

**STUDENT E-LEARNING PORTAL**

**21IT902 – ADVANCED APPLICATION DEVELOPMENT**

***Submitted by***

**KANIGA R A (727722EUAI026)**

***in partial fulfilment for the award of the degree***

***of***

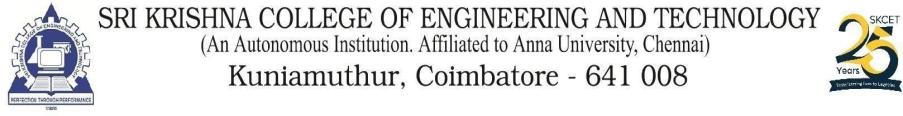
**BACHELOR OF TECHNOLOGY IN**

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

**SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(An Autonomous Institution, Affiliated to Anna University Chennai -600025)**

AUGUST 2024



**BONAFIDE CERTIFICATE**

Certified that this project report titled “STUDENT E-LEARNING PORTAL” is the bonafide work of KANIGA R A (727722EUAI026) who carried out the project work under my supervision.

**SIGNATURE SIGNATURE**

**Dr.VENKATALAKSHMI Mr.SENTHIL KUMAR**

**HEAD OF THE DEPARTMENT SUPERVISOR**

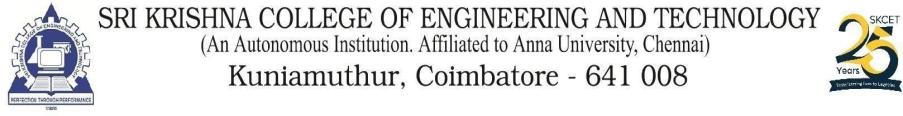
Professor Assistant Professor

Computer Science and Design, Computer Science and Design

|  |
| --- |
| Sri Krishna College of Engineering and Sri Krishna College of Engineering and |
| Technology, Kuniyamuthur, Technology,Kuniyamuthur, |
| Coimbatore-641008 Coimbatore-641008 |
|  |

Submitted for the Project viva-voce examination held on \_\_\_\_\_\_\_\_\_

**INTERNAL EXAMINER EXTERNAL EXAMINER**



# 1.ACKNOWLEDGEMENT

At this juncture, we take the opportunity to convey our sincere thanks and gratitude to the management of the college for providing all the facilities to us.

We wish to convey our gratitude to our college principal**, Dr. S. Sophia** for forwarding us to do our project and offering adequate duration to complete our project.

We would like to express our grateful thanks to **Dr. VenkataLakshmi** Head of the department, Department of Information Technology for her encouragement and valuable guidance to this project.

We extend my gratitude to our beloved guide **Mr S SENTHIL KUMAR, ME.**, **MBA., (PhD).,** Assistant Professor, Department of Information Technology for her constant support and immense help at all stages of the project.

**2.TABLE OF CONTENTS**

##### CHAPTER NO. CONTENTS

|  |  |
| --- | --- |
| 1 | **Abstract** |
| 2 | **List of Tables** |
| 3 | **Introduction** |
| 4 | **Literature Review** |
| 5 | **System Design** |
| 6 | **Testing and Results** |
| 7 | **Conclusion** |
| 8 | **Outputs** |

|  |  |  |
| --- | --- | --- |
| 1 | Detailed System Requirements |  |
| 2 | Functional Requirements Breakdown |  |
| 3 | Frontend Components and Their Responsibilities: |  |
| 4 | Backend API Endpoints |  |
| 5 | Database Schema with Entity-Relationship Details |  |
| 6 | Test Cases and Results |  |
| 7 | Comparative Analysis of Previous Work |  |
|  |  |  |

**3.1INTRODUCTION**

The E-learning portal project aims to create an interactive and user-friendly platform that allows users to access online courses, manage their profiles, make payments, and contact administrators. The project involves building a comprehensive system using Django for the backend and React for the frontend. The Django admin.py configuration is tailored to manage critical administrative functions, such as handling payments, managing teacher profiles, and overseeing user contact messages.

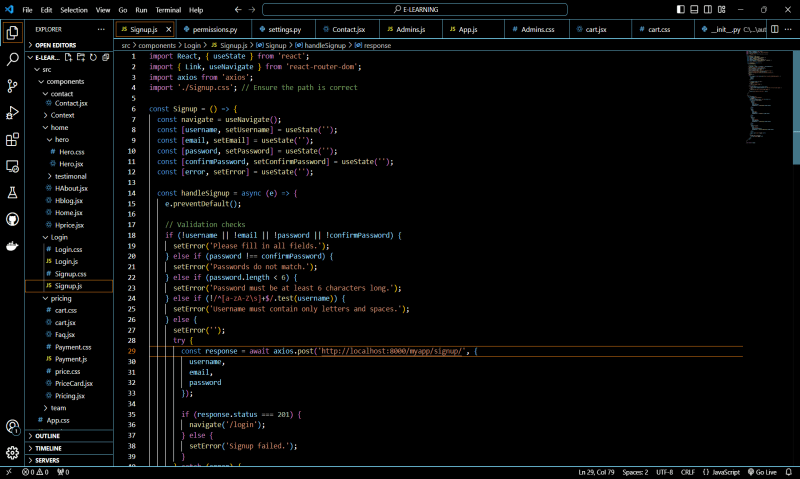
**3.1PROJECT PLAN**

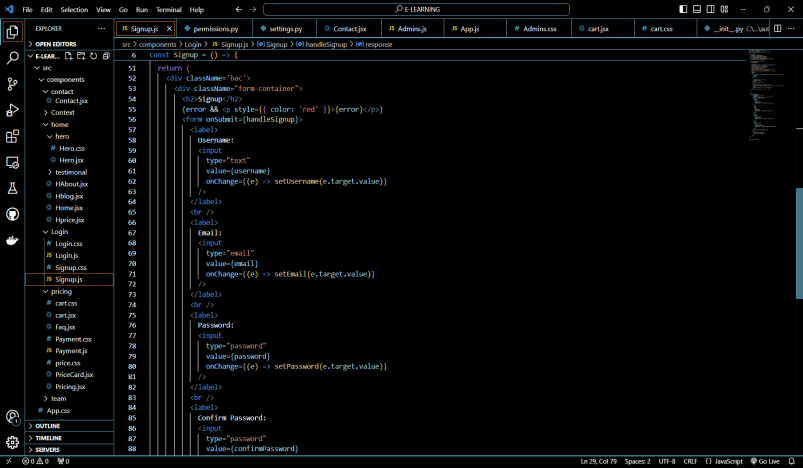
The project is organized into several phases, each focusing on a specific aspect of the system's development. The key components include user authentication, course management, payment processing, and administrative controls.

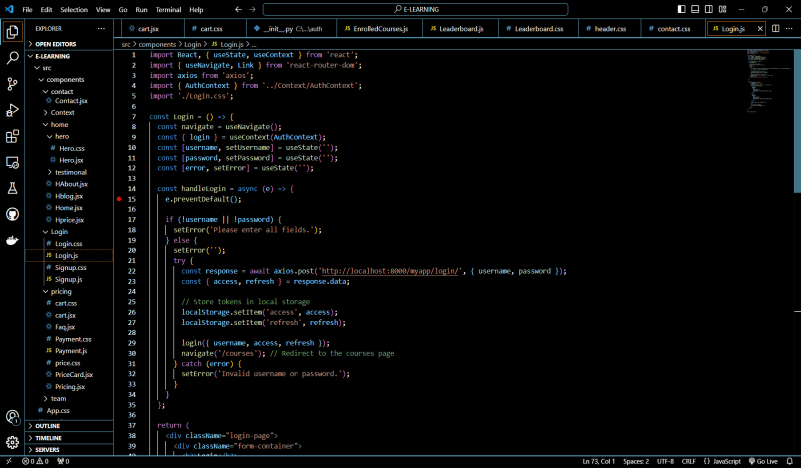
**3.1.1FRONTEND DEVELOPMENT (REACT)**

1. **Creating the Login and Registration Forms**:

Implemented user authentication using React. Components for Login and Signup were developed to handle user input, validation, and interaction with the backend**.**

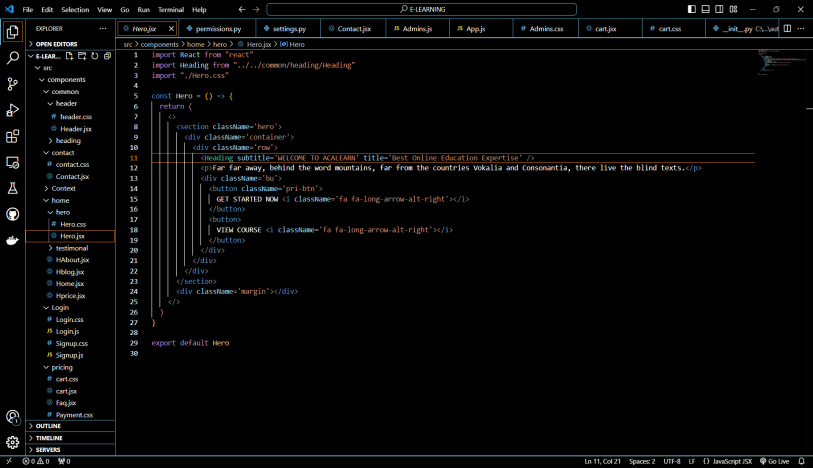
****

****

****

1. **Global State Management and Component Creation:**

For managing user data across the application. Key components include Home, About, CourseHome, Contact, and Payment.



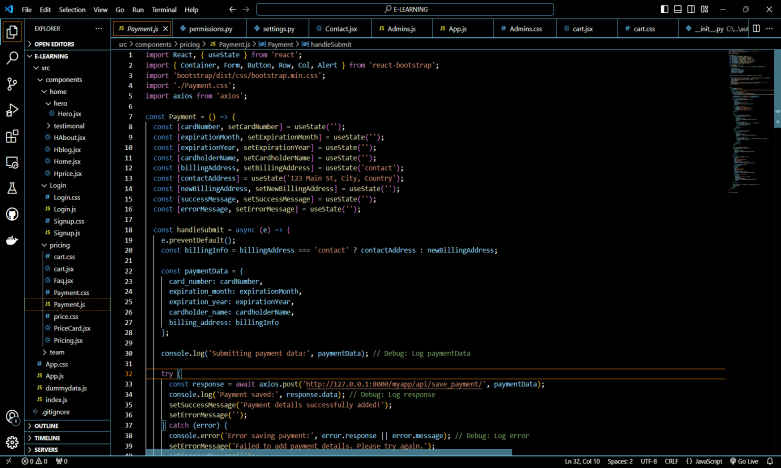
1. **Homepage Component Design:**

The Home component serves as the landing page, featuring a clean and intuitive design. It directs users to various parts of the portal.

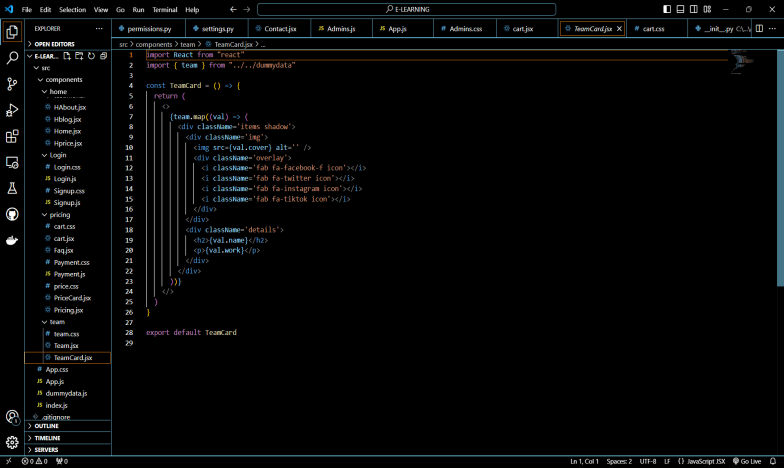
**FRONTEND:**

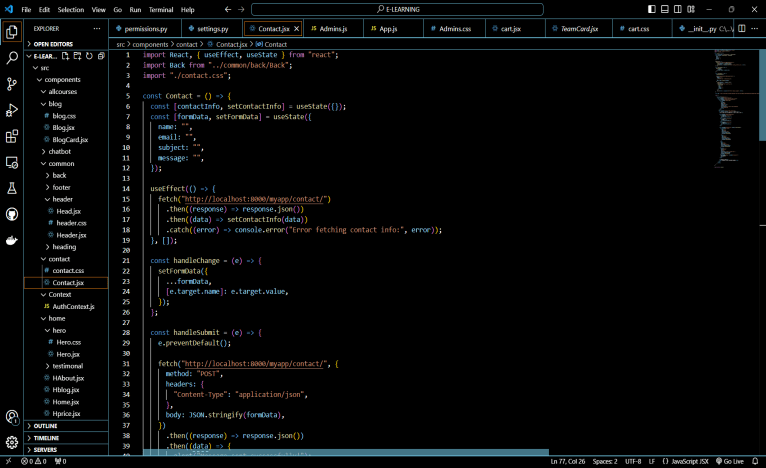
* [All Courses](http://localhost:3000/courses)
* [About](http://localhost:3000/about)
* [Team](http://localhost:3000/team)
* [Pricing](http://localhost:3000/pricing)
* [Journal](http://localhost:3000/journal)
* [Contact](http://localhost:3000/contact)
* Enrolled Courses
* Modules
* Card

