based content storage using Cloudinary, and an inbuilt IDE for interactive database learning.

EduVantage is designed to address the limitations of traditional E-learning platforms by integrating a practical coding environment alongside course content. The IDE feature allows students to execute database queries in real-time, bridging the gap between theoretical learning and hands-on practice. Additionally, an admin panel is provided to manage courses, users, and categories, ensuring smooth system operation.

By leveraging modern web technologies, secure payment gateways, and efficient cloud storage, EduVantage aims to enhance accessibility, flexibility, and engagement in digital learning. The platform not only simplifies the course delivery process for educators but also enriches the learning experience for students through interactive features and a well-organized content structure. Furthermore, the platform ensures a secure and seamless user experience by implementing authentication mechanisms and data encryption.

## 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. **Facilitate Remote Learning:** Enable students to access high-quality educational content anytime and from anywhere, promoting flexibility in learning.
2. **Simplify Course Management:** Provide teachers with an intuitive interface to create, upload, and manage courses effortlessly.
3. **Seamless Course Purchase & Payment Processing:** Integrate Razorpay for secure and efficient course transactions, ensuring smooth financial operations.
4. **Interactive SQL Learning Environment:** Incorporate an inbuilt IDE to allow students to write and execute SQL queries in real-time, bridging the gap between theory and practice.
5. **Efficient Content Storage & Delivery:** Utilize Cloudinary for cloud-based storage, enabling fast and secure access to course materials, including videos and documents.
6. **Role-Based Access & Administration:** Implement admin, teacher, and student roles with controlled access to ensure smooth platform management.
7. **Secure & Scalable Architecture:** Develop a robust system using the MERN stack, ensuring high scalability, security, and performance.
8. **User-Friendly Interface:** Design an intuitive and responsive UI for easy navigation and interaction across all devices.
9. **Improve Engagement & Learning Outcomes:** Enhance the digital education experience by integrating interactive and practical learning components.
10. **Ensure Data Security & Privacy:** Implement authentication and encryption mechanisms to protect user data and transactions.

## 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 & Enrol 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:** React.js (JavaScript Library for UI Development)
- **Backend**: Node.js with Express.js (Server-side Framework)
- **Database**: MongoDB (NoSQL Database for storing user, course, and transaction data)
- **Operating System:** Windows 10/11, macOS, or Linux
- **Internet Connection:** Stable broadband connection for real-time testing & cloud integration

### 2.2.2 TESTING TOOLS

- **Postman:** API testing and debugging
- **Jest & Mocha:** Unit testing for JavaScript (Backend & Frontend)

### 2.2.3 DEPLOYMENT & HOSTING SERVICES

- **Web Hosting:** Vercel / Render / Heroku
- **Database Hosting:** MongoDB Atlas (Cloud database service)
- **Domain Management:** Namecheap / GoDaddy
- **Version Control:** Git & GitHub for 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

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.
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. **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.
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.
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.
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:

1. **Remote Learning Accessibility:** Students can access courses anytime, anywhere, removing geographical barriers.
2. **Effortless Course Management:** Teachers can easily upload, structure, and update courses with a user-friendly interface.
3. **Secure Payment Gateway:** Integrated Razorpay ensures safe and seamless transactions for course purchases.

4. **Hands-on SQL Practice:** An inbuilt IDE allows students to write and execute queries in real-time.
5. **Cloud Storage for Content:** Cloudinary is used for efficient and secure storage of course materials.
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:

**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.

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.

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. **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

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.

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.

3. **Scalability & Performance Optimization**

- Platforms must handle thousands of concurrent users without slowing down [9].
- 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

1. Secure payment processing via Razorpay API.
2. Automated invoice generation after payment.

### 5.1.3 COURSE MANAGEMENT

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

### 5.1.4 IDE FUNCTIONALITY

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

### 5.1.5 AUTHENTICATION & SECURITY

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

## 5.2 NON-FUNCTIONAL REQUIREMENTS

### 5.2.1 PERFORMANCE REQUIREMENTS

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

### 5.2.2 SECURITY REQUIREMENTS

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

### 5.2.3 USABILITY REQUIREMENTS

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

### 5.2.4 SCALABILITY REQUIREMENTS

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

### 5.2.5 AVAILABILITY & RELIABILITY

1. 99.9% uptime guarantee with cloud-based deployment.
2. Automatic failover mechanisms to prevent downtime.

### 5.2.6 MAINTAINABILITY & EXTENSIBILITY

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

### 5.2.7 COMPLIANCE REQUIREMENTS

1. Follows GDPR for data protection and privacy.
2. Payment system complies with PCI-DSS security standards.
3. 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:

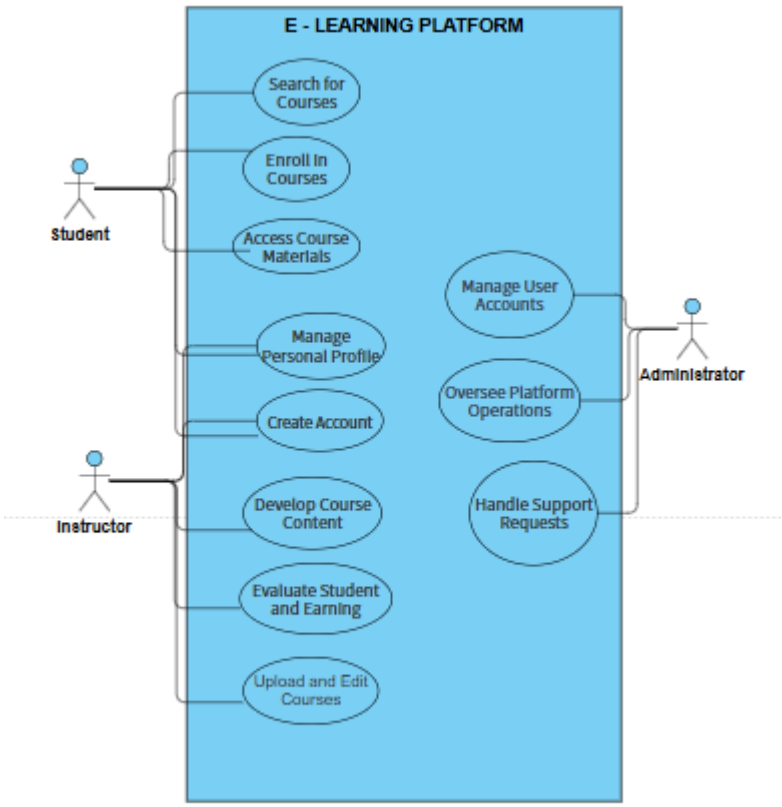


**Figure 6.1.1: Use Case Diagram**

## 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.

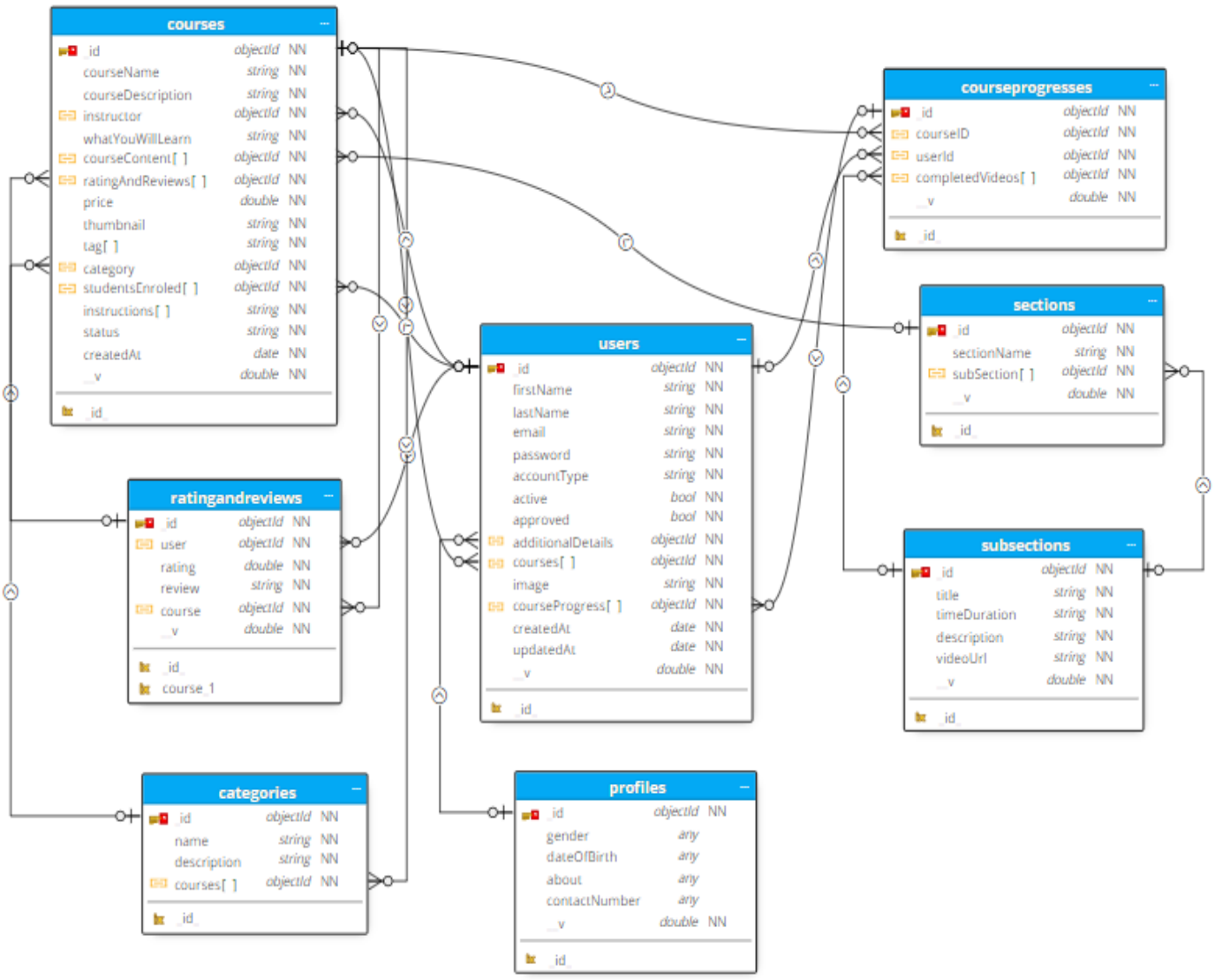


**Figure 6.2.1: ER Diagram**

## 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).

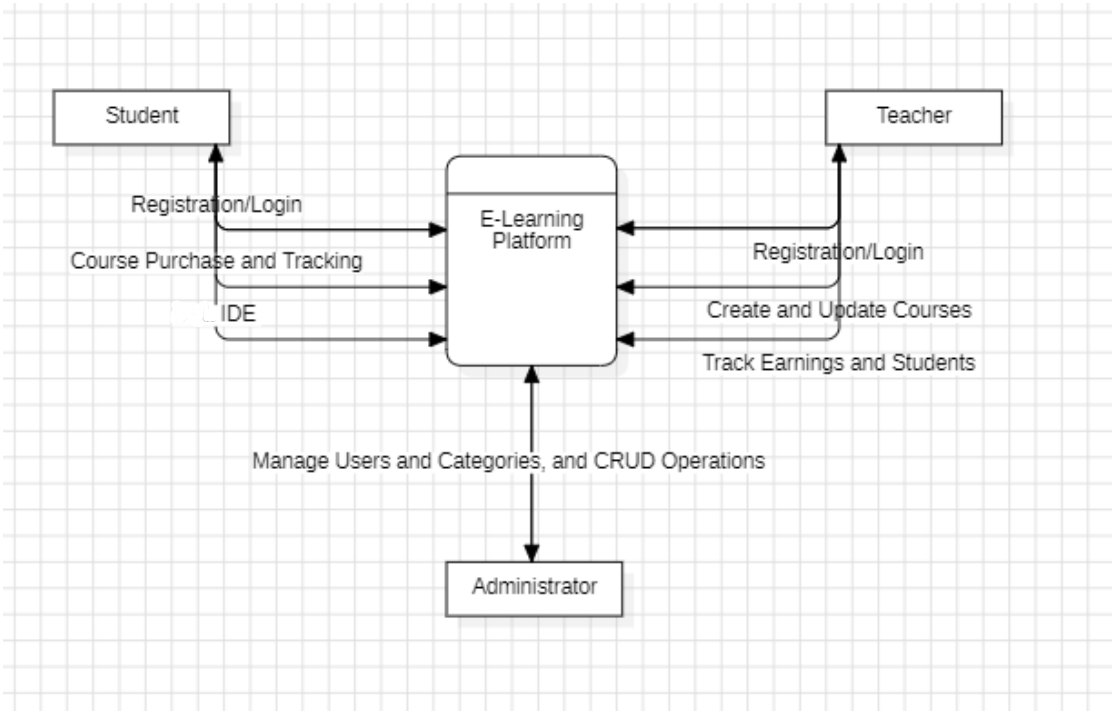


**Figure 6.3.1: Level 0 Data Flow Diagram**

### 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.

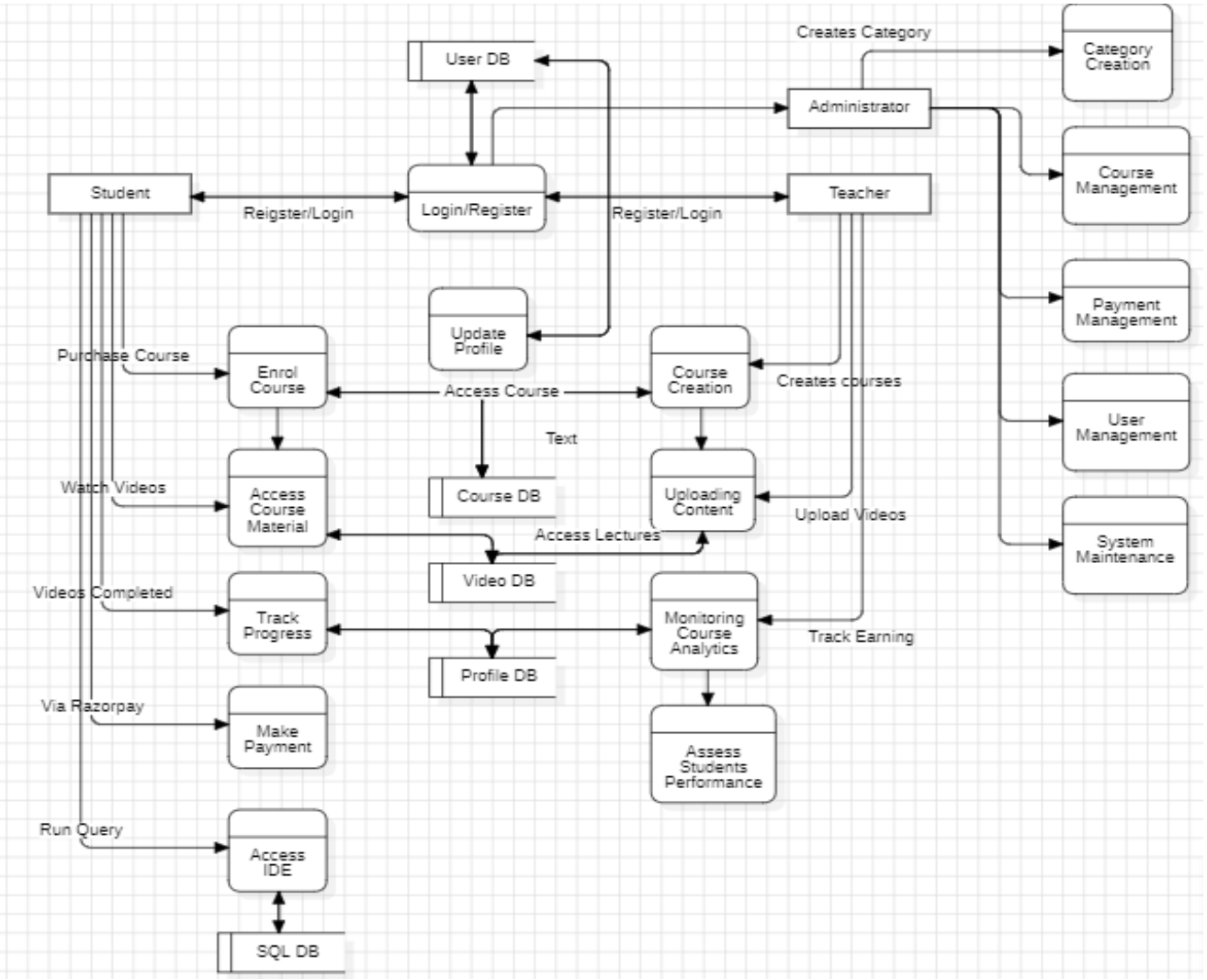


**Figure 6.3.2: Level 1 Data Flow Diagram**

# CHAPTER 7

# OUTPUT SCREENS

## 7.1 OUTPUT SCREENS

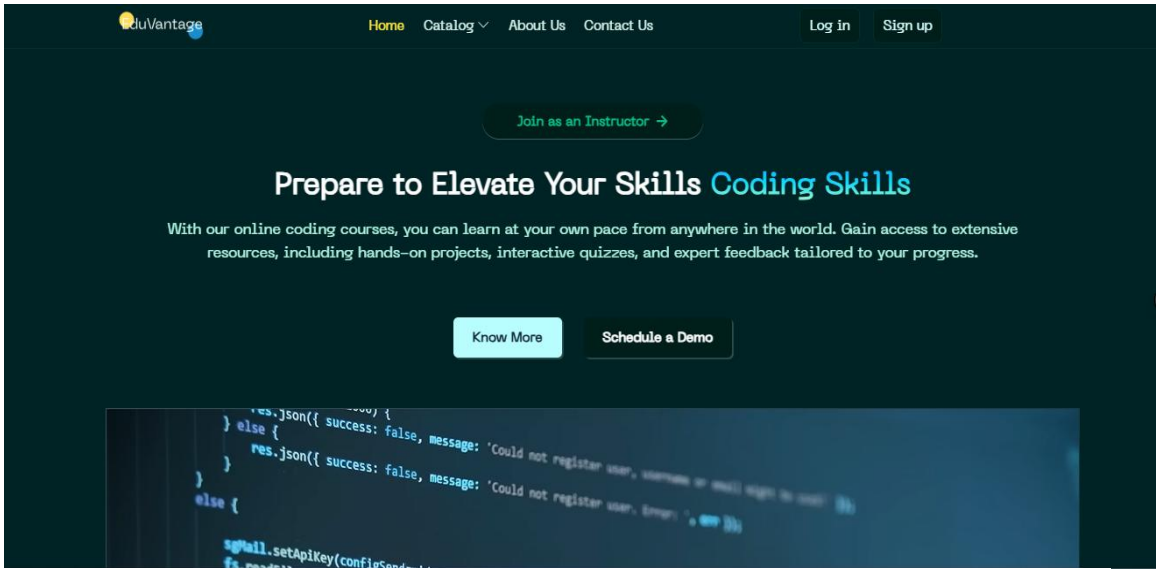


**Figure 7.1.1 Home Page**


**Figure 7.1.2 About Us Page**

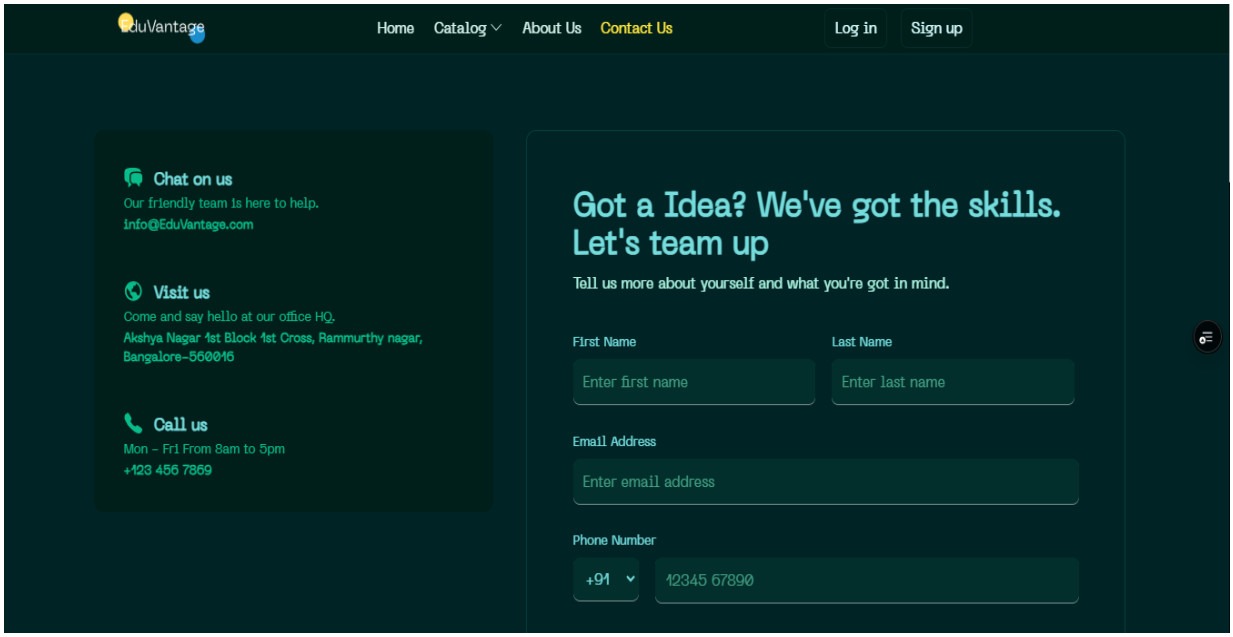


**Figure 7.1.3 Contact Us Page**

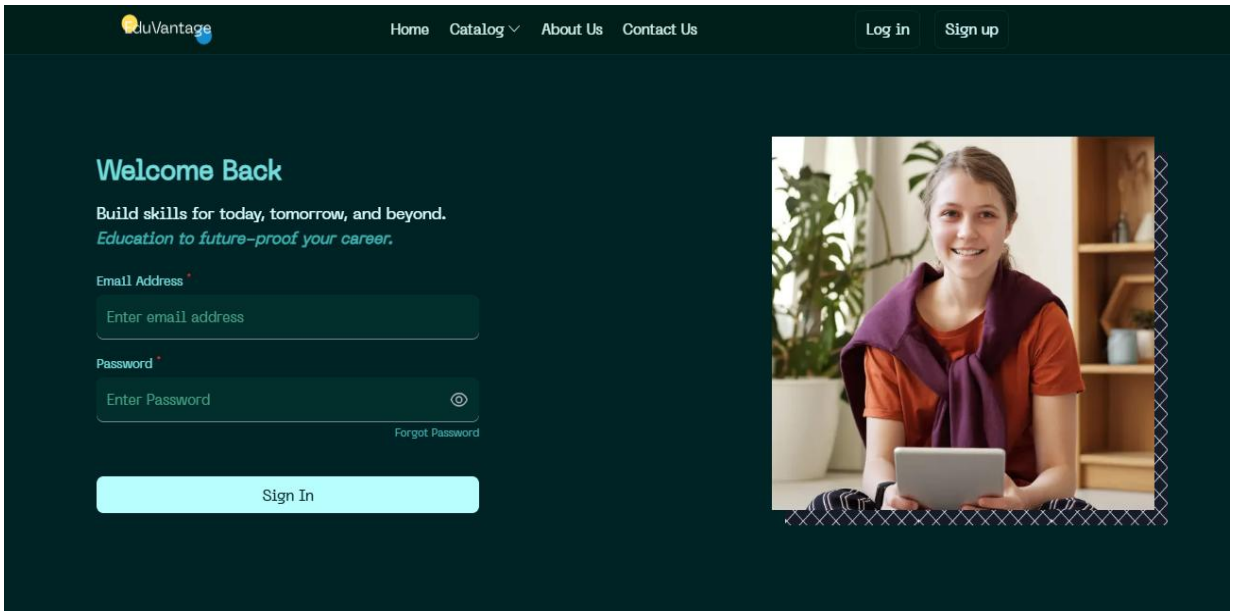


**Figure 7.1.4 Login Page**

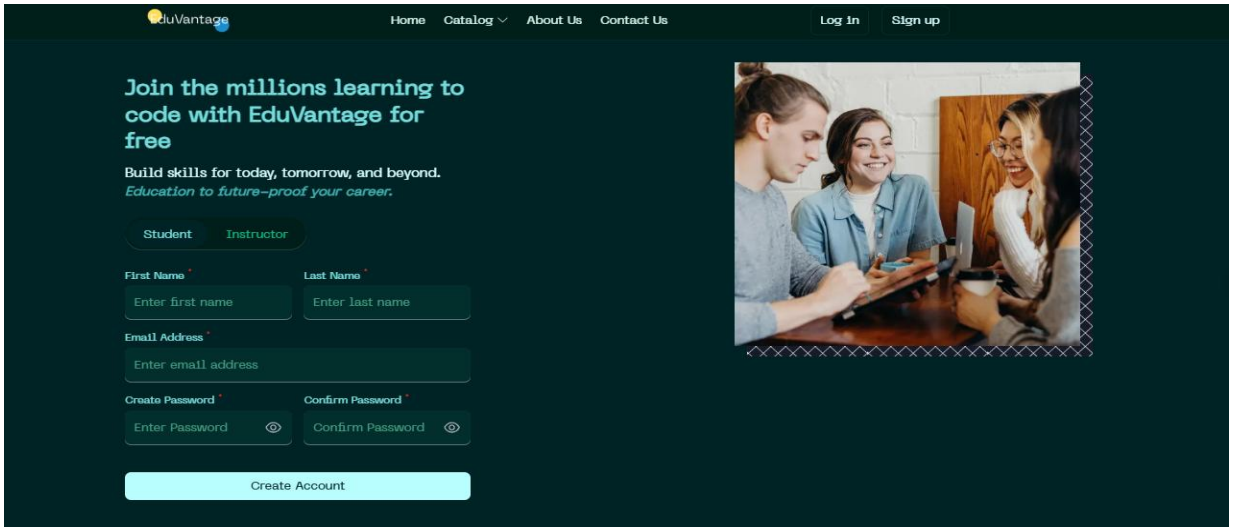


