**PROF FINDER**

**COURSE REVIEW PLATFORM USING AI**

**System Architecture for Course Review Platform with AI**

Here's the system architecture for your Course Review Platform, outlining the different components and how they interact with each other:

**1. Frontend (React + TailwindCSS + Redux)**

* **Description**: The frontend is the user-facing part of the platform, developed with React.js and styled using TailwindCSS and Chakra UI for responsiveness and aesthetics. It communicates with the backend via REST APIs or WebSockets to fetch data such as courses, reviews, and user profiles.
  + **Components**:
    - **Login/Signup Page**: Handles user authentication (JWT-based).
    - **Dashboard**: Displays personalized course recommendations and review activity.
    - **Course Browsing Page**: Displays course listings with filter and search options.
    - **Course Details Page**: Displays course description, prerequisites, reviews, and ratings.
    - **Review Section**: Allows users to submit and view reviews.
    - **Profile Management**: Allows users to edit and manage their profiles and academic preferences.

**2. Backend (Node.js + Express)**

* **Description**: The backend is built using Node.js and Express.js to serve APIs to the frontend. The backend handles user authentication, course management, and interactions with the AI features for personalized recommendations and sentiment analysis.
  + **Components**:
    - **User Authentication**:
      * JWT Authentication for secure login, signup, and role-based access.
    - **API Endpoints**: RESTful API to handle CRUD operations for courses, users, and reviews.
      * **User Management**: Sign up, login, profile update.
      * **Course Management**: Create, read, update, delete courses.
      * **Review Management**: Post, update, delete reviews.
    - **Authorization Middleware**: Ensures that only authorized users can access specific resources (e.g., admin panel).

**3. Database (MongoDB)**

* **Description**: MongoDB is used to store persistent data such as user profiles, courses, and reviews. The database is designed to be flexible and scalable to handle various types of data.
  + **Collections**:
    - **Users**: Stores user details such as profile, academic preferences, and review history.
    - **Courses**: Stores course details including the title, description, instructor, difficulty, and reviews.
    - **Reviews**: Stores reviews submitted by users, including sentiment score, rating, and review text.

**4. AI Features (Recommendation Engine + Sentiment Analysis)**

* **Description**: AI-powered features are integrated into the platform to offer personalized course recommendations and analyze reviews. These features help students receive suggestions based on their preferences and academic history.
  + **Components**:
    - **Course Recommendation Engine**: Uses collaborative filtering and content-based filtering algorithms to suggest courses based on student preferences and previous reviews.
    - **Sentiment Analysis**: Applies NLP techniques to analyze reviews and determine whether they are positive, neutral, or negative.
    - **Personalized Suggestions**: Recommends courses based on students’ review history and academic preferences5..

**5.Real-Time Updates (WebSockets)**

* **Description**: The platform supports real-time updates for reviews and recommendations. This is crucial for displaying new reviews or real-time course suggestions as soon as users post their reviews or interact with the platform.
  + **Components**:
    - **WebSocket Server**: Handles live communication between users and the server.
    - **Client-Side WebSocket**: Updates the frontend with new reviews or recommendations.

**6. Admin Interface**

* **Description**: The admin panel allows administrators to manage courses, reviews, and users. Admins have full control over course listings, review moderation, and analytics.
  + **Components**:
    - **Course Management**: Admins can add, update, or delete courses.
    - **User Management**: Admins can manage user roles, delete users, and monitor their activities.
    - **Review Moderation**: Admins can approve, delete, or flag inappropriate reviews.

**SYSTEM ARCHITECTURE**

User Interface (React + Redux) <----> Admin Interface (Admin Panel)

|

| (HTTP Requests / WebSockets)

|

Backend (Node.js + Express) <----> Database (MongoDB)

|

(JWT Auth / REST API)

|

AI Features ( Recommender)

|

(Sentiment Analysis, Collaborative Filtering, etc.)