**PRICING:**

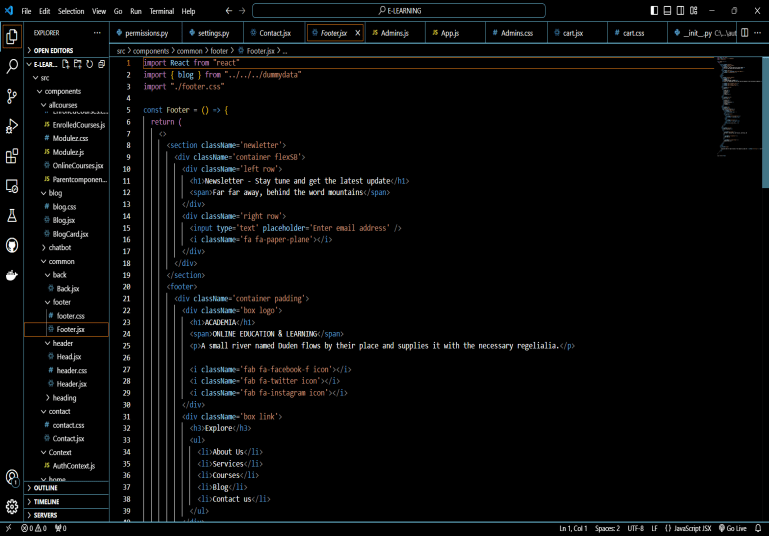


**TEAM:**

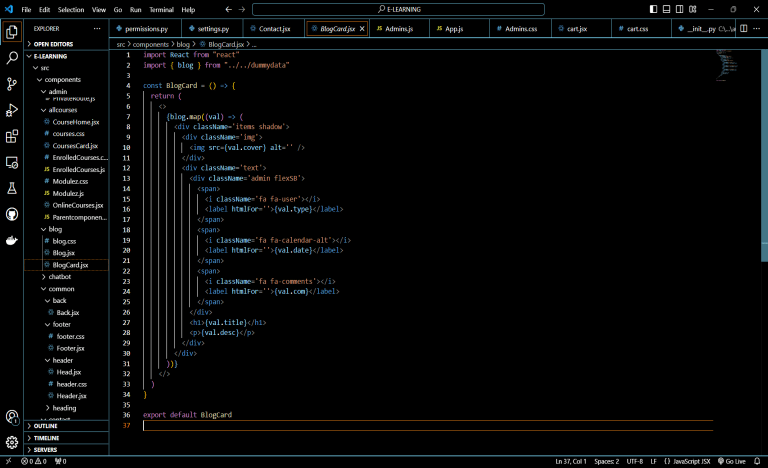


**CONTACT :**

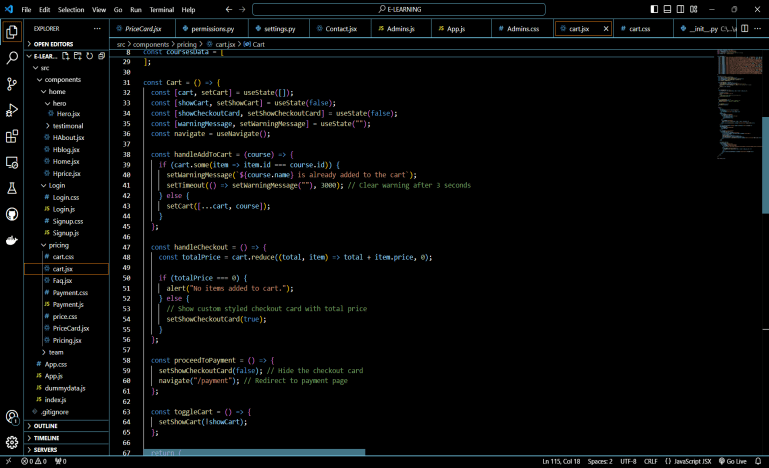
**FOOTER:**



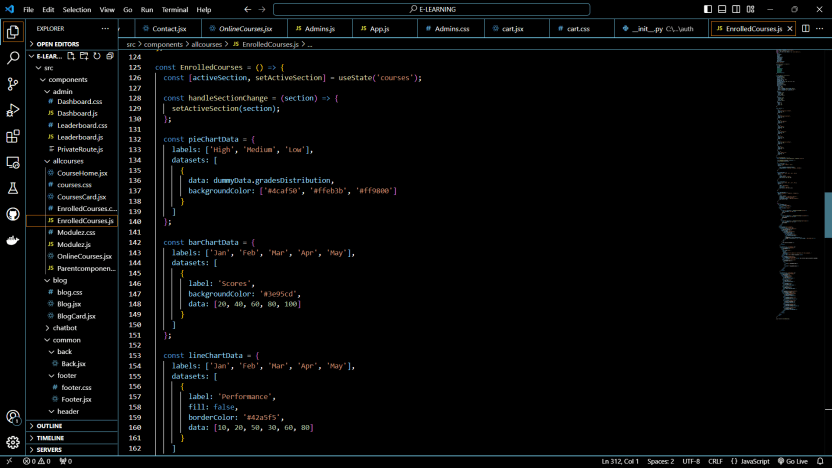
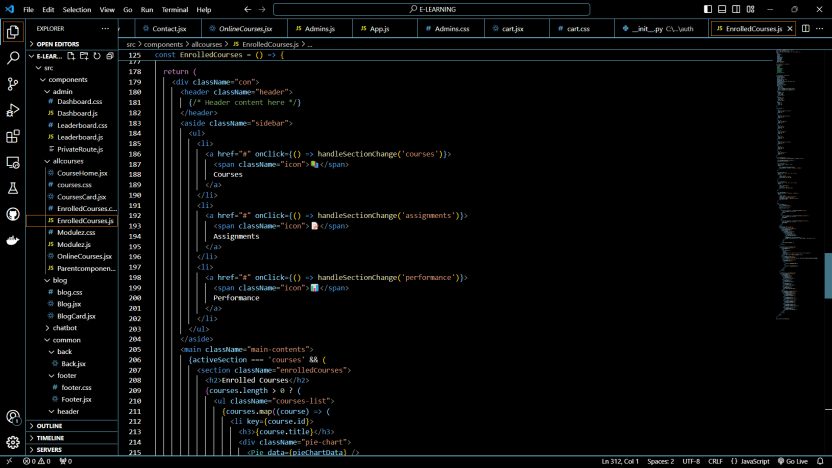
**BLOG:**



