

**PROJECT REPORT OF INDUSTRY ORIENTED HANDS-ON
EXPERIENCE (IOHE)**

ON

EduElevate

submitted in partial fulfilment of the requirements for the award of degree of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Manan Arora

2110990834

Supervised By:

Dr. Gifty Gupta

Mentor



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**CHITKARA UNIVERSITY INSTITUTE OF ENGINEERING AND TECHNOLOGY
CHITKARA UNIVERSITY, PUNJAB, INDIA**

DECLARATION

I hereby certify that the work which is being presented in the project report entitled EduElevate in partial fulfilment of requirement for the award of the degree of Bachelor of Engineering (Computer Science and Engineering) submitted in the department of Computer Science and Engineering at Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India, is an authentic record of my own work carried out under the supervision of Dr. Gifty Gupta. The matter presented in this project report has not been submitted in any other university/institute for the award of any degree.

Place : Rajpura

Date : 17May,2025

Manan Arora

2110990834

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

Dr. Gifty Gupta

Mentor

Department of Computer Science and Engineering

Chitkara University Institute of Engineering and Technology,

Chitkara University, Punjab, India

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to Chitkara University for providing me with the opportunity to undertake this project as a part of the Industry Oriented Hands-On Experience (IOHE) program. This project has been a valuable learning experience and has helped me bridge the gap between academic knowledge and real-world application.

I extend my sincere thanks to my mentor, Dr. Gifty Gupta for their constant guidance, valuable feedback, and encouragement throughout the project. Their support was instrumental in helping me navigate challenges and complete the project successfully.

I would also like to thank all the faculty members of the Department of Computer Science and Engineering for their continuous support and for creating a conducive learning environment.

Lastly, I am thankful for the opportunity to independently execute this project, which helped me enhance my problem-solving skills, technical understanding, and confidence as a software developer.

ABSTRACT

In the era of digital transformation, online education platforms have become essential in providing accessible and flexible learning opportunities to students across the globe. EduElevate is an innovative full-stack EdTech platform developed with the aim of delivering a user-centric, scalable, and secure learning management experience. This platform allows instructors to create and manage educational content while enabling students to learn through structured courses, engaging videos, quizzes, and community interaction.

The core features of EduElevate include a robust authentication system using JWT and bcrypt for security, a comprehensive course management system, Cloudinary-based video hosting, OTP verification via node mailer, and role-based dashboards for students and instructors. The student dashboard provides access to course progress, profile editing, and enrolled courses. Instructors are empowered with tools to add, view, and manage their courses seamlessly. The platform also includes dynamic interfaces like a homepage, about and contact pages, and a course exploration section.

To ensure data persistence and integrity, a well-structured MongoDB database schema was implemented, consisting of collections for users, profiles, course categories, OTPs, ratings, reviews, and more. Frontend development was carried out using React.js and Tailwind CSS, offering a modern and responsive UI/UX, while Node.js and Express.js form the backbone of the backend. Also used react-native for android app development.

This report presents the detailed implementation methodology, technologies used, challenges encountered, and outcomes achieved during the development of the platform. With continuous development and optimization, EduElevate aims to scale further by integrating advanced features such as live sessions, discussion forums, and intelligent course recommendations. The project reflects a comprehensive hands-on experience with industry-relevant technologies and practices in software development.

Table of Contents

S. No.	Contents	Page No.
	Declaration	
	Certificate	
	Acknowledgment	i
	Abstract	ii
	Table of Contents	iii
	List of Figures	iv
	List of Tables	v
	List of Abbreviations	vi
1	Introduction	1
1.1	Background	1
1.2	Problem Statement	1
1.3	Objectives of the Project	2
1.4	Scope of the Project	2
1.5	Methodology	3
1.6	Organization of the Report	4
2	System Analysis	5
2.1	Requirement Analysis	5
2.2	Feasibility Study	6
2.3	System Architecture	6
2.4	Tools and Technologies Used	7
3	System Design	7
3.1	Overview of System Design	7
3.2	Data Flow Diagrams / UML Diagrams	7-8
3.3	Database Design	9
3.4	Interface Design	10
4	Implementation	11
4.1	Modules Description	11
4.2	Development Tools	12
4.3	Implementation Steps	13
4.4	Screenshots and Explanation	14-29
5	Testing and Results	30
5.1	Testing Methodology	30
5.2	Test Cases and Results	30-31
5.3	Performance Analysis	31
5.4	Final Output	32
6	Conclusion and Future Scope	32-33

List of Figures

Fig No.	Figure Title	Page No.
1	Login page (Student/Instructor)	13
2	Signup page (Student/Instructor)	13
3	Home page (i)	14
4	Home page (ii)	15
5	Home page (iii)	16
6	Footer	16
7	Aboutus page (i)	17
8	Aboutus page(ii)	17
9	Contactus page	18
10	User profile page after login	18
11	Course details page	19
12	Razorpay payment gateway	19
13	Enrolled courses	20
14	Video lectures after enrolment	20
15	Instructor dashboard	21
16	Courses list created by Instructor	21
17	Add courses section (i)	22
18	Add courses section (ii)	23
19	Android app home page	24
20	Course details page	25
21	Course content page	26
22	Video lecture page	27

List of Tables

S. No.	Table Title	Page No.
1	Tools and Technologies used	7
2	Database collections	8
3	Development tools	11
4	Test cases and results	29

List of Abbreviations

Abbreviation	Full Form
UI	User Interface
UX	User Experience
API	Application Programming Interface
OTP	One-Time Password
JWT	JSON Web Token
CRUD	Create, Read, Update, Delete
DB	Database
IDE	Integrated Development Environment
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
MERN	MongoDB, Express.js, React.js, Node.js
DBMS	Database Management System
SDK	Software Development Kit
HTTP	HyperText Transfer Protocol

1. Introduction

EduElevate is an innovative online learning platform designed to bridge the gap between traditional education and digital learning. It provides a wide range of courses, progress tracking, and video lectures to help users learn effectively. The platform caters to both students and professionals looking to enhance their skills in various domains.

1.1 Background

With the exponential rise of the internet and digital tools, online learning has emerged as a transformative approach to education. Traditional classroom learning methods are increasingly being complemented or even replaced by virtual platforms that offer greater flexibility, accessibility, and scalability. The demand for robust e-learning solutions has surged, particularly in the wake of the COVID-19 pandemic, which necessitated remote education across the globe.

EduElevate is designed in response to this demand, offering a centralized platform where instructors can create and publish their courses, and students can easily access quality educational content from anywhere at any time. It integrates modern frontend technologies like React.js and Tailwind CSS for an engaging user experience, while leveraging backend technologies like Node.js, Express.js, and MongoDB for scalable and secure data handling. The platform also aims to simplify the learning process through intuitive navigation, multimedia content support, and user-specific dashboards.

By combining innovative technologies and user-centric design, EduElevate aims to bridge the gap between learners and educators, ensuring efficient knowledge transfer in a digital format.

1.2 Problem Statement:

In the digital age, accessing quality education remains a challenge due to high costs, limited resources, and the absence of an engaging learning environment. Traditional education methods often fail to adapt to modern learning needs, making it difficult for learners to acquire new skills efficiently. EduElevate aims to solve these challenges by providing an interactive, affordable, and accessible learning platform.

1.3 Objectives of the Project

The primary objective of the EduElevate project is to develop a comprehensive and user-friendly online learning platform that bridges the gap between instructors and learners. The platform aims to simplify the process of course creation, delivery, and consumption in a structured and efficient manner.

The specific objectives of the project are:

- To create a responsive and modern web interface using React.js and Tailwind CSS.
- To develop a robust backend system using Node.js and Express.js for handling API requests, authentication, and business logic.
- To design a flexible and scalable database schema using MongoDB to store and manage data related to users, courses, categories, reviews, and more.
- To implement authentication and role-based access control (RBAC) to differentiate between students and instructors.
- To provide instructors with an intuitive dashboard for uploading course content, managing lectures, and tracking learners.
- To offer students an organized dashboard to view purchased courses, update profile information, and give feedback through ratings and reviews.
- To integrate secure payment functionality for course enrolment (if applicable).
- To ensure good user experience by maintaining responsiveness, accessibility, and error handling throughout the platform.
- To develop a basic Android version of the application using React Native for wider accessibility (with static content in the current version).

