**How Our AI Tutor Works Under the Hood**

At its core, it’s a pretty straightforward setup with two main parts: a friendly **frontend** that you interact with in your browser, and a smart **backend** that does all the heavy lifting. I have used more like open sourced STT and TTS for the enterprise scalable applications need to use more advanced APIs like Whisper for STT. Due to limitations over these I managed myself to use following as mentioned.

**The Journey of a Question: From Voice to Answer**

It all starts with you, the user. Here’s how your question travels through the system and comes back as a helpful answer.

**1. You Start the Conversation** Everything kicks off when you click the "Start Listening" button. The app uses your computer's microphone to capture what you're saying. That audio clip is immediately sent to Google's Speech Recognition service, which is incredibly fast at turning your voice into plain text. The text then pops up in the app, giving you a chance to quickly edit it or fix any little mistakes before sending it off.

**2. The Frontend Sends Your Message** Once you hit "Send," our Streamlit frontend takes over. It packages up your text query along with a unique session\_id. That ID is super important—it’s how the app remembers the conversation it’s having specifically with you, so it doesn't get your chat mixed up with someone else's. This package is then sent as a request to our backend.

**3. The Backend Gets to Work** Our backend, built with FastAPI, is the brains of the operation. When it receives your request, it immediately starts a process called Retrieval-Augmented Generation (RAG). Instead of just making up an answer, it does some research first:

* **First, it understands the context.** It looks at your chat history and uses Google's Gemini AI to rephrase your latest question. For example, if you ask "What about that?", it might rephrase it to "What is the process of feature scaling in machine learning?" based on your previous messages.
* **Next, it finds relevant information.** The backend takes this refined query and searches its "library"—a PDF on Machine Learning that we’ve already processed into a specialized Chroma vector database. It finds and pulls out the most relevant snippets of text that are likely to contain the answer.
* **Finally, it formulates the perfect answer.** The AI takes your original question, the chat history, and the relevant text it just found and hands it all over to the Gemini model. With all this rich context, the model can generate a helpful, accurate answer. It also figures out an appropriate "emotion" for the tutor (like 'explaining' or 'happy') to give the conversation a more personal touch.

**4. The Answer Comes Back to You** The backend sends the answer text and the emotion back to the frontend. Instantly, the tutor's reply appears in the chat window, and the mascot's emoji changes to match the tone.

At the same time, the app sends the answer text to Google's Text-to-Speech service. The audio is streamed back and plays automatically, so you don't just read the tutor's response—you hear it, bringing the whole conversation to life.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 1 The backend and frontend flow diagram