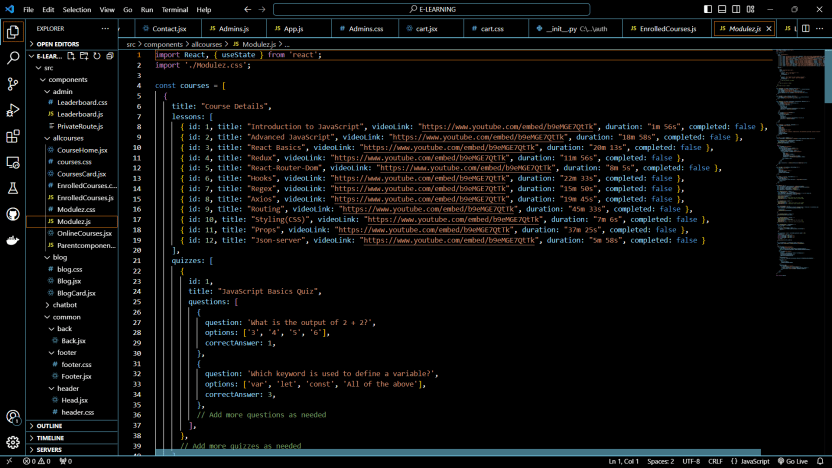
**COURSES :**

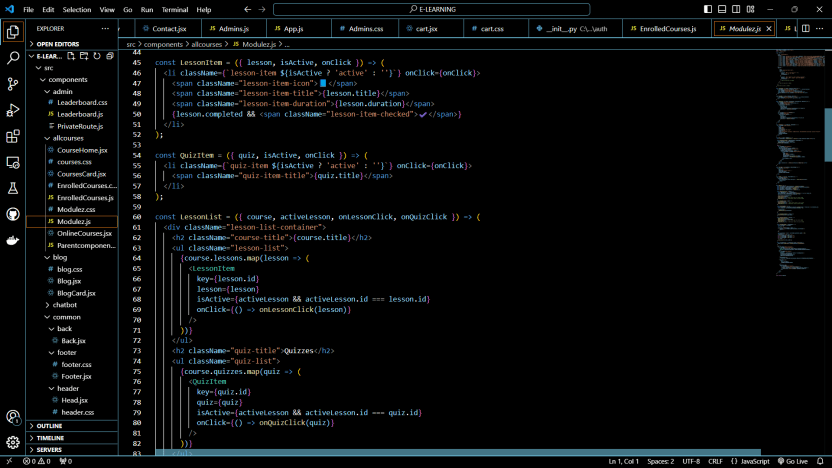


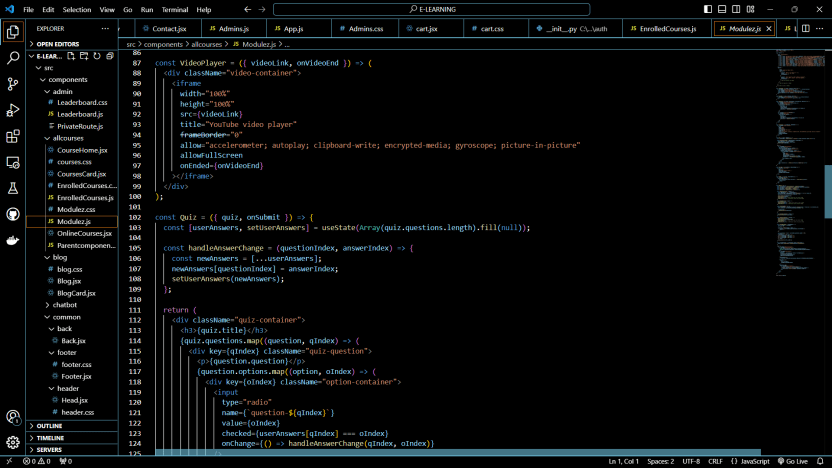
**ENROLLED COURSES:**

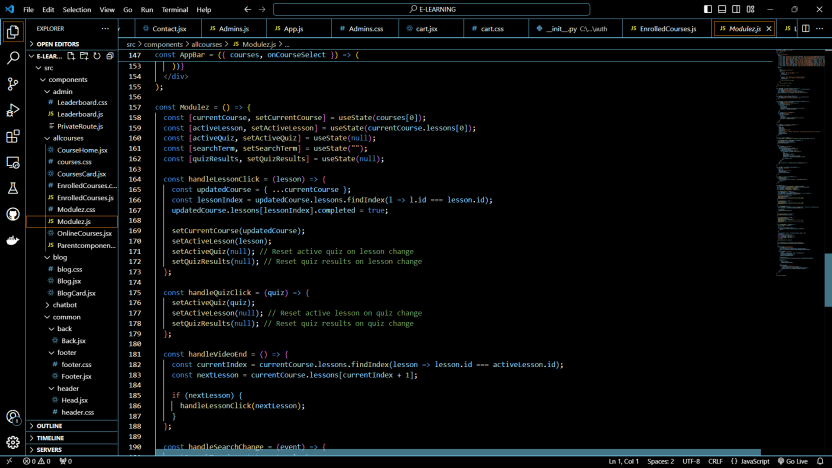


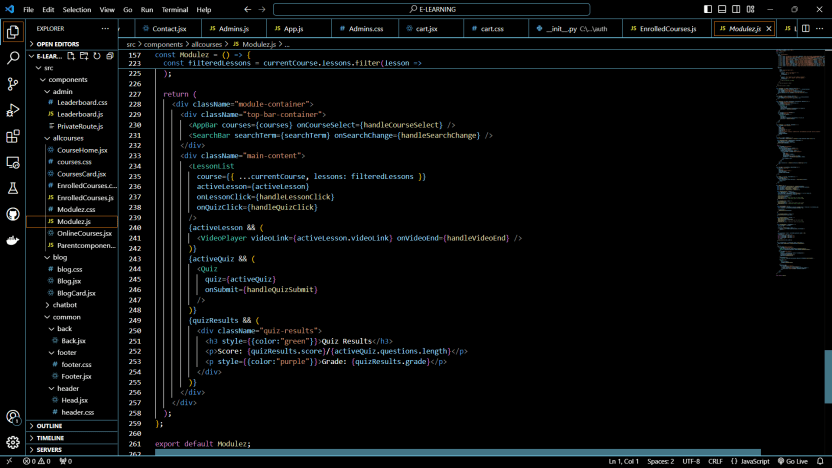
**MODULES:**



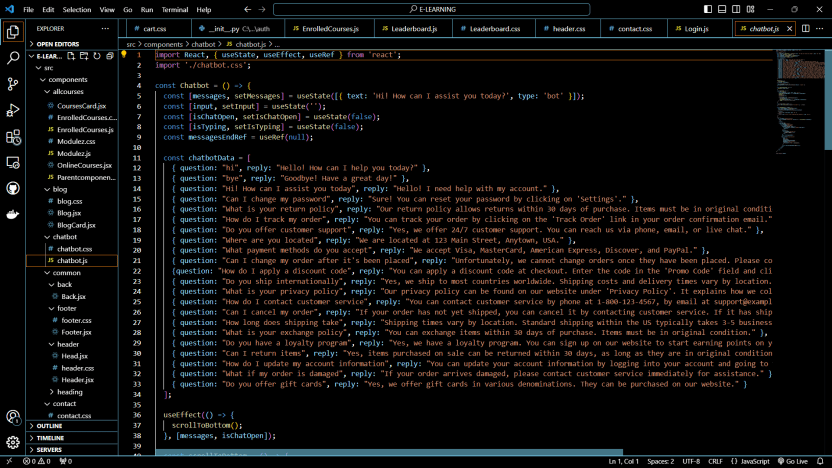


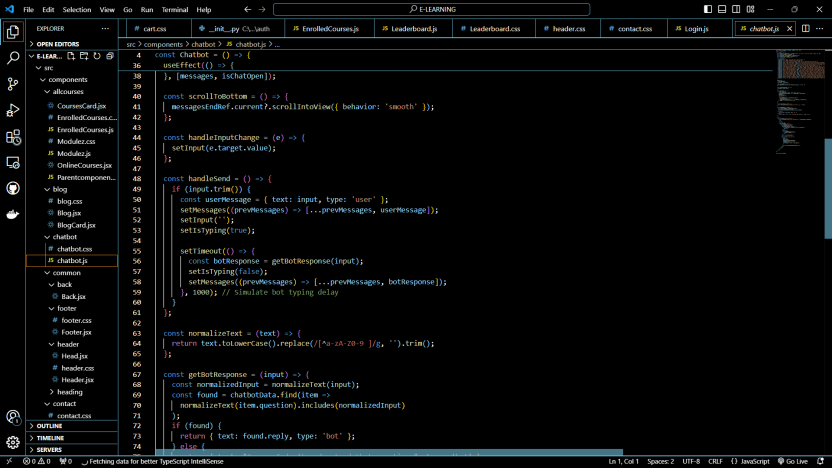


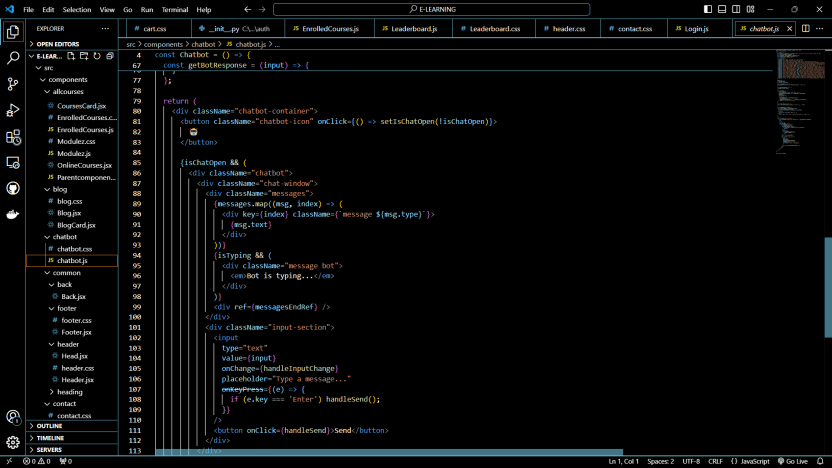


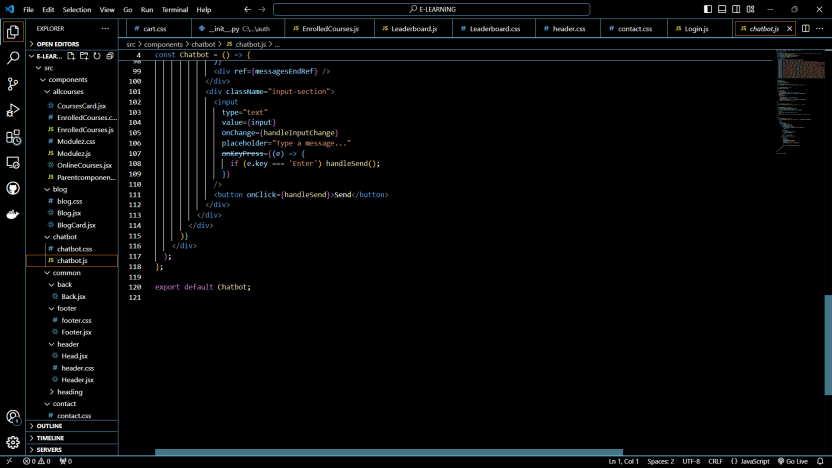


**CHATBOT:**



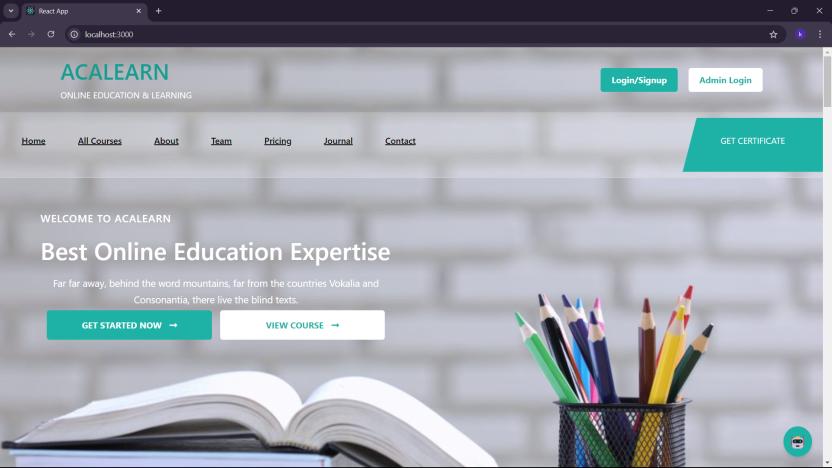


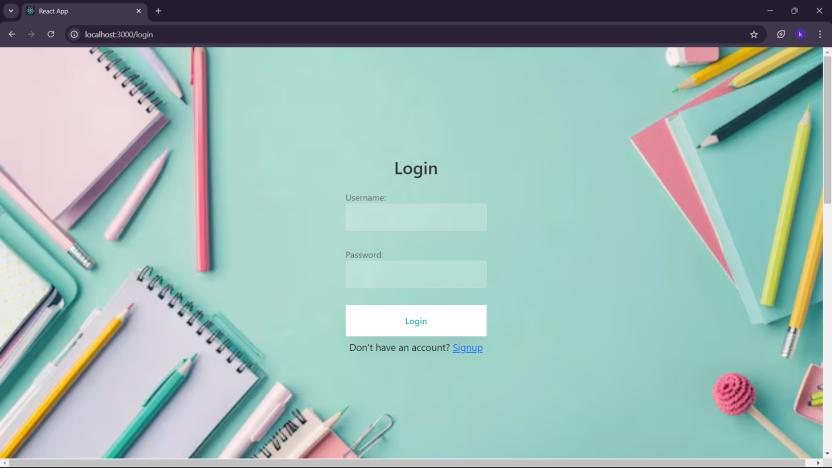


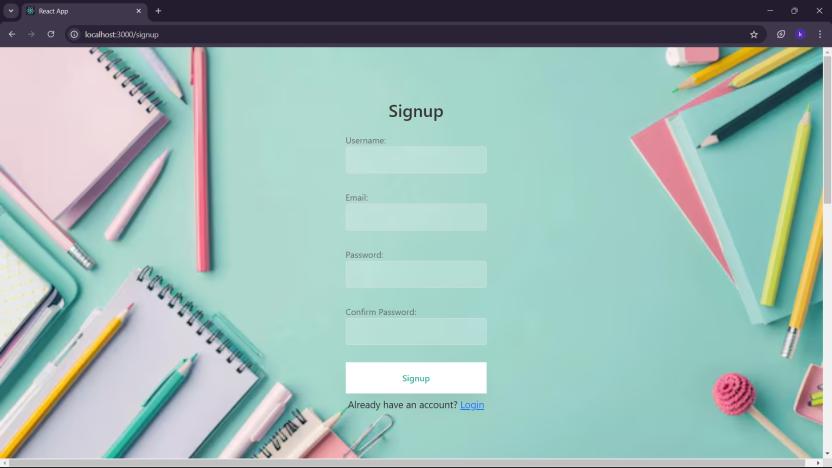


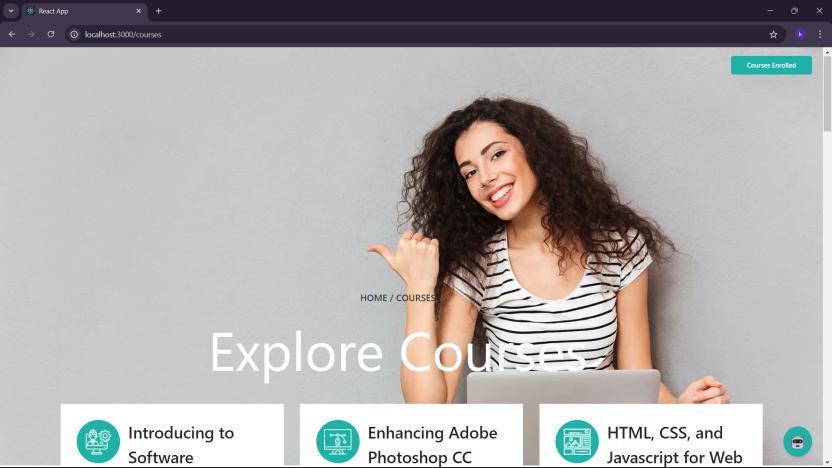
1. **Intergrating Bootstrap:**

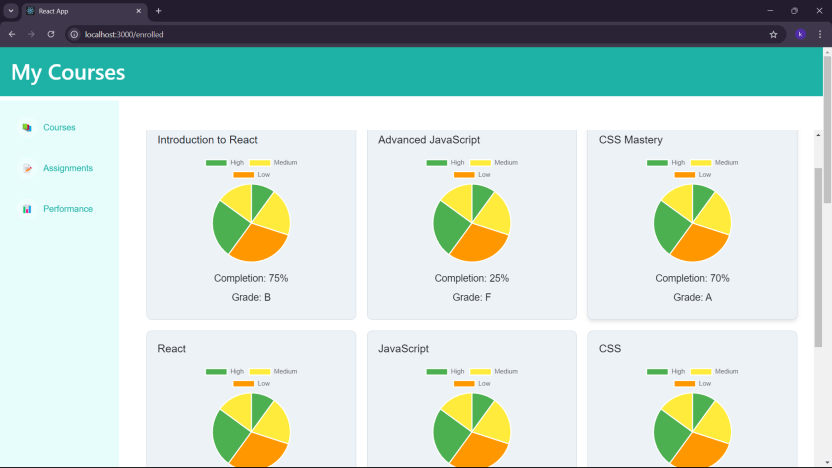
Bootstrap was incorporated to enhance the UI/UX, ensuring a responsive design across all devices.