1.4 Scope of the Project

The scope of the EduElevate project encompasses the development of a full-stack e-learning web application that serves as a dynamic platform for both students and instructors. The project aims to provide essential features such as user authentication, course creation, course enrolment, video content delivery, progress tracking, and profile management.

The platform is designed to cater to multiple user roles — primarily students and instructors — allowing instructors to upload and manage their own courses while students can browse, enrol, and learn at their own pace. The system supports multimedia content, including video hosting and rich course descriptions, ensuring a comprehensive learning experience.

Additionally, the backend infrastructure ensures secure data handling using MongoDB, with Express.js and Node.js managing APIs and business logic. The frontend is built using React.js and Tailwind CSS to ensure a responsive and interactive user interface. An Android application with static content has also been developed using React Native to explore cross-platform accessibility.

Though currently limited to core learning features, the project lays the foundation for integrating advanced functionalities such as quizzes, assignments, live classes, discussion forums, and certificates in future updates, thereby expanding the overall utility and impact of the platform.

Potential Benefits of EduElevate

- **Enhanced Learning Experience:**
 - Users can access high-quality content tailored to their learning needs.
 - A variety of assessment tools ensure continuous improvement.
- **Skill Development and Career Growth:**
 - Helps learners prepare for job opportunities through industry-relevant courses.
- **Secure and Scalable Platform:**
 - Ensures data security and privacy for users.
 - Scalable architecture to support a growing user base.

Target Audience

- Students looking for additional learning resources.
- Working professionals seeking upskilling opportunities.
- Instructors wanting to share knowledge with a wider audience.

1.5 Methodology

The methodology adopted for the development of the **EduElevate** platform follows a structured and iterative approach, combining both frontend and backend development processes. The project is implemented using the **Agile Software Development Model**, which supports incremental progress, continuous feedback, and flexibility in development.

i. Requirement Gathering and Analysis

The first phase involved understanding the problem, identifying the key features of an online learning platform, and gathering the functional and non-functional requirements. Discussions were held to decide the key user roles (students and instructors), data flow, and overall system behaviour.

ii. System Design

This phase focused on planning the architecture of the application, including frontend-backend interaction, database schema design, and role-based access control. Tools like Figma and flow diagrams were used to visualize the UI layout and system workflow.

iii. Development

The development was carried out in two main parts:

- **Frontend Development:** Using **React.js** and **Tailwind CSS**, dynamic and responsive interfaces were built. Pages such as Home, Signup, Login, Dashboard (for both students and instructors), Course Details, and Profile were created.
- **Backend Development:** Using **Node.js** and **Express.js**, APIs were created for authentication, user profile management, course creation, and course consumption. **MongoDB** was used as the database, with collections designed for users, profiles, courses, categories, OTPs, and more.
- **Mobile App Development:** A simple Android version using **React Native** was created to show basic static data and interface for future expansion.

iv. Testing

Each module was tested independently to ensure proper validation and error handling. Functional testing, UI testing, and form validation were performed to ensure the platform's reliability and responsiveness.

v. Deployment

The web application was deployed using **Render** for the frontend and the backend API, allowing remote access and testing on real-world networks.

vi. Feedback and Iteration

Based on periodic evaluations and personal review, improvements were made in UI/UX, validation, routing, and database interactions. The system remains under development for future modules like quizzes, assignments, and integrated payment support.

1.6 Organization of the Report

This project report is organized into six chapters, each highlighting a specific aspect of the development lifecycle of the EduElevate platform:

- **Chapter 1: Introduction** – Provides a comprehensive overview of the project, including background, problem statement, objectives, scope, methodology, and the overall structure of the report.
- **Chapter 2: System Analysis** – Covers requirement gathering, feasibility study, system architecture, and a detailed explanation of the tools and technologies used during the development.

- **Chapter 3: System Design** – Describes the system's architecture in terms of design diagrams, database schema, and interface planning that shaped the overall layout and flow of the application.
- **Chapter 4: Implementation** – Focuses on the implementation process, module-wise descriptions, tools used, step-by-step development process, and user interface screenshots with explanations.
- **Chapter 5: Testing and Results** – Discusses the testing methodology, test cases, performance evaluation, and final output obtained from the system.
- **Chapter 6: Conclusion and Future Scope** – Summarizes the outcomes of the project and outlines potential future enhancements like assignments, quizzes, and live classes.

Each chapter has been written to document the development journey of the platform in a structured and systematic manner.

2. System Analysis

2.1 Requirement Analysis

To successfully build the EduElevate platform, both functional and non-functional requirements were identified:

Functional Requirements:

- User authentication (Sign up, Login, OTP verification, Reset password).
- Role-based access control (Student/Instructor).
- Course creation and management for instructors.
- Course browsing and enrolment for students.
- Video previews, course progress tracking.
- Student dashboard for personal profile and settings.
- Instructor dashboard with course management options.
- Contact form for user queries.

Non-Functional Requirements:

- Responsive user interface across devices.
- Secure data handling (JWT, bcrypt.js).
- Fast performance and minimal load times.
- Reliable email services for OTP and password resets.
- Scalability to handle more users and courses in future.

2.2 Feasibility Study

A feasibility study was conducted to evaluate the practicality of implementing the project within constraints of time, budget, and available technology.

- **Technical Feasibility:**

The use of React.js, Node.js, Express.js, MongoDB, and Tailwind CSS makes the development technically feasible. All technologies are open-source and have strong community support.

- **Operational Feasibility:**

The application meets the real-world needs of users seeking to learn or teach online. Students can easily navigate, enrol in courses, and instructors can publish content with minimal training.

- **Economic Feasibility:**

As all tools and services used (except optional paid AWS) are free or have generous free tiers, the project is economically feasible for individual development.

2.3 System Architecture

The platform follows a **client-server architecture**. It is a MERN stack-based web application:

- **Frontend (Client Side):** Developed using React.js and Tailwind CSS. The frontend interacts with backend APIs to fetch or send data.
- **Backend (Server Side):** Built with Node.js and Express.js, responsible for handling authentication, course management, OTP services, and database interactions.
- **Database:** MongoDB is used as the NoSQL database to store users, profiles, OTPs, course data, ratings, and progress.
- **Additional Services:**
 - **Email Service:** Nodemailer is used to send OTPs and reset password links.
 - **Cloud Storage (optional for future scope):** AWS S3 integration planned for video storage.

2.4 Tools and Technologies Used

Tool/Technology	Purpose
React.js	Frontend UI development
Tailwind CSS	Styling and responsive design
Node.js	Backend runtime environment
Express.js	Backend web framework
MongoDB	Database to store all application data
Mongoose	MongoDB object modeling tool
Nodemailer	Sending OTPs and emails
JWT (jsonwebtoken)	Secure user authentication
bcrypt.js	Password hashing
React Router	Frontend routing
Axios	API communication
React Native	Static Android application for demonstration
Postman	API testing and development
Razorpay	Payment gateway integration
Cloudinary	Cloud storage for image uploads (e.g., profile pictures, course thumbnails)

