Full Stack Development with MERN

1. Introduction

Project Title: Online Learning Platform using MERN
Team Members:

- Bala Murugan V – Front-end Developer
- Balavasan S – Full-stack Developer
- Prince Immanuel J – Back-end Developer
- Ragul A – Technical Lead

2. Project Overview

Purpose: This project aims to create a responsive and user-friendly online learning platform. It allows students to access courses, take quizzes, and monitor progress. Administrators can manage course content and track student engagement.
Features:

- User registration and login with JWT authentication.
- Course management (create, update, delete courses).
- User dashboard for progress tracking.
- Admin dashboard for content management.
- Real-time communication using live messaging for interactive sessions.

3. Architecture

- **Frontend:** Built using React, the frontend provides a dynamic user interface. It includes components for user authentication, course browsing, and content consumption, leveraging React Router for seamless navigation.
- **Backend:** The backend is built with Node.js and Express.js. It handles requests, manages authentication, and connects with MongoDB for data storage. Controllers manage course data and user interactions.
- **Database:** MongoDB stores information about users, courses, quizzes, and progress. It is structured with collections for each major feature (e.g., users, courses), enabling quick retrieval and updates.4. Setup Instructions

4. Setup Instructions

Prerequisites:

- Node.js v14+
- MongoDB v4+
- (Optional) npm or yarn for package management

Installation:

1. Clone the repository: git clone <repository-url>
2. Navigate to both the backend and frontend directories and install dependencies:
   For Backend: cd backend
   npm install
   For Frontend: cd ../frontend
   npm install

3. Set up environment variables by creating a .env file in the backend directory with database connection strings, JWT secret, etc.

5. Folder Structure

Client:
The frontend directory contains:

- src/components - Contains reusable React components.
- src/pages - Different pages (e.g., CourseList, Login, Register).
- src/utils - Utility functions and API calls.
- src/styles - CSS files for styling.

Server:
The backend directory is organized as follows:

- config - Database connection configuration.
- controllers - Functions to handle business logic.
- routers - Defines routes for various API endpoints.
- middlewares - Middleware for authentication and error handling.
- schemas - MongoDB schemas and models.

6. Running the Application

Frontend:
Start the frontend server: npm start
Backend:
Start the backend server: npm start

7. API Documentation

The backend exposes several endpoints, including:

- POST /api/auth/login - User login with JWT authentication.
- POST /api/auth/register - User registration.
- GET /api/courses - Fetch available courses.

Example Response:

```
{
  "status": "success",
  "data": {
    "courses": [
      {
        "id": "course-id",
        "title": "Course Title",
        "description": "Course description",
      }
    ]
  }
}
```

8. Authentication

Authentication is handled using JWT tokens. After login, users receive a token stored in local storage. Protected routes require a valid token for access, which is verified

using middleware.

9. User Interface

The interface includes:

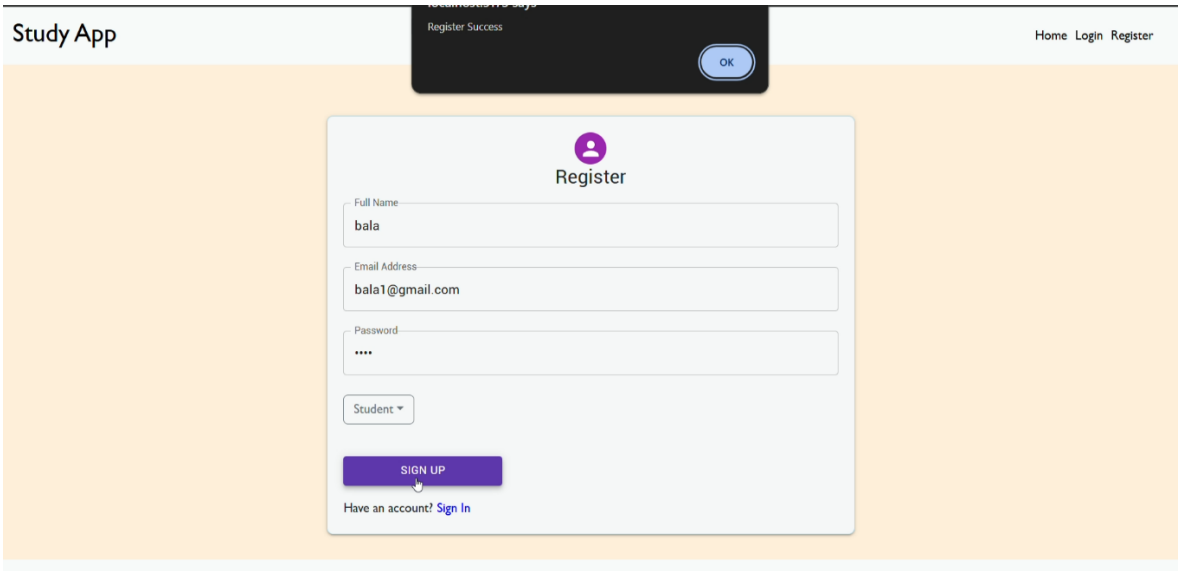
- A login and registration form.
- Course catalog page displaying available courses.
- Dashboard page where users track their learning progress.

