Online Learning Platform using MERN - Project Documentation

1. Introduction

- **Project Title**: Online Learning Platform using MERN
- **Team members:**
    - Dharmalingam Praveen Kumar
    - Akash K
    - Gunalan K
    - Ashok Kumar P

2. Project Overview

- **Purpose**:
  The goal of this project is to create an online learning platform that offers a wide variety of courses. Students can access courses, track progress, and earn certificates, while instructors can upload and manage courses, and admins can oversee the platform's operation.
- **Features**:
    - **User Registration**: Allows students and instructors to create and manage accounts.
    - **Course Management**: Instructors can create, update, and delete courses.
    - **Student Dashboard**: Students can browse, enroll in, and track courses.
    - **Interactive Learning**: Includes discussions, quizzes, and assignments.
    - **Certification**: Students receive certificates upon course completion.
    - **Payment Gateway**: For purchasing paid courses.
    - **Admin Dashboard**: Admins can manage users, courses, and platform settings.

3. Architecture

- **Frontend**:
  The frontend is built using **React.js** for creating a dynamic and responsive user interface. The application uses **Material-UI** and **Bootstrap** for styling and provides an interactive user experience.

- **Backend**:
  The backend is built using **Node.js** and **Express.js**, which manage server-side operations, including user authentication, course management, and communication between the frontend and database.
- **Database**:
  The database is **MongoDB**, a NoSQL database that stores user information, courses, and progress in a flexible format. The backend interacts with the database using **Mongoose**.

4. Setup Instructions

- **Prerequisites**:
  1. **Node.js**: JavaScript runtime to run backend code.
  2. **MongoDB**: Database to store user and course data.
  3. **Vite**: For frontend development and build optimization.
  4. **React.js**: Library for building the user interface.
- **Installation**:
  1. Clone the repository:

     https://github.com/praveenkumar262/Online-Learning-Platform-Mern.git

  2. Navigate to the project directory:

     cd OnlineLearningPlatform

  3. Install frontend

     dependencies: cd frontend

     && npm install

  4. Install backend dependencies:

     cd ../backend && npm install

  5. Set up environment variables for both frontend and backend (e.g., MongoDB connection URL, JWT secret, etc.).

5. Folder Structure

- **Client (Frontend)**:
  - `src/`: Contains the React components and pages for the platform.
  - `public` : Static assets like images and fonts.

- **Server (Backend)**:
  - `controllers/`: Logic for handling requests.
  - `models/`: Defines the database schema.
  - `routers/`: API route handlers.
  - `config` : Database and environment variable configurations.

6. Running the Application

- **Frontend**:
  To run the frontend development server, navigate to the `frontend` folder and use:
  `npm start`
- **Backend**:
  To run the backend development server, navigate to the `backend` folder and use:
  `npm start`

7. API Documentation

- **POST /login**:
  - **Request**:
    - `email`: User's email.
    - `password`: User's password.
  - **Response**:
    - `token`: JWT token for authentication.
- **GET /courses**:
  - **Request**: None.
  - **Response**: List of all available courses.
- **POST /courses**:
  - **Request**:
    - `course_name`: Name of the course.
    - `description`: Course description.
  - **Response**:
    - `course`: Newly created course object.

8. Authentication

- **JWT Authentication**:
  Users log in using their email and password. If successful, a JWT token is generated and sent to the user. This token is used for subsequent API requests to authenticate and authorize access to protected routes.

9. User Interface

- **Login Page**:
  Simple and clean UI for user login with form validation.

- **Dashboard**:
  Displays enrolled courses, course progress, and the ability to enroll in new courses.

- **Course Page**:
  Shows course details, modules, and the option to start or continue learning.

10. Testing

- **Unit Testing**:
  - Jest and React Testing Library for testing React components.
  - Mocha and Chai for backend API testing.

- **Integration Testing**:
  Ensures that frontend and backend systems communicate effectively, especially for user authentication and course enrollment.

11. Screenshots or Demo

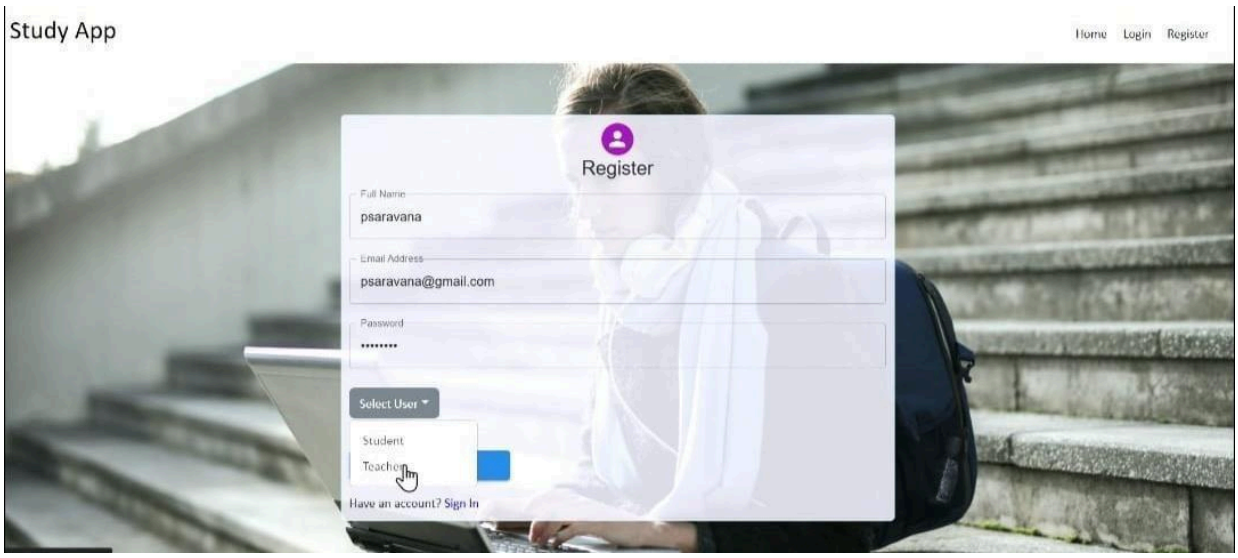

Figure 1- student and teacher sign up page

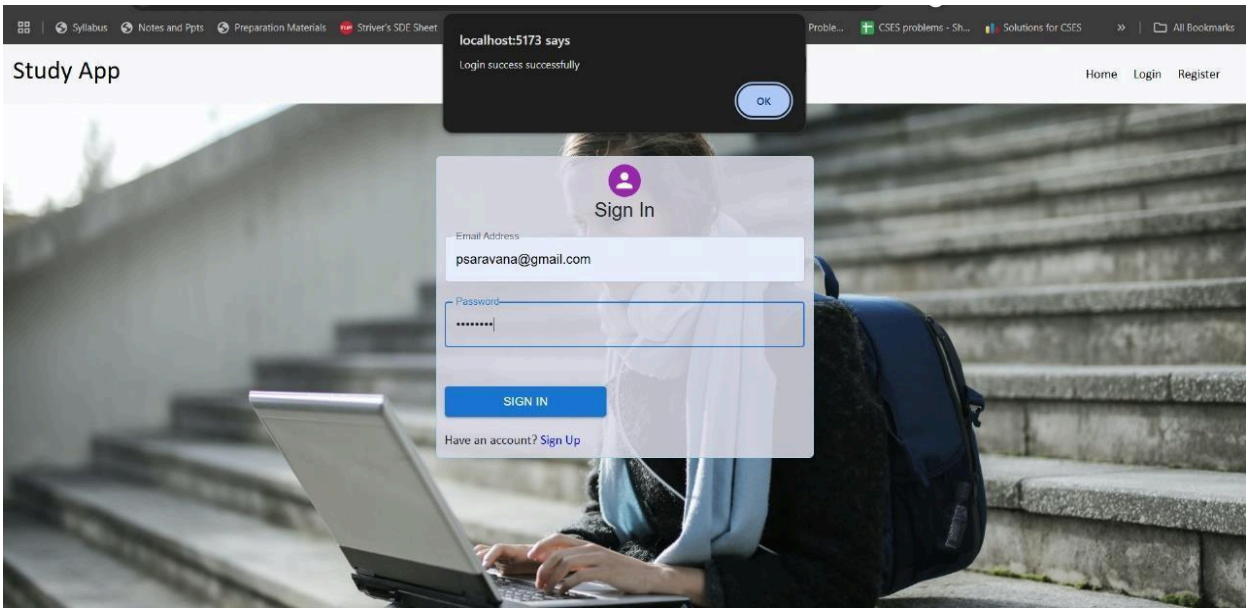

Figure 2- Teacher sign in page

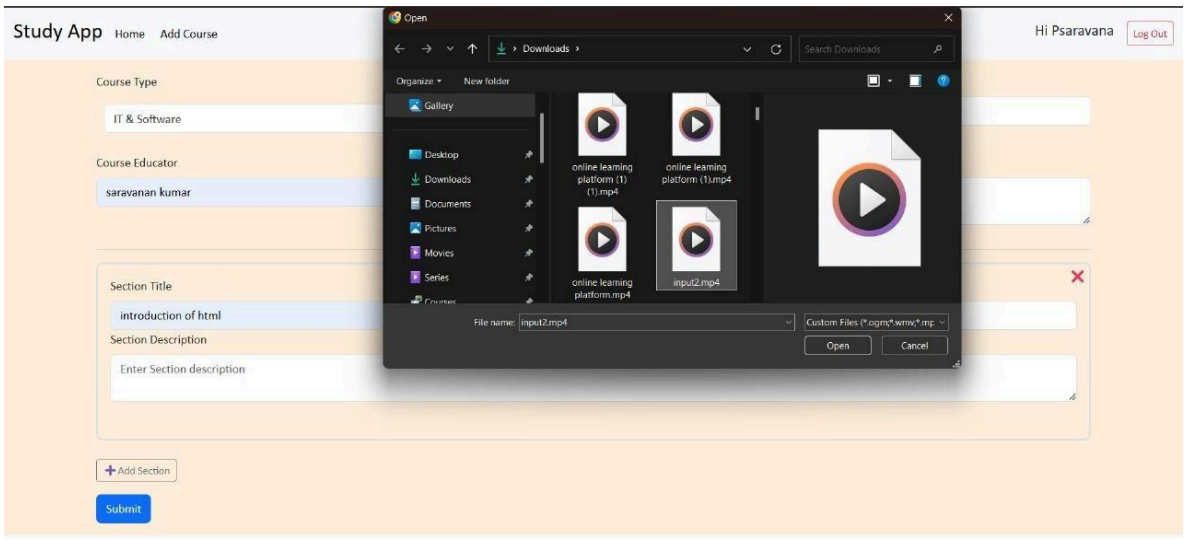

Figure 3- Adding Course video

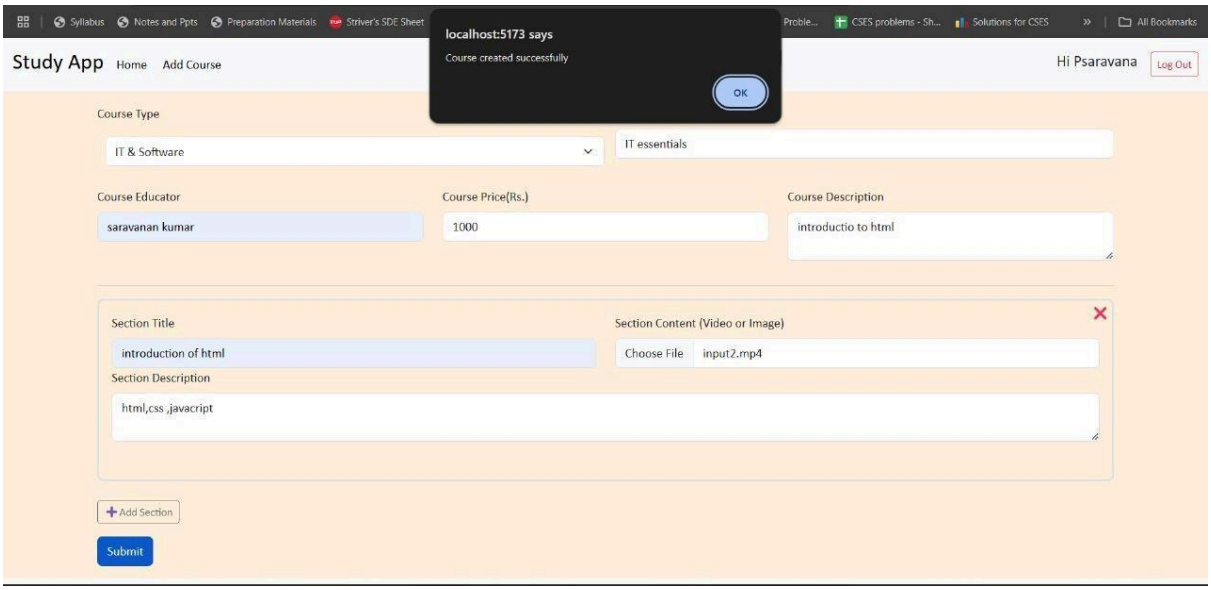

Figure 4-Adding new course by teacher

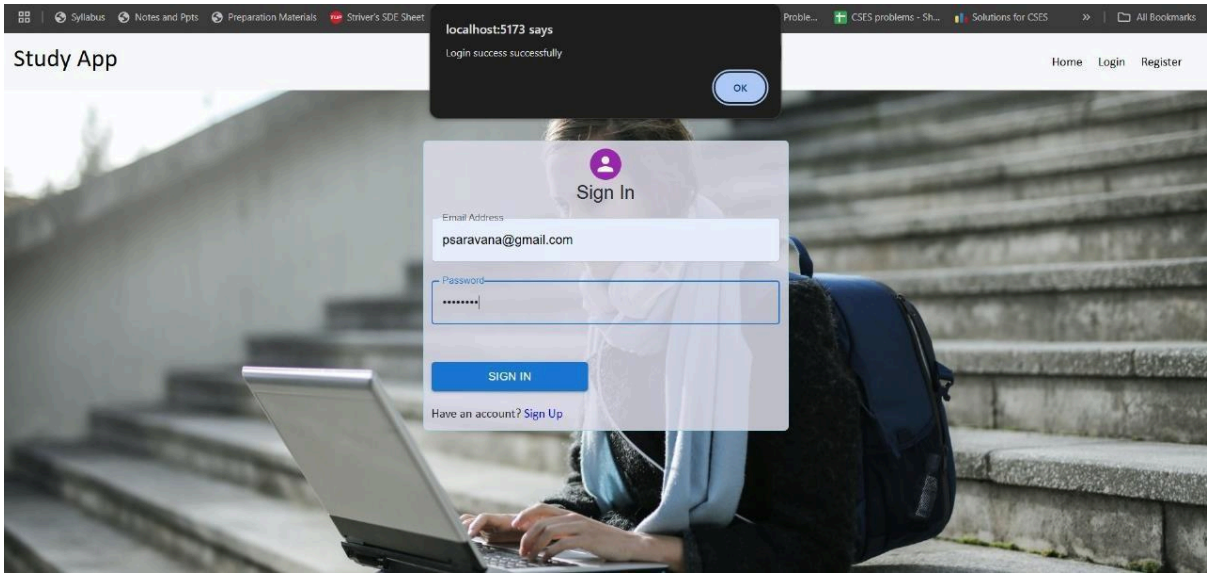

Figure 5-Student Login

Figure 6- Enrolling course

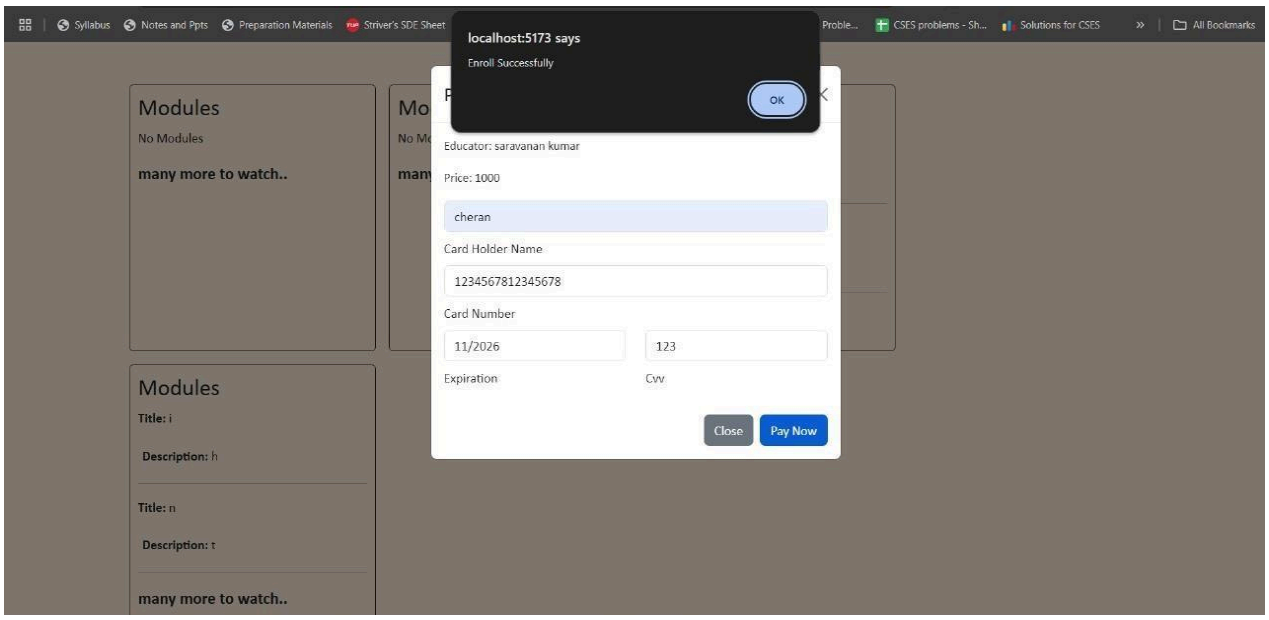

Figure 7- Payment details

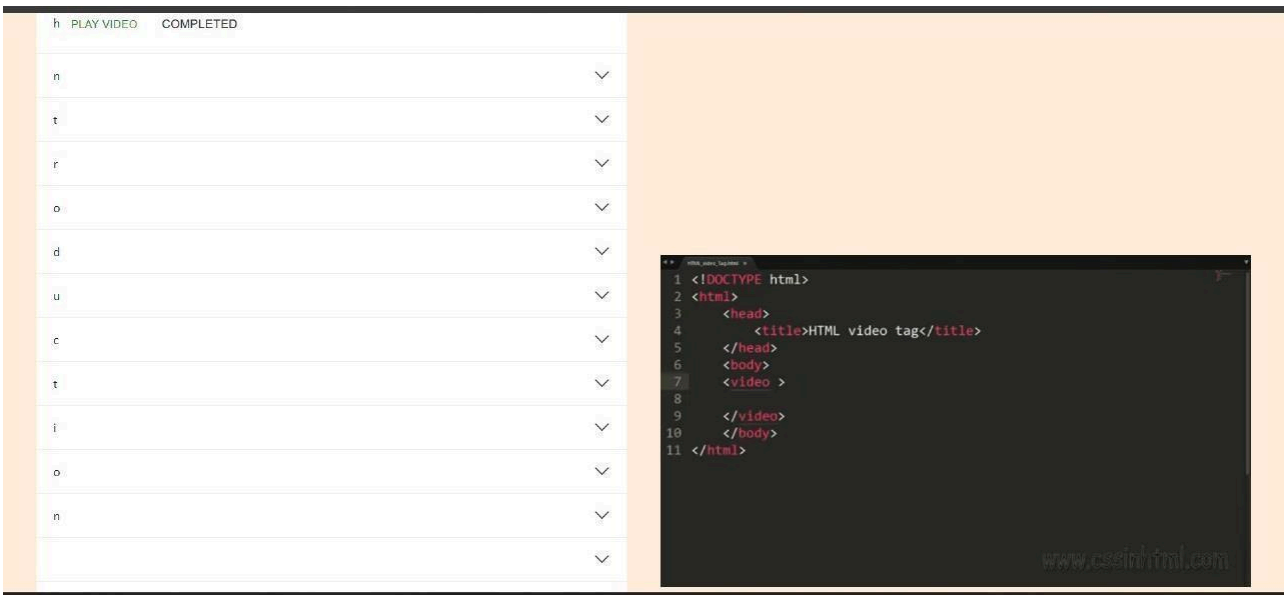

Figure 8- watching Course Video

Figure 9-Delete course by teacher

12. Known Issues

- Document any bugs, limitations, or areas needing improvement, such as performance issues with large data sets or specific edge cases in the shopping cart functionality.

13. Future Enhancements

- **Mobile App**:
  A mobile version of the platform for better accessibility.
- **Advanced Analytics**:
  Add detailed analytics and reporting features for both students and instructors.