Table 1. Tools and Technologies Used

3. System Design

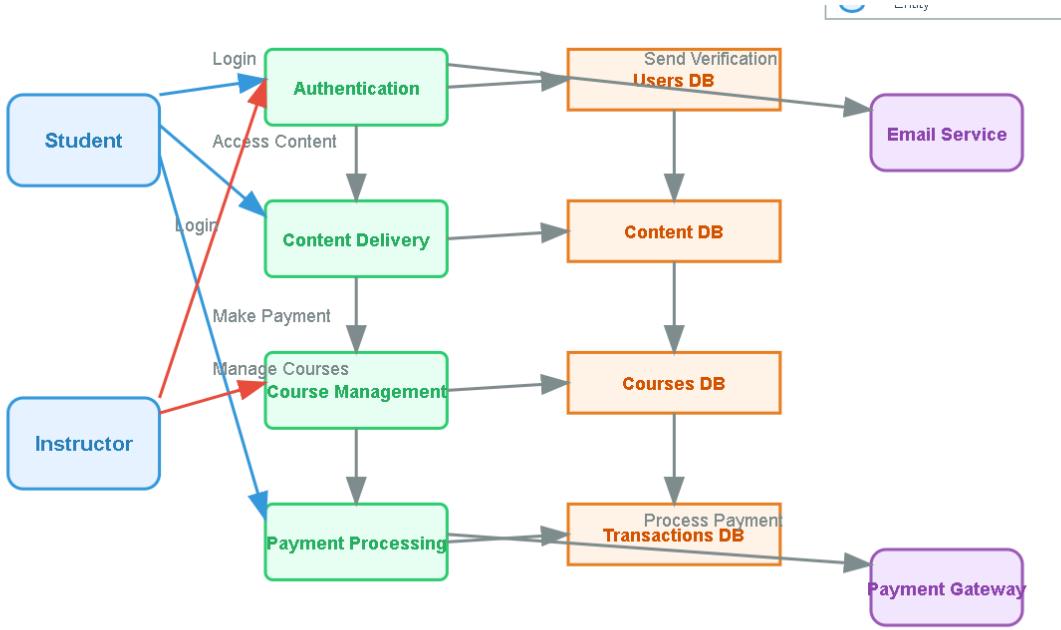
3.1 Overview of System Design

System design is a crucial phase where the structure and components of the application are planned to ensure efficiency, reliability, and scalability. EduElevate is designed as a full-stack web application using the MERN stack (MongoDB, Express.js, React.js, Node.js). It follows a **modular** and **component-based** architecture that separates concerns and promotes maintainability.

The system supports two types of users – **Students** and **Instructors**, with dashboards and functionalities tailored to each. The backend is RESTful, and all communication between client and server happens via secure HTTP requests.

3.2 Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) provides a high-level graphical representation of the flow of data within the EduElevate platform. It highlights how data moves between users (students and instructors), system processes, data stores, and external systems. This diagram helps stakeholders understand the logical flow of information, ensuring system functionalities are well defined and interconnected.



Components of the DFD

1. External Entities

- Student: A primary user who can register/login, access course content, and make payments.
- Instructor: An admin user who can log in and manage (create/update/delete) courses.

2. Processes

- Authentication: Handles login and registration for both students and instructors.
- Content Delivery: Provides access to course videos and learning material.
- Course Management: Allows instructors to upload, update, and delete course content.
- Payment Processing: Manages the student's course payment workflow.

3. Data Stores

- Users DB: Stores registered user data including authentication credentials.
- Content DB: Stores all course materials such as video lectures, documents, etc.
- Courses DB: Maintains the course metadata such as titles, prices, and enrolled users.
- Transactions DB: Logs payment transactions and associated user details.

4. External Systems

- Email Service: Sends email verifications during sign-up and communication alerts.

- Payment Gateway: A third-party service that securely processes student payments.

Data Flows Explained

- **Student Data Flows (Blue Arrows):**
 - A student can log in through the Authentication module.
 - Once authenticated, the student accesses content through the Content Delivery process.
 - To enroll in paid courses, the student proceeds to Payment Processing.
- **Instructor Data Flows (Red Arrows):**
 - Instructors authenticate and access the Course Management module.
 - They can manage the Courses DB via the Course Management process.
- **System Data Flows (Gray Arrows):**
 - All processes interact with respective databases for reading/writing data.
 - Authentication communicates with the Email Service for verification.
 - Payment Processing interacts with the Payment Gateway to complete financial transactions.
- **Inter-Process Flows:**
 - Each process connects to the next logically. For example, Authentication leads to Content Delivery, then to Course Management, and finally to Payment Processing.

3.3 Database Design

The database is designed using MongoDB with the following key collections:

Main Collections:

Collection	Description
Users	Stores basic user data including name, email, password (hashed), account type (student/instructor), authentication tokens, and course progress references.
Profiles	Stores additional user information such as date of birth, gender, bio, contact number, and profile picture. Linked via user ID.
Courses	Contains metadata for each course including title, description, price, tags, thumbnail, category, students enrolled, average ratings, and instructor reference.
Sections	Each course is divided into multiple sections. This collection stores section titles and is linked to a specific course.

Collection	Description
Subsections	Stores content inside each section such as title, description, time duration, and video URL. Linked to their parent section.
Ratings	Stores ratings and reviews provided by users for each course. Contains user ID, course ID, rating value, and review content.
Categories	Defines different categories to organize courses (e.g., Web Development, Data Science). Each course belongs to a category.
OTPs	Stores OTPs (One-Time Passwords) for secure login, signup, and password reset workflows. Includes email and OTP expiration timestamp.
CourseProgress	Tracks the progress of a student in a specific course, storing completed sections/subsections, user ID, and course ID.

Table 2. Database collections

Each collection uses appropriate indexing and references to ensure faster lookups and data consistency.

3.4 Interface Design

The user interface is built using **React.js** and styled with **Tailwind CSS**, ensuring responsiveness and modern design principles.

Key Interface Components:

- **Home Page:** Highlights featured courses, platform benefits, and call-to-action buttons.
- **About Page:** Displays static content about the platform.
- **Contact Page:** Includes a form with validation and toast notifications on submission.
- **Student Dashboard:**
 - *My Profile:* Edit personal info.
 - *Settings:* Change password, profile image, delete account.
 - *Logout:* Session termination.
- **Instructor Dashboard:**
 - *My Courses:* View created courses.
 - *Add Course:* Form for title, description, price, tags, thumbnail, etc.
 - *Course Sections:* Upload content and videos.

The design follows usability principles with consistent navigation, clear call-to-action buttons, and error handling for better user experience.

4. Implementation

4.1 Modules Description

The EduElevate project is divided into several interconnected modules that handle specific functionality:

1. Authentication Module

- Handles user signup/login for students and instructors.
- OTP verification via email using **Nodemailer**.
- Passwords are securely hashed using **bcrypt.js**.
- JWT-based token system is used for user session management.

2. User Profile Module

- Allows users to update personal information like name, gender, bio, date of birth, and profile image.
- Profile pictures are uploaded and managed using **Cloudinary**.

3. Course Management Module

- Instructors can create, update, and manage courses.
- Each course consists of multiple sections and subsections (videos, descriptions).
- Course thumbnails and video uploads are handled via **Cloudinary**.

4. Enrolment and Progress Tracking Module

- Students can enroll in courses after successful payment.
- The course progress is tracked and stored in the CourseProgress collection.

5. Payment Integration Module

- Uses **Razorpay** for secure and fast online payments.
- Generates Razorpay orders, verifies them on the backend, and enrolls students on success.

6. Ratings and Reviews Module

- Allows students to post reviews and rate the courses.
- Each rating is associated with both the course and the user for traceability.

7. Frontend Module

- Built using **React.js** and styled with **Tailwind CSS**.
- Uses **React Router** for navigation and **Axios** for communicating with the backend APIs.

8. Mobile App (React Native)

- A basic Android application was created using **React Native**.
- It currently displays static course content as a prototype.

4.2 Development Tools

Tool/Technology	Purpose
React.js	Frontend development
Tailwind CSS	Styling and responsive UI
Node.js	Backend runtime
Express.js	Web server and routing
MongoDB	NoSQL database
Mongoose	MongoDB modeling
Nodemailer	Sending emails with OTP
bcrypt.js	Password encryption
JWT (jsonwebtoken)	Secure authentication
React Router	Client-side routing
Axios	API communication
React Native	Mobile app (prototype)
Postman	API testing and debugging
Razorpay	Payment gateway integration

Tool/Technology	Purpose
Cloudinary	File and image hosting

Table 3. Development Tools

4.3 Implementation Steps

1. Setup Project Structure

- Initialized Git repositories for both frontend and backend.
- Installed required dependencies using npm.

2. Database Design and Connection

- Designed MongoDB schema for Users, Profiles, Courses, OTPs, etc.
- Connected backend to MongoDB using Mongoose.

