**1. User Roles and Authentication:**

* **Admin**: Manages users (students, teachers), quiz templates, feedback, community moderation, system-wide configurations.
* **Teacher**: Can create classes, track student progress, interact with students in the community, assign custom quizzes, and view quiz results.
* **Student**: Has a personalized dashboard that provides skill recommendations, allows quiz-taking, tracks progress, and interacts with other users in the community.

**Technologies**:

* **React** for the frontend (authentication, dashboard, community).
* **Express.js** for backend APIs (user management, authentication, skill tracking).
* **JWT** or **OAuth** for authentication and session management.

**2. Student Profile and Learning Path:**

* When students sign up, they complete a detailed profile that includes:
  + **Strengths and Interests**: Allows the AI system to recommend learning pathways based on this data.
  + **Current Study**: Helps tailor learning content that matches the student’s current education level.
  + **Goals**: Long-term and short-term goals for personalized recommendations.

**Feature**: Based on this data, the AI system generates a **personalized learning pathway** that suggests courses and content to help the student progress.

**3. Daily Quizzes and Skill Assessment:**

* **Quiz Customization**: Students can set the frequency of quizzes (daily, every 2 days) and define their skill focus. The quiz content is dynamically generated based on the student’s profile.
  + **Quiz Features**: AI-generated questions (15-50 questions), personalized to each student's skill level and interests.
  + **Quiz Notifications**: Push notifications or reminders are sent when it's time for the quiz.
* **Results and Feedback**:
  + After each quiz, the system provides a detailed breakdown of performance by skill category (e.g., critical thinking, problem-solving).
  + If a student performs poorly in a particular area, their strength score for that skill decreases on the profile, triggering more quizzes and recommendations to improve.

**Technologies**:

* **AI** with Python (e.g., GPT or other NLP models) to dynamically generate quizzes and provide feedback.
* **React Notifications API** for reminders and notifications.
* **Express.js** and **MongoDB** to store quiz results and update user progress.

**4. AI-Powered Intelligent Support System:**

* This system will provide real-time feedback and answers to students’ questions (similar to a chatbot).
  + **Question Answering**: Students can ask questions related to course content, and the system will provide AI-generated responses.
  + **English Pronunciation and Grammar Check**: Students can input text, and the AI checks grammar, pronunciation (if integrated with speech-to-text APIs), and provides suggestions for improvement.

**Technologies**:

* **Python AI Models** (e.g., GPT for question-answering, grammar checks).
* **Speech-to-Text API** for pronunciation check (Google Cloud Speech-to-Text, Amazon Polly).
* **React** for the frontend interface where students interact with the AI support system.

**5. Essay Grading System:**

* Students can submit essays or written assignments, and the AI system will grade based on factors like:
  + Grammar and spelling.
  + Structure and argumentation.
  + Content relevance to the topic.
* **Feedback Generation**: Detailed feedback provided after grading, helping students improve their writing skills.

**Technologies**:

* **NLP models** (such as BERT or GPT) for essay grading and feedback generation.

**6. Skill Tracking and Progress Monitoring:**

* The system continuously tracks students' progress based on quizzes, assignments, and activities.
  + **Strength Reduction**: If a student performs poorly in a skill area consistently, the system updates the strength percentage in their profile.
  + **Improvement Suggestions**: Based on progress, the system recommends courses or quizzes to focus on specific weak areas.

**Technologies**:

* **MongoDB** to store progress data.
* **Python AI models** for analysis and progress recommendations.

**7. Community and Collaboration:**

* A community section where students and teachers can discuss topics, ask questions, and collaborate.
* **Features**:
  + Discussion boards.
  + Question and answer sessions.
  + Teacher moderation capabilities to guide students.

**Technologies**:

* **WebSocket** or **GraphQL Subscriptions** for real-time messaging.
* **React** for building the community interface.

**8. Teacher Dashboard:**

* Teachers can create classes and add students.
* **Progress Monitoring**: Teachers can track quiz participation, performance, and areas where students need improvement.
* Teachers can also interact in the community, answering questions or giving feedback on essays or quizzes.

**Technologies**:

* **React** for the teacher dashboard interface.
* **Express.js** for backend APIs related to teacher-student interaction and progress tracking.

**9. Admin Panel:**

* Admins can manage users, view system stats (like the number of quizzes taken, performance metrics), and moderate community discussions.
* **System-wide Settings**: Admins can adjust settings for quiz generation, feedback algorithms, and community management.

**Technologies**:

* **React Admin Dashboard** for managing users and system settings.
* **Express.js** and **MongoDB** for backend operations.

**10. AI-Generated Content:**

* **Quiz Generation**: Use AI models to dynamically create questions based on the student’s profile, strengths, and past quiz performance.
* **Feedback Generation**: AI-powered feedback for quizzes, essays, and general performance, which can adapt over time as students improve.

**Technologies**:

* **Python AI/ML** libraries for NLP and question generation (e.g., GPT-3, NLTK, or spaCy for linguistic tasks).
* **React** and **Express.js** for seamless integration between AI and the frontend.

**Overall Technology Stack:**

* **Frontend**: React (with Bootstrap or Material UI for UI components)
* **Backend**: Express.js (Node.js) for API routes and logic
* **Database**: MongoDB (NoSQL for storing user data, quiz results, etc.)
* **AI/ML**: Python (for NLP, essay grading, quiz generation, etc.)
* **Authentication**: JWT or OAuth (with roles for Admin, Teacher, Student)
* **Real-Time Communication**: WebSocket or GraphQL for community chats
* **Notification System**: Push notifications (Browser API or mobile notification services)

This complete system not only personalizes learning for students but also tracks their progress, continuously adjusts recommendations, and fosters a collaborative environment with AI-generated quizzes and feedback, making it a well-rounded educational platform.