Project Documentation: Online Learning Platform

Project Overview

This project aims to build an online learning platform where students can enroll in courses and instructors can create and manage course content. The platform supports multiple user roles (students, instructors, and approvers), with role-based access control powered by JSON Web Token (JWT) authentication. The platform also incorporates a course approval system managed by users with the 'approver' role.

Features

- **Authentication & Authorization**: Secure JWT-based login system for different user roles.
- **User Roles**: Students, instructors, and approvers with different permissions.
- **Course Management**: Instructors can create and manage courses, while approvers review and approve courses.
- **Enrollment**: Students can browse approved courses and enroll in them.
- **Progress Tracking**: Track student progress in enrolled courses.

System Architecture

The system follows a standard client-server architecture. Here's a breakdown of the core components:

1. **Frontend (UI/UX)**:
    - Built with HTML, CSS, and JavaScript, providing user-friendly interfaces for each role (student, instructor, and approver).
    - Handles user interaction, form submission, and data presentation.
2. **Backend (API)**:
    - The backend is developed using Node.js with Express for RESTful API development.

    - Handles user authentication, course creation, enrollment, and approval logic.
3. **Database**:
    - Uses a relational database like MySQL or PostgreSQL to store user data, course details, and enrollment records.

Core Components and Functionality

1. Authentication and Authorization

- **JWT-Based Authentication**:
    - Upon successful login, the system generates a JWT for the user, which is stored client-side and used for subsequent requests.
    - Each token contains information about the user role (student, instructor, or approver) to determine access rights.
- **User Roles**:
    - **Students**: Can view and enroll in approved courses, and track their progress.
    - **Instructors**: Can create, update, and delete courses but need approval from approvers for courses to be published.
    - **Approvers**: Review and approve/reject pending courses created by instructors.

2. User Management

- **Registration & Login**:
    - Users can register with a unique email and password.
    - Passwords are hashed using bcrypt for security.
- **Profile Management**:
    - Users can update their personal details and change their passwords.

3. Course Management

- **Create, Update, and Delete Courses**:
    - Instructors can create new courses with a 'Pending' status by default.

    - They can also update course details and delete their own courses.
- **Course Approval**:
    - Approvers have the ability to review pending courses and change their status to 'Approved', making them visible to students.

4. Content Management

- **Upload Course Materials**:
    - Instructors can upload course-related materials like PDFs, videos (to be implemented), and other resources.
- **View Course Content**:
    - Students enrolled in a course can access all course materials.

5. Enrollment & Progress Tracking

- **Enroll in Approved Courses**:
    - Students can browse and enroll in approved courses.
- **Track Progress**:
    - The system tracks which course materials have been completed by each student and displays progress as a percentage.

6. Security

- **JWT Authentication**:
    - JWT ensures that each request is made by an authenticated user and provides role-based access control.
- **HTTPS**:
    - Secure communication with HTTPS to protect sensitive information during data transmission.

Database Design

The platform uses a relational database for storing and managing user, course, and enrollment data. Below is the simplified entity-relationship (ER) diagram.

ER Diagram

```
+----------------+       +----------------+       +------------------+
|    Users       |       |    Courses     |       |   Enrollment     |
+----------------+       +----------------+       +------------------+
| user_id (PK)   |       | course_id (PK)  |       | enrollment_id (PK)|
| name           |       | title           |       | user_id (FK)      |
| email          |       | description     |       | course_id (FK)    |
| password_hash  |       | instructor_id (FK)|      | progress          |
| role           |       | status (Pending/Approved) | date_enrolled    |
+----------------+       +----------------+       +------------------+

Relationships:
- Users (Instructors) can create Courses (1:M)
- Users (Students) can enroll in Courses (M:M)
```

User Flow Diagrams

1. User Registration and Login

```
+---------+       +---------+       +---------+
|  User   | -----> |  Front  | -----> |  API    |
|         | <----- |  End    | <----- |         |
+---------+       +---------+       +---------+
            (1) User submits registration details
            (2) Frontend sends details to API
            (3) API registers user and returns JWT upon login
```

2. Course Creation and Approval Flow

```
+------------+      +---------+      +---------+      +------------+
| Instructor | ----> |  Front  | ----> |  API    | ----> |  Approver  |
|            |       |  End    |       |         |       |            |
+------------+       +---------+       +---------+       +------------+
  (1) Instructor creates a new course
  (2) Course status is 'Pending'
  (3) Approver reviews and updates status to 'Approved'
```

Endpoints Documentation

Endpoint	Method	Description	Role
/register	POST	Registers a new user	All users
/login	POST	Authenticates a user	All users
/courses	GET	Fetches all approved courses	Student
/courses/pending	GET	Fetches all pending courses	Approver
/courses	POST	Creates a new course	Instructor
/courses/:id/approve	PATCH	Approves a pending course	Approver
/courses/:id/enroll	POST	Enrolls a student in a course	Student
/courses/:id/progress	GET	Fetches progress for enrolled course	Student

---

Technology Stack

- **Frontend**: HTML, CSS, JavaScript (React)

- **Backend**: Node.js, Express.js
- **Database**: MySQLL
- **Authentication**: JWT for secure authentication
- **Password Hashing**: Bcrypt for password security

Conclusion

This project provides a robust foundation for an online learning platform with secure authentication and role-based access control. It implements essential features such as course creation, enrollment, and progress tracking, offering a scalable solution for educational content delivery. Future improvements include video integration, enhanced UI, and additional security features.