3. Implemented Authentication

- Developed routes and logic for registration, login, OTP verification.
- Secured user sessions using JWT tokens.

4. Built Course Management Functionality

- Instructors can add/edit courses, sections, and subsections.
- File uploads (thumbnails, videos) managed using Cloudinary.

5. Integrated Razorpay Payments

- Backend generates Razorpay orders.
- On payment success, enrollment and confirmation handled via webhooks and API.

6. Frontend Development

- Designed responsive UI using Tailwind CSS.
- Used Axios for making backend API calls and managing user data.

7. Implemented Reviews and Ratings

- Students can rate and review courses.
- Ratings are used to calculate course popularity.

8. Built Mobile App (Prototype)

- Created a basic Android app using React Native.
- Displays sample course content with static data.

4.4 Screenshots and Explanation

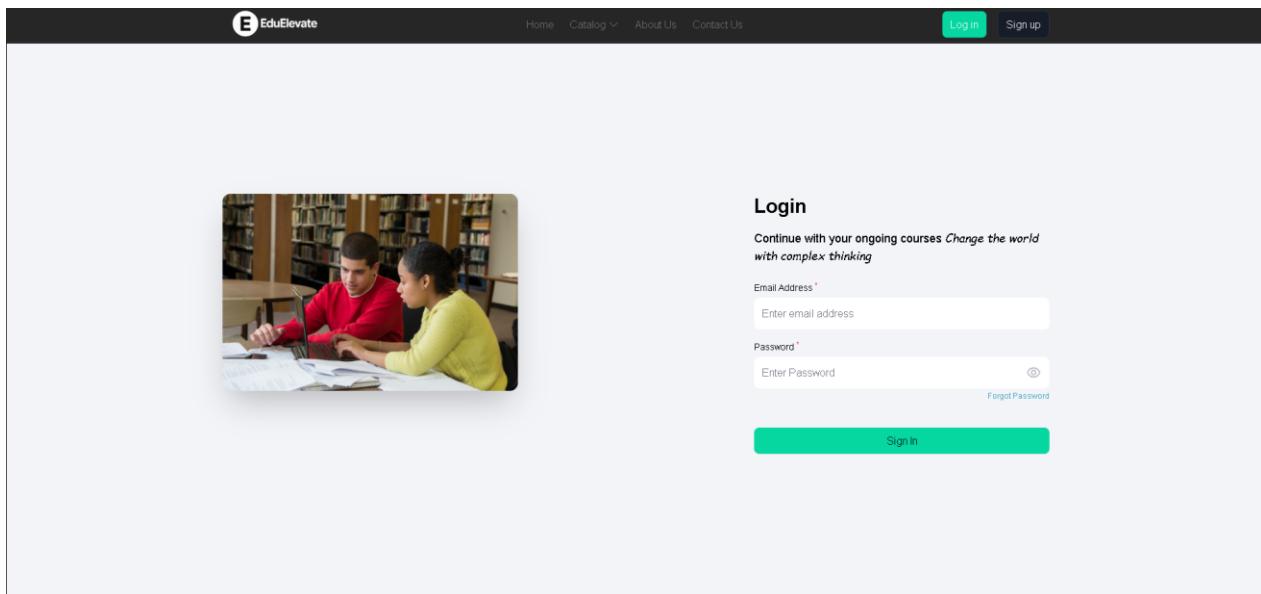


Figure 1. Login Page (Student/Instructor)

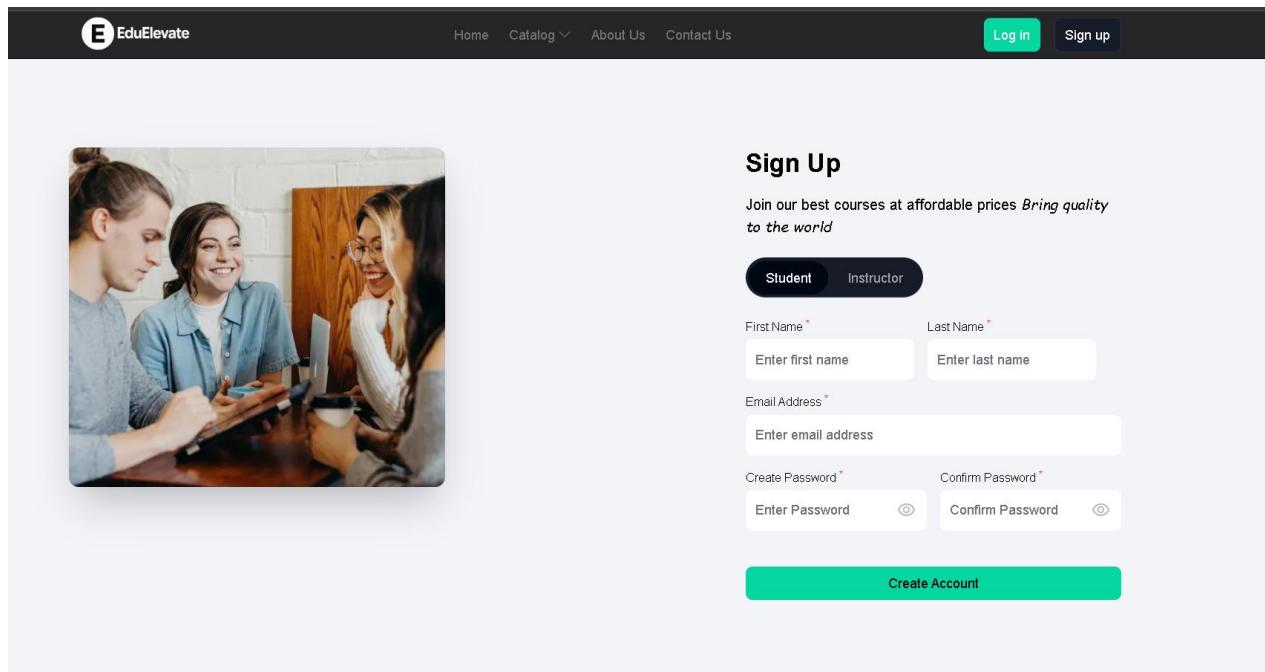


Figure 2. Signup Page (Student/Instructor)

