EduTutor AI

Project Name: EduTutor AITeam ID: NM2025TMID04162Team Leader: Ponmalar VTeam Members: Rakshana S, Mahalakshmi M, Padmasri P

Project Overview

1. Purpose
   EduTutor AI aims to revolutionize education by providing a personalized, AI-driven tutoring platform that supports students, educators, and institutions. By leveraging advanced AI and real-time data, EduTutor AI delivers tailored learning experiences, optimizes educational resources, and fosters academic success. For educators, it acts as a decision-making tool, offering insights, predictive analytics, and simplified content to enhance teaching strategies. The platform bridges technology, education, and community engagement to create inclusive, efficient, and impactful learning environments.
2. Features

Conversational Interface: Natural language interaction for students and educators to ask questions, receive guidance, and access resources.
Content Summarization: Simplifies complex educational materials, research papers, and textbooks into concise, actionable summaries.
Performance Forecasting: Predicts student performance and learning outcomes using historical and real-time data.
Personalized Learning Tips: Recommends daily study tips and strategies based on individual student behavior and progress.
Feedback Loop: Collects and analyzes student and educator feedback to improve content delivery and platform functionality.
KPI Forecasting: Projects key performance indicators (KPIs) to help educators and institutions track academic progress and plan interventions.
Anomaly Detection: Identifies unusual patterns in student performance or engagement for early intervention.
Multimodal Input Support: Accepts text, PDFs, and CSVs for analysis, content processing, and forecasting.
Streamlit/Gradio UI: Provides an intuitive dashboard for students, educators, and administrators.

3. Architecture

Frontend (Streamlit): Interactive dashboards for file uploads, chat, feedback forms, and report visualization.
Backend (FastAPI): REST framework handling content processing, chat, personalized tips, reports, and embeddings.
LLM Integration (IBM Watsonx Granite): Powers summarization, personalized learning tips, and report generation.

Vector Search (Pinecone): Stores and searches educational content embeddings for efficient retrieval.
ML Modules: Forecasting and anomaly detection using Scikit-learn for performance analytics.

4. Setup Instructions
   Prerequisites

Python 3.9 or later
pip and virtual environment tools
API keys for IBM Watsonx and Pinecone
Internet access

Installation Process

Clone the repository.
Install dependencies from requirements.txt.
Configure .env credentials for IBM Watsonx and Pinecone.
Run the FastAPI backend and Streamlit frontend.
Upload educational content (e.g., PDFs, CSVs) and interact with the platform.

5. Folder Structure

app/ – FastAPI backend logic
app/api/ – Modular API routes
ui/ – Streamlit frontend components
edu_dashboard.py – Main dashboard entry
granite_llm.py – Watsonx Granite integration
content_embedder.py – Embedding and Pinecone storage
kpi_forecaster.py – KPI forecasting for academic performance
anomaly_checker.py – Anomaly detection for student engagement
report_generator.py – Educational progress reports

6. Running the Application

Launch the FastAPI backend server.
Run the Streamlit dashboard.
Navigate via the sidebar.
Upload educational content (documents or CSVs).
Interact with the assistant to view summaries, predictions, and personalized tips.

7. API Documentation

POST /chat/ask: Returns AI-generated responses to user queries.
POST /upload-content: Uploads and embeds educational documents.
GET /search-content: Semantic search for educational materials.

GET /get-learning-tips: Personalized study tips for students.
POST /submit-feedback: Stores student and educator feedback.

8. Authentication

Current: Open environment for demo purposes.
Secure Deployments:
Token-based authentication (JWT, API keys).
OAuth2 with IBM Cloud credentials.
Role-based access (student, educator, administrator).

Planned Enhancements: User session tracking and personalized history.

9. User Interface

Minimalist, user-friendly design:
Sidebar navigation for easy access.
KPI visualizations with summary cards for academic progress.
Tabbed layouts for chat, learning tips, and forecasting.
Real-time form handling for feedback and uploads.
PDF report downloads for progress tracking.

10. Testing
    Testing Phases

Unit Testing: Validates prompt functions and utilities.
API Testing: Uses Swagger UI and Postman for endpoint testing.
Manual Testing: Tests file uploads, chat functionality, and output accuracy.
Edge Cases: Tests large files, malformed inputs, and invalid API keys.

11. Known Issues & Future Enhancements
    Known Issues

API rate limitations.
Handling large educational datasets.

Future Enhancements

Advanced authentication with multi-factor support.
Integration of additional ML models for deeper analytics.
Mobile app for on-the-go learning and tutoring.

Thank You!