**Figure 7.1.5 Sign Up Page**

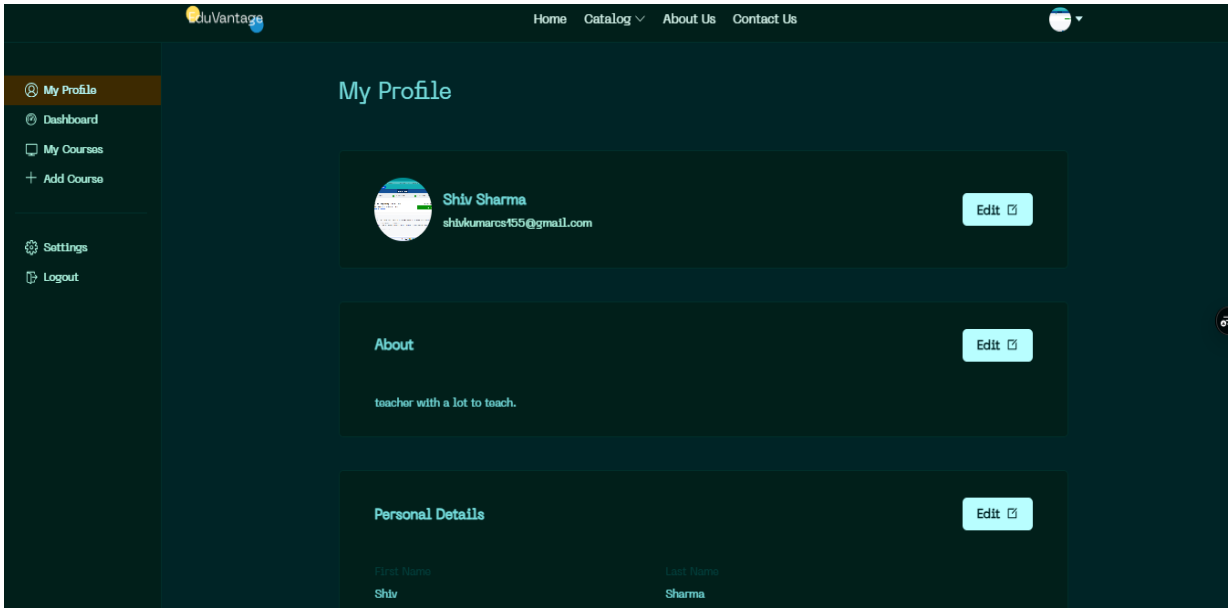


**Figure 7.1.6 Teacher's Dashboard**

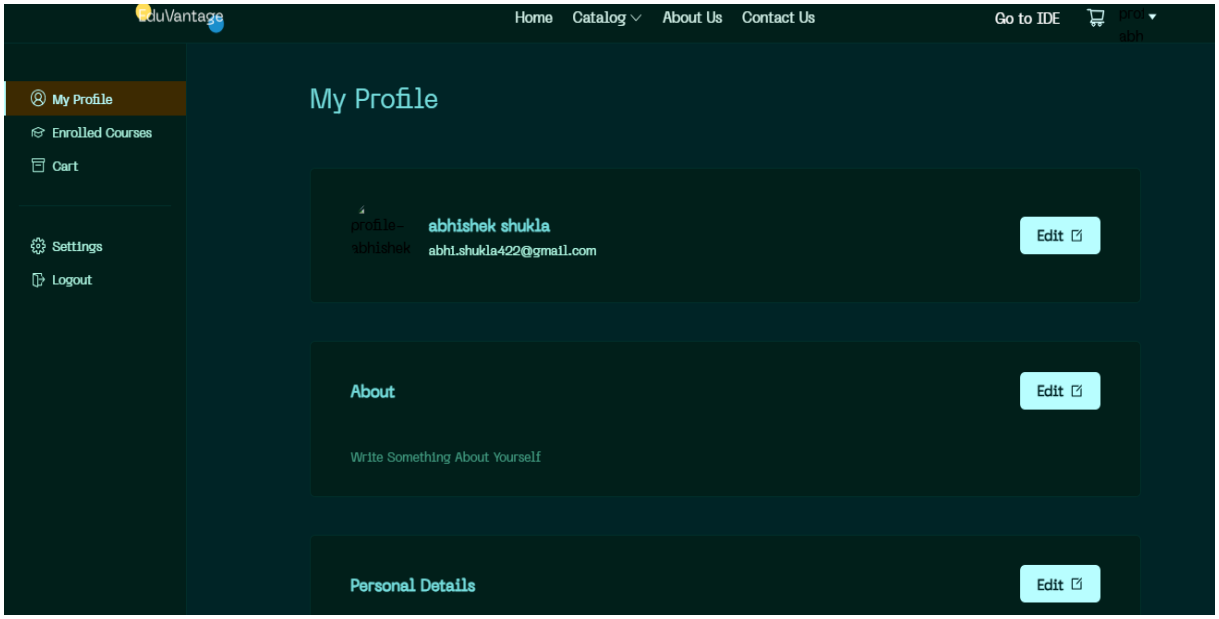


**Figure 7.1.7 Student's Dashboard**

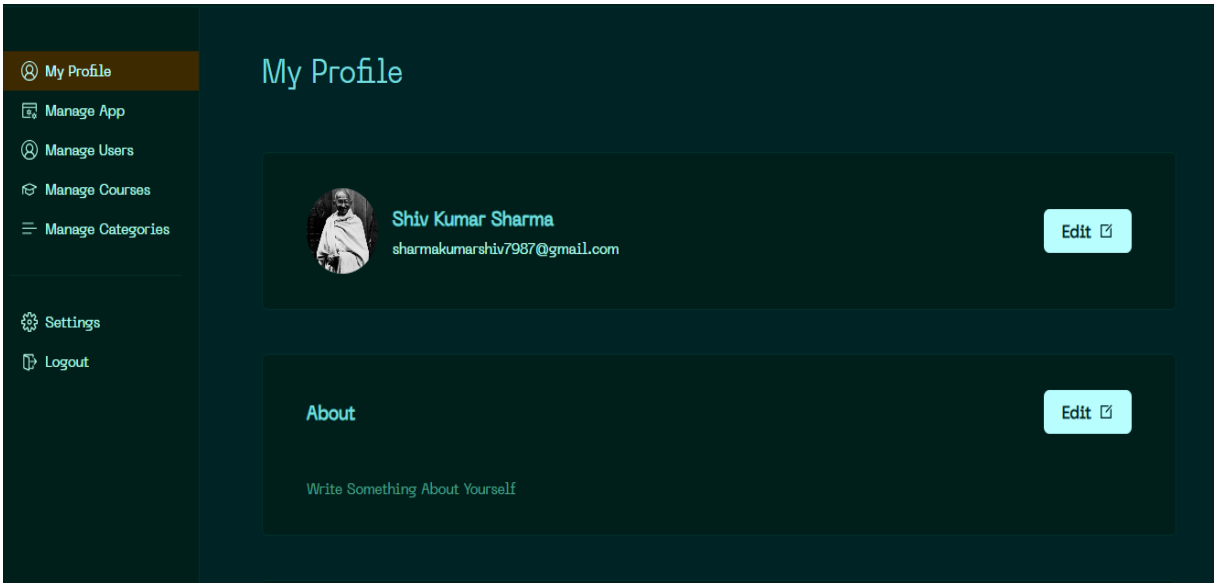


**Figure 7.1.8 Admin's Dashboard**

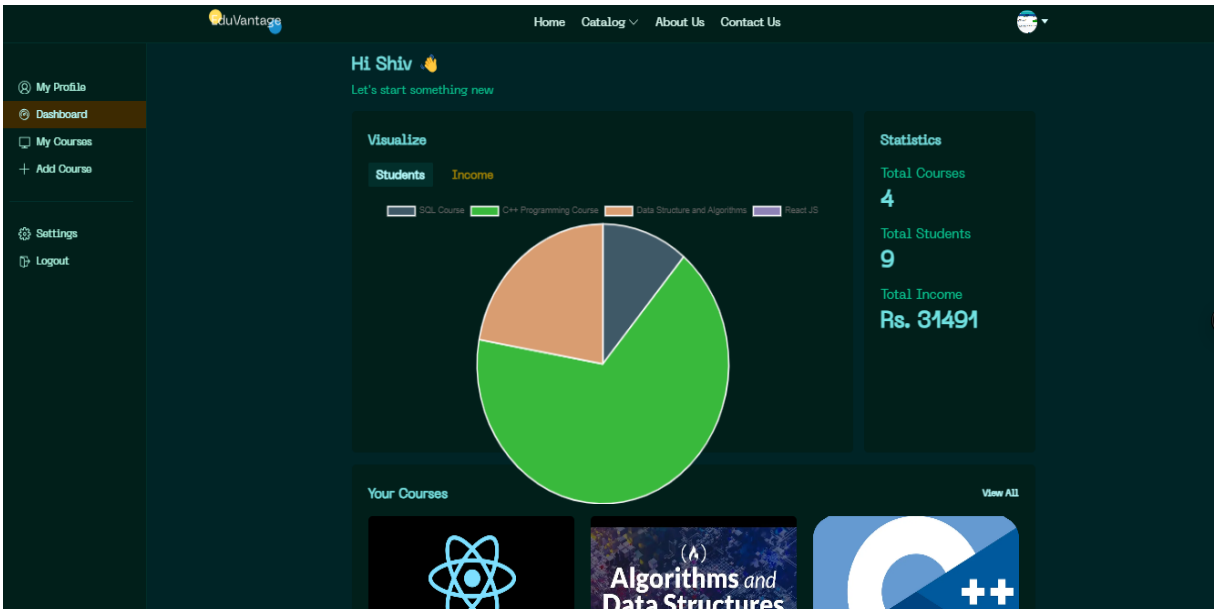


**Figure 7.1.9 Teacher's Stats Page**

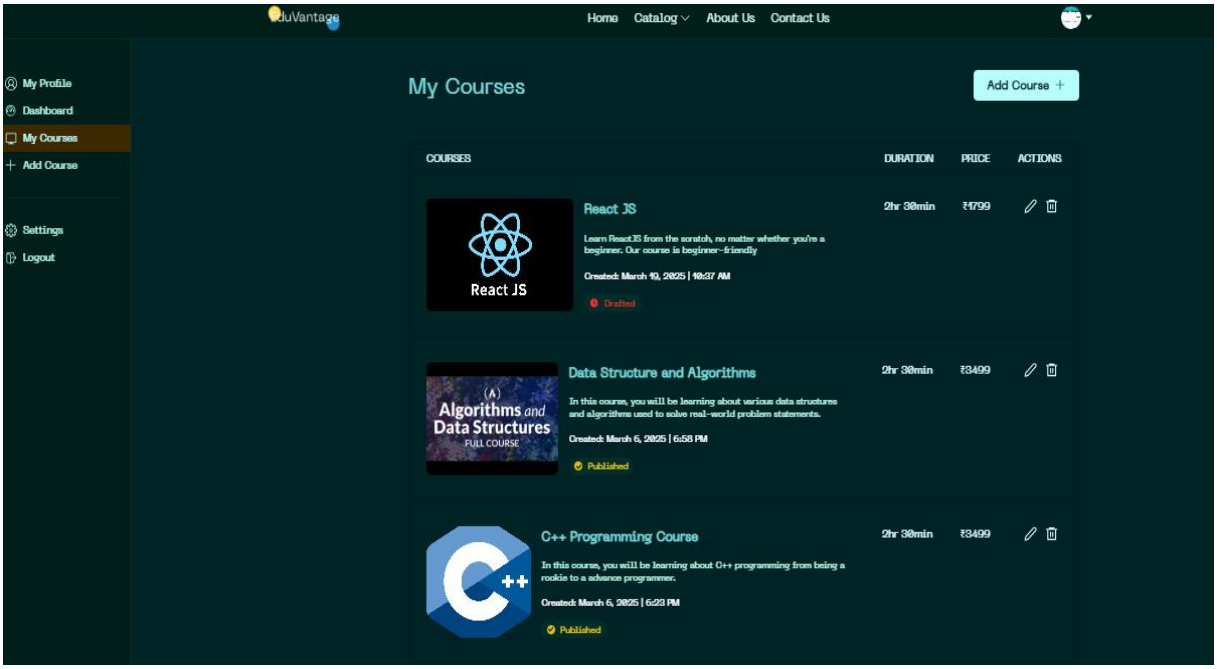


**Figure 7.1.10 Teacher's My Course Page**

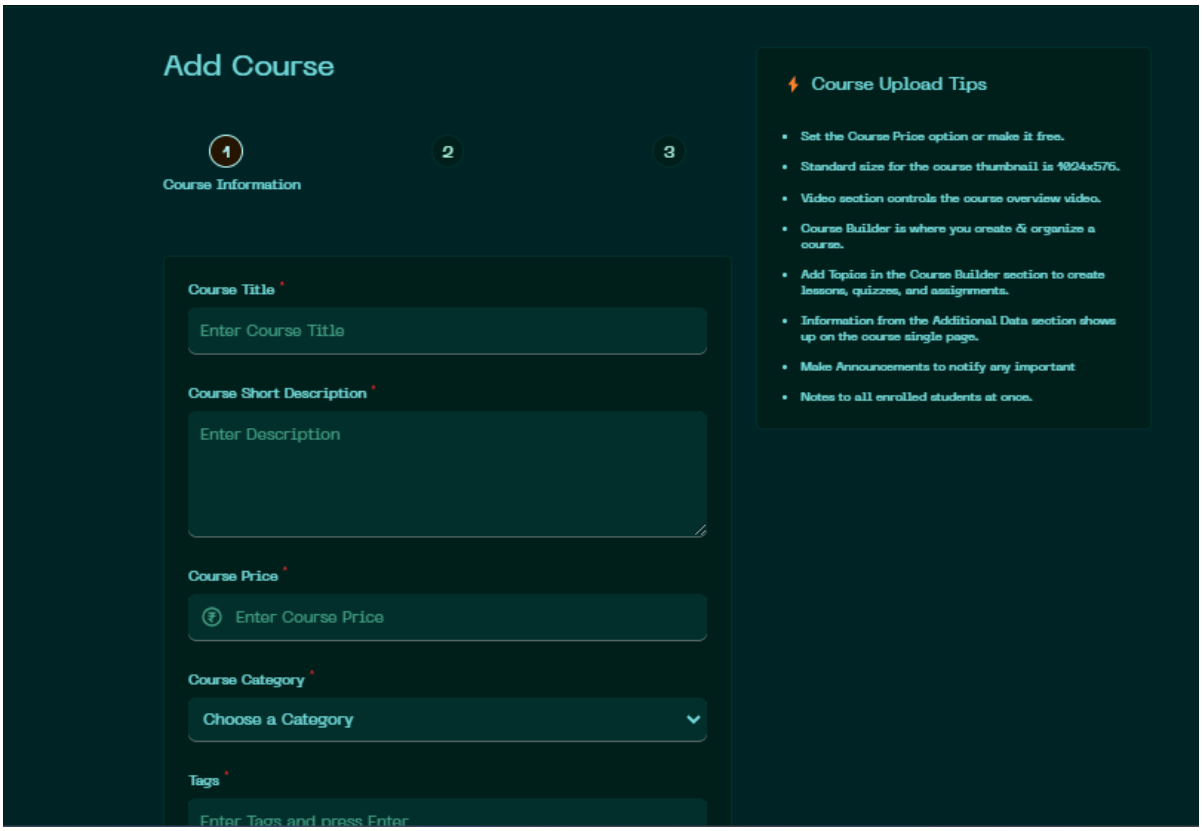


**Figure 7.1.11 Add Course Page**

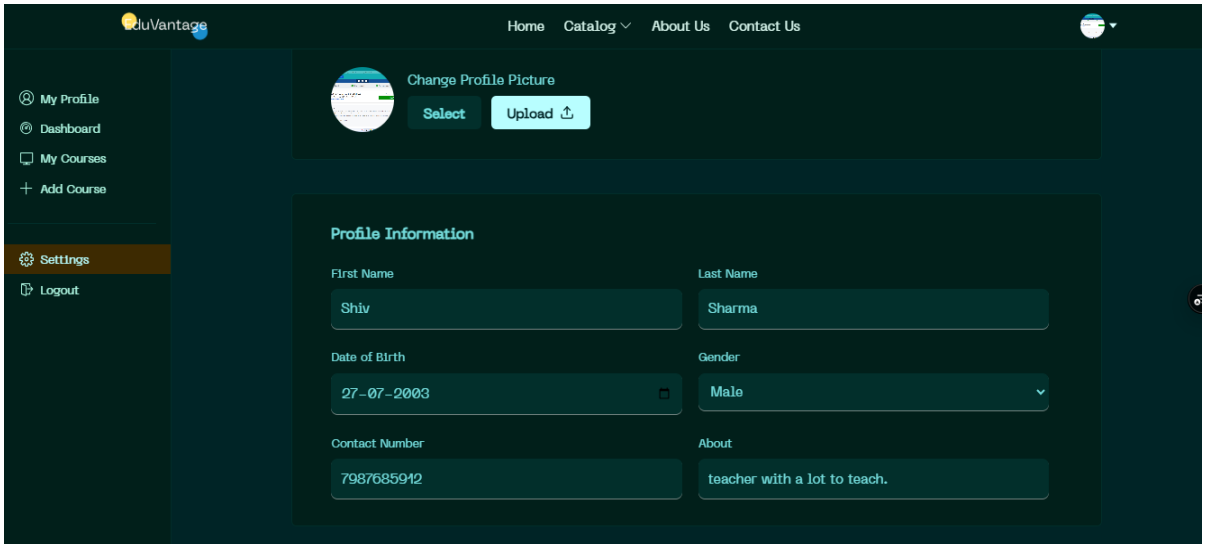


**Figure 7.1.12 Update Profile Page**

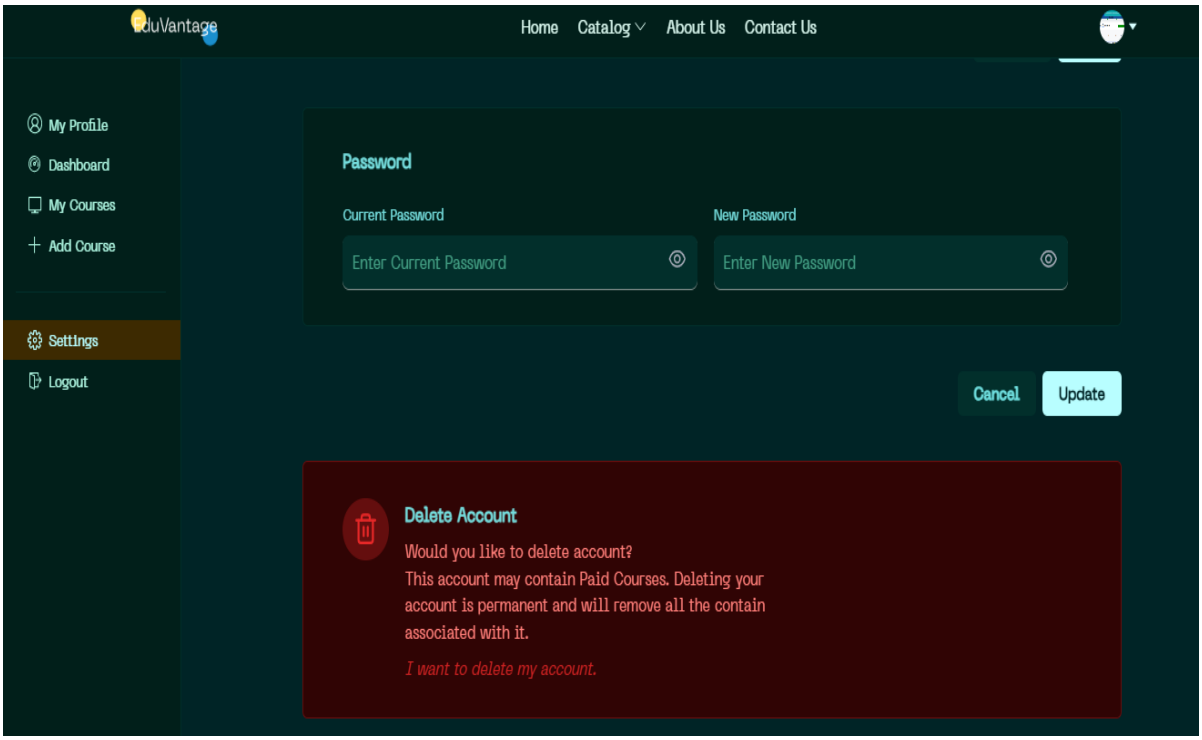


**Figure 7.1.13 Update Password and Delete Account Page**

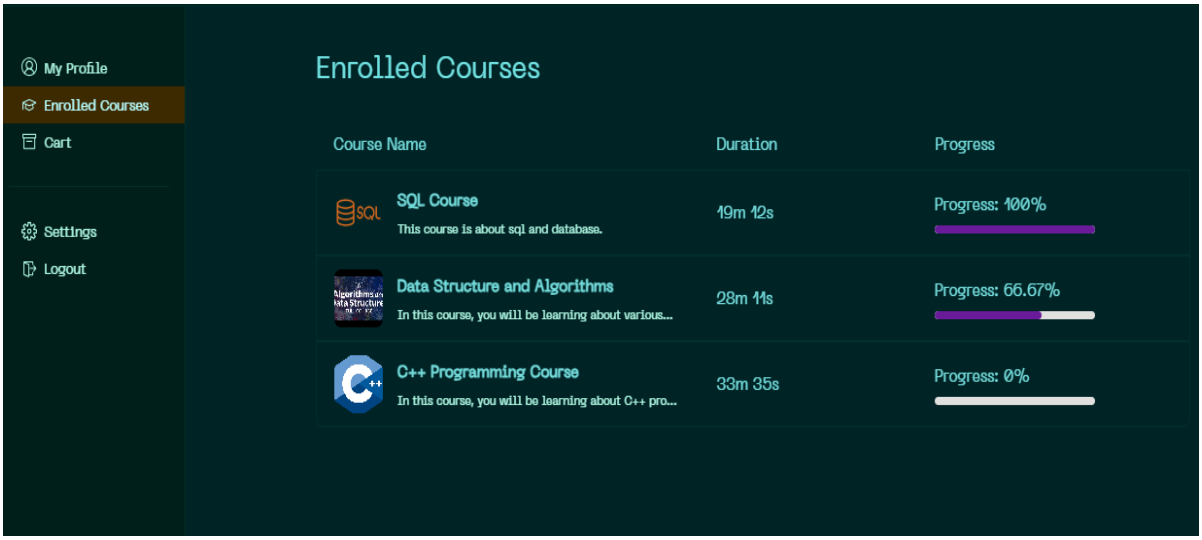


**Figure 7.1.14 Enrolled Course Page**

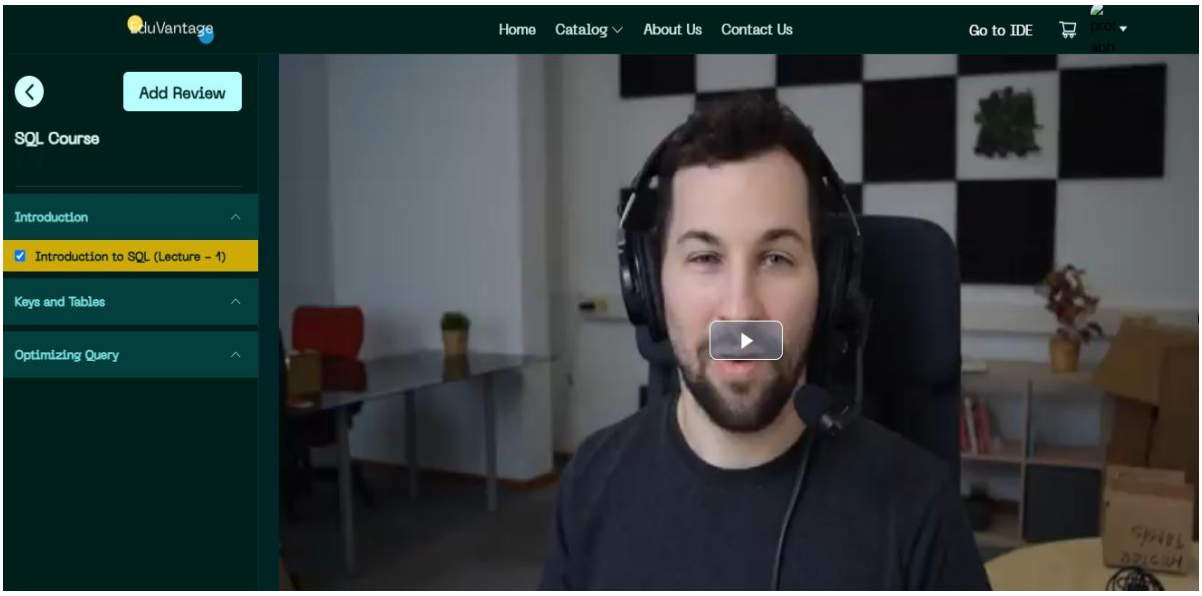


