

PROJECT

Report on

Online Learning Platform using MERN

SUBMITTED FOR:THE ASSIGNED PROJECT IN ONLINE LEARNING PLATFORM

USING MERN DURING NAAN MUDHALVAN TRAINING PROGRAM

SUBMITTED TO

Naan Mudhalvan Upskilling Platform

SUBMITTED BY: Team-9956 DHANALAKSHMI SRINIVASAN COLLEGE OF
ENGINEERING AND TECHNOLOGY MAHABALIPURAM

TEAM MEMBER 1: (TEAM LEADER)

NAME: SUMIT KUMAR

NM ID: AF46002C10364DC949743B7B0B139C0E

DEPARTEMENT: CSE (FINAL YEAR)

ROLE: TEAM LEAD AND BACKEND

TEAM MEMBER 2:

NAME: VICKY KUMAR

NM ID: 400CD28E35745A593A04672DC984CB3F

DEPARTEMENT: CSE

ROLE: TESTING, DEPLOYMENT AND DOCUMENTATION

TEAM MEMBER 3:

NAME: PRINCE KUMAR RAM

NM ID: 69D76643769948F97D54AA4DC97CDFE7

DEPARTEMENT: CSE

ROLE: FRONTEND DEVELOPER

TEAM MEMBER 4:

NAME: ROHIT KUMAR PANDEY

NM ID: AD5A285F50E4A66807845905EB06C0A0

DEPARTEMENT: CSE

ROLE: UI AND UX DESIGNER

CERTIFICATE OF ORIGINALITY

This is to certify that the project report entitled Online Learning Platform using MERN is submitted to Naan Mudhalvan Upskilling Platform is an original work carried out by

SUMIT KUMAR, Reg no:310521104124,

VICKY KUMAR, Reg no:310521104124,

PRINCE KUMAR RAM, Reg no:310521104124,

ROHIT KUMAR PANDEY, Reg no:310521104124,

under my guidance. The matter embodied in this project is genuine work done by the student.

ACKNOWLEDGEMENT

This is the great opportunity to acknowledge and to thank all those people without whose support and help this project would have been impossible. I would like to add a few helpful words for the people who are part of this project in numerous ways. First among them my respected teachers who have unending support for choosing this project must also thank the respected guide provided by naan mudhalvan team for their valuable suggestions while working on this project. I would like to thank all faculty members of my institution for their cheerful readiness to help me during my project. I would also like to thank all my classmates who helped me with this great idea and valuable suggestion. Finally, I would like to thank my parents for their blessing without which I could not do anything in my life. Last, but not least, I would like to thank all the people who directly helped me in doing this project.

Thank You.

DECLARATION

I declares that project is submitted by our team ("Online Learning Platform Using MERN") for the project assigned by the Naan Mudhalvan Upskilling Platform

It is not a copy or duplicate one.

All the information given is correct as per our knowledge.

Project Description

Online Learning Platform (OLP)

Overview

The **Online Learning Platform (OLP)** is a robust, web-based educational tool designed to provide learners with a wide variety of courses, resources, and interactive features for effective online learning. Aimed at individuals of all ages and backgrounds, the platform offers a flexible, accessible, and engaging way to learn new skills, advance careers, or pursue personal development. With a user-friendly interface and rich set of features, OLP caters to both learners and instructors, fostering a collaborative and interactive learning environment.

Key Features

1. User-Friendly Interface:

The platform is designed with simplicity and ease of navigation in mind, ensuring that users of all technical abilities can quickly find courses and access educational materials. A clean layout, intuitive controls, and clear course categorizations enhance the overall experience.

2. Course Management:

Instructors can upload and manage courses through a dedicated dashboard. This includes creating lesson plans, uploading video lectures, reading materials, assignments, and quizzes. Learners can browse, enroll in, and track their course progress.