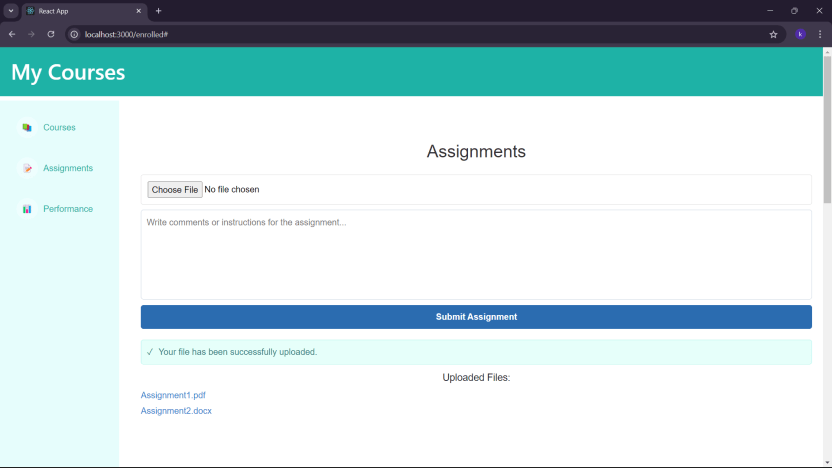


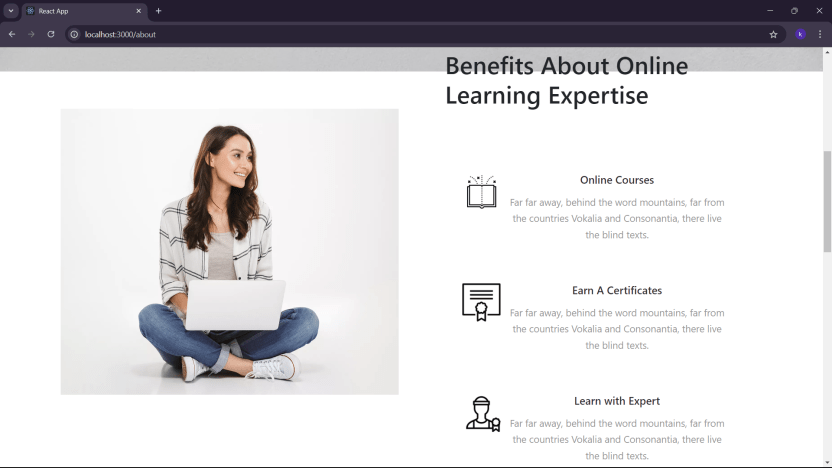


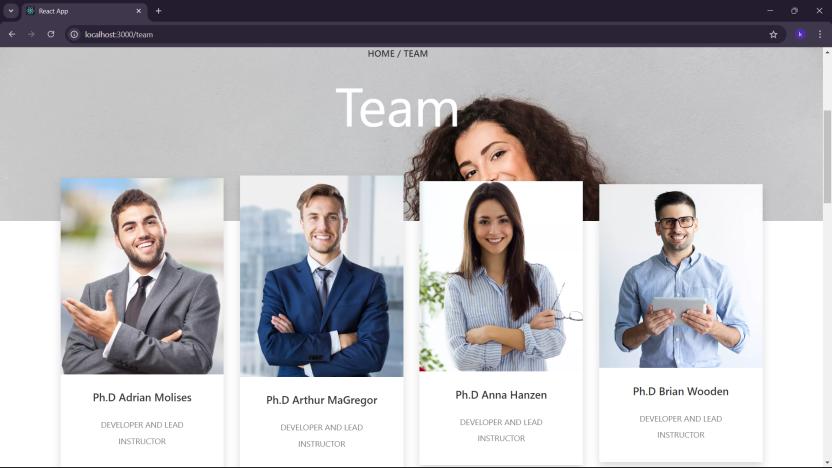


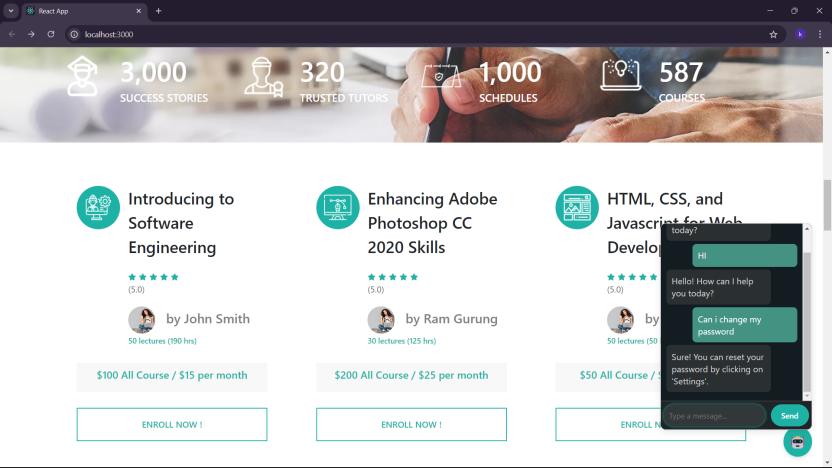


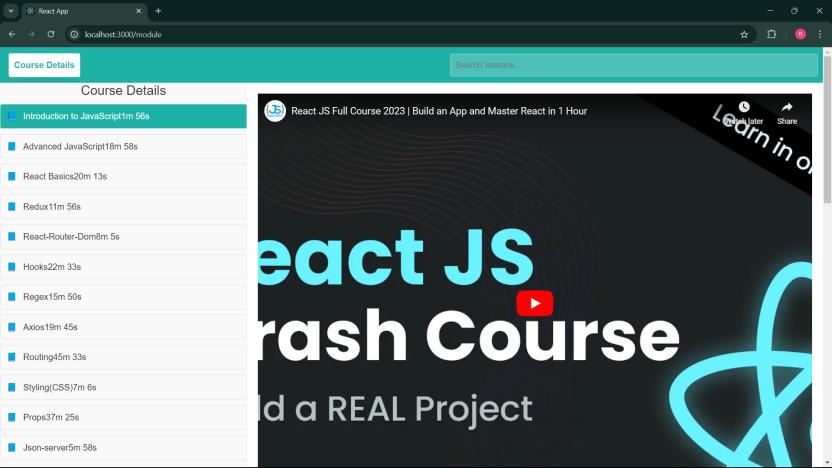


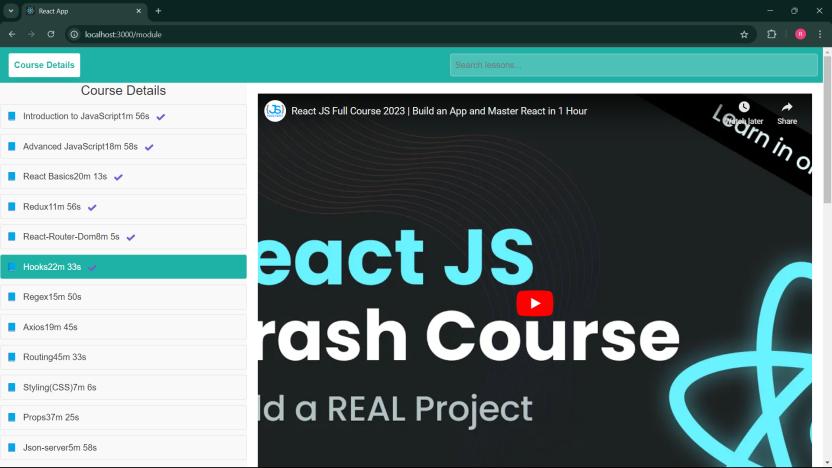


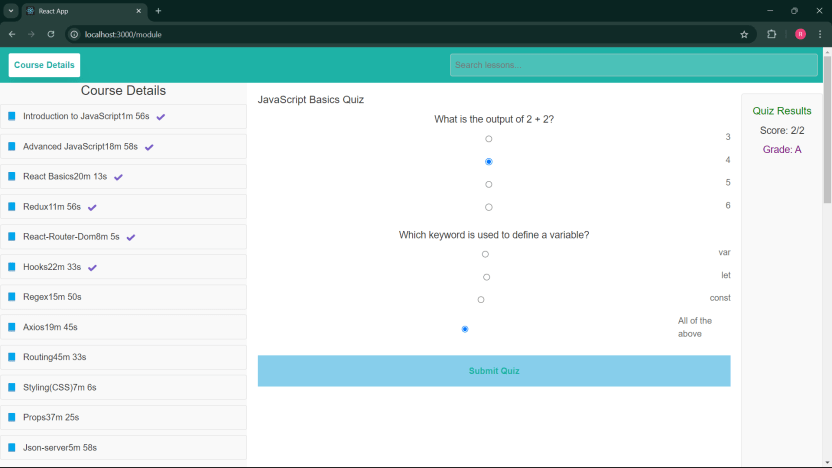


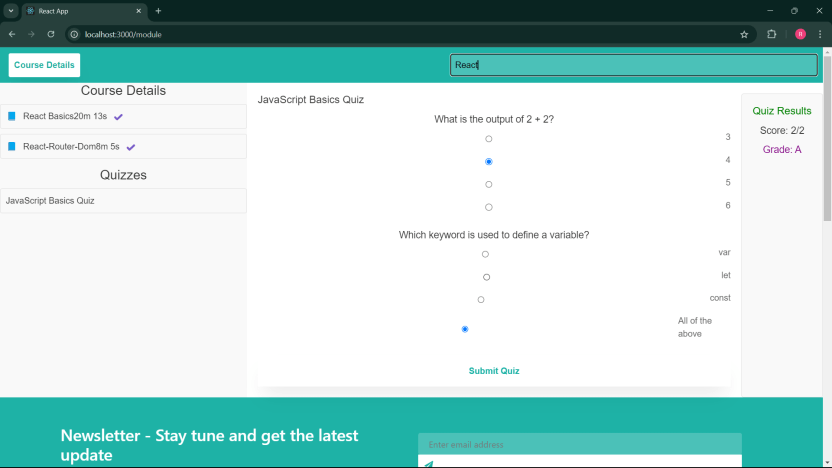


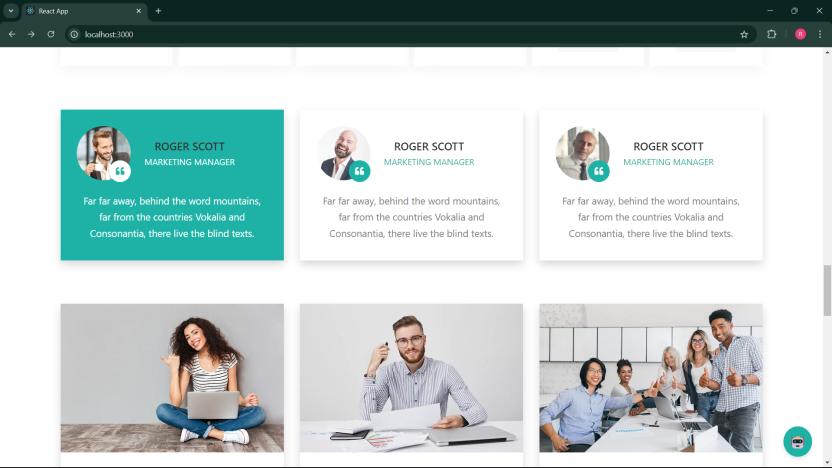


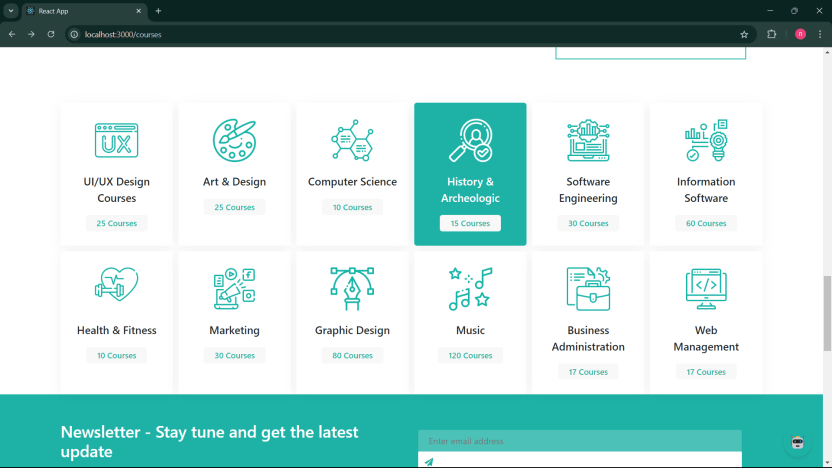


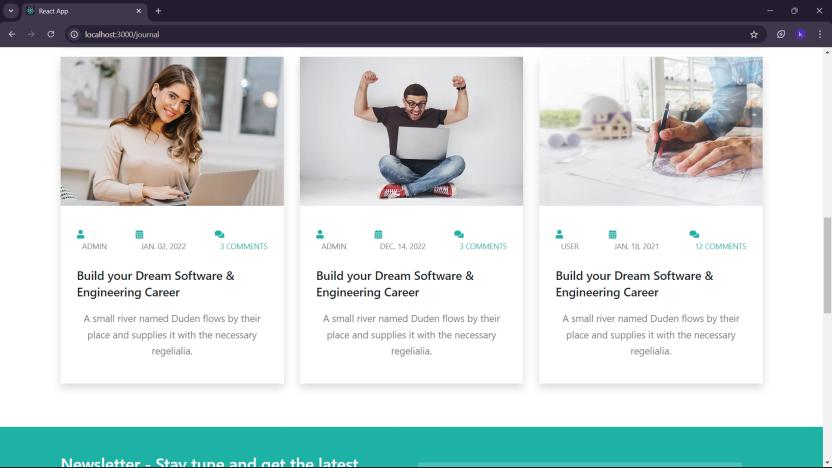


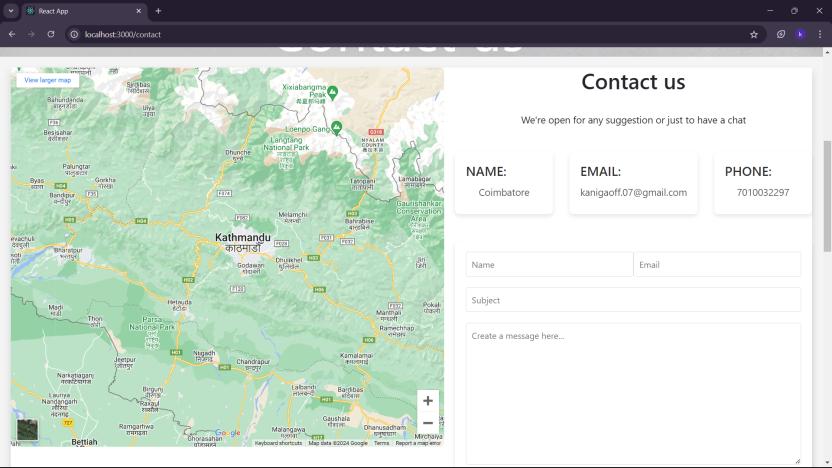


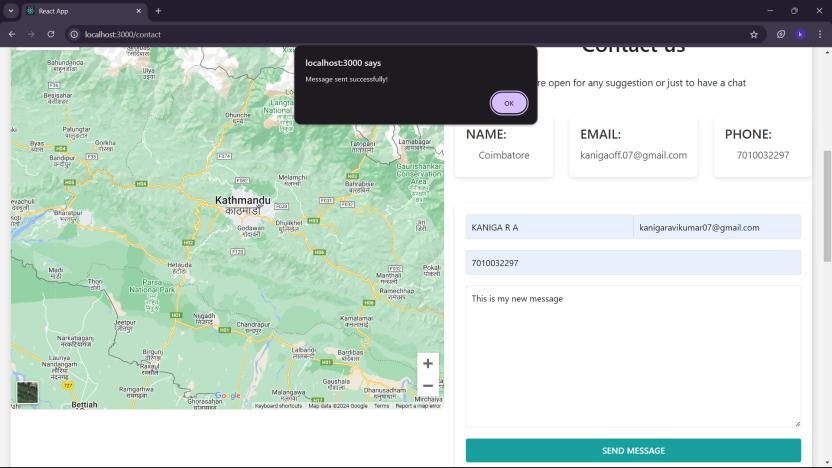


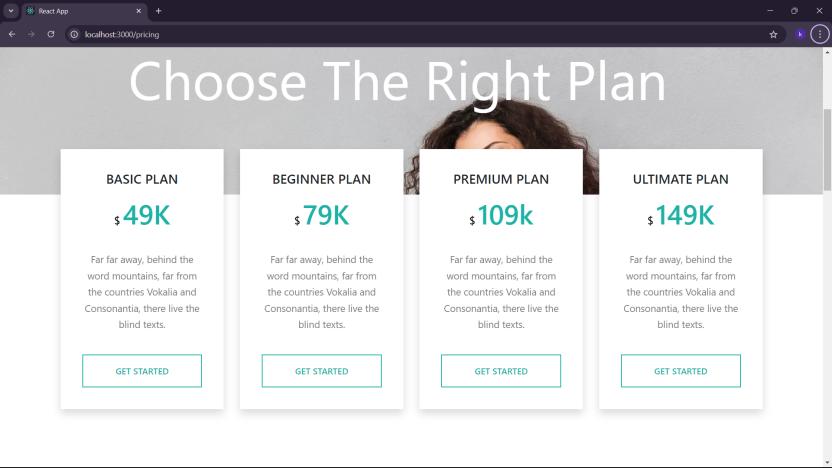


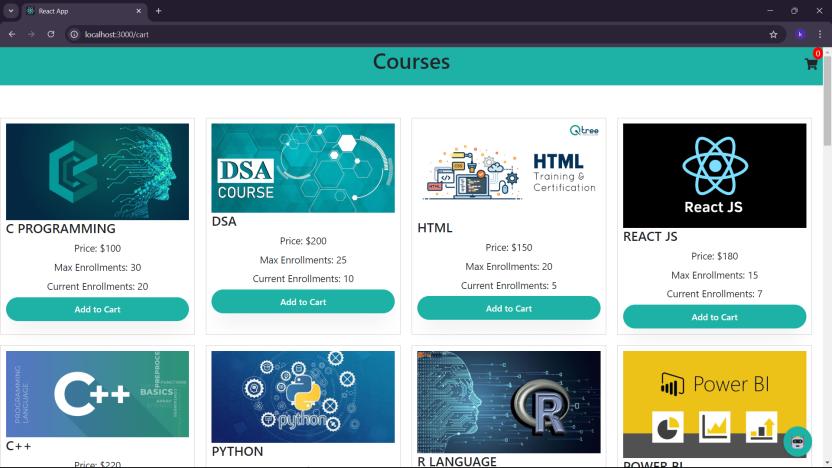