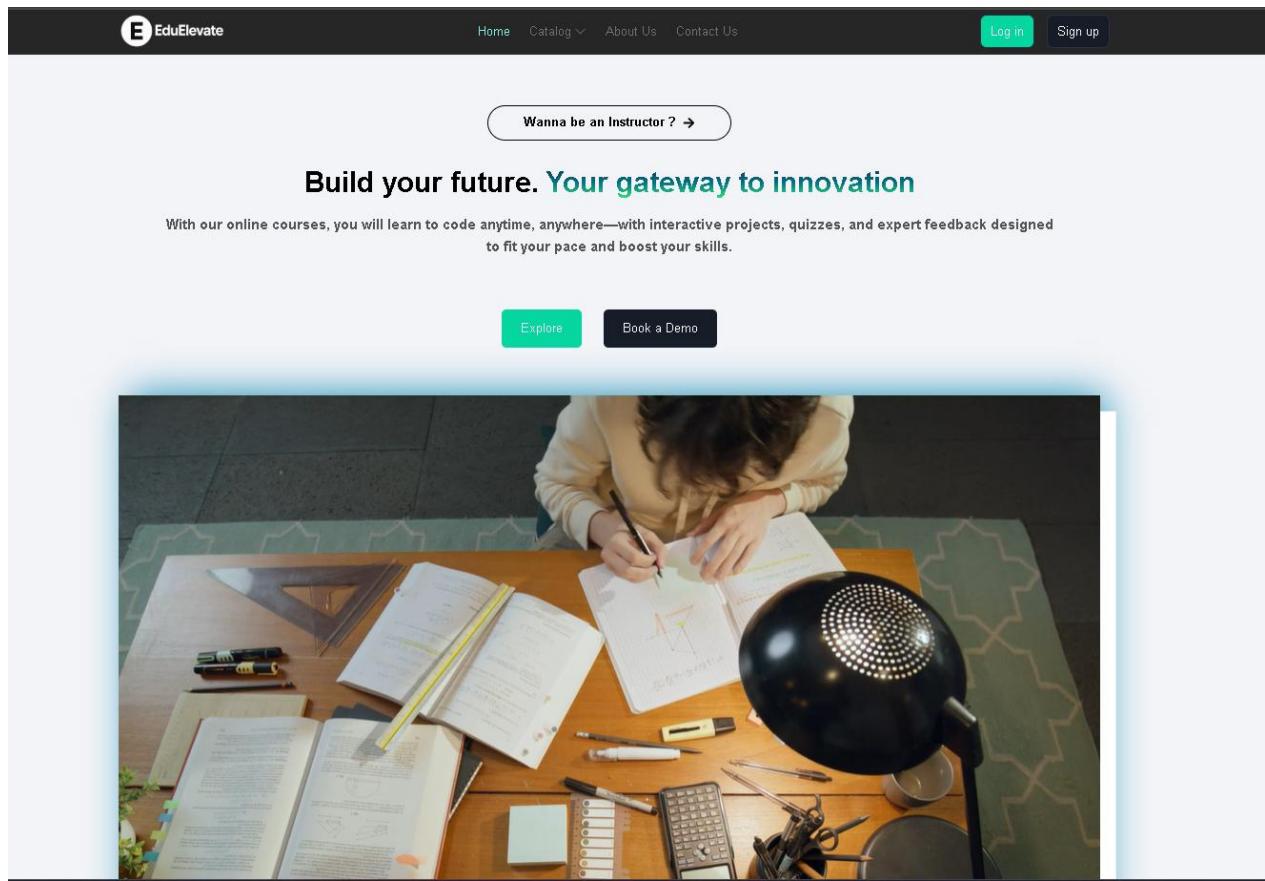


Figure 3. Home page (i)

The screenshot displays a section of a learning platform's website. On the left, there is a vertical stack of four circular icons, each associated with a mission statement:

- Quality**: Giving quality education is our responsibility.
- Leadership**: Leadership qualities for company ready.
- Accessibility**: Content available anytime, provide flexibility.
- Solve the problem**: Different paths to solve the problem.

To the right of these statements is a photograph of a woman sitting at a desk, working on a laptop and writing in a notebook, with a cup of coffee nearby.

Below this section is a header with the text "Learn like a Pro" and the tagline "Your Brain Will Thank You Later".

Below the header is a navigation bar with five items: "Free", "New to coding", "Most popular", "Skills paths", and "Career paths".

The main content area features three course cards, each with a title, a brief description, and two status indicators at the bottom:

- Learn HTML**: This course covers the basic concepts of HTML including creating and structuring web pages, adding text, links, images, and more. Status: Beginner, 6 Lesson.
- Learn CSS**: This course explores advanced topics in HTML5 and CSS3, including animations, transitions, and layout techniques. Status: Beginner, 6 Lesson.
- Responsive Web design**: This course teaches responsive web design techniques, allowing web pages to adapt to different devices and screen sizes. Status: Beginner, 6 Lesson.

Figure 4. Home page (ii)

Get the skills you need for a job that is in demand.

The modern EduElevate is the dictates its own terms. Today, to be a competitive specialist requires more than professional skills.

[Learn More](#)

Transform your Future with Expert-Led Learning

Learn from the best—our instructors are seasoned developers who live and breathe to share knowledge, and they're here to guide you every step of the way.

[Give it a try →](#)

[Get Details](#)

