ONLINE LEARNING PLATFORM

TEAM MEMBERS

- Aravinth S - 2021503006
- Pratap V M - 2021503536
- Puviyarasan V K - 2021503038
- Nithish T – 2021503032
- Santhosh Kumar S - 2021503302

Purpose

The purpose of this project is to create an **Online Learning Platform** that bridges the gap between learners and educators, aproviding an efficient and interactive environment for e-learning. The platform enables learners to explore courses, enroll in classes, and track their progress while allowing educators to create, manage, and deliver courses seamlessly.

- Offer a comprehensive and interactive learning experience.
- Provide educators with tools to manage courses and track learner performance.
- Ensure secure user interactions with real-time communication.
- Leverage a scalable MERN stack architecture to handle large user bases and diverse content.

Features

- **Course Listings:** Detailed course information, including syllabus, duration, instructor details, and pricing.
- **Search and Filters:** Search courses by category, instructor, price, ratings, and language.
- **User Registration and Profiles:**
    - Learners: Profile with enrolled courses, progress tracking, and certifications.
    - Educators: Profile with course creation and earnings dashboard.
- **Course Management:** Educators can add, update, and delete courses.
- **In-app Communication:** Chat and discussion forums for learners and instructors.
- **Progress Tracking:** Interactive dashboards showing learners' progress, quiz scores, and milestones.
- **Admin Panel:** Manage user accounts, course approvals, and platform analytics.

- **Payment Gateway Integration:** Secure payment system for course purchases.
- **Notifications:** Real-time notifications for course updates, reminders, and announcements.
- **Analytics for Educators:** Insights into course engagement, ratings, and earnings.

Architecture

Frontend

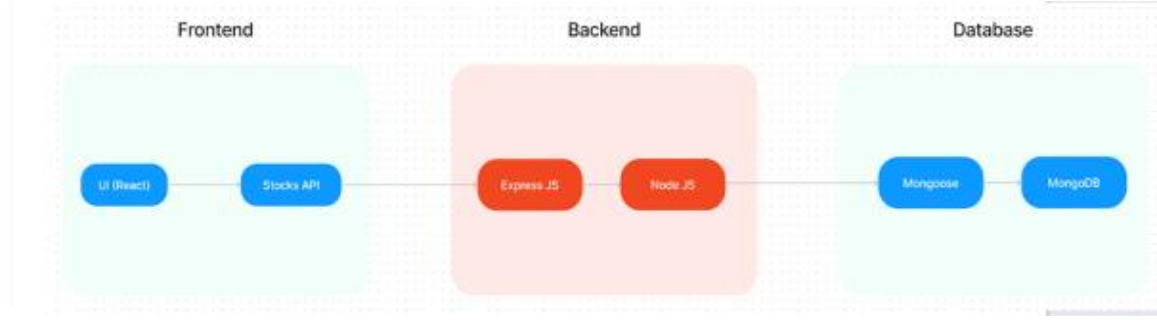
- Built with React, styled using Bootstrap and Material UI for a responsive, user-friendly interface.
- Features include course exploration, dashboard management, and progress visualization.
- Axios handles API requests to ensure smooth communication with the backend.

Backend

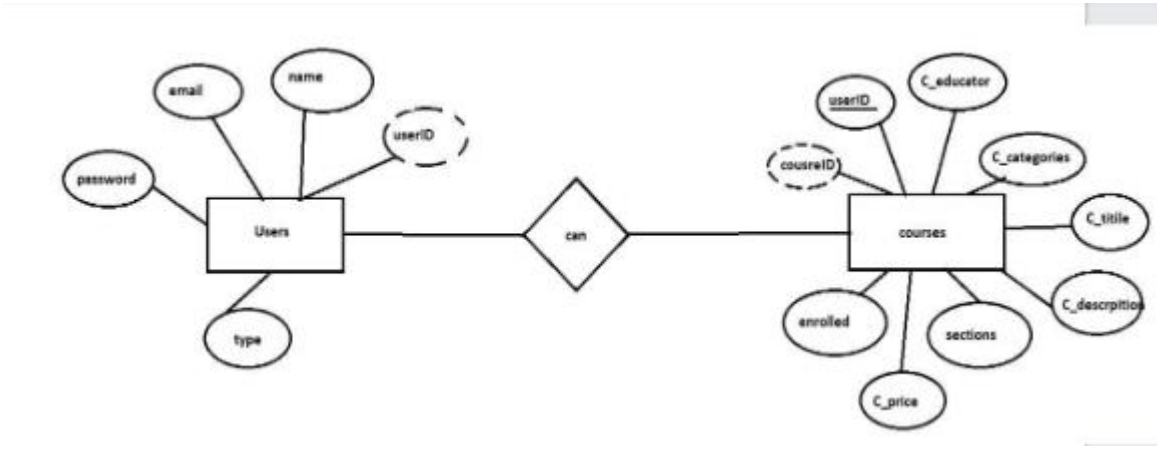
- Developed with Node.js and Express.js for server logic and API management.
- Handles user authentication, course management, communication tools, and payment processing.