**Figure 7.1.15 Video Lecture Page**

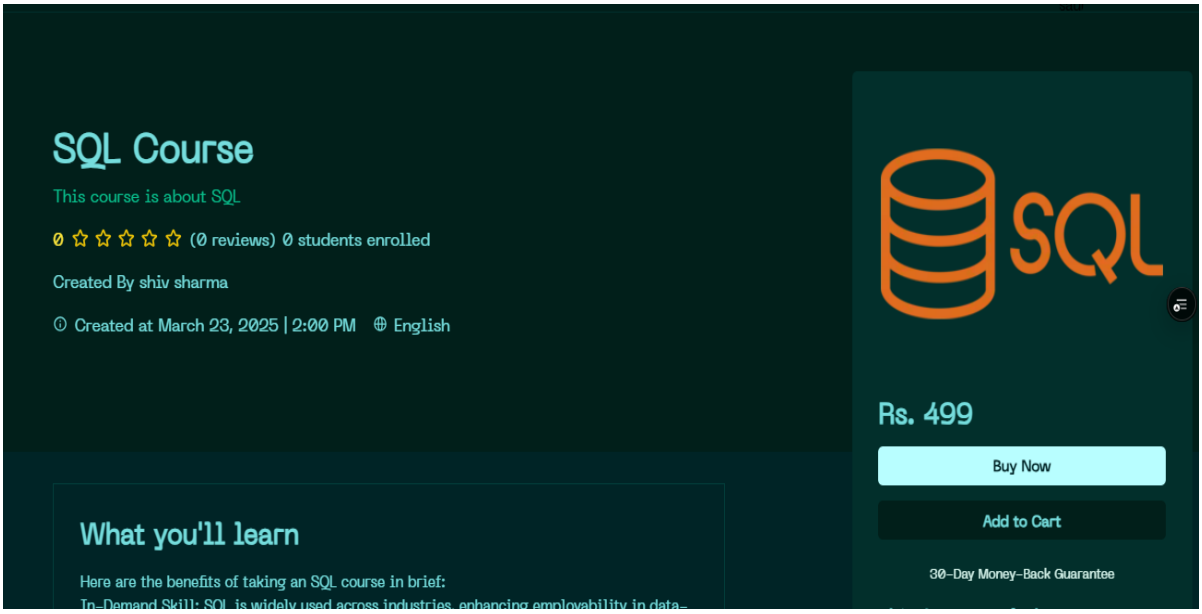


**Figure 7.1.16 Buy Course Page**

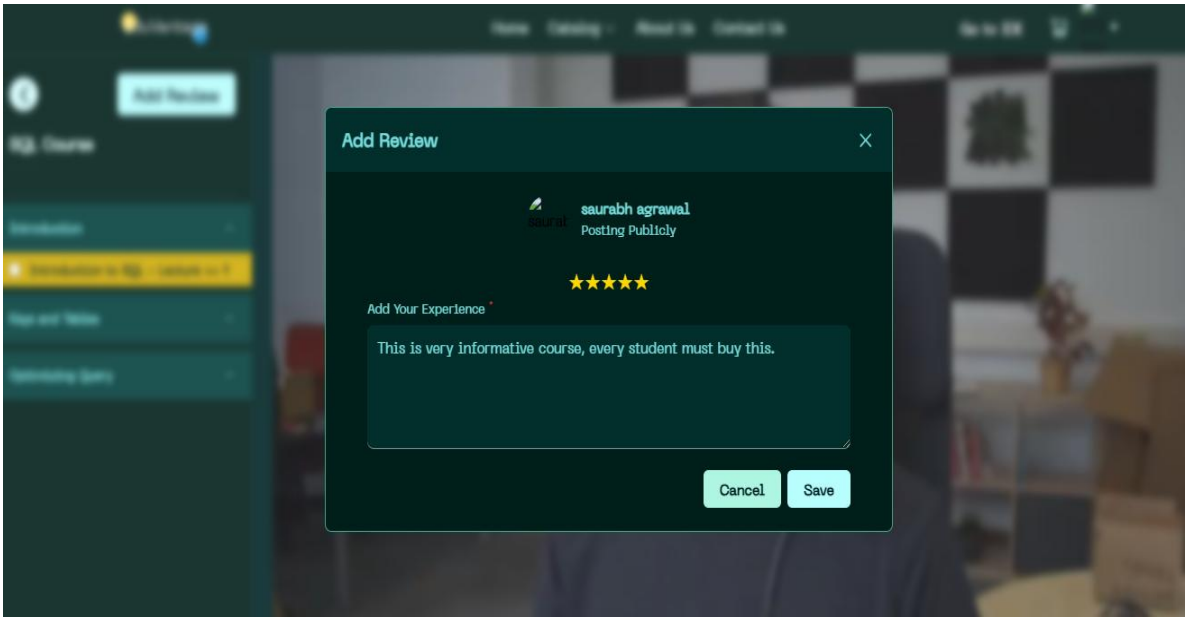


**Figure 7.1.17 Give Review Page**

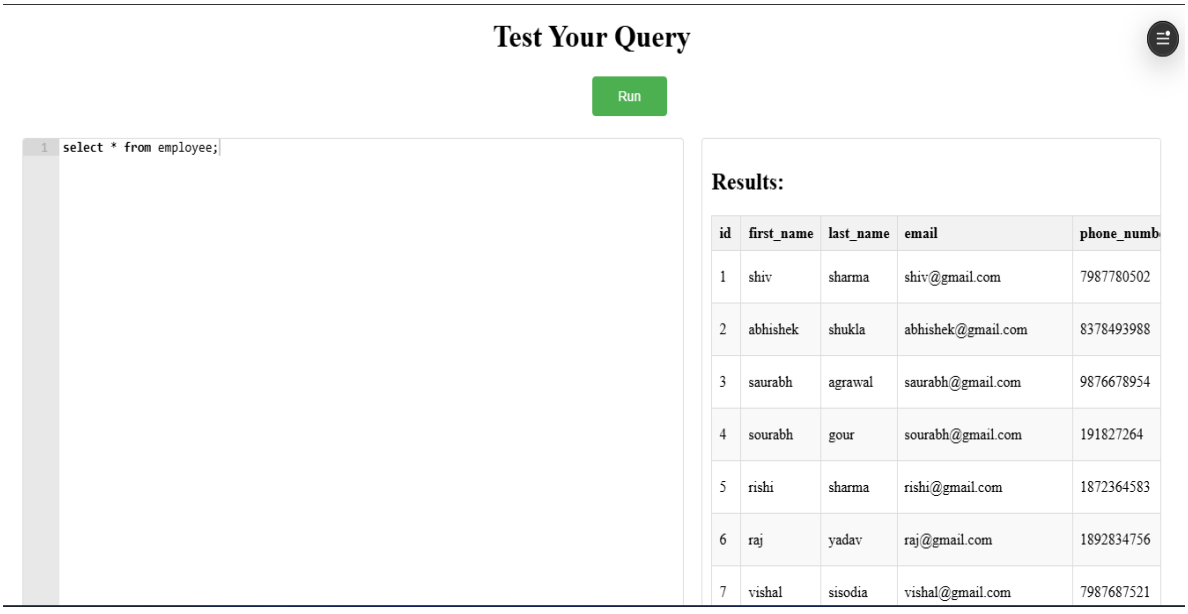


**Figure 7.1.18 IDE Page**

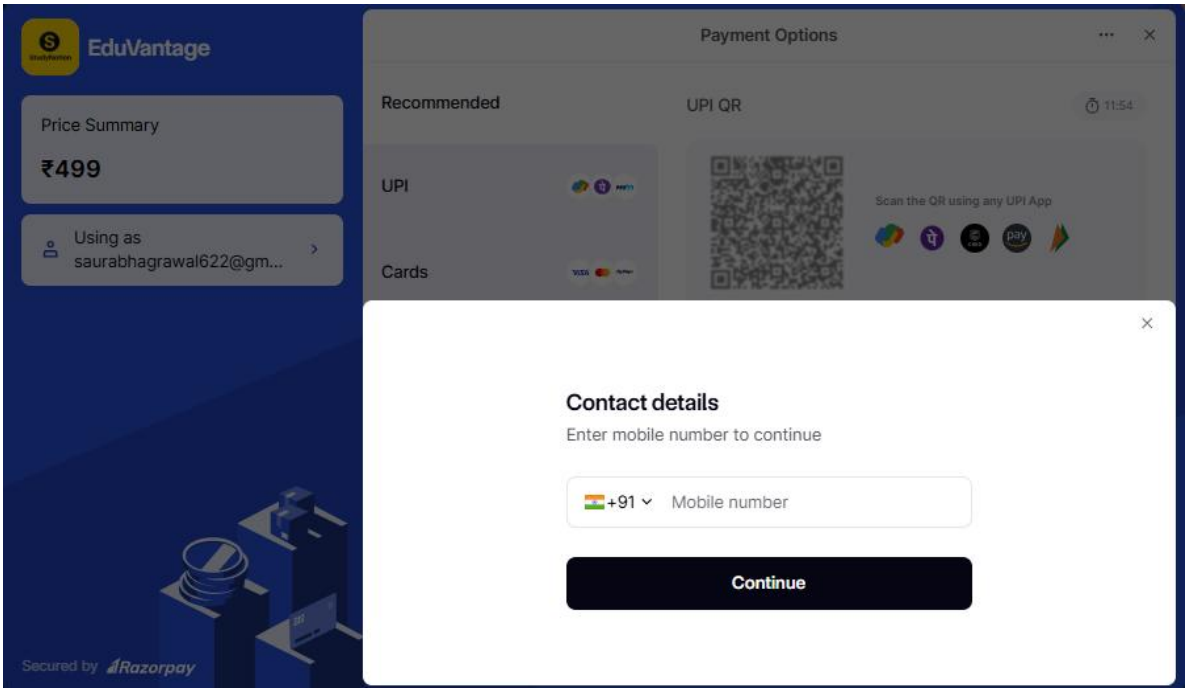


**Figure 7.1.19 Razorpay Interface Page**

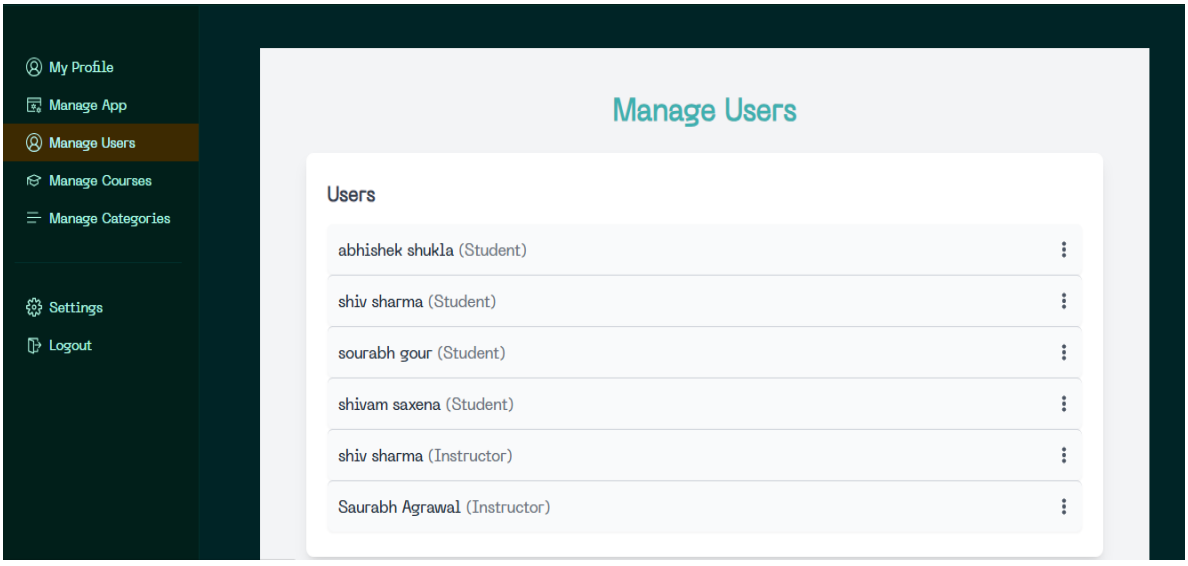


**Figure 7.1.20 Manage Users Page**

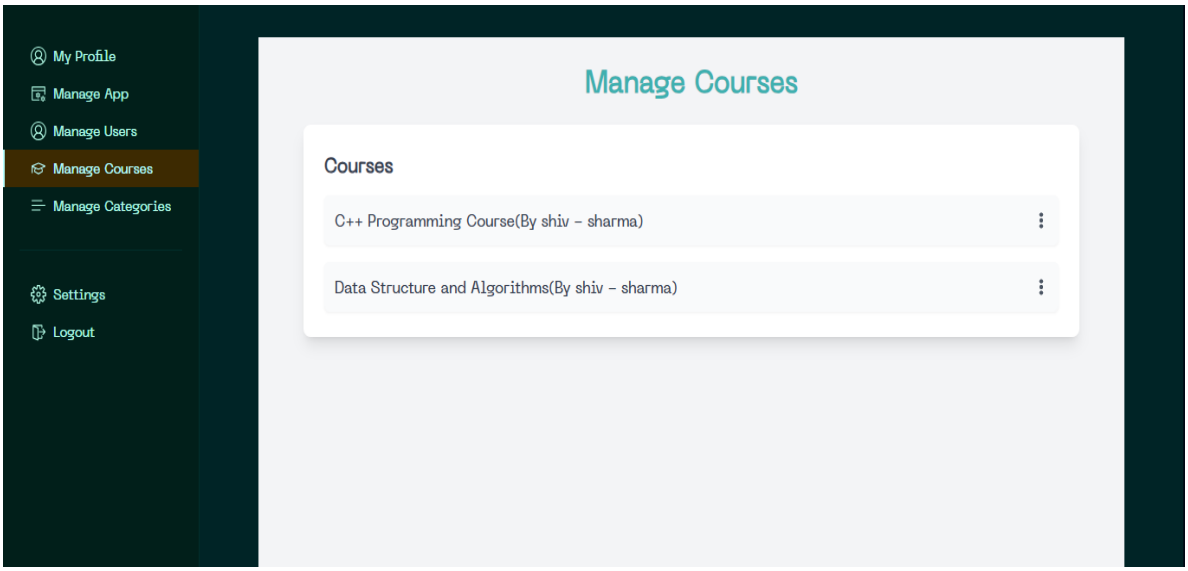


**Figure 7.1.21 Manage Courses Page**

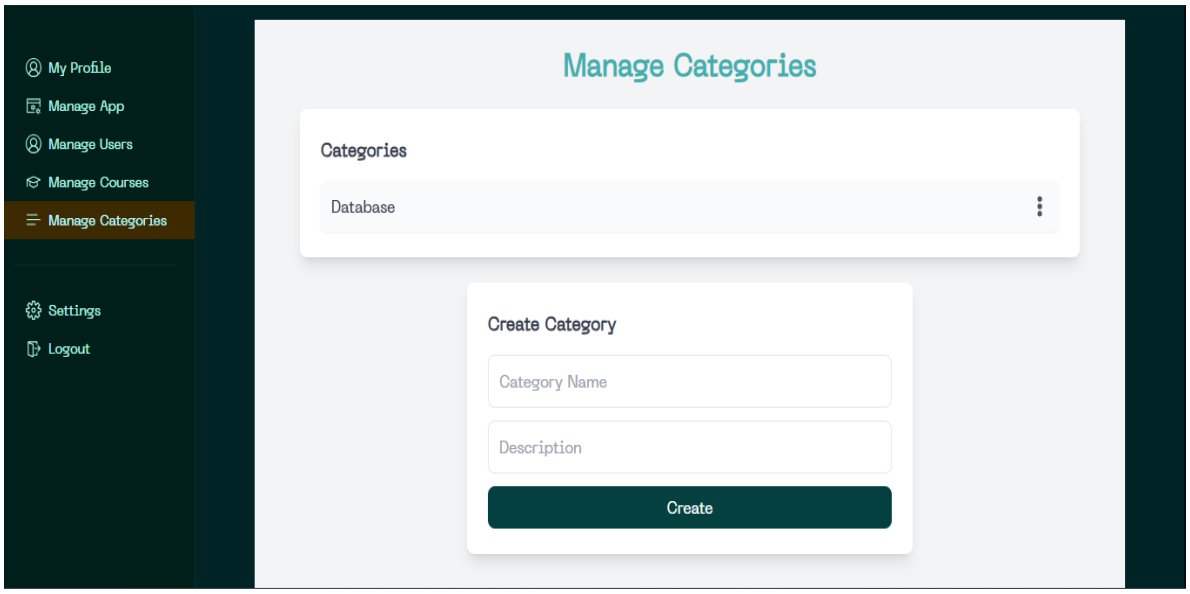


**Figure 7.1.22 Manage Categories Page**

# 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. To ensure seamless operation, the project is deployed using cloud-based services that provide scalability, reliability, and efficient resource management.