3. Interactivity:

OLP promotes engagement through interactive elements like discussion forums, live webinars, and group chats. These features encourage learners to ask

questions, share insights, and collaborate with their peers and instructors, fostering a dynamic learning environment.

4. Certification:

Learners who complete courses or meet specific milestones can earn certificates or badges, which can be downloaded or added to professional portfolios. These certifications are valuable for career advancement or as a record of achievement.

5. Accessibility:

The platform is fully responsive, ensuring content is accessible across a wide range of devices, from desktops to smartphones and tablets. This multi-device compatibility allows learners to study at their convenience, wherever they are.

6. Self-Paced Learning:

Courses are designed to be flexible, enabling learners to study at their own pace. This flexibility is perfect for individuals with busy schedules or those who prefer to set their own learning goals.

7. Payment & Subscription Models:

While many courses are free, premium content is available via various pricing models, such as one-time payments or subscription plans. Secure payment processing is integrated to allow for seamless transactions.

Scenario-Based Case Study: Learning a New Skill

- User Registration:**

Sarah, a student looking to learn web development, creates an account on OLP. She provides basic information, including her email address and password.

- Browsing Courses:**

Upon logging in, Sarah is greeted with a well-organized homepage that displays a wide array of courses. She filters the list by category and difficulty, eventually choosing the "Web Development Fundamentals" course.

- Course Enrollment:**

Sarah reads through the course description, reviews the syllabus, and decides to enroll. She gains immediate access to the course materials, which include video lectures, written resources, and assignments.

- Learning Progress:**

As Sarah proceeds through the course at her own pace, the platform remembers her progress, allowing her to resume from where she left off after a break.

- Interactive Elements:**

Throughout the course, Sarah participates in live webinars, engages in forum discussions, and interacts with her peers and instructors. She can also ask questions and receive feedback.

- **Course Completion and Certification:**
After completing the final exam and all assignments, Sarah receives a digital certificate of completion, which she can download or share on LinkedIn.
- **Advanced Courses and Payment:**
Sarah discovers an advanced course in web development that requires a subscription. She makes a payment using the platform's secure payment system and gains access to additional premium content.
- **Instructor Role:**
John, a web developer, creates and uploads advanced courses on web development, monitors course progress, and provides support to learners throughout the course.
- **Admin Oversight:**
The platform's administrator oversees course content, user management, and platform maintenance, ensuring smooth operation and quality control.

Technical Architecture

The **Online Learning Platform** uses a client-server model with a robust and scalable architecture, ensuring seamless user interaction and efficient data management. The technical stack includes:

1. Frontend:

The frontend is built using modern web technologies and libraries to deliver an engaging and responsive user interface.

- **React.js:** Provides a dynamic and interactive experience, making it easier to build user interfaces.
- **Material UI & Bootstrap:** These UI frameworks are used to enhance the visual appeal and responsiveness of the platform, ensuring that the platform works well on both desktop and mobile devices.
- **Axios:** This library is used to handle asynchronous HTTP requests, enabling seamless communication with the backend via RESTful APIs.

2. Backend:

The backend is powered by the **Express.js** framework, which is a lightweight and efficient tool for building RESTful APIs and handling server-side logic.

- **Node.js:** A JavaScript runtime that allows for scalable backend processing.
- **Express.js:** Handles API requests, user authentication, and data routing.

- **MongoDB:** A NoSQL database that stores user data, course materials, user progress, and other dynamic content. MongoDB provides scalability and flexibility, making it a great choice for an online platform with evolving content.
3. **Payment Gateway Integration:**
The platform integrates with secure payment systems, such as **Stripe** or **PayPal**, to process payments for paid courses or subscriptions.
 4. **Real-Time Features:**
To enhance the interactivity of the platform, features like live webinars and real-time chat are implemented using **WebSockets** and **Socket.IO**.