Database

- MongoDB stores user profiles, course data, transaction records, and chat messages.
- Flexible document-based storage enables scalability for diverse content and large user volumes.



ER-Diagram



Setup Instructions

Prerequisites

- Node.js (version 14+)
- MongoDB (local or remote server setup)
- NPM (Node Package Manager)

Installation

1. **Clone the Repository:**

   ```
   git clone <repository-url>
   ```

2. **Navigate into Project Folder:**

   ```
   cd online-learning-platform
   ```

3. **Install Dependencies:**
   - **Client:**

     ```
     cd client && npm install
     ```

   - **Server:**

     ```
     cd server && npm install
     ```

4. **Set Environment Variables:**
   - Create a `.env` file in the server directory with values for `MONGO_URI`, `JWT_SECRET`, etc.

Authentication and Authorization

- **JWT (JSON Web Tokens):** Secure authentication for users with tokens stored on the client side.
- **Role-based Access Control:** Differentiated access for learners, educators, and admins.

Folder Structure

Client

- **`/src`:** Contains all frontend files.
    - **`/components`:** Reusable components (e.g., Navbar, Footer, CourseCard).
    - **`/pages`:** Pages for the app (Home, Course Details, and Dashboard).
    - **`/services`:** Axios API calls and services.
    - **`/styles`:** CSS or SCSS files for styling.

Server

- **`/controllers`:** Handles logic for routes (e.g., user authentication, course management).
- **`/models`:** MongoDB schemas (e.g., User, Course, Transaction).
- **`/routes`:** API endpoints (e.g., `/auth`, `/courses`).
- **`/middleware`:** Custom middleware (e.g., authentication, error handling).

Running the Application

Frontend

- **Navigate to the client directory:** `cd client`
- **Run the frontend server:** `npm start`

Backend

- **Navigate to the server directory:** `cd server`
- **Run the backend server:** `npm start`

Testing

- **Unit Testing:** Test components, services, and utility functions in both frontend and backend.
- **Integration Testing:** Validate interactions between frontend and backend APIs.
- **End-to-End Testing:** Use tools like Jest, Mocha, or Cypress for core functionalities (e.g., course purchase, user registration).

Known Issues

- **Video Streaming:** May experience delays without integrating an advanced content delivery network (CDN).
- **Mobile Responsiveness:** Some components may need optimization for smaller screens.

Future Enhancements

- **Gamification:** Add badges, leaderboards, and rewards for learners.
- **Mobile Application:** Native apps for Android and iOS platforms.
- **AI-driven Recommendations:** Suggest personalized courses based on user interests and progress.
- **Integration with Video Platforms:** Support live classes through Zoom or Google Meet.
- **Multi-language Support:** Expand course accessibility with translations.

Conclusion

This **Online Learning Platform** provides a holistic e-learning ecosystem, catering to the needs of learners and educators with intuitive design, secure transactions, and advanced analytics. Built on a scalable MERN stack architecture, it ensures robust performance and lays the groundwork for future enhancements to revolutionize digital education.

References

- GitHub Link: https://github.com/Aravinths132003/Online-Learning-Platform.git
- Demo Video Link: https://drive.google.com/file/d/1wTty4VxLh4iDXPErbl_gc-4PJ6xlFXem/view?usp=drive_link