# ONLINE LEARNING PLATFORM USING MERN

Siva Jegadeesh C B (2021503559)

Prasanna M (2021503535)

Babith Sarish S (2021503009)

Ajaykumar K (2021503003)

Santhosh D (2021503047)

### Objective:

1. Provide Accessible Learning

* Allow students to access quality education from anywhere, anytime.
* Ensure the platform is user-friendly and works across devices.

1. Support Teachers in Course Management

* Help teachers create, edit, and organize courses easily.
* Provide tools to monitor enrollments and student progress.

1. Deliver Personalized Learning

* Enable students to learn at their own pace with progress tracking.
* Offer features like course filtering for tailored experiences.

1. Streamline Certification and Assessments

* Issue digital certificates after course completion.
* Facilitate exams and assignments within the platform.

1. Enable Paid and Free Courses

* Support both free and premium courses with a secure payment system.

1. Ensure Platform Administration

* Allow admins to oversee users and manage course listings.
* Maintain platform integrity and resolve user issues efficiently.

1. Foster Interaction and Engagement

* Provide discussion forums and live webinars for collaboration.
* Encourage student-teacher and peer-to-peer communication.

1. Build a Scalable and Robust System

* Use modern technologies for a scalable and high-performance platform.
* Ensure secure and efficient data storage and management.

### Project Description :

The **Online Learning Platform (OLP)** is a comprehensive web-based application designed to offer an interactive and engaging learning experience. It connects students and teachers, allowing for the creation, management, and consumption of educational content. The platform features both free and paid courses across various topics, with functionalities for progress tracking, interactive learning, and certification upon completion.

* **For Students**: The platform provides easy access to courses with filtering options by category, difficulty, and popularity. Students can enroll in courses, track their learning progress, and receive certificates upon completion. They can also interact with instructors and peers through discussion forums and live webinars.
* **For Teachers**: Educators can create and manage courses, add sections, and monitor student enrollments and progress. They can also delete courses when necessary and provide valuable insights into student performance.
* **For Admins**: Administrators oversee platform operations, manage user roles, monitor course listings, and ensure smooth functioning of the system. They also handle user issues and maintain the integrity of the platform.

The platform is built using modern technologies such as **Node.js**, **Express.js**, **MongoDB**, and **React.js**, ensuring a seamless and scalable learning experience.

### Tech Stack Overview:

#### Frontend

#### React.js: For building dynamic and responsive user interfaces.

#### Vite: A fast build tool for rapid development and optimized production builds.

#### Material UI and Bootstrap: Libraries for creating professional, user-friendly, and visually appealing designs.

#### Axios: For handling API requests and facilitating seamless communication with the backend.

#### Backend

* **Node.js**: A powerful JavaScript runtime for building scalable server-side applications.
* **Express.js**: A minimalist web application framework for routing, middleware, and API development.

#### Database

* **MongoDB**: A NoSQL database for storing structured and unstructured data efficiently.
* **Mongoose:** An ODM library for defining schemas and handling database interactions.

#### Authentication / Authorization

* **JSON Web Tokens (JWT)**: A secure and stateless authentication method for user and admin authentication, enabling safe transmission of user credentials and authorization data.

#### Database Hosting

* **MongoDB Atlas**: A fully managed cloud service for hosting MongoDB databases, offering automated backups, scaling, and high availability to ensure robust and secure data storage.
* **Localhost (for development):** Accessible at <http://localhost:5172>.

#### Additional Libraries

* **JSON Web Tokens (JWT**): For secure user authentication.
* **bcrypt.js:** For password hashing and secure storage.