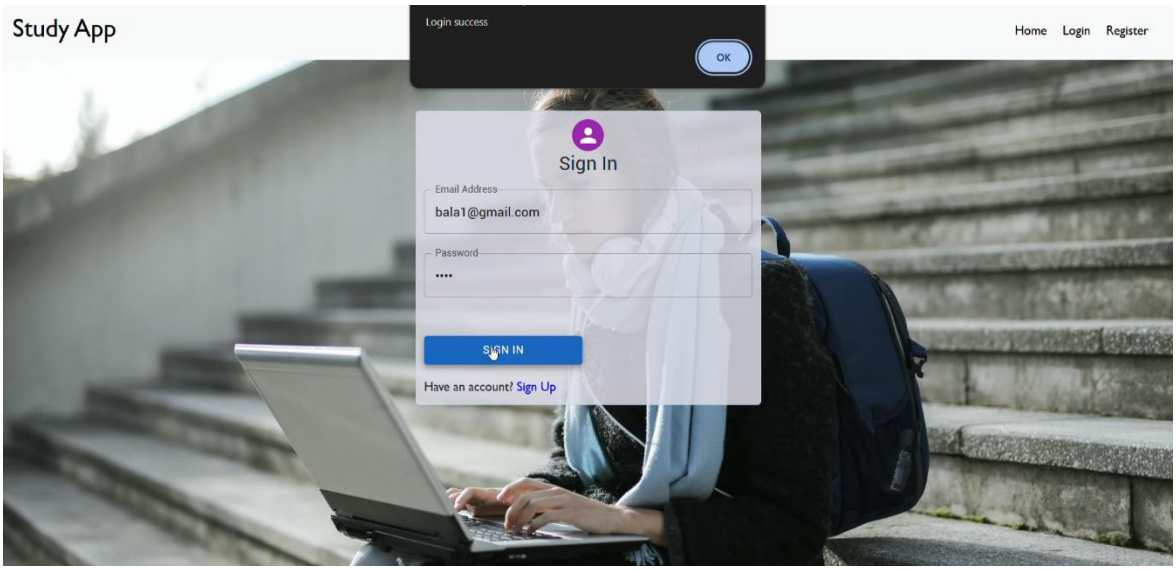
10. Testing

Testing is performed using tools like Jest and Postman for API testing. Unit tests cover component functionality and API response validation.

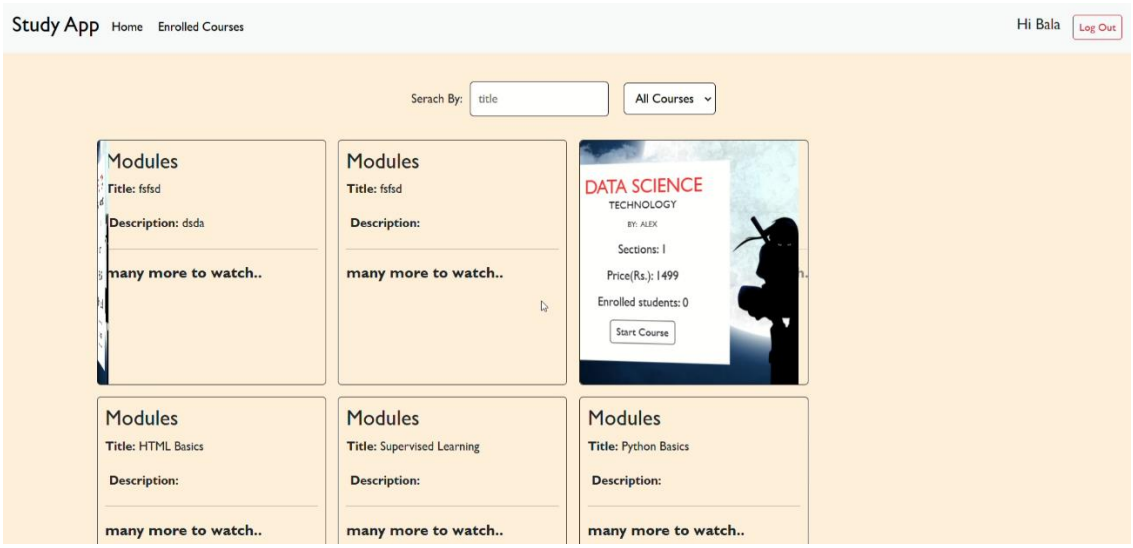
11. Screenshots

- A login and registration form.