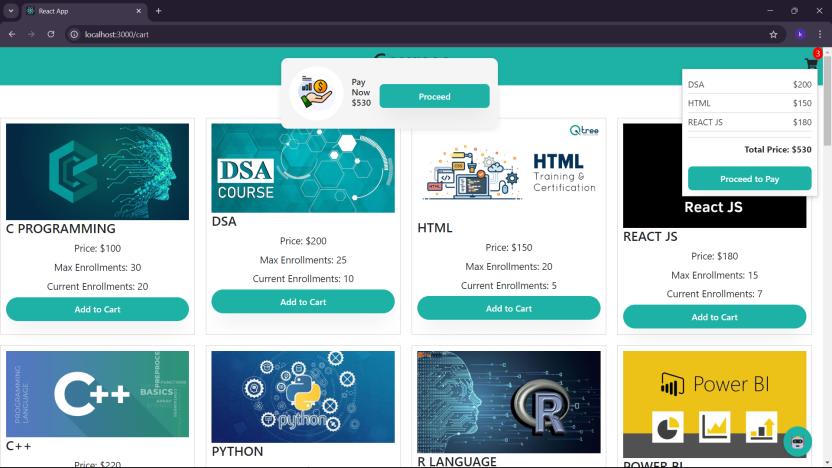


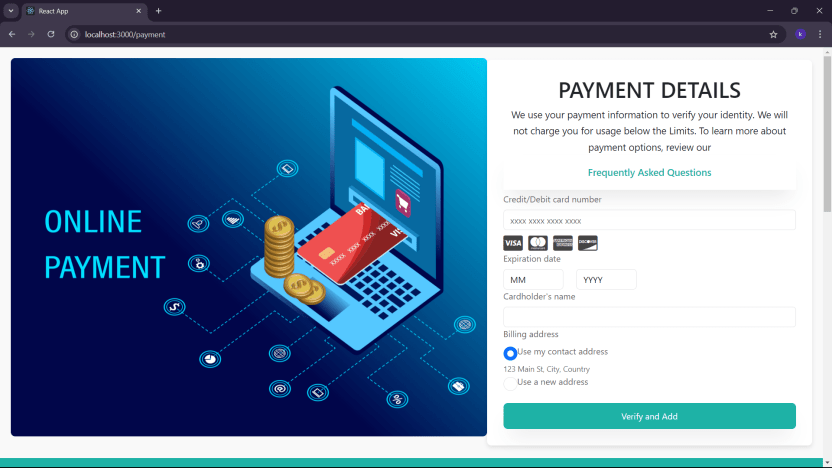


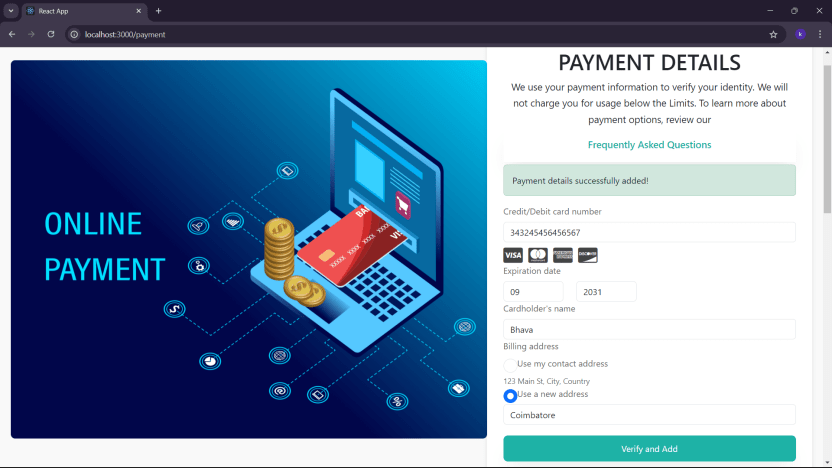




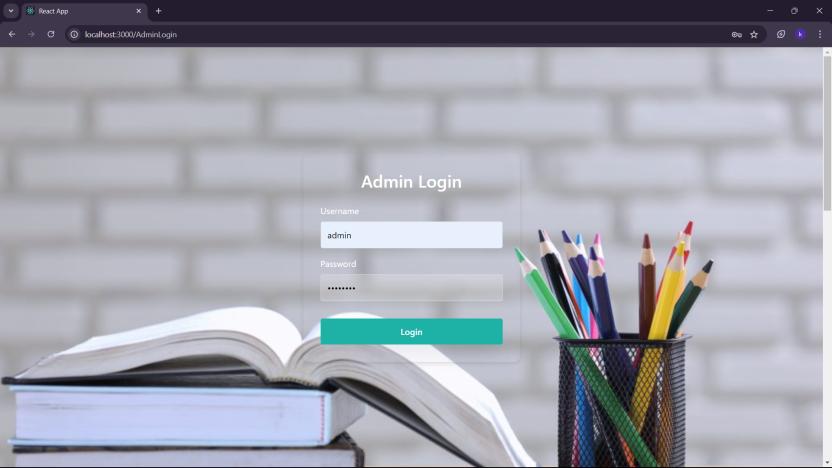


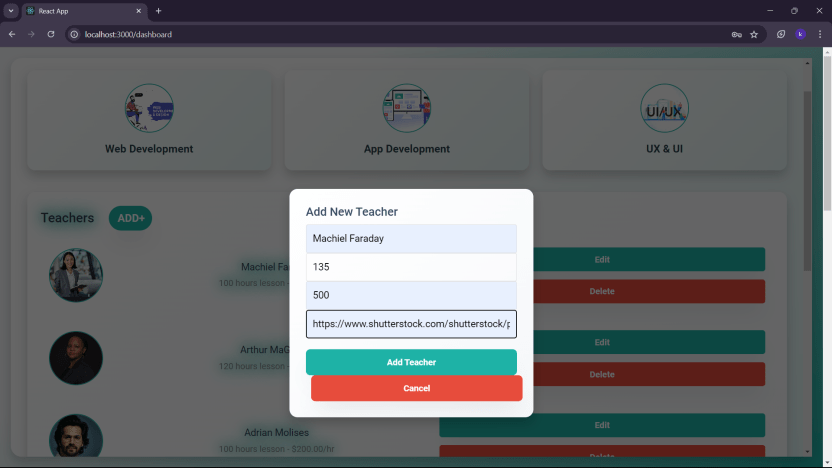


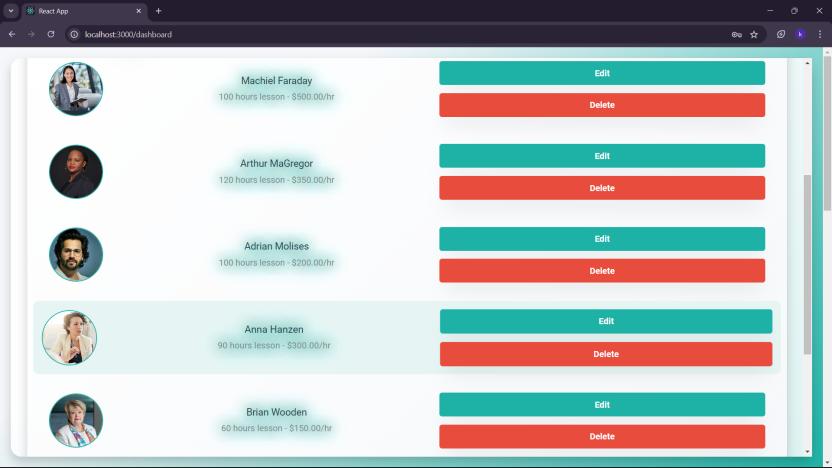


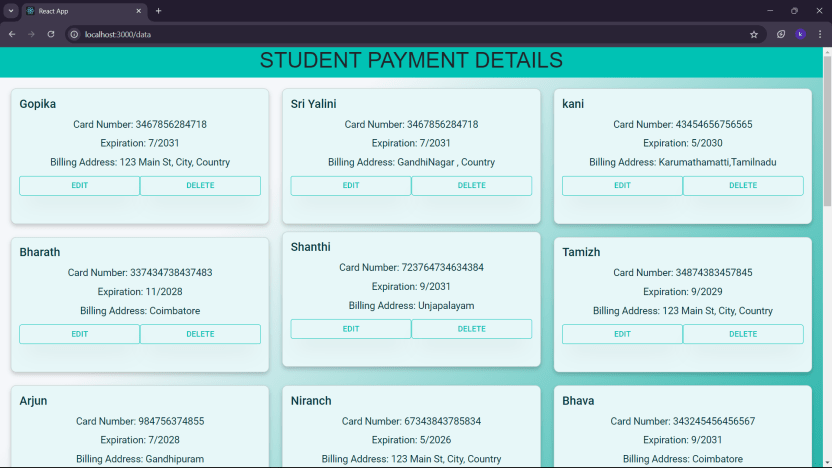


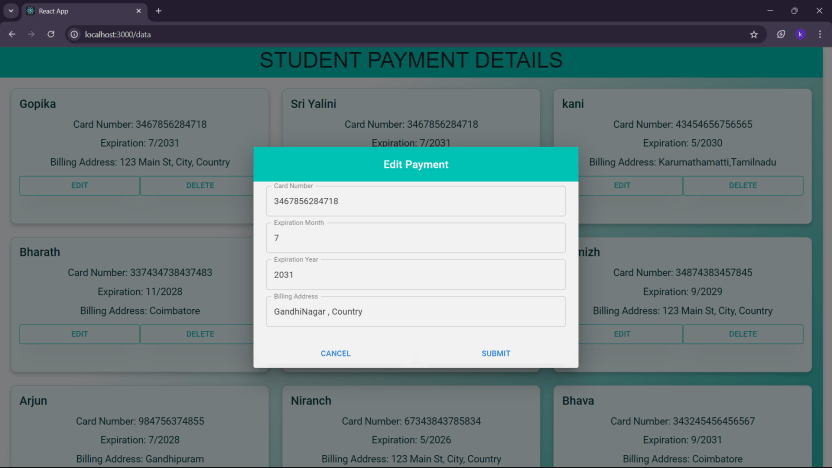
**ADMIN:**

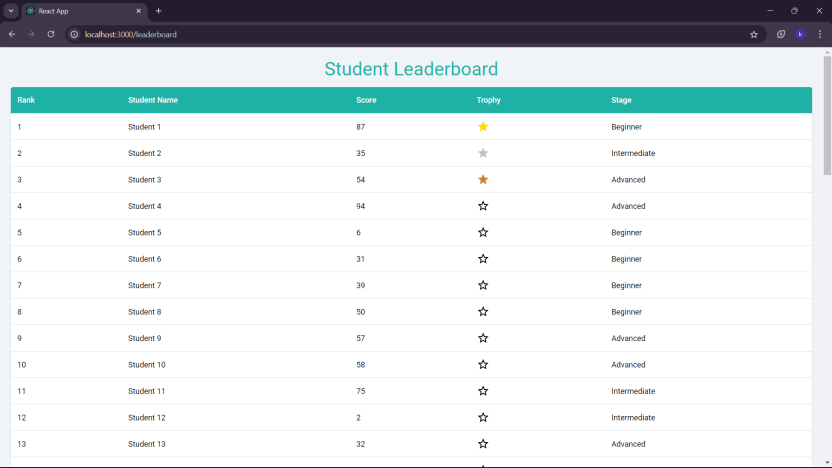


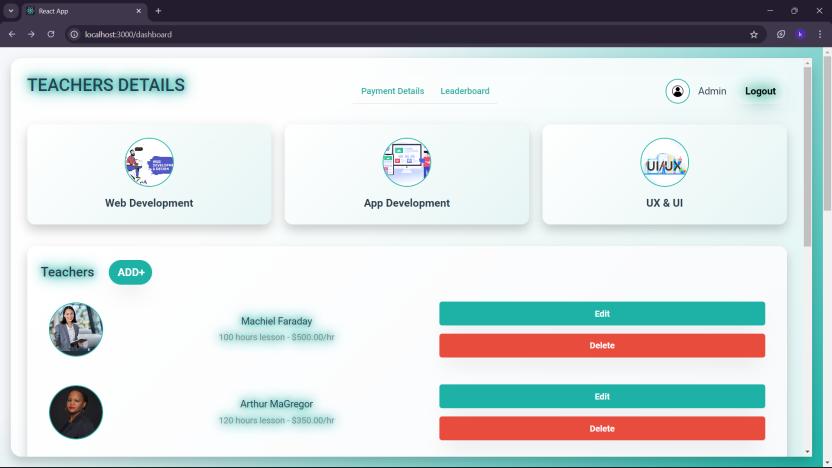










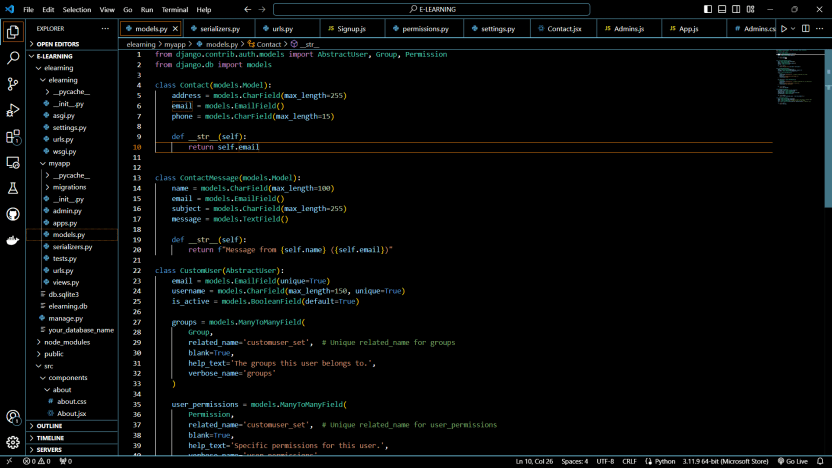


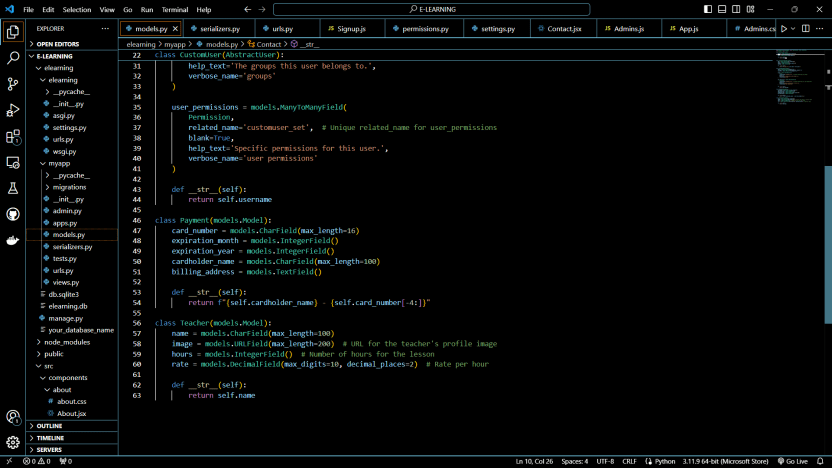
**3.1.2BACKEND DEVELOPMENT (DJANGO):**

1. **Database Schema Design:**

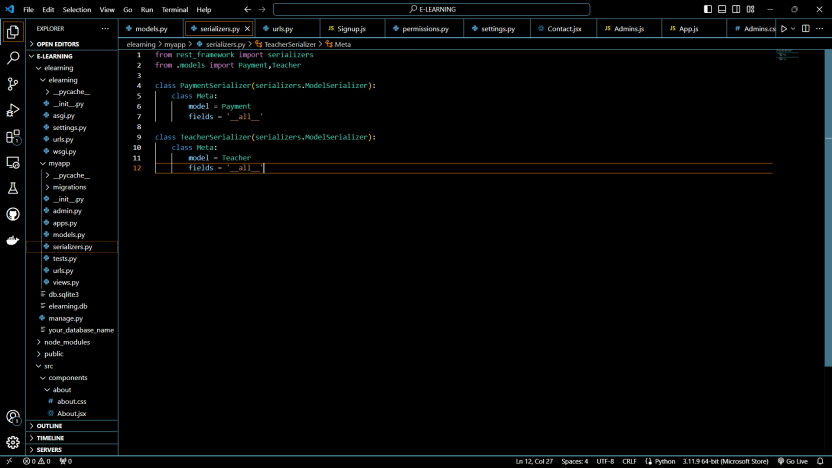
Models such as CustomUser, Payment, Teacher, Contact, and ContactMessage were created to structure the database. These models ensure data consistency and facilitate efficient data management.

**MODELS.PY**



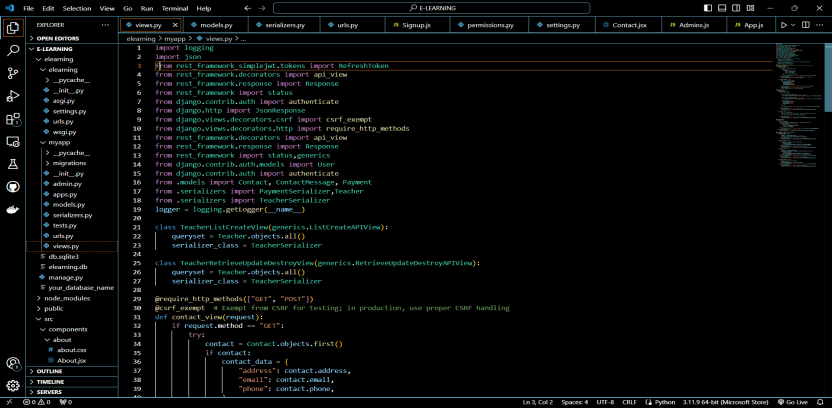


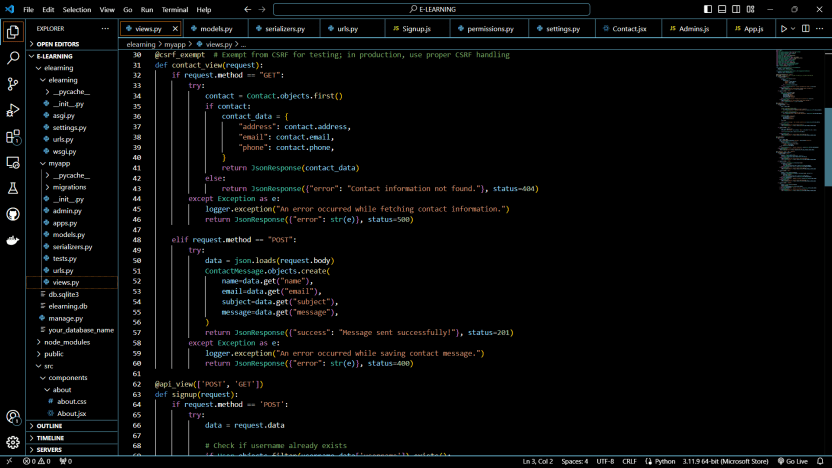
**SERIALIZERS.PY**

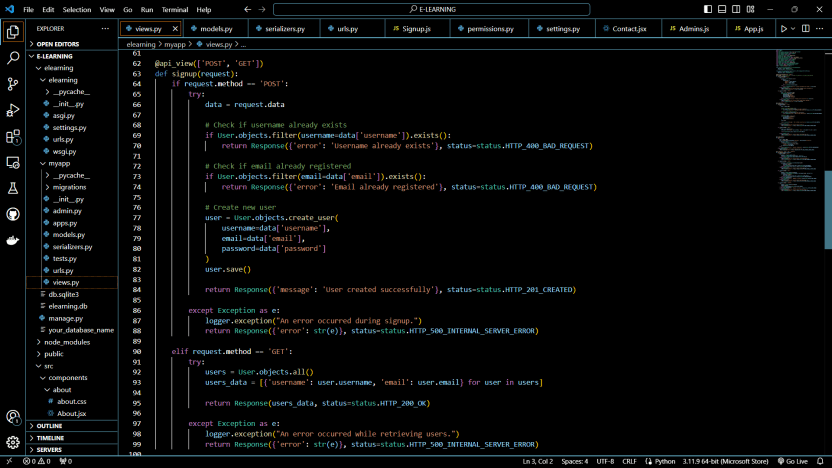


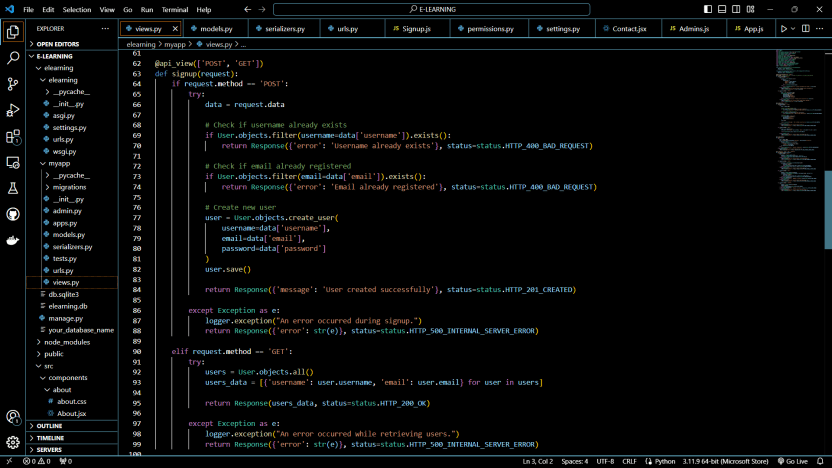
1. **Backend Setup:**

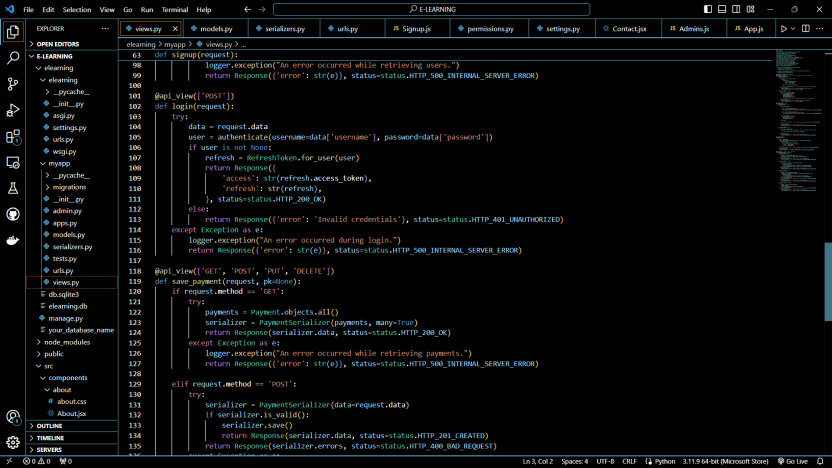
The Django backend was set up with a focus on scalability and security. This involved configuring settings, installing necessary packages, and establishing the admin interface for data management.

