EduTutor AI: Personalized Learning with Generative AI and LMS Integration

---

1. Introduction

Project title : EduTutor AI-Personalized Learning Platform

Team members:

- Priya Dharshini B
- Kayathri J
- Krithika sri R
- Preethi B
- Sunil P

2. Project Overview

Purpose

The purpose of EduTutor AI is to transform traditional education into a personalized, adaptive, and data-driven learning experience. Many existing educational systems follow a uniform approach that fails to accommodate the diverse learning needs of students. EduTutor AI addresses this gap by leveraging generative AI models and seamless LMS integration to provide tailored learning paths for each learner. For students, the platform delivers dynamic quizzes, adaptive feedback, and personalized recommendations that align with their pace and knowledge level, keeping them both engaged and challenged. For educators, it offers real-time insights, diagnostic assessments, and performance analytics, enabling smarter teaching decisions and targeted interventions. By combining AI-powered adaptability with Google Classroom integration, EduTutor AI ensures that learning remains curriculum-aligned, engaging, and outcome-focused, ultimately improving both teaching efficiency and student success.

Features

Dynamic Quiz Generation – Quizzes are automatically created using Granite LLM based on course material and synced topics.

Real-Time Feedback – Students receive instant evaluation and explanations for their answers, helping them learn faster.

Diagnostic Testing – Initial assessments identify knowledge gaps and establish personalized learning paths.

Adaptive Quizzing – Quiz difficulty adjusts dynamically according to the learner's performance and progress.

Personalized Learning Pathways – Recommendations guide students on which topics to review, practice, or advance.

Educator Dashboard & Analytics – Teachers gain access to performance metrics, quiz history, and actionable insights via Pinecone vector database.

Google Classroom Integration – Courses, assignments, and student rosters sync seamlessly to maintain curriculum consistency.

Secure Authentication – OAuth2 and JWT ensure secure login and session management.

Scalable Modular Architecture – The system is built to easily integrate new AI models and LMS platforms.

Cross-Platform Accessibility – A responsive web interface ensures smooth access across desktops, tablets, and mobile devices.

3. Architecture

EduTutor AI follows a modular, scalable architecture designed for adaptability and seamless LMS integration.

Frontend: A React-based web interface offering dashboards for students and educators, with real-time quizzes, feedback, and analytics.

Backend: Python/Django services managing authentication, APIs, quiz logic, and integration with AI models.

AI Layer: IBM Watsonx and Granite LLM handle quiz generation, diagnostic testing, and adaptive evaluations.

Database: Pinecone vector database stores performance insights; relational storage holds user data and quiz history.

Integration Layer: Google Classroom API syncs classes, students, and subjects for curriculum alignment.

Authentication & Security: OAuth2 for Google Classroom, JWT for secure sessions, and role-based access control.

Deployment: Cloud-ready, containerized with Docker, and scalable via modular microservices.

4. Setup Instructions

Prerequisites

Python 3.9 or higher for backend services and AI integration.

Node.js version 14 or higher for the frontend React application.

Google Classroom API credentials for course and student data synchronization.

IBM Watsonx API key for quiz generation and adaptive assessment.

Pinecone API key for vector database access and performance analytics.

Docker (optional) for containerized deployment and scalability.

Installation process

1. Clone the Project **–** Download the repository to your local machine.

2. Backend Setup – Install dependencies, configure environment variables, and initialize the database.

3. Frontend Setup – Install frontend dependencies and connect the interface to backend services.

4. Launch Application – Start backend and frontend services, then open the application in a web browser.

5. Login and Sync – Use Google Classroom credentials to synchronize courses, students, and subjects.

6. Verify – Check student dashboards for quizzes and feedback, and educator dashboards for performance insights.

5. Folder Structure

backend/ – Contains APIs, business logic, AI integration, authentication, and database operations.

frontend/ – React-based interface for dashboards, quizzes, and analytics visualization.

config/ – Holds configuration files, environment variables, and deployment settings.

docs/ – Documentation, guides, and references for developers and users.

tests/ – Unit and integration test cases for backend and frontend components.

scripts/ – Utility scripts for automation, deployment, and data migration.

assets/ – Static resources including images, media files, and icons.

6. Run the Application

1. Start Backend – Launch the backend services to handle APIs, AI interactions, and database operations.

2. Start Frontend – Open the React-based interface for students and educators.

3. Login – Use Google Classroom credentials to access courses and student data.

4. Student Dashboard – Attempt personalized quizzes, view real-time feedback, and track progress.

5. Educator Dashboard – Monitor class performance, quiz history, and learning analytics.

This ensures a fully integrated platform where students receive adaptive learning and educators gain actionable insights.

7. API Documentation

EduTutor AI provides RESTful APIs for seamless interaction between frontend, backend, AI services, and LMS integrations.

Key Endpoints:

Authentication – Handles secure login and role-based access for students and educators.

Quiz Management – Generates quizzes, submits responses, and retrieves quiz history.

Google Classroom Integration – Syncs courses, students, and subjects for curriculum alignment.

Insights & Analytics – Provides student-level and class-level performance metrics.

All APIs are secure, scalable, and modular, enabling future expansion and integration with other LMS platforms or AI mode

8. Authentication

EduTutor AI uses a secure authentication system to manage access for students and educators.

OAuth2 Integration – Students and educators log in using Google Classroom credentials, ensuring seamless access and course synchronization.

JWT (JSON Web Tokens) – Maintains secure sessions and protects API endpoints.