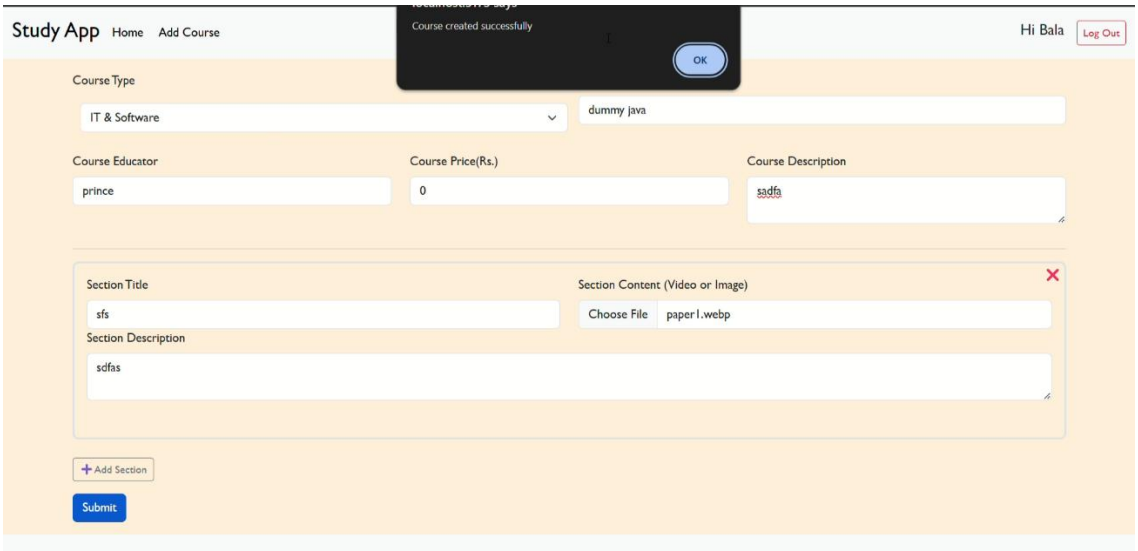


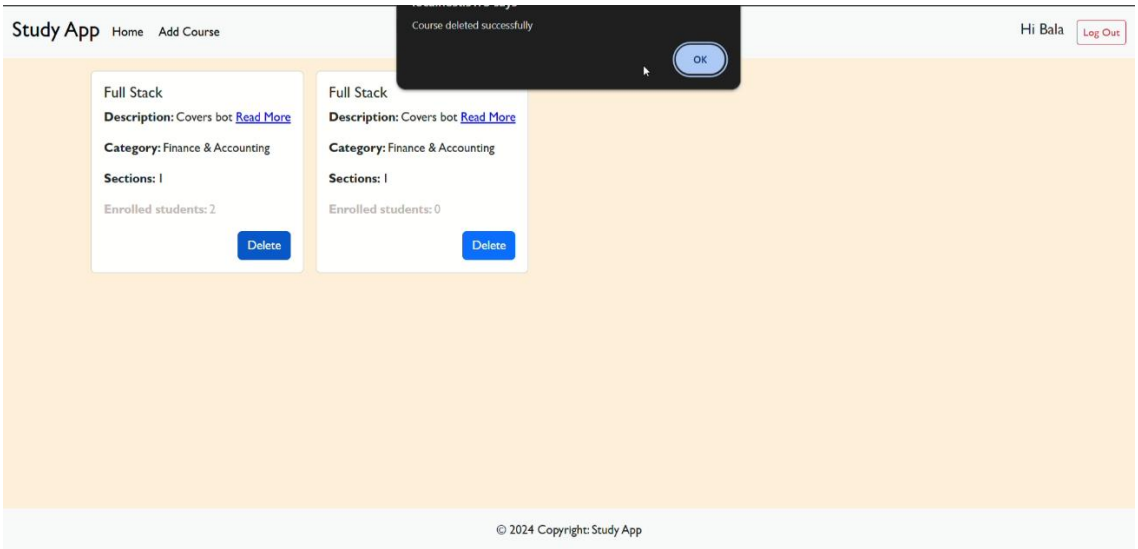


- Course catalog page displaying available courses.

