Edu Tutor AI: Personalized Learning System

1. Introduction

Project Title: Edu Tutor AI – Personalized Learning Assistant
Team Members:
- M. Arthi (23051601802112010)
- T. Divya Dharshini
- A. Gurupriya
- S. Atchaya

2. Project Overview

Purpose:
Edu Tutor AI empowers students and educators by providing personalized learning pathways, adaptive study plans, and real-time academic support. Leveraging AI and data-driven insights, the system identifies individual learning gaps, offers tailored content, and engages students with interactive lessons and assessments. For educators, it acts as a teaching assistant that provides class performance analytics and helps in optimizing lesson plans.

Key Features

- **Conversational Interface:** Natural language interaction allowing students to ask questions, get explanations, and receive guidance.
- **Curriculum Summarization:** Summarizes textbooks and course materials into study guides.
- **Learning Progress Forecasting:** Estimates student performance trends.
- **Personalized Study Tips Generator:** Recommends study habits and tasks.
- **Student Feedback Loop:** Collects feedback to adjust curriculum.
- **KPI Forecasting for Educators:** Helps plan teaching interventions.
- **Anomaly Detection in Performance:** Flags drops in performance.
- **Multimodal Input Support:** Supports PDFs, CSVs, and other documents.
- **User Interface:** Streamlit or Gradio dashboard for interaction.

3. System Architecture

Frontend: Streamlit-based interactive dashboard.
Backend: FastAPI framework.
LLM Integration: IBM Watsonx Granite for content summarization and generation.
Vector Search: Pinecone for semantic search.
ML Modules: Scikit-learn for forecasting and anomaly detection.

4. Setup Instructions

Prerequisites:
- Python 3.9+, pip, virtual environment
- API keys for IBM Watsonx and Pinecone

Installation Process:
- Clone repository
- Install dependencies
- Configure .env file

- Start FastAPI backend
- Launch Streamlit frontend

5. Running the Application

- Start backend server
- Launch frontend dashboard
- Upload study materials, interact with chat, view progress predictions, download reports.

6. API Documentation

- POST /chat/ask: Submit queries
- POST /upload-doc: Upload materials
- GET /search-docs: Search concepts
- GET /get-study-tips: Get study advice
- POST /submit-feedback: Submit feedback

7. Authentication and User Interface

Future plans include token-based authentication, OAuth2, and role-based access.
The UI focuses on clarity, simplicity, KPI visualizations, and intuitive interaction flows.

8. Testing Strategy and Future Enhancements

- Unit Testing, API Testing, Manual Testing, Edge Case Handling.

Future: User session management, learning history tracking, multimedia support.