

Education Platform with MERN Stack - Project Documentation

Introduction

This document provides a detailed overview of the education platform built using the MERN stack. The platform is designed to offer a variety of online courses to users, enabling them to learn different subjects at their own pace. It consists of multiple modules for users, admins, and course management. This platform includes a front-end user interface, back-end services, and a database using MongoDB for storing course and user data.

Project Structure

The MERN stack includes MongoDB, Express.js, React.js, and Node.js. The project is divided into several parts:

- Front-end: Built using React.js, responsible for rendering UI and handling user interactions.
- Back-end: Node.js and Express.js handle the business logic, API endpoints, and routing.
- Database: MongoDB is used to store and manage user and course data.
- Authentication: JWT tokens and Passport.js are used for user authentication and authorization.

Features

The platform has the following key features:

1. User Registration & Login: Users can sign up, log in, and access courses based on their subscription.
2. Course Catalog: Users can browse through various courses with details like description, instructor, and pricing.
3. Admin Dashboard: Admins can manage courses, user accounts, and course subscriptions.
4. Video Lessons: Each course contains video lessons that users can view as part of their learning experience.

5. Course Progress Tracking: Users can track their progress, and admins can see student performance.
6. Search and Filter: Users can search for courses based on categories, difficulty, and ratings.

User Module

The User module includes the following functionality:

1. User Registration: Users can create an account by providing their name, email, and password. Passwords are hashed and stored securely.
2. Login/Logout: After registration, users can log in and access their personalized course dashboard. JWT tokens are used for authentication.
3. Browse Courses: Users can view all available courses and filter them based on categories, duration, or pricing.
4. Enroll in Courses: Users can enroll in a course, which allows them to access all the content provided within it.
5. Course Progress: The platform tracks the user's progress in each course, including completed lessons and overall completion percentage.

Admin Module

The Admin module allows platform administrators to manage the entire system. Key features include:

1. Admin Authentication: Admins can log in using their credentials, which are stored securely.
2. Manage Users: Admins can view, edit, and delete user accounts, including changing roles or resetting passwords.
3. Course Management: Admins can add, update, or delete courses from the platform. Each course has details such as title, description, instructor, price, and lesson videos.
4. Course Enrollment: Admins can manage which users are enrolled in which courses and can

assign course completion certificates.

5. Report Generation: Admins can generate reports on user activity, course performance, and revenue from paid courses.

Database Schema

The database is structured using MongoDB to store data related to users, courses, and enrollments.

The key collections are:

1. Users: Stores user data, including name, email, hashed password, and enrolled courses.
2. Courses: Contains details about each course such as title, description, instructor, and lessons.
3. Enrollments: Tracks which users are enrolled in which courses and their progress.
4. Reports: Stores information on user activities and course completion.

API Endpoints

The platform exposes several REST API endpoints for interaction between the front-end and back-end.

Some of the key endpoints include:

1. POST /api/users/register: Registers a new user.
2. POST /api/users/login: Logs in a user and returns a JWT token.
3. GET /api/courses: Retrieves a list of all available courses.
4. GET /api/courses/{id}: Retrieves details of a specific course.
5. POST /api/courses/enroll: Allows a user to enroll in a course.
6. GET /api/users/{id}/progress: Retrieves the progress of a user in their enrolled courses.