Edu tutor AI: Intelligent Educational Assistant for Personalized Training

Project Documentation

Introduction:

Project Title: Edu tutor AI personalized training

Team ID: NM2025TMID12550

Team Leader: srihari k

Team Members: ARUN KUMAR A,RIDHAN A ,ANITHAN RR

Team Size: 4

1. Project Overview:

Education AI is an intelligent educational assistant that leverages advanced AI technologies to provide personalized tutoring, adaptive learning paths, performance tracking, and content recommendations. The project aims to make educational support accessible and tailored to the unique needs of individual learners through an intuitive interface built on Python.

2. Architecture:

Frontend (Streamlit)

Interactive dashboard for students and educators.

Real-time progress monitoring, tutoring sessions, and chatbot interface.

Backend (FastAPI)

Handles API requests.

Manages user data, authentication, and AI model execution.

3. Setup Instructions:

Python 3.9+

Pip or Conda

AI service API key (OpenAI, IBM Watson, etc.)

Pinecone API key for vector search

Git

4. Folder Structure:

app/ — Contains backend FastAPI logic including routers, models, and integrations.

app/api/ — Modular API routes for chat, feedback, report, and document vectorization.

ui/ — Frontend Streamlit components and layouts.

main_dashboard.py — Main startup script for the Streamlit dashboard.

5. Running the Application:

Launch FastAPI server to expose backend endpoints.

Run the Streamlit dashboard for the web interface.

Navigate via sidebar to access tutoring, content upload, chat assistant, and reports.

Upload documents or CSVs for embedding; all interactions are real-time.

6. API Documentation:

POST /chat/ask — Accepts queries and returns AI-generated responses.

POST /upload/doc — Uploads and embeds educational documents.

GET /search/doc — Returns semantically relevant educational content.

POST /submit/feedback — Stores user feedback for improvement.

All endpoints are documented and testable via Swagger UI.

7. Authentication:

JWT-based authentication system.

User registration and login functionality.

Role-based access control (Student, Teacher, Admin).

8. User Interface:

Streamlit-based dashboard with simple, interactive design.

Student/teacher data input forms.

AI chatbot for educational queries.

Visualizations of student progress and learning analytics.

9. Testing:

Unit testing of core functionalities and models.

Integration testing to ensure backend-frontend communication.

AI validation for tutoring accuracy and performance.

Tools: Pytest, Postman.

10. Screenshot:

Screenshots and demo links are provided on the project’s GitHub repository showcasing UI and features.

11. Known Issues:

Dependency on availability of external AI services.

Limited dataset diversity may affect certain subjects.

Performance variations for large student datasets.

API usage costs for AI and vector database.

12. Future Enhancements:

Integration with Learning Management Systems (LMS).

Multi-language support for diverse learners.

Advanced personalized learning recommendations via deep learning.

Mobile app versions for better accessibility.

Real-time collaborative learning features.

AI-driven assessment and grading assistance.

13. Objectives & Scope:

Objectives:

Provide AI-driven personalized education support.

Enhance student engagement and performance.

Assist educators with insightful data and resources.

Scope:

Covers students, teachers, and educational institutions.

Supports multiple subjects and learning styles.

14. Problem Statement:

Challenges in current education systems:

Lack of personalized learning paths.

Limited access to real-time educational assistance.

Difficulty in monitoring individual student progress effectively.

Edu tutor AI aims to solve these by providing AI-powered personalized tutoring and resource management.

15. Use Case Scenarios:

Student Use Case: Student asks subject questions AI tutor explains concepts and generates practice exercises.

Teacher Use Case: Teacher uploads curriculum AI summarizes and suggests teaching materials.

Administrator Use Case: Monitor cohort performances and generate analytics reports.

16. Technology Stack:

Frontend: Streamlit

Backend: FastAPI

Database: MongoDB / PostgreSQL

AI / LLM: OpenAI GPT / IBM Watson AI

Vector Search: Pinecone

17. Performance Metrics:

API response times (target <1 second)

Tutoring accuracy via user feedback

Engagement and learning improvement metrics

Scalability to handle concurrent users

18. Limitations:

AI is an aid, not a replacement for expert teaching.

Model accuracy depends on training data quality.

Requires stable internet connectivity for AI queries.

Cost constraints linked to third-party AI service usage.

19. References:

OpenAI API Documentation

IBM Watson AI Documentation

Pinecone Vector DB Documentation

Literature on AI in education

20. Conclusion:

The Edu tutor AI project highlights how intelligent educational assistants can transform learning by offering personalized, scalable, and effective support to students and educators alike, advancing the future of education through AI technologies.