Benefits of the Platform

- **Flexibility:** Learners can access courses anytime, anywhere, and at their own pace.
- **Interactive Learning:** Engaging discussions, live webinars, and interactive quizzes provide a well-rounded learning experience.
- **Scalability:** The platform's architecture supports a growing number of users and courses without sacrificing performance.
- **Security:** Strong security measures ensure that user data and payment information are kept secure.
- **Personalization:** With its intuitive UI and user-specific recommendations, the platform offers a customized learning journey for each user.

PRE-REQUISITES:

Here are the key prerequisites for developing a full-stack application using Node.js, Express.js, MongoDB, React.js:

✓ Node.js and npm:

Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the server-side. It provides a scalable and efficient platform for building network applications.

Install Node.js and npm on your development machine, as they are required to run JavaScript on the server-side.

Download: <https://nodejs.org/en/download/>

Installation instructions: <https://nodejs.org/en/download/package-manager/>

✓ Express.js:

Express.js is a fast and minimalist web application framework for Node.js. It simplifies the process of creating robust APIs and web applications, offering features like routing, middleware support, and modular architecture.

Install Express.js, a web application framework for Node.js, which handles server-side routing, middleware, and API development.

Installation: Open your command prompt or terminal and run the following command:

```
npm install express
```

✓ MongoDB:

MongoDB is a flexible and scalable NoSQL database that stores data in a JSON-like format. It provides high performance, horizontal scalability, and seamless integration with Node.js, making it ideal for handling large amounts of structured and unstructured data.

Set up a MongoDB database to store your application's data.

Download: <https://www.mongodb.com/try/download/community>

Installation instructions: <https://docs.mongodb.com/manual/installation/>

✓ React.js:

React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications.

Install React.js, a JavaScript library for building user interfaces.

Follow the installation guide: <https://reactjs.org/docs/create-a-new-react-app.html>

✓ HTML, CSS, and JavaScript: Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.

✓ Database Connectivity: Use a MongoDB driver or an Object-Document Mapping (ODM) library like Mongoose to connect your Node.js server with the MongoDB database and perform CRUD (Create, Read, Update, Delete) operations. To Connect the Database with Node JS go through the below provided link:
<https://www.section.io/engineering-education/nodejs-mongoosejs-mongodb/>

✓ Front-end Framework: Utilize Reactjs to build the user-facing part of the application, including entering complaints, status of the complaints, and user interfaces for the admin dashboard.

For making better UI we have also used some libraries like material UI and bootstrap.

✓ Version Control: Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.

Git: Download and installation instructions can be found at:

<https://git-scm.com/downloads>

✓ Development Environment: Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.

- Visual Studio Code: Download from <https://code.visualstudio.com/download>

To run the existing Video Conference App project downloaded from GitHub:

Follow below steps:

Clone the Repository:

- Open your terminal or command prompt.
- Navigate to the directory where you want to store the e-commerce app.
- Execute the following command to clone the repository:

git clone: <https://github.com/awdhesh-student/complaint-registry.git>

Install Dependencies:

- Navigate into the cloned repository directory:
cd complaint-registry
- Install the required dependencies by running the following commands:

```
cd frontend  
npm install  
cd ..\backend  
npm install
```

Start the Development Server:

- To start the development server, execute the following command:
npm start
- The online complaint registration and management app will be accessible at
<http://localhost:3000>

You have successfully installed and set up the online complaint registration and management app on your local machine. You can now proceed with further customization, development, and testing as needed.

Roles and Responsibilities:

The project has two types of users – Agents and Customer and third is Admin which takes care to all the user whether it is Agent or simple user. The roles and responsibilities of these two types of users can be inferred from the API endpoints defined in the code. Here is a summary:

Customer/Ordinary:

- Create an account and log in to the system using their email and password.
- Browse and fill the form of your complaint or any issues for the agent to solve.
- After filling your complaint, he/she can view the status of complaint in the status section.
- He/She can connect to the agent directly by sending message and talk more about the complaints by using chat window.
- Manage their profile information, including personal details and shipping addresses.