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <title>This is myPage</title>
5 </head>
6 <body>
7 <h1><a href
8
9
10
11
```



Become an instructor

Instructors from around the world teach millions of students on EduElevate. We provide the tools and skills to teach what you love.

[Start Teaching Today →](#)

Figure 5. Home page (iii)

Resources	Plans	Subjects	Languages	Career building
Articles	Paid memberships	AI	Bash	Career paths
Blog	For students	Cloud Computing	C++	Career services
Chart Sheet	Business solutions	Code Foundations	C#	Interview prep
Code challenges		Computer Science	Go	Professional certification
Docs	Forums	Cybersecurity	HTML & CSS	-
Projects	Chapters	Data Analytics	Java	Full Catalog
Videos	Events	Data Science	JavaScript	Beta Content
Workspaces		Data Visualization	Kotlin	
Affiliates		Developer Tools	PHP	
Company		DevOps	Python	
About		Game Development	R	
Careers		IT	Ruby	
Affiliates		Machine Learning	SQL	
Support		Math	Swift	
Help center		Mobile Development		
		Web Design		
		Web Development		

Privacy Policy | Cookie Policy | Terms

Made with ❤️ Manan © 2025 EduElevate

Figure 6. Footer

Driving Innovation in Online Education for a Brighter Future

EduElevate is at the forefront of driving innovation in online education. We're passionate about creating a brighter future by offering cutting-edge courses, leveraging emerging technologies, and nurturing a vibrant learning community.

We are passionate about revolutionizing the way we learn. Our innovative platform **combines technology, expertise, and community to create an unparalleled educational experience.**

5K
10+
200+
50+

Figure 7. Aboutus page (i)

5K
Active Students

10+
Mentors

200+
Courses

50+
Awards

World-Class Learning for Anyone, Anywhere

EduElevate partners with more than 275+ leading universities and companies to bring flexible, affordable, job-relevant online learning to individuals and organizations worldwide.

[Learn More](#)

Curriculum Based on Industry Needs

Save time and money! The Belajar curriculum is made to be easier to understand and in line with industry needs.

[Learn More](#)

Our Learning Methods

EduElevate partners with more than 275+ leading universities and companies to bring

Certification

EduElevate partners with more than 275+ leading universities and companies to bring

Rating "Auto-grading"

EduElevate partners with more than 275+ leading universities and companies to bring

Ready to Work

EduElevate partners with more than 275+ leading universities and companies to bring

Figure 8. Aboutus page (ii)

Figure 9. Contactus page

Figure 10. User profile page after Login

The screenshot shows the course details page for 'MERN Stack' on EduElevate. At the top, there's a navigation bar with links for Home, Catalog, About Us, Contact Us, a shopping cart icon, and a user profile icon. The main title 'MERN Stack' is displayed in large bold letters. Below it, a brief description states: 'This course will help you to get hand on practice on MERN stack web development.' A rating of 'NaN ★ ★ ★ ★ ★ (1 reviews)' and '0 students enrolled' is shown. The course was created by 'Manan Arora' on May 7, 2025, at 0:18 PM, and is available in English. To the right, there's a promotional image of a hand interacting with a futuristic digital interface, followed by the price 'Rs. 499' and a 'Go To Course' button. A sidebar on the left lists 'What you'll learn' with the subtext 'Deeply learn core concepts, Realtime project creation'. Another sidebar on the right lists 'This Course Includes' with 'MVC Architecture' and a 'Share' button.

Figure 11. Course Details page