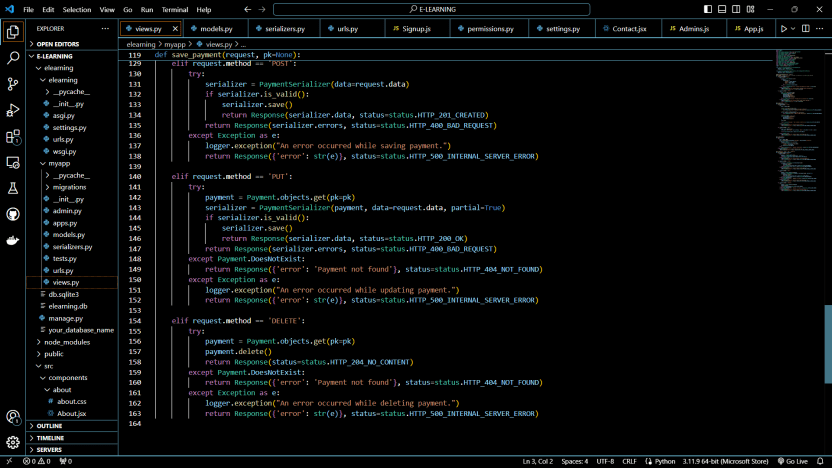
**VIEWS.PY:** 



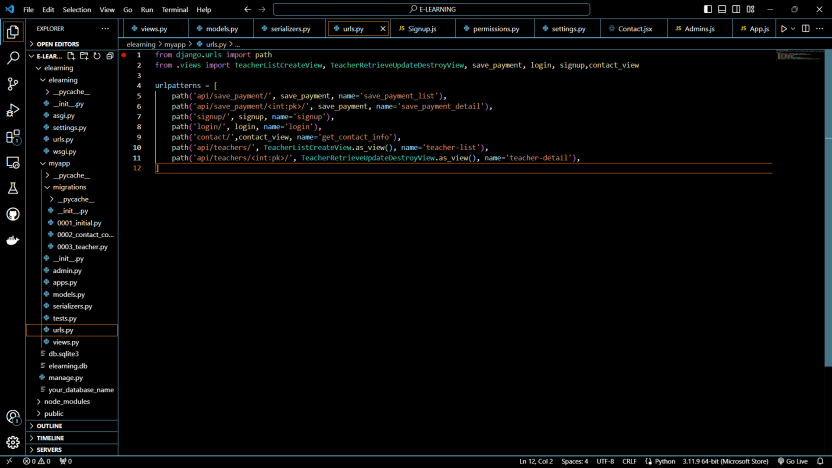


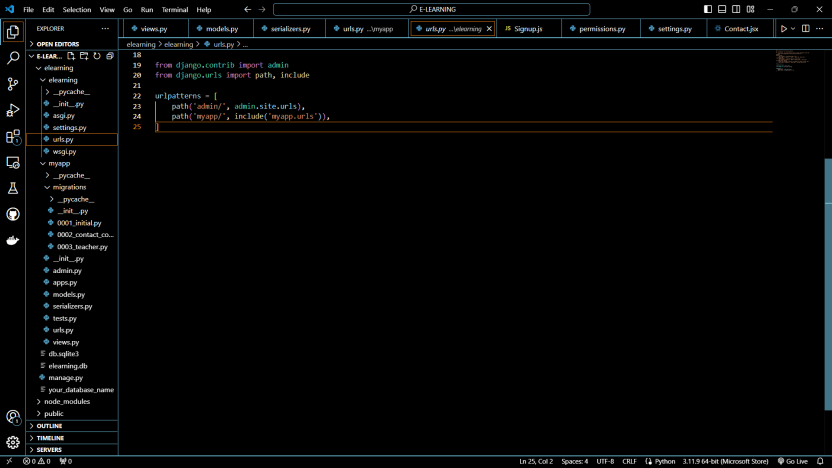




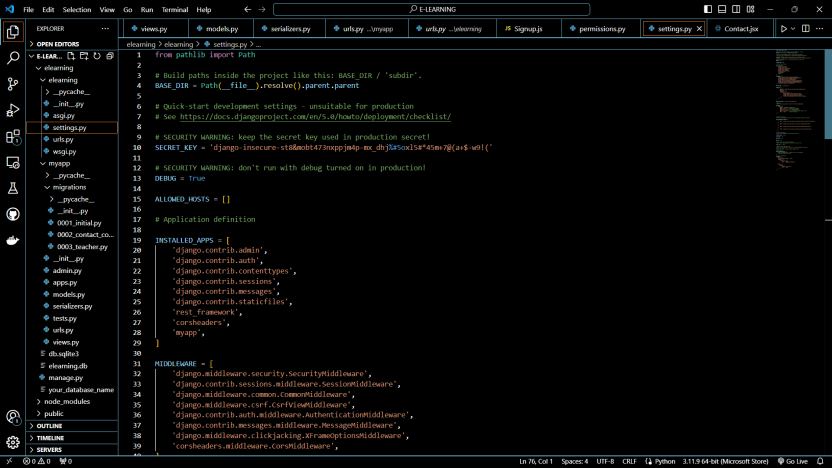


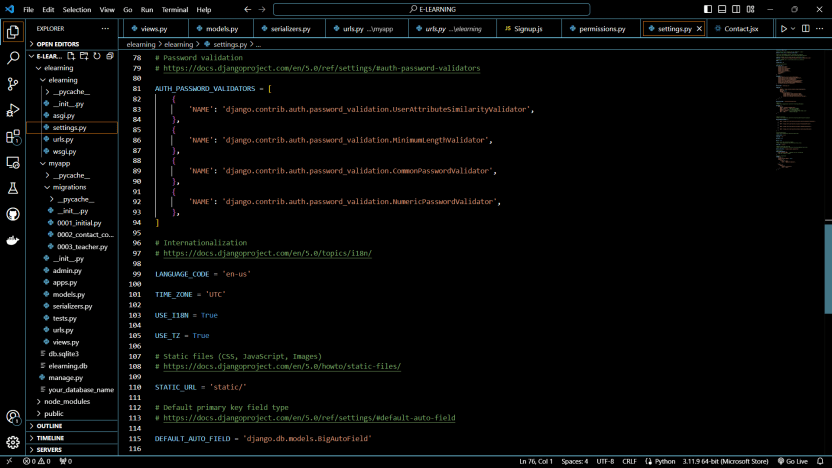
**URLS.PY**

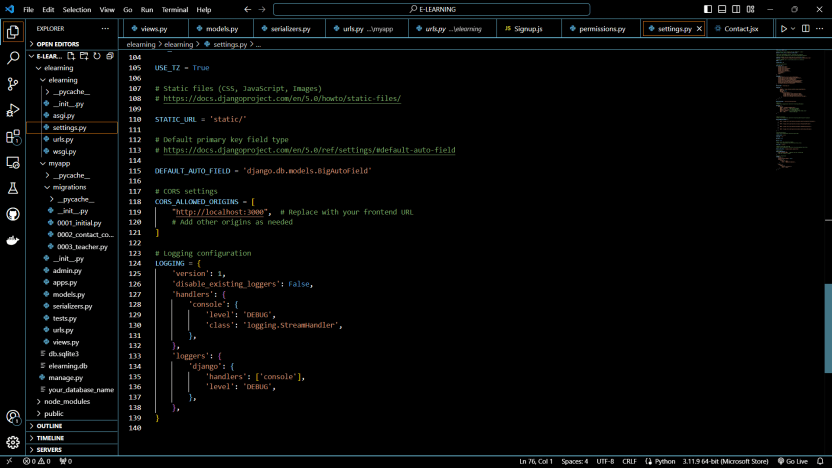




**SETTINGS.PY**



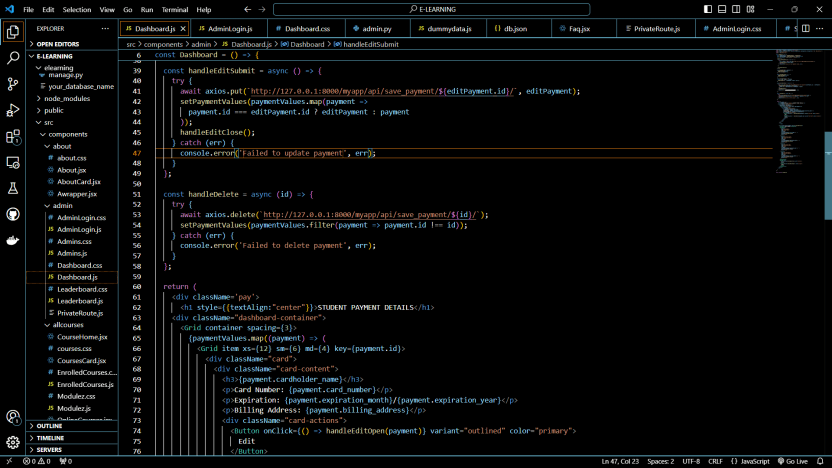


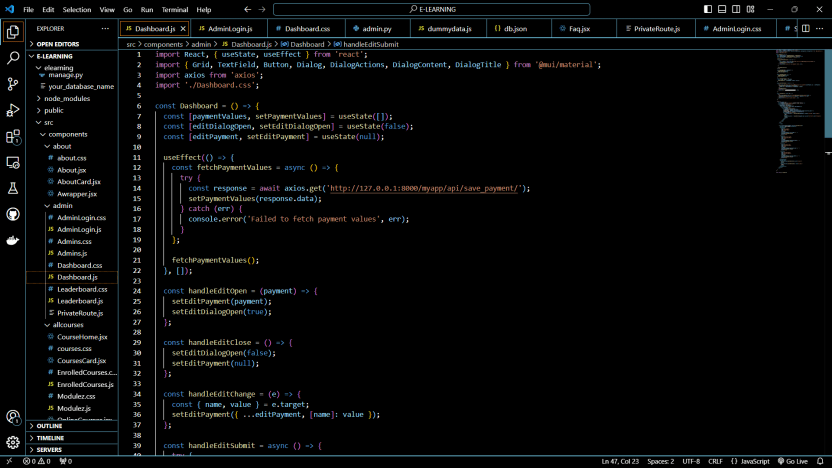


1. **CRUD Operations Implementation:**

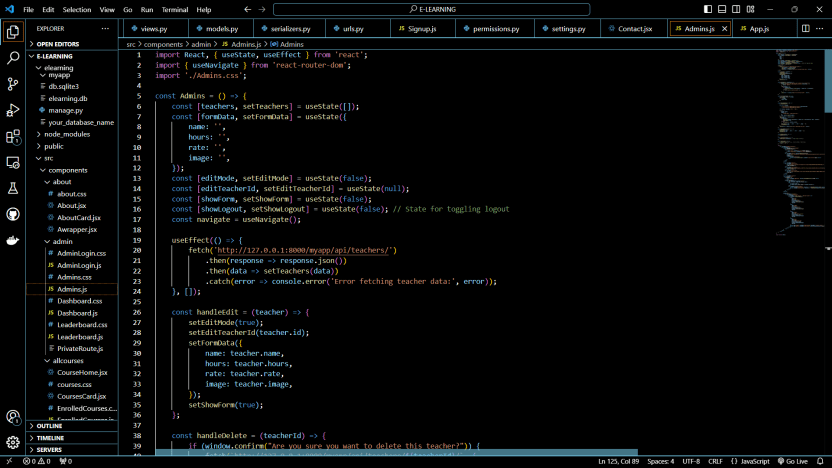
Implemented CRUD operations for managing the data related to teachers and payments. Django's ORM was leveraged to simplify database interactions.