Agent:

- Create an account and log in to the system using their email and password.
- Manage all the complaints assigned by the Admin.
- He/She can connect directly to the user of the complaint by sending the message through chat window.

- If complaints are resolved, he can change the status by clicking the button a
- Interact with customers by responding to inquiries, resolving issues, and addressing feedback.

Admin:

- Manage and monitor the overall operation of the complaint registering platform.
- Monitor and moderate all the complaints that are coming from the user
- Easily assigned the complaints to the desired agent.
- Manage user as well as agents accounts.
- Implement and enforce platform policies, terms of service, and privacy regulations.
- Continuously improve the platform's functionality, user experience, and security measures.

Project Flow:

Milestone 1: Project setup and configuration.

- Folder setup:
 - Create frontend and
 - Backend folders
-
- Installation of required tools:
 1. Open the frontend folder to install necessary tools

For frontend, we use:

- React JS
- Bootstrap
- Material UI
- Axios

2. Open the backend folder to install necessary tools

For backend, we use:

- cors
- express
- mongoose

Milestone 2: Backend Development

- Setup express server
- Create index.js file in the server (backend folder).
- define port number to access it.
- Configure the server by adding cors, body-parser.
- Configure MongoDB
- Import mongoose.
- Add Database URL to the config.js file.
- Connect the database to the server.
- Create a ‘schema’ file in the server to store all the DB models.
- Add authentication
- Create the “User Schema” model for the MongoDB.
- Define registration & login activities.
- Using Axios library, make request from the frontend and vice-versa.

Milestone 3: Web Applications development

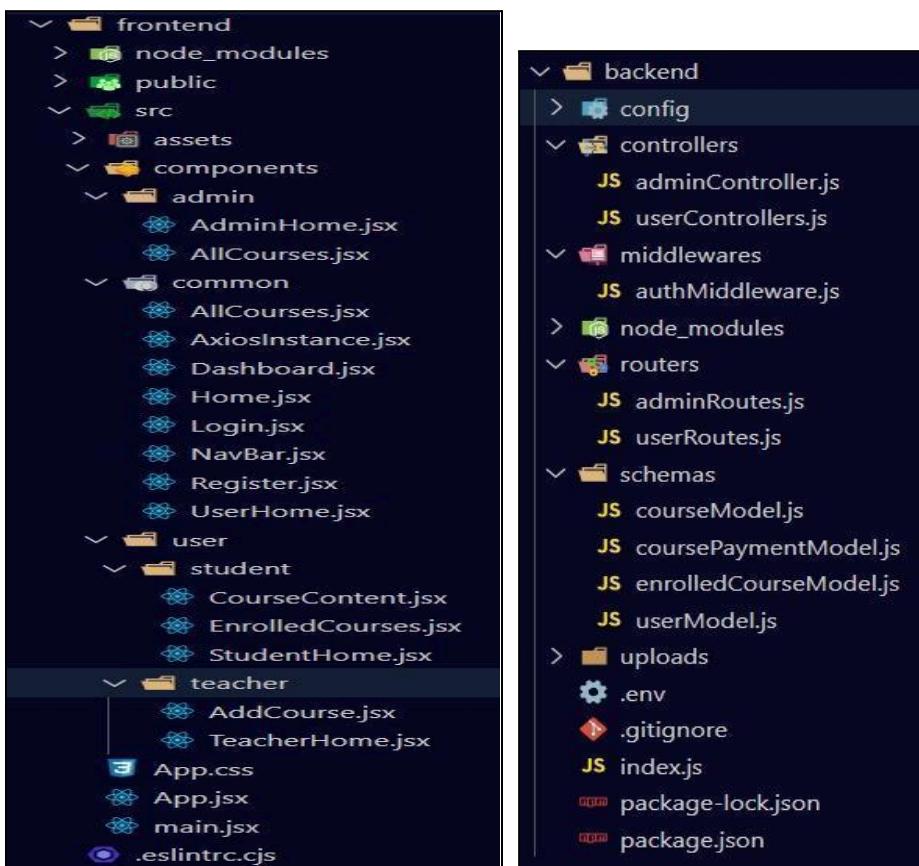
- Add complaints and the status of the complaints
- Create the “complaint_schema” model for the MongoDB to register complaint
- Defined well RESTful API for better fetching the records
- Sending message to agent through chat window
- Create the “message” model for the MongoDB to sending more description about the complaint easily.
- It can keep records by using various attributes present in rest of the collections.