In this project, the frontend is hosted on Vercel, a platform optimized for fast and automated deployment of React applications. The backend is deployed on Render, which supports Node.js applications and provides automatic scaling, making it a suitable choice for hosting APIs. The database is managed using MongoDB Atlas, a cloud-based NoSQL database, ensuring high availability and secure storage of application data.

Additionally, Cloudinary is used for media storage, allowing efficient management of course materials such as videos and images. The Razorpay payment gateway is integrated to handle secure transactions for course purchases.

This chapter details the deployment process, environment setup, domain configuration, and continuous deployment strategy to ensure a smooth and professional production environment.

## 8.2 DEPLOYMENT ARCHITECTURE

1. **Frontend:** React.js (Hosted on Vercel)
2. **Backend:** Node.js & Express.js (Hosted on Render)
3. **Database:** MongoDB (Cloud-based, e.g., MongoDB Atlas)
4. **Cloud Storage:** Cloudinary (For media storage)
5. **Payment Gateway:** Razorpay

## 8.3 DEPLOYMENT OF FRONTEND

Vercel is a cloud platform optimized for frontend applications. It offers automatic deployments and global CDN distribution for fast performance.

Steps to Deploy React Frontend on Vercel

1. Push the project to GitHub/GitLab/Bitbucket.
2. Import the repository into Vercel Dashboard.
3. Configure Environment Variables (API URLs, Razorpay Keys, Cloudinary Keys).
4. Click Deploy.

Once deployed, Vercel provides a public URL for the frontend application.