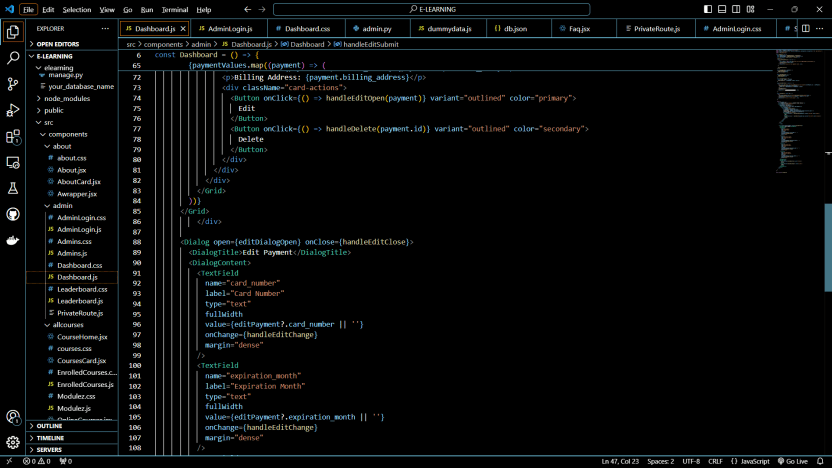
**TEACHERS:**

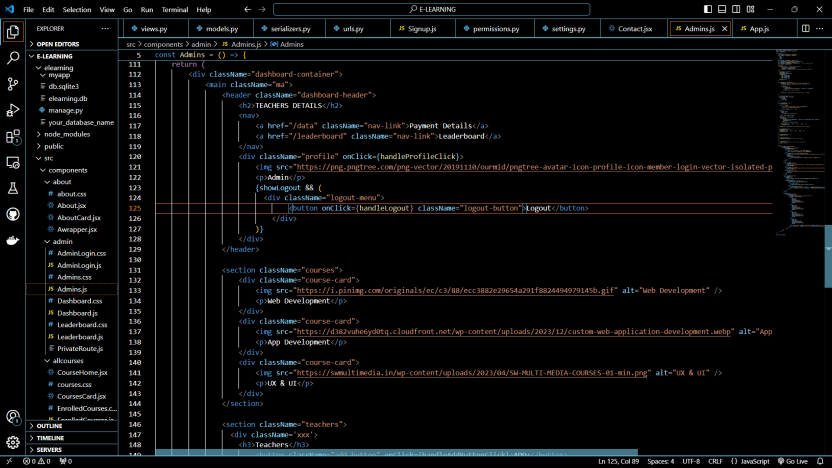


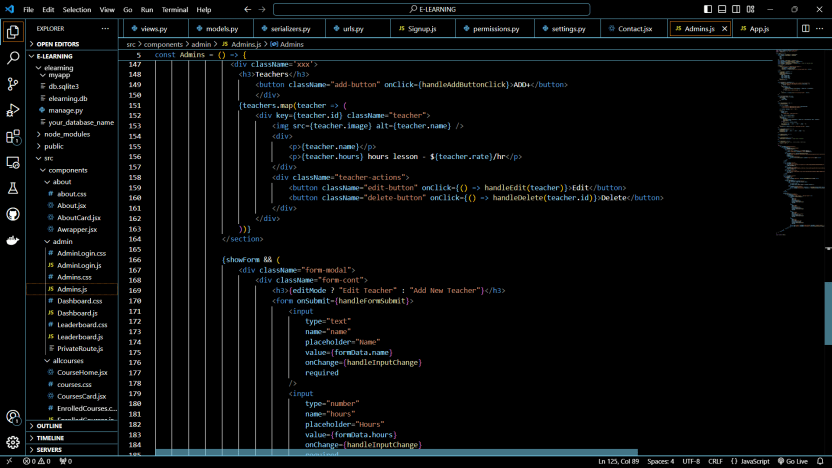


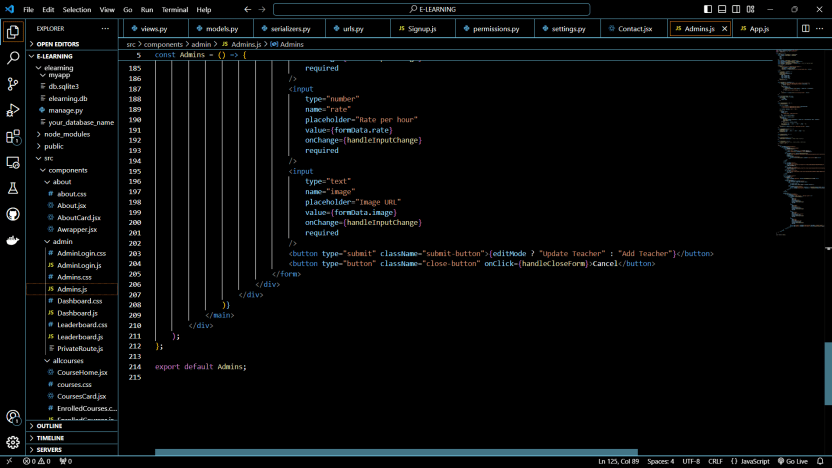
**PAYMENT:**





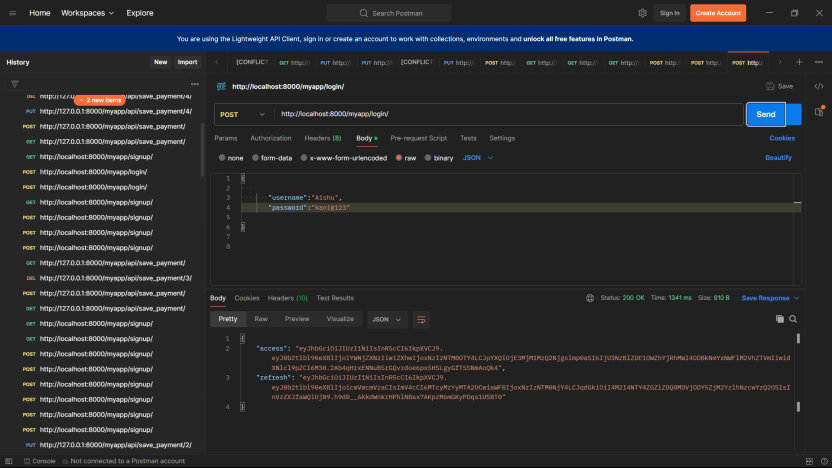


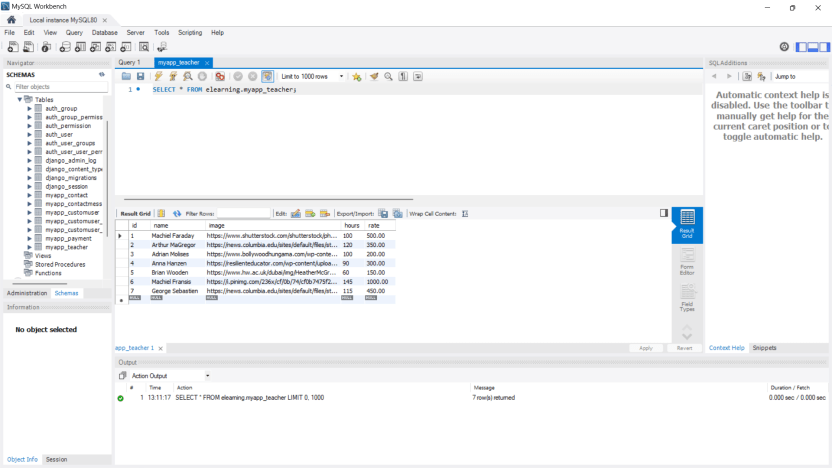


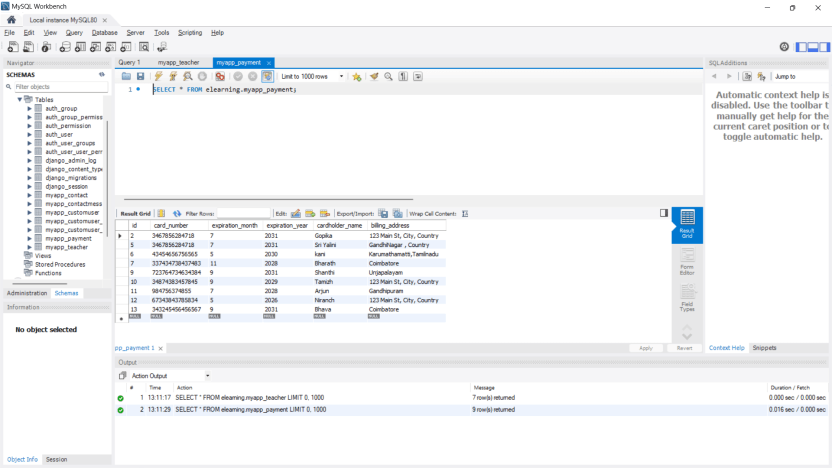


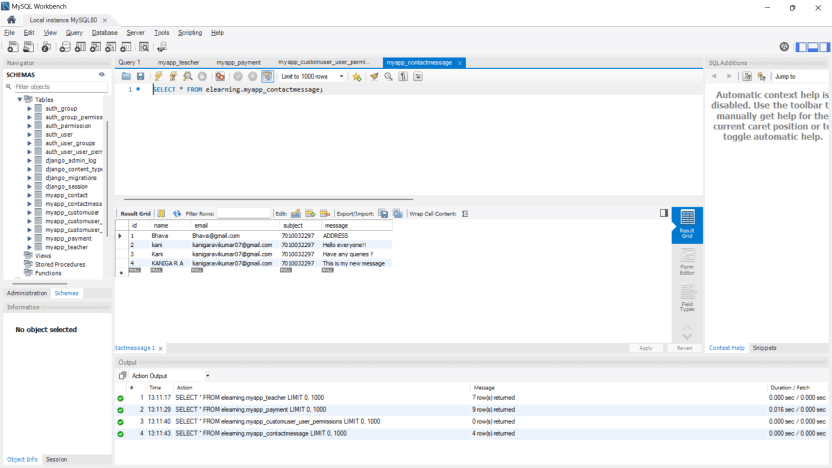
1. **Security Implementation:**

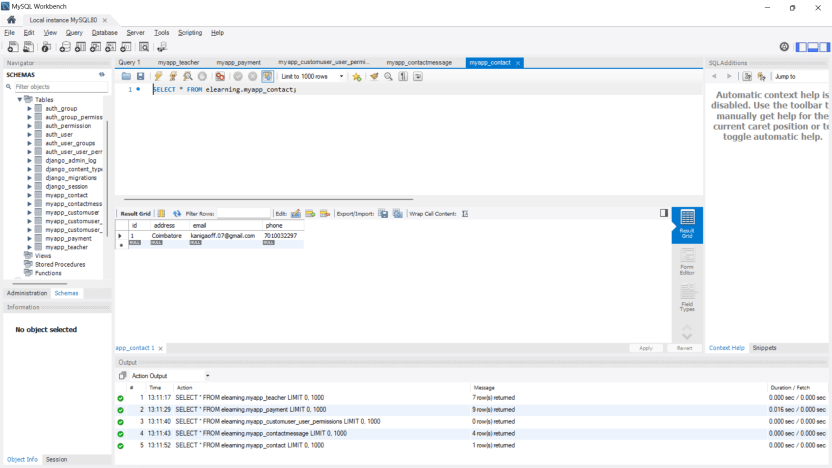
Django's built-in security features, such as form validation, CSRF protection, and password hashing, were utilized. Additionally, Spring Security was integrated to enhance security across the microservices architecture.

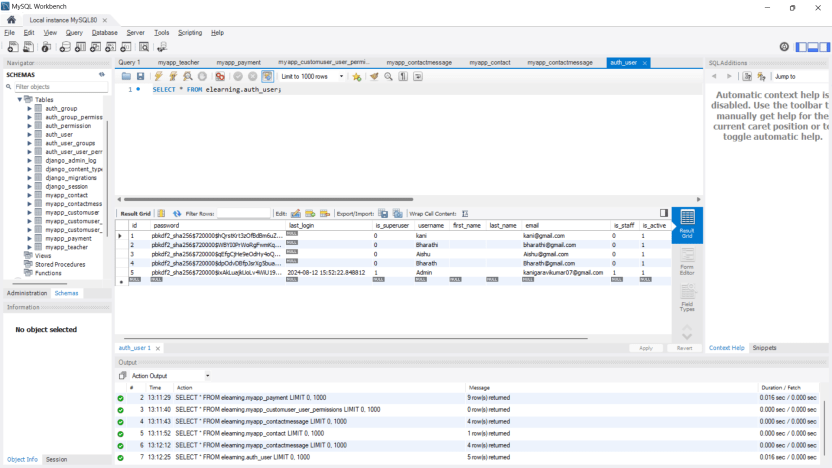












**5. Frontend-Backend Integration:**

The React frontend was integrated with the Django backend through the API Gateway. This setup ensures smooth communication between the client and server, allowing users to access real-time data.

**CustomUser:**

Extends Django's AbstractUser to include custom fields such as email and username, both unique, ensuring user identification and authentication.

**Payment:**

Manages payment information, including card details, expiration dates, and billing addresses. The \_\_str\_\_ method returns the masked card number for security.

**Teacher:**

Stores teacher profiles, including name, profile image URL, lesson hours, and hourly rates. This information is essential for course management and teacher assignment.

**Contact & ContactMessage:**

Contact holds general contact information, while ContactMessage manages individual messages from users. These models facilitate user communication with administrators.

**Microservices:**

The backend was modularized using a microservices architecture. Services like Teacher Management, Payment Processing, and User Management were developed independently and communicated through APIs. Configured an API Gateway using Spring Cloud Gateway to route requests to the appropriate microservices. This gateway serves as a single entry point for frontend requests.

**CONCLUSION:**

This full-stack E-learning portal project successfully integrates a React frontend with a Django backend, utilizing a microservices architecture. The project is structured to be scalable, maintainable, and secure, with a focus on providing a seamless user experience. The Django admin interface is customized to allow administrators to manage essential data efficiently, ensuring the smooth operation of the portal.