The screenshot shows a payment options overlay for the 'Java Programming' course on EduElevate. The overlay has a dark background with a light blue header bar. It displays a 'Price Summary' showing '₹399' and an option to 'Using as +91 70098 71559'. Below this, there are sections for 'Recommended', 'UPI', 'Cards', 'Netbanking', 'Wallet', and 'Pay Later'. Under 'UPI', there's a field to 'Add a new card' with fields for 'Card Number', 'MM / YY', and 'CVV'. A checkbox 'Save this card as per RBI guidelines' is present. At the bottom right of the overlay is a 'Continue' button. The background of the page shows the course details for 'Java Programming', including its description, rating, creator, and course content sections.

Figure 12. Razorpay Payment Gateway

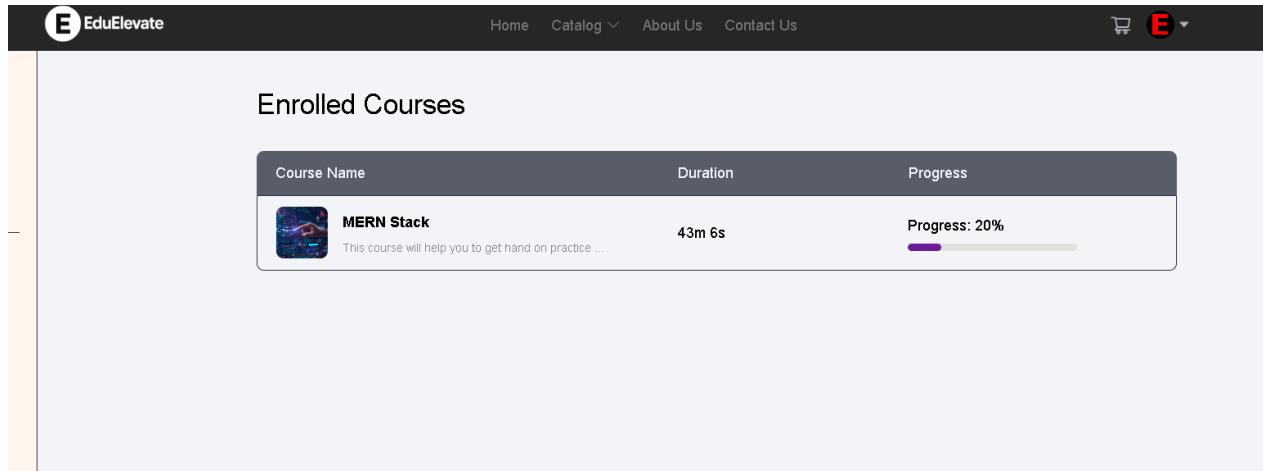


Figure 13. Enrolled Courses after Payment

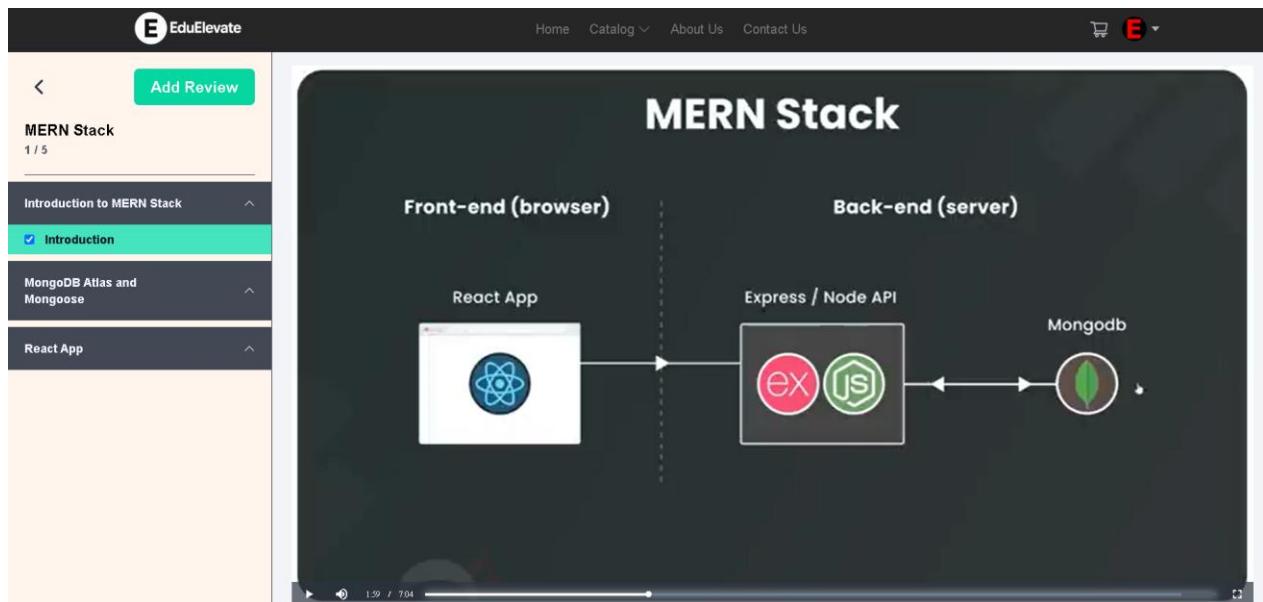


Figure 14. Video Lectures after enrolment

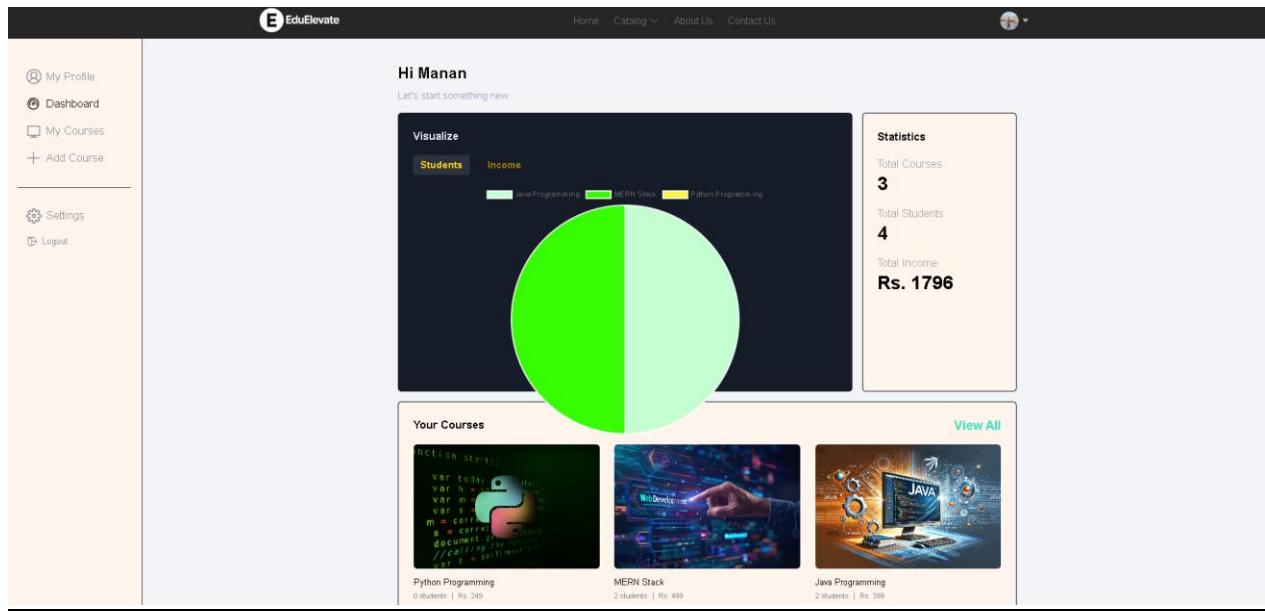


Figure 15. Instructor Dashboard

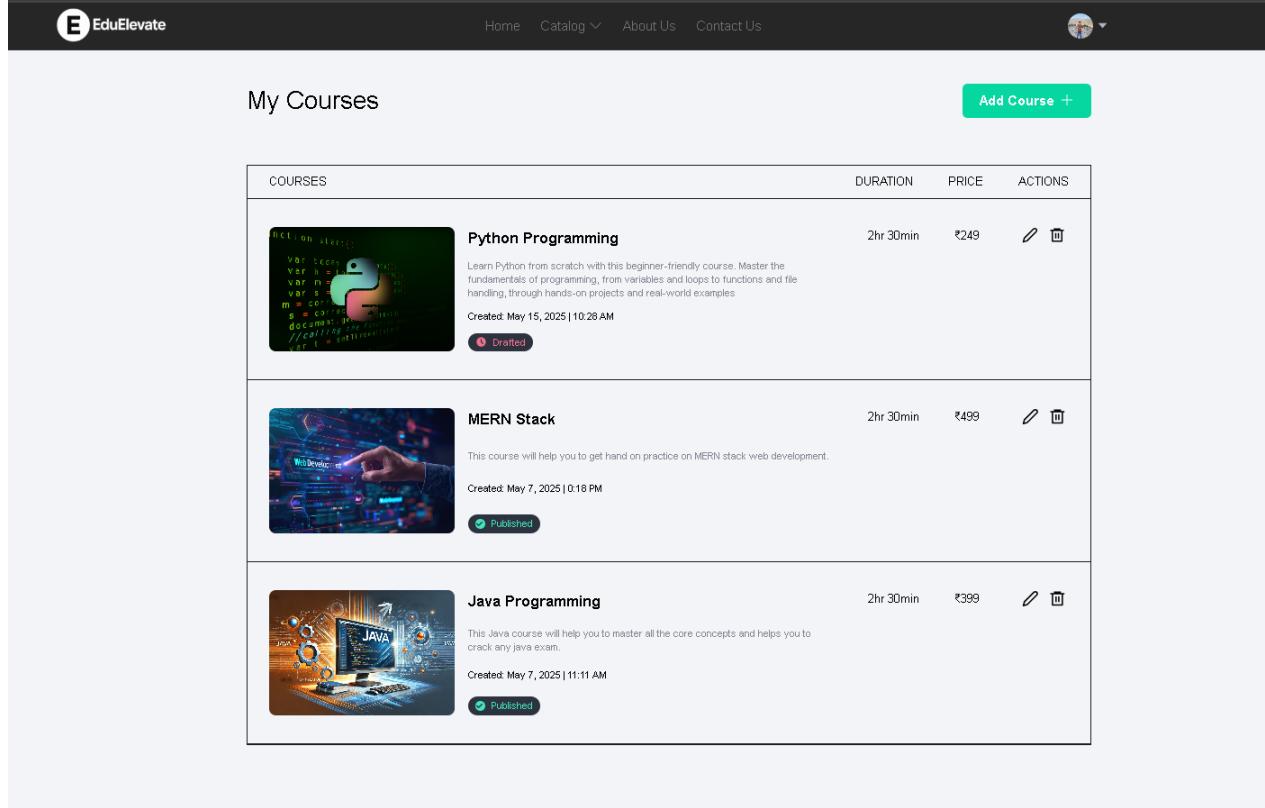


Figure 16. Courses List created by Instructor

The screenshot shows the 'Add Courses' section of the EduElevate platform. The interface is divided into several input fields:

- Course Title ***: A text input field with placeholder text "Enter Course Title".
- Course Short Description ***: A text input field with placeholder text "Enter Description".
- Course Price ***: A text input field with placeholder text "Enter Course Price".
- Course Category ***: A dropdown menu with placeholder text "Choose a Category".
- Tags ***: A text input field with placeholder text "Enter Tags and press Enter".
- Course Thumbnail ***: A large input field with a placeholder "Drag and drop an image, or click to **Browse** a file". It includes a circular "X" icon and notes about aspect ratio (16:9) and file size (1024x576).
- Benefits of the course ***: A text input field with placeholder text "Enter benefits of the course".
- Requirements/Instructions ***: A text input field with placeholder text "Enter Requirements/Instructions".

A green "Next >" button is located at the bottom right of the form.

Figure 17. Add Courses Section (i)

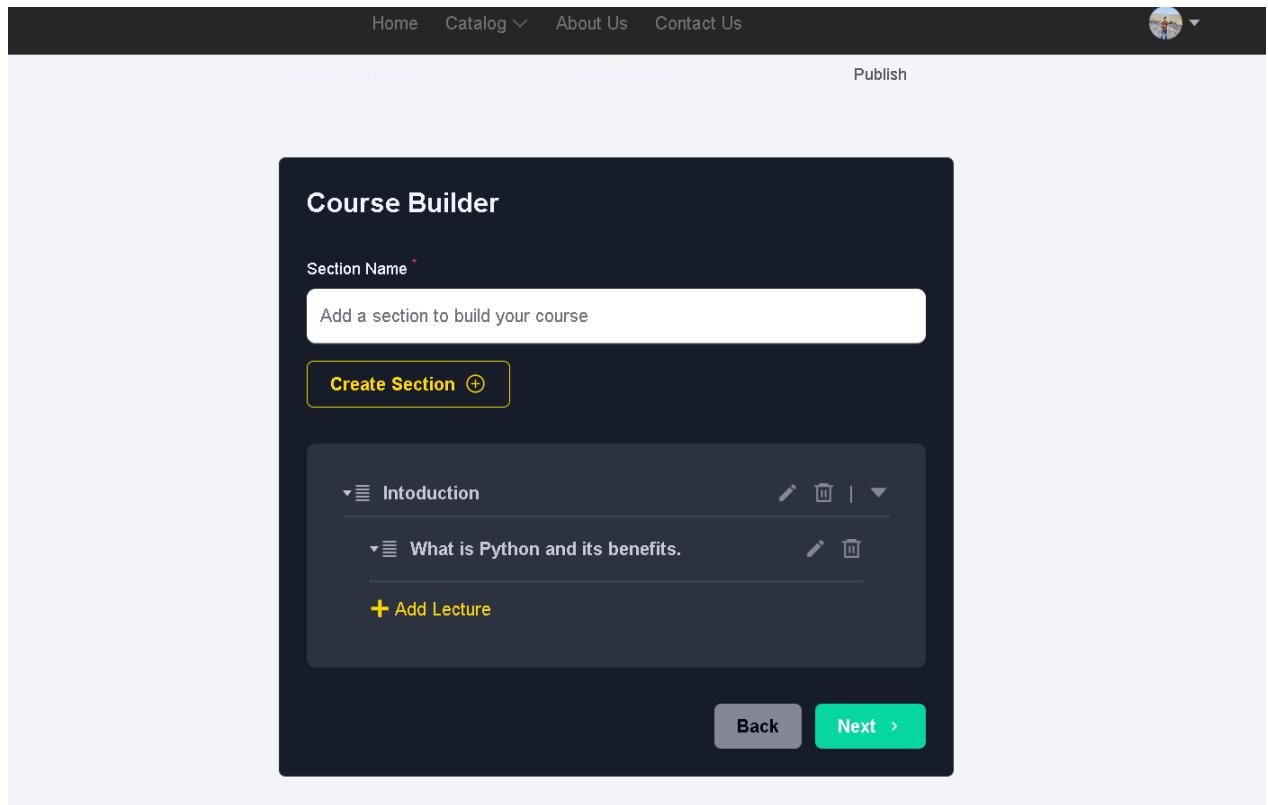


Figure 18. Add Courses Section (ii)



Figure 19. Android App Home Page



About Course

Master Python from scratch with hands-on projects and real-world examples. Learn core concepts like variables, data types, numbers, and casting. Perfect for beginners aiming to build a solid foundation in pr...

Course Content

- 1 Introduction 
- 2 Variables 
- 3 Data Types 
- 4 Loops 
- 5 Casting 



Figure 20. Course details Page

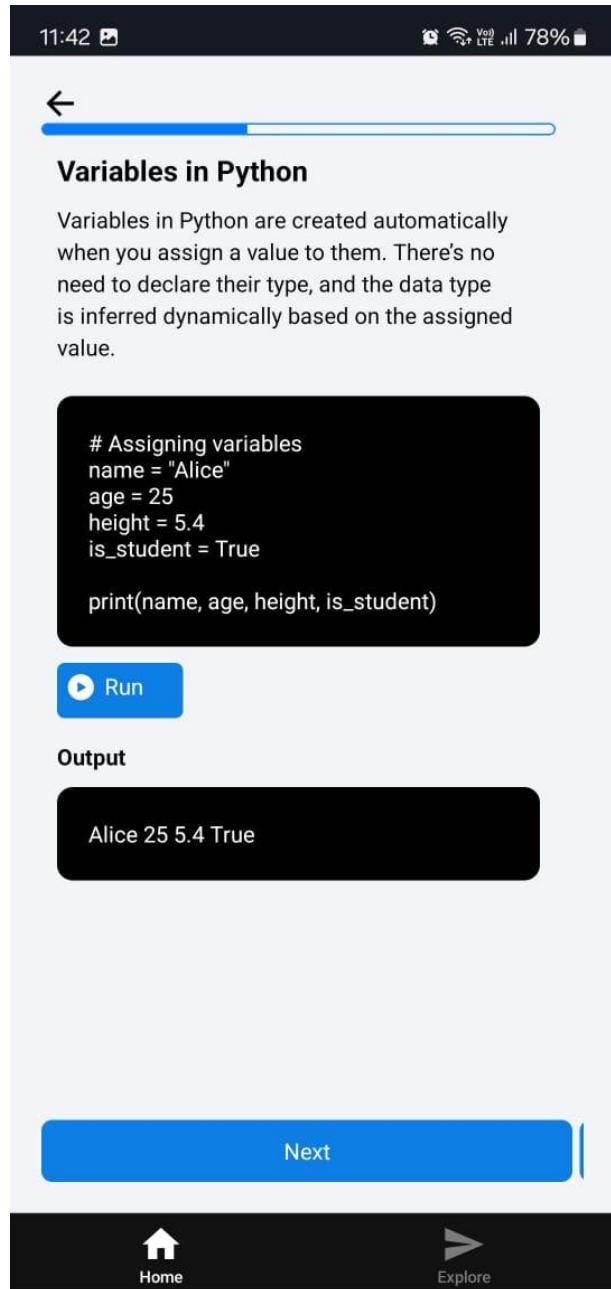


Figure 21. Couse Content Page

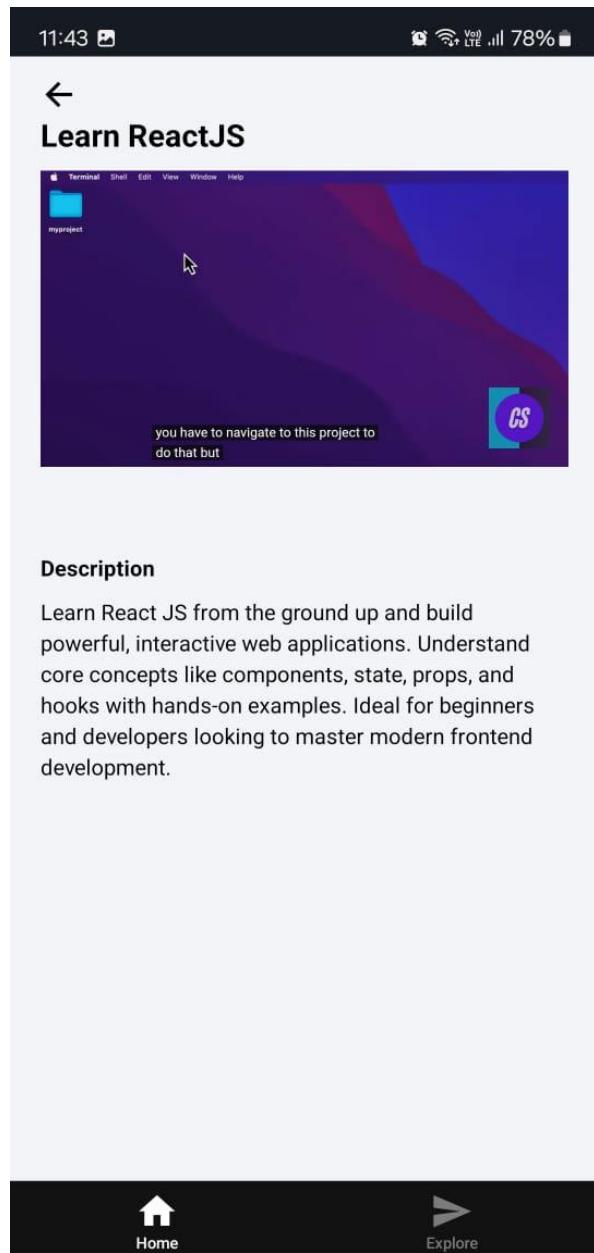


Figure 22. Video Lecture Page

5. Testing and Results

5.1 Testing Methodology

The EduElevate platform was tested using both **manual** and **automated** testing approaches to ensure functionality, reliability, and security. The testing phases included:

- **Unit Testing:** To verify the correctness of individual functions and components (e.g., user authentication, payment processing).
- **Integration Testing:** To ensure different modules (e.g., frontend ↔ backend ↔ database) work together as expected.
- **End-to-End (E2E) Testing:** To simulate real-user scenarios like course enrollment and content viewing.
- **UI/UX Testing:** To validate layout responsiveness and user interaction flow on multiple devices.
- **API Testing:** Postman was used extensively to test all backend routes including edge cases, errors, and validations.
- **Cross-Browser Testing:** To check compatibility on Chrome, Firefox, and Microsoft Edge.

5.2 Test Cases and Results

Test Case	Description	Expected Result	Actual Result	Status
TC001	Student Signup with Email & OTP	Account should be created successfully	As Expected	Pass
TC002	Instructor Course Creation	Course should be saved with all details	As Expected	Pass
TC003	Razorpay Payment Process	Payment should succeed and user enrolled	As Expected	Pass
TC004	Rating & Review Submission	Review should be stored and displayed	As Expected	Pass