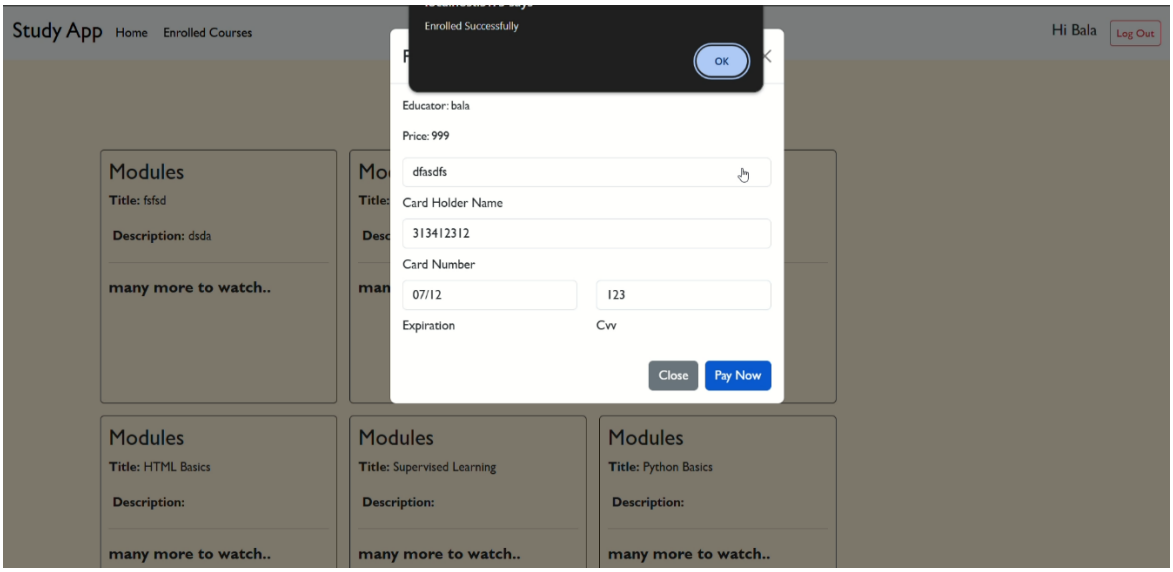


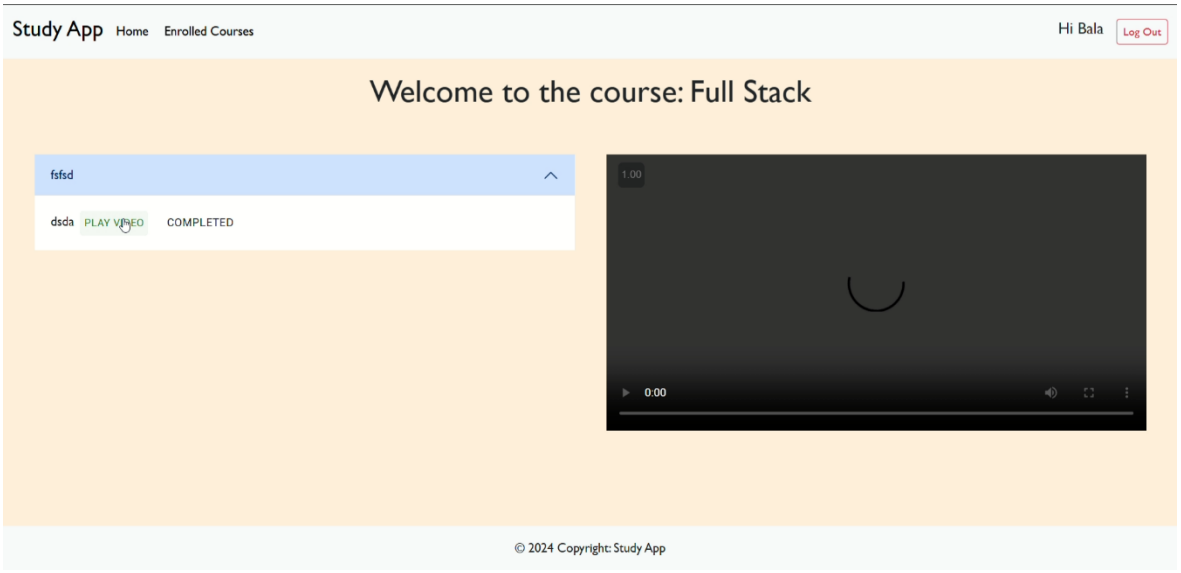
- Create Course and Delete page.





- Payment and play video page





.

12. Known Issues

- Token expiration may log users out unexpectedly.
- Occasional delays in real-time messaging.

13. Future Enhancements

- Add support for quizzes and assignments within courses.
- Implement a payment gateway for premium courses.
- Enhance real-time chat with video conferencing.