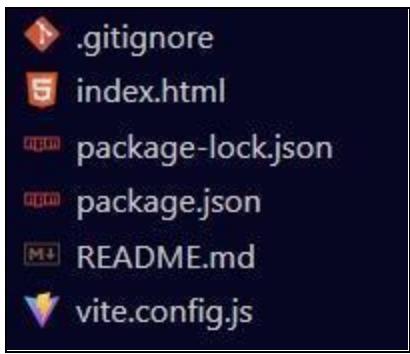
- The chat window can adjust himself with the latest message arrived at the chat window.
- CRUD operation are done by admin
- Admin has the right to perform deletion, updating or add user in the database.
- He can assign the issues or complaints by himself to the right agents present in collection.

Project Structure

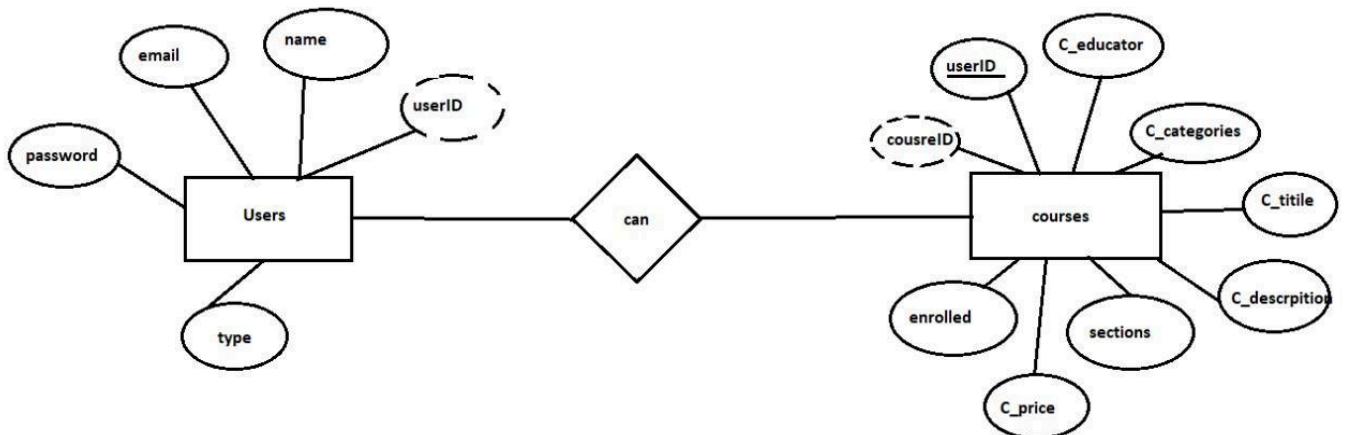
The first image is of frontend part which is showing all the files and folders that have been used in UI development

The second image is of Backend part which is showing all the files and folders that have been used in backend development





ER DIAGRAM



Here there is 2 collections namely users, courses which have their own fields in

Users:

1. _id: (MongoDB creates by unique default)
2. name
3. email
4. password
5. type

Courses:

1. userID: (can act as a foreign key)
2. _id: (MongoDB creates by unique default)
3. C_educator
4. C_categories
5. C_title
6. C_description
7. sections
8. C_price
9. enrolled

Setup Instructions

Prerequisites:

Ensure you have the following installed:

- Node.js (v14 or higher)
- npm or yarn

Installation:

Clone the repository:

```
git clone  
https://github.com/Sumitrazz/StudyApp-naan-mudhalvan-Project.git
```

1. cd StudyApp-naan-mudhalvan-Project

Install dependencies:

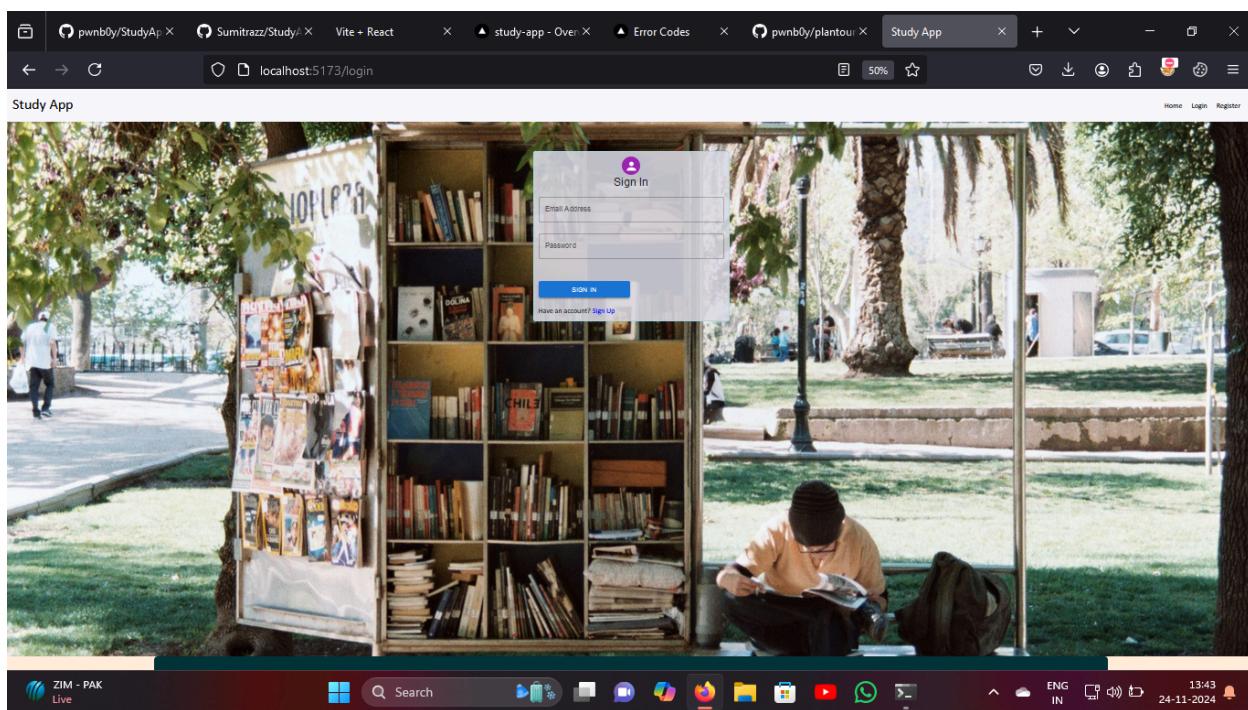
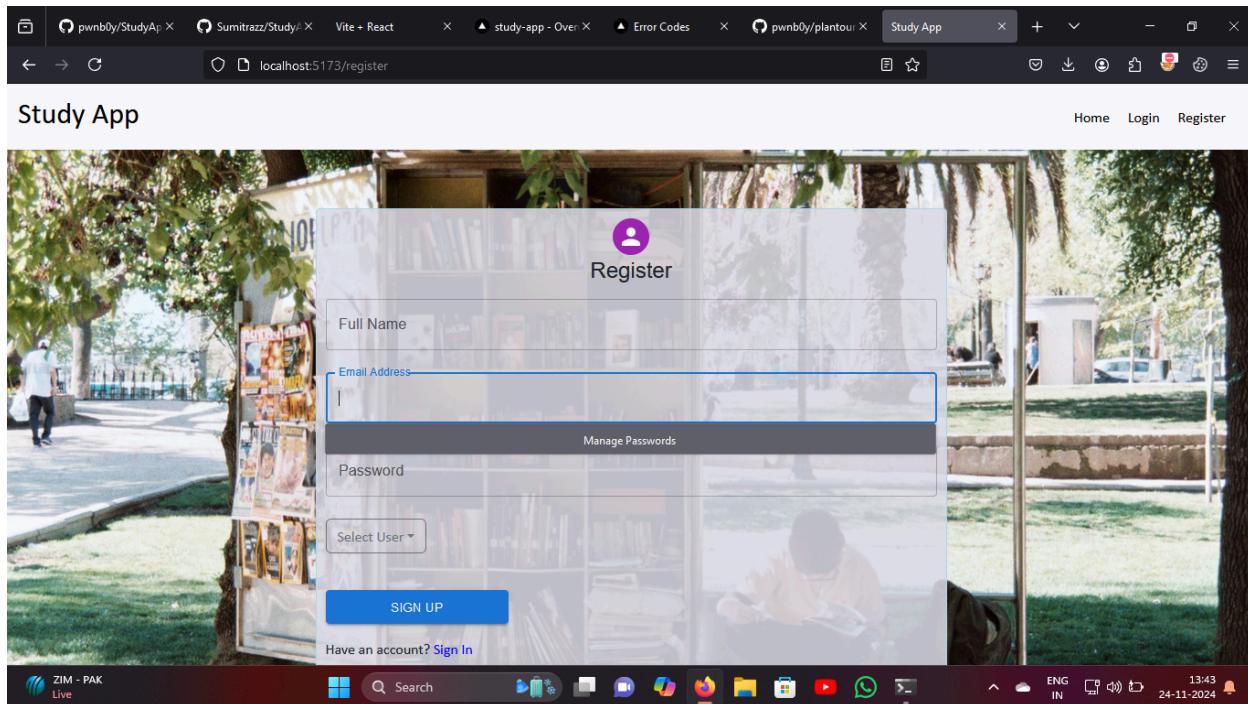
```
npm install  
# or  
yarn install
```