Role-Based Access Control – Ensures that students and educators can access only the features relevant to their roles, such as quizzes for students and analytics for educators.

This authentication framework guarantees security, privacy, and controlled access while maintaining a smooth user experience.

9. User Interface

EduTutor AI provides a responsive and intuitive web interface tailored for both students and educators.

- **Student Dashboard**

Displays personalized quizzes and learning paths.

Provides real-time feedback and performance tracking.

Highlights recommended topics and adaptive learning suggestions.

- **Educator Dashboard**

Shows class performance analytics, quiz history, and individual student progress.

Enables monitoring of learning trends and identification of areas needing attention.

Supports data-driven instructional planning.

- **Design Principles**

Clean and modern interface with easy navigation.

Mobile-friendly layout for accessibility across devices.

Modular design to accommodate future feature expansions.

The interface ensures that students stay engaged while educators have actionable insights at their fingertips.

---

10. Testing

EduTutor AI includes **comprehensive testing** to ensure reliability and accuracy across all components:

- **Unit Testing** – Validates individual functions in the backend and frontend to ensure they perform as expected.
- **Integration Testing** – Confirms that the backend, frontend, AI services, and LMS integration work together seamlessly.
- **Performance Testing** – Measures responsiveness of quiz generation, feedback delivery, and dashboard analytics under varying loads.

- **User Acceptance Testing (UAT)** – Ensures the platform meets the requirements of students and educators in real-world scenarios.

Regular testing ensures **high-quality performance, security, and stability**, providing a smooth learning experience.

---

11. Known Issues

While EduTutor AI is stable and fully functional, the following **known issues** exist:

- **Occasional Sync Delays** – Google Classroom course and student data may take a few moments to fully synchronize.
- **AI Response Latency** – Quiz generation and adaptive feedback may experience slight delays during high server load.
- **Browser Compatibility** – Older browsers may not fully support all interface features; recommended browsers are the latest versions of Chrome, Firefox, or Edge.
- **Limited Offline Access** – The platform requires an active internet connection for AI services and Google Classroom integration.
- **Scalability Under Heavy Load** – Extremely large class sizes may slightly impact dashboard responsiveness, pending future optimization.

These issues are being addressed in ongoing development to enhance stability, speed, and usability.

---

12. Future Enhancements

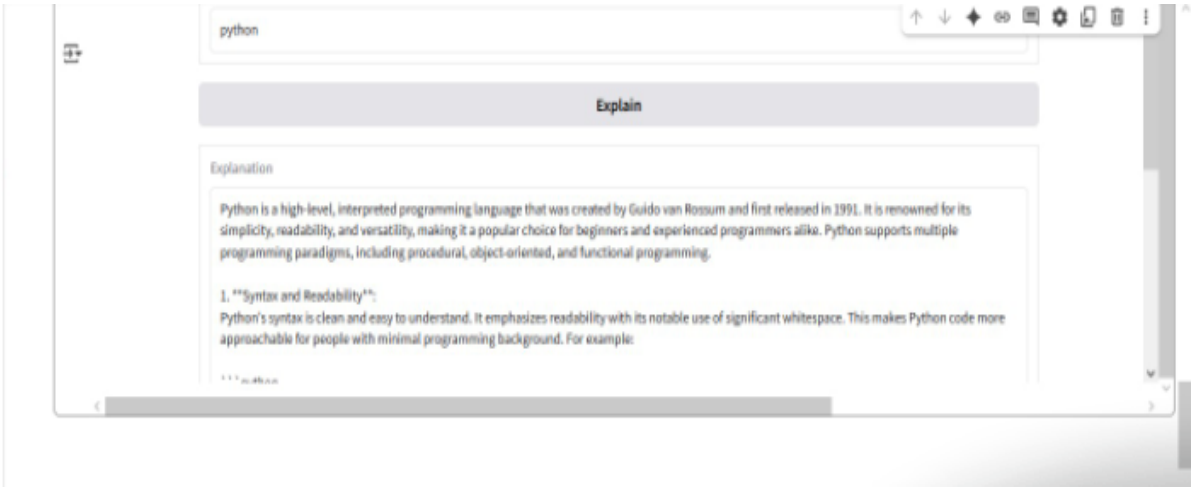
Future plans for EduTutor AI include:

- **Expanded LMS Integration** – Adding support for other learning management systems beyond Google Classroom.
- **Mobile Application** – Native mobile apps for iOS and Android to improve accessibility.
- **Enhanced AI Capabilities** – More advanced personalized recommendations, predictive analytics, and automated content suggestions.
- **Offline Mode** – Limited offline functionality for quizzes and progress tracking.
- **Gamification Features** – Badges, leaderboards, and rewards to boost student engagement.
- **Advanced Analytics** – Detailed trend analysis, predictive performance metrics, and curriculum recommendations for educators.

- **Multilingual Support** – Enabling learning in multiple languages for broader accessibility.

These enhancements aim to **expand the platform's reach, engagement, and educational impact** while keeping it future-ready and scalable.

OUTPUT


python
Explain
Explanation
Python is a high-level, interpreted programming language that was created by Guido van Rossum and first released in 1991. It is renowned for its simplicity, readability, and versatility, making it a popular choice for beginners and experienced programmers alike. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming.
1. **Syntax and Readability**:
Python's syntax is clean and easy to understand. It emphasizes readability with its notable use of significant whitespace. This makes Python code more approachable for people with minimal programming background. For example: