Project Title

Edututor AI

Team Members

- V. Priyaroshini
- Lakshmi
- Napisha Ansalin
- Nithyaselvi
- Prasannadevi

1. Introduction

Edututor AI is an innovative educational assistant designed to enhance learning experiences for students, teachers, and educational institutions. By combining artificial intelligence with interactive tools, it provides personalized guidance, automated assessments, and instant educational support. The project focuses on making education more **accessible, engaging, and intelligent.**

2. Project Overview

Purpose

The purpose of Edututor AI is to act as a **24/7 intelligent tutor** that supports students in learning different subjects, helps teachers with content creation and evaluation, and assists institutions in improving academic performance. It bridges the gap between human teaching and technology by offering real-time guidance, doubt clarification, and resource recommendations.

Features

- **Conversational Interface**
  - *Key Point:* Natural language interaction
  - *Functionality:* Students and teachers can ask questions in plain language and receive accurate answers or explanations.
- **Automated Lesson Summarization**
  - *Key Point:* Simplified understanding of study materials
  - *Functionality:* Converts long chapters or notes into concise summaries.
- **Question & Answer Generator**
  - *Key Point:* Smart exam preparation
  - *Functionality:* Creates multiple-choice, short, and essay-type questions automatically from study materials.

- **Personalized Learning Path**
  - *Key Point:* Adaptive tutoring
  - *Functionality:* Tracks student progress and suggests customized study plans.
- **Assessment & Feedback**
  - *Key Point:* Continuous evaluation
  - *Functionality:* Generates quizzes, evaluates answers, and provides instant feedback.
- **Multimodal Input Support**
  - *Key Point:* Flexible data handling
  - *Functionality:* Accepts text, PDFs, images, and voice for learning assistance.
- **Report Generation**
  - *Key Point:* Academic insights
  - *Functionality:* Provides performance reports and study recommendations for students and teachers.

3. Architecture

- **Frontend (Streamlit/Gradio):**
  Interactive web-based dashboard with modules for chat, uploads, quizzes, and reports.
- **Backend (FastAPI):**
  Handles API endpoints for content summarization, Q&A, performance analysis, and report generation.
- **AI/LLM Integration (Watsonx/OpenAI models):**
  Provides natural language understanding, question generation, summarization, and adaptive tutoring.
- **Database & Vector Search (Pinecone/FAISS):**
  Stores learning materials, embeddings, and student progress for quick retrieval and personalization.

- **Database & Vector Search (Pinecone/ FAISS):**
  Stores learning materials, embeddings, and student progress for quick retrieval and personalization.
- **ML Modules (Performance Prediction & Adaptive Learning):**
  Predicts student performance trends and adapts content delivery accordingly.

4. Setup Instructions

Prerequisites:

- Python 3.9+
- pip & virtual environment
- API keys for AI models (IBM Watsonx / OpenAI)
- Internet access

Installation:

1. Clone the repository
2. Install dependencies from `requirements.txt`
3. Configure `.env` file with API credentials
4. Run backend server with FastAPI
5. Launch frontend using Streamlit
6. Upload study material and interact with the tutor

5. Folder Structure

Copy code

```
app/ - Backend logic (API, models, AI integration)
ui/ - Frontend dashboard and pages
document_processor.py - Summarizes and embeds study material
qa_generator.py - Creates practice questions
report_generator.py - Builds student performance reports
learning_path.py - Suggests adaptive study plans
```

6. Running the Application

1. Start FastAPI server for backend endpoints.
2. Launch Streamlit dashboard for web interface.
3. Upload study notes or books.
4. Interact with AI tutor for explanations, summaries, and quizzes.
5. Download reports and track progress.

7. User Interface

- Sidebar navigation (Chat, Notes, Quizzes, Reports)
- Student performance dashboards
- Real-time Q&A chat support
- Quiz & assessment panels
- PDF/Report download feature

8. Testing

- **Unit Testing:** For summarization and Q&A modules
- **API Testing:** Using Swagger/Postman
- **Manual Testing:** Chat responses, quiz evaluation, and reports
- **Edge Case Handling:** Large files, mixed input types, incomplete questions

9. Future Enhancements

- Voice-based tutoring
- Multilingual support for regional languages
- Integration with LMS (Learning Management Systems)
- Gamified learning for better student engagement