## 8.4 DEPLOYMENT OF BACKEND

Render is a cloud platform that supports Node.js applications with automatic scaling and free hosting for small applications.

Steps to Deploy Node.js Backend on Render

1. Push the Backend Code to GitHub
2. Log in to Render.com
3. Click New Web Service and connect the GitHub repository.
4. Choose Node.js as the runtime.
5. Select Auto Deploy (for automatic updates on GitHub push).
6. Set up Environment Variables (MongoDB URI, Cloudinary API keys, Razorpay keys).
7. Click Deploy and wait for the build process to complete.

Once done, Render provides a public URL for the backend API.

## 8.5 DATABASE SETUP

MongoDB Atlas is a cloud-based database for secure and scalable data storage.

Steps to Set Up MongoDB Atlas

1. Create an Account on MongoDB Atlas

2. Create a New Cluster and Choose a Free Tier
3. Connect to the Database
4. Copy the MongoDB Connection String and update it in Render's environment variables

## 8.6 DOMAIN CONFIGURATION

For a professional setup, a custom domain can be linked to the Vercel frontend.

Steps to Connect a Custom Domain

1. Purchase a Domain (from Namecheap, GoDaddy, etc.).
2. Add the Domain in Vercel Dashboard.
3. Set CNAME record to Vercel’s provided URL.

Now, the frontend is accessible via the custom domain.

## 8.7 CONTINUOUS DEPLOYMENT

Both Vercel and Render support auto-deployments on every push to the GitHub repository.

Steps to Enable Continuous Deployment

1. Ensure GitHub repository is linked to Vercel (frontend) and Render (backend).
2. Every time code is pushed to GitHub, the changes are automatically deployed.
3. Monitor deployments via the Vercel and Render dashboards.