Personalised Tutor :

📚 AI-Powered Tutor & Quiz App

Project Overview

The AI-Powered Tutor & Quiz App is an educational platform that leverages artificial intelligence to provide personalized learning experiences. The application allows users to ask questions on various academic subjects and receive tailored explanations based on their learning preferences. It also generates quizzes to help users test their knowledge.

🛠️ Tools & Technologies

- **FastAPI**: Backend framework for creating the API endpoints
- **Streamlit**: Frontend framework for building the user interface
- **LangChain**: AI orchestration framework for handling LLM interactions
- **OpenAI API**: Powers the AI tutoring and quiz generation capabilities
- **Python**: Primary programming language
- **Pydantic**: Data validation and settings management
- **Uvicorn**: ASGI server for running the FastAPI application
- **Python-dotenv**: For managing environment variables

🏗️ Architecture & Components

Component Architecture

The application follows a client-server architecture with three main components:

1. **Streamlit Frontend (app.py)**

- User interface for interacting with the application
- Collects user preferences and questions
- Displays personalized explanations and quizzes

2. **FastAPI Backend (main.py)**
   - RESTful API endpoints for tutoring and quiz generation
   - Handles request validation and error management
   - Integrates with the AI engine
3. **AI Engine (ai_engine.py)**
   - Core logic for generating personalized tutoring content
   - Handles LLM prompt construction and response parsing
   - Includes error handling and fallback mechanisms

Workflow

1. User selects their learning preferences in the Streamlit UI (subject, level, learning style, etc.)
2. User enters a question or requests a quiz
3. Streamlit frontend sends the request to the FastAPI backend
4. Backend validates the request and calls the appropriate AI engine function
5. AI engine constructs a prompt based on user preferences
6. LangChain processes the prompt using the OpenAI API
7. Responses are formatted and returned to the frontend
8. Frontend displays the personalized explanation or quiz to the user

🔍 Key Features

- **Personalized Learning**: Adapts explanations to the user's learning style, background knowledge, and proficiency level

- **Multi-Subject Support**: Covers Mathematics, Physics, Computer Science, History, Biology, and Programming
- **Quiz Generation**: Creates customized quizzes to test knowledge retention
- **Multi-Language Support**: Provides explanations in various languages (English, Hindi, Spanish, French)
- **Learning Style Adaptation**: Tailors content for Visual, Text-based, and Hands-on learners

📋 Implementation Details

Frontend Implementation

The Streamlit application provides an intuitive interface with:

- Subject and learning preference selectors in the sidebar
- Text area for entering questions
- Buttons for triggering explanations and quizzes
- Formatted display of AI-generated content

Backend Implementation

The FastAPI backend implements:

- RESTful endpoints for tutoring and quiz generation
- Request validation using Pydantic models
- Error handling with appropriate HTTP status codes
- CORS middleware for cross-origin requests

AI Engine

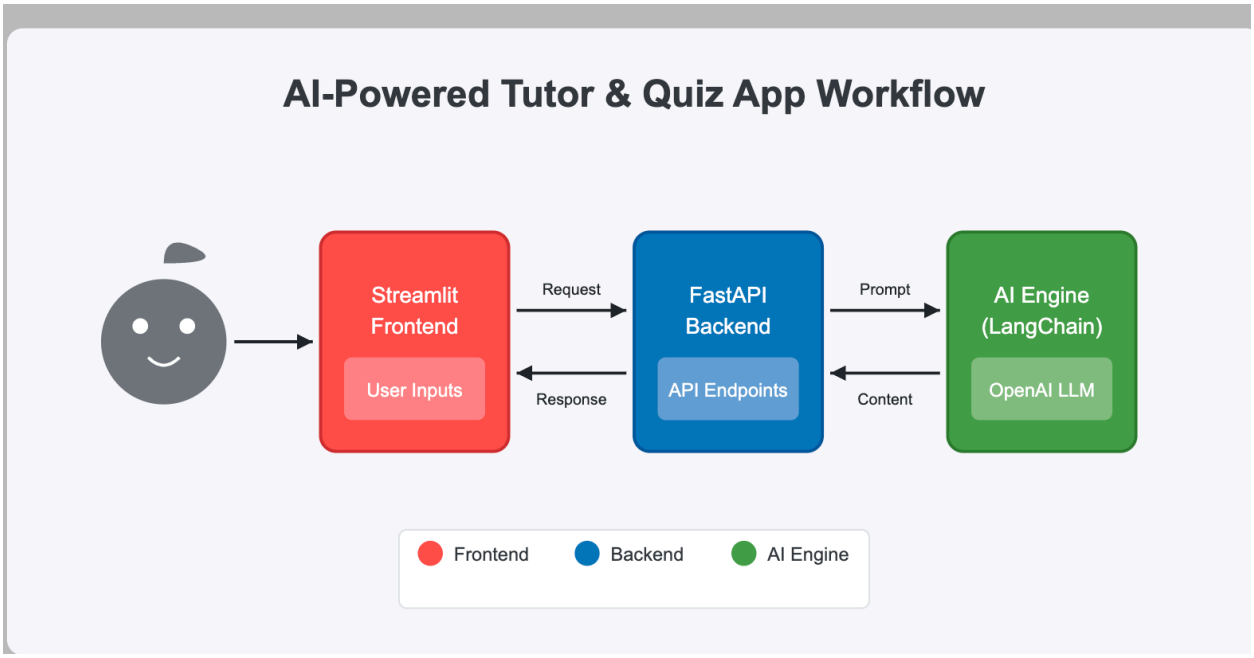
The AI engine handles:

- Prompt engineering for effective LLM interactions
- Response parsing and formatting
- Fallback mechanisms for handling errors

- Adaptation of content based on learning styles

🚀 Getting Started

1. Install dependencies: `pip install -r requirements.txt`
2. Set up your `.env` file with your OpenAI API key
3. Start the backend: `uvicorn main:app --reload`
4. Launch the frontend: `streamlit run app.py`



How to Use the AI Tutor & Quiz App

Chatting with AI Tutor

1. **Set Preferences**
   Select subject, level, learning style, language and background knowledge
2. **Ask Your Question**
   Type your question in the text area
   "Explain Newton's Second Law of Motion"
3. **Get Explanation**
   Click "Get Explanation 🧠" button
   Receive personalized explanation

Personalized Learning Experience

Generating a Quiz

1. **Set Your Preferences**
   Select your subject of interest
   Choose your learning level
2. **Generate the Quiz**
   Click "Generate Quiz 📝" button
   AI will create custom multiple-choice quiz
3. **Take the Quiz**
   Answer the multiple-choice questions
   Test your knowledge on the subject

Adaptive Knowledge Assessment

AI-Powered Tutor & Quiz App Architecture