AI Learning Assistant for Students and Programmers

Problem Statement

Students and programmers, especially in developing regions like Cameroon, often struggle with accessing personalized and interactive study aids. They face challenges in summarizing complex learning materials, understanding programming concepts from lengthy online content such as YouTube tutorials or PDF notes, and staying engaged with learning materials due to information overload. Manual note summarization and repetitive reading consume time and reduce productivity. Additionally, not all students can effectively learn through text, and there is a growing need for multimodal (text + audio) learning tools that adapt to user needs.

Proposed Solution

The proposed solution is an AI-powered Learning Assistant web application that helps students and programmers summarize learning materials and convert them into audio for easier understanding and accessibility. The assistant will process user inputs like text, uploaded documents, or links to YouTube videos and automatically generate concise summaries, highlight key points, and provide a natural voice narration of the content. It will also support explaining code snippets, suggesting improvements, and answering programming-related questions using advanced language models. Retrieval-Augmented Generation (RAG) will be integrated to allow the assistant to fetch accurate and contextually relevant answers from stored course materials, ensuring personalized and precise results.

Objectives

- To build an intelligent assistant that summarizes long study materials into concise, easy-to-understand content.
- To enable text-to-speech functionality for auditory learning.
- To integrate Retrieval-Augmented Generation (RAG) for accurate contextual responses.
- To provide an interactive chatbot interface for code-related explanations and debugging assistance.
- To create a user-friendly interface accessible through the web using React and Flask.
- To promote efficient, AI-assisted learning for students and programmers in various fields.

Key Features

- **Summarization Engine:** Automatically summarizes notes, PDFs, and articles using LLMs.
- **YouTube Course Summarizer:** Extracts and summarizes key insights from video transcripts.
- **Text-to-Speech (TTS):** Converts summaries or explanations into natural audio.
- **AI Chat Assistant:** Provides Q&A; support for programming and technical topics.
- **RAG Integration:** Retrieves contextual information from stored learning materials.
- **Code Understanding:** Explains snippets, optimizes code, and detects logical errors.
- **User Dashboard:** Allows saving, reviewing, and replaying previous summaries and explanations.

Tools & Technologies

- **Frontend:** React.js (for dynamic and responsive UI)
- **Backend:** Flask (for API handling and AI model integration)
- **AI/ML Models:** Hugging Face Transformers (LLMs for summarization and chat), LangChain (for orchestration and RAG)
- **Text-to-Speech:** Pyttsx3 or gTTS (for generating natural audio)
- **Vector Database:** FAISS or Chroma (for storing embeddings in RAG)
- **Programming Language:** Python
- **Other Tools:** Git, GitHub, VScode, Postman