Set up environment variables:

Create a `.env` file in the root directory and add the following:

```
MONGO_URI=your_mongo_db_uri  
JWT_SECRET=your_jwt_secret
```

Live Demo



Why SimplyBook.me is an ideal fit for your educational and training businesses scheduling

- Reminders**
Don't let your students forget a date and time for a lesson or course. Remind them where they need to be in time to get them with automated and customizable reminders.
- Calendar Sync**
Don't let teaching and meeting times clash with your personal life and vacation time. Synchronize your personal and professional calendar to avoid double booking yourself. Each teacher can sync their personal calendars with their teaching schedules.
- Intake Forms**
If you need student or pupil information before they join your classes, you can request and create lesson specific intake forms to ensure they are taking the most appropriate course.
- Kiosk**
Let your students book themselves into available classes on site with a security enhanced kiosk system. They can log in and join a class quickly. Once logged in, the system will forget all of their details and login information when they have finished.
- Code Colours for Teachers**
Create colour coordinated schedules for your teaching staff to more easily identify upcoming lessons, and overbooked schedules.
- Group Booking**
Allow companies and organizations to group book important courses and seminars for multiple staff members with the group bookings feature.
- Daily Reports**
Maintain detailed daily reports on your teaching schedule and external meeting, get it all on a report created for a daily overview of your trainer schedules.

LEARN ANYTHING

Benefits of Online Learning Expertise

- Flexible Learning**
Learn at your own pace with access to materials anytime, anywhere.

StudyApp - Stay tune and get the latest update

For far away, behind the word mountains

Enter email address

© 2024 Copyright: StudyApp