### Use Case:

### **1. Student Use Cases**

### Register and log in to access the platform.

### Browse and filter courses by name, category, difficulty level, or popularity.

### Enroll in free or paid courses (with a payment system for premium content).

### Access course content, including video lectures, assignments, and reading materials.

### Track progress and resume learning from where they left off.

### Participate in discussions through forums and live webinars.

### Complete courses and download certificates of completion after passing exams..

### **2. Teacher Use Cases**

### Create new courses by adding descriptions, syllabi, and sections.

### Edit or update existing course content as needed.

### Delete courses if there are no enrollments or for other valid reasons.

### Monitor enrollments and track the progress of students in their courses.

### **3. Admin Use Cases**

### Oversee the platform to ensure smooth operations.

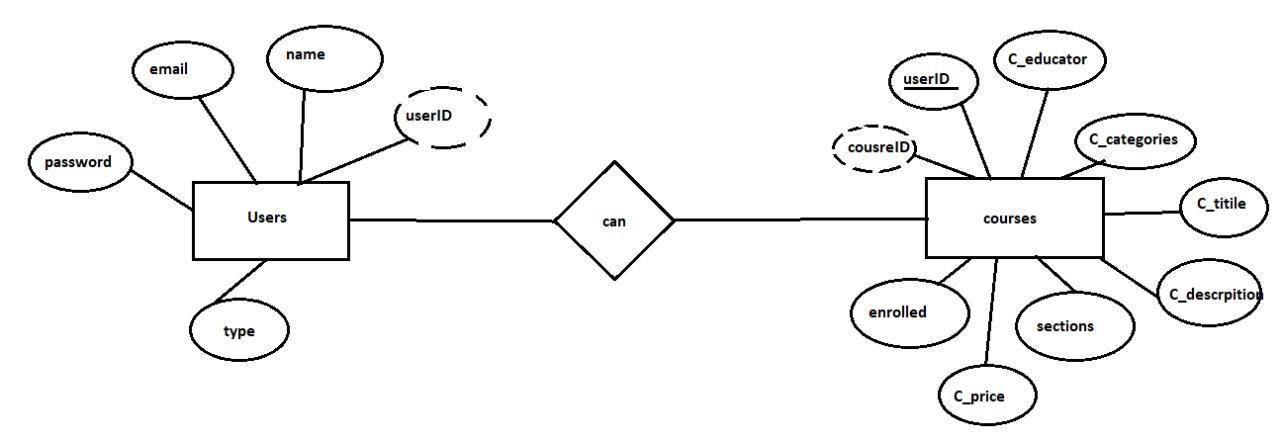
### Manage users (add, edit, or remove students or teachers).

### Monitor course listings and student enrollments.

### Handle user issues and maintain platform integrity.

### Modify or update course content as required.

### **Database Design / ER Diagram**



### **Relationships**

### User ↔ Enrollments

### A user can enroll in multiple courses.

### (One-to-many relationship: One user can have multiple enrollments)

### Courses ↔ Enrollments

### A course can have multiple students enrolled.

### (One-to-many relationship: One course can have many enrollments)

### Enrollments ↔ Payments

### Enrollments are linked to payments, especially for paid courses.

### (One-to-many relationship: One enrollment can have one payment)

### User ↔ Payments

### A user makes payments for courses.

### (One-to-many relationship: One user can make multiple payments)

### Courses ↔ Certificate

### A certificate is generated when a user completes a course.

### (One-to-many relationship: One course can issue many certificates)

### **Explanation**

### User is the central entity that interacts with various other entities, such as enrolling in courses, making payments, and receiving certificates.

### Courses represent the courses offered on the platform. They are linked to Users (specifically teachers who create the courses) and Enrollments (which track which students are enrolled).

### Enrollments record which students are enrolled in which courses and are also connected to Payments, especially for paid courses.

### Payments capture the financial transactions for paid courses, linking to both the User (who made the payment) and the Courses (which the payment is for).

### Certificate represents the successful completion of a course by a student, issued after the final exam or assessment.

### Features:

#### Student

* User Registration and Login.
* Browse and filter courses by category, name, difficulty, or popularity.
* Enroll in free and paid courses with a secure payment gateway.
* Access course materials, including video lectures, assignments, and reading materials.
* Track learning progress with the ability to resume courses.
* Participate in interactive discussion forums and live webinars.
* Receive digital certificates upon course completion and exam success.

#### Teacher

#### Create new courses with detailed descriptions, syllabi, and sections.

#### Edit or update existing course content.

#### Delete courses when necessary (e.g., no enrollments).

#### Monitor student enrollments and track their progress.

#### Add additional sections to enhance course content.

#### Admin

#### Oversee platform operations and ensure smooth functionality.

#### Manage users (students and teachers): add, edit, or remove accounts.

#### Monitor and manage course listings.

#### Track student enrollments and overall platform activity.

#### Resolve user issues and maintain platform security and integrity.

**General Platform**

### User-friendly interface for seamless navigation.

### Mobile-friendly design for access across devices.

### Secure authentication using JWT (JSON Web Tokens).

### Real-time updates with efficient database communication.

### Discussion forums and webinars for collaborative learning.

### Progress tracking to enhance personalized learning.

### Integration of both free and paid courses with a payment system.

### Scalable and robust architecture for future growth.

### **Conclusion**

The Online Learning Platform (OLP) is a robust, scalable, and user-friendly solution designed to bridge the gap between learners and educators in the digital age. By providing features such as interactive learning, personalized progress tracking, and seamless course management, it ensures an engaging and efficient educational experience. The platform's modern tech stack, including React.js, Node.js, Express.js, and MongoDB, guarantees reliable performance, security, and scalability. Its inclusive design supports the needs of students, teachers, and administrators, making it a versatile tool for both free and paid learning environments. With its emphasis on accessibility, collaboration, and continuous improvement, the OLP is well-positioned to transform the way education is delivered, empowering learners and educators worldwide to achieve their goals effectively.

### References

Github Link: <https://github.com/babith02/Online-Learning-Platform-using-MERN>

Demo Video Link: <https://docs.google.com/document/d/1UmdYRA864xc9c-f0zOiukTjeZksHyoi2Ew_DJpAzYd0/edit?usp=sharing>