Test Case	Description	Expected Result	Actual Result	Status
TC005	Unauthorized API Access	Should return 401 Unauthorized	As Expected	<input checked="" type="checkbox"/> Pass
TC006	File Upload with Cloudinary	File should upload and return URL	As Expected	<input checked="" type="checkbox"/> Pass
TC007	Video Lecture Playback	Should stream without buffering	As Expected	<input checked="" type="checkbox"/> Pass
TC008	Mobile App Navigation	Static content should be viewable	As Expected	<input checked="" type="checkbox"/> Pass

Table 4. Test Cases and Results

All test cases passed successfully. The platform showed stability and accuracy across modules.

5.3 Performance Analysis

- API Response Time:** Most API calls returned within 150–300 ms under normal load.
- Page Load Speed:** Home and Dashboard pages loaded within 1–2 seconds on average.
- Razorpay Integration:** Payments were completed and verified within 3–5 seconds.
- Database Operations:** CRUD operations with MongoDB were optimized using indexed fields, ensuring fast read/write.

Overall, the application performed efficiently without any major bottlenecks.

5.4 Final Output

The final outcome of the project met most of the proposed objectives:

- Fully functional student-instructor portal**
- Course creation and management**
- Razorpay payment gateway integration**
- Cloudinary-powered media uploads**
- User profile management and progress tracking**
- Responsive web design with smooth UI/UX**
- React Native demo app with static content**

While the platform does not yet include **Assignments** and **Quizzes**, the backend structure is flexible enough to accommodate these features in future updates.

6. Conclusion and Future Scope

Conclusion

The EduElevate project successfully demonstrates the development of a full-fledged edtech platform using the MERN stack and associated technologies. The platform allows instructors to create and manage courses, while students can explore, purchase, and track their learning progress. It integrates secure authentication, payment processing through Razorpay, and media uploads via Cloudinary. The use of modern tools like React.js, Tailwind CSS, and MongoDB helped ensure responsive design, modular structure, and robust data handling.

The platform was tested thoroughly for functionality, performance, and user experience. A companion mobile application was also developed using React Native to showcase the project's adaptability across platforms (though currently limited to static content).

Through this project, essential concepts of full-stack development, REST API design, third-party service integration, and responsive UI creation were implemented effectively.

Future Scope

There are several enhancements and additional features that can be incorporated in the future to improve the platform:

- **Assignments and Quizzes:** Implementing interactive quizzes and assignment submission features to enhance student engagement and assessment.
- **Mobile App with Full Functionality:** Extending the React Native app to integrate with the backend for dynamic data access and course management.
- **Live Classes Integration:** Adding support for real-time video sessions using tools like Zoom SDK or WebRTC.
- **Discussion Forums:** Creating a community-based forum for doubt resolution and peer learning.
- **Certificates:** Generating auto-certificates upon course completion with verifiable links.
- **Analytics Dashboard:** For both students and instructors to view insights like progress, earnings, and performance metrics.
- **Multi-language Support:** Making the platform accessible to a wider audience by supporting different regional languages.

These additions would not only increase the platform's functionality but also make it more competitive and useful for a broader range of learners and educators.

