AI Interview Copilot – Full Project Documentation

Live Speech to Intelligent Interview Answers (Full■Stack Project)

1. Project Overview

AI Interview Copilot is a full■stack Generative AI application designed to help users practice real■time interview questions. The system listens to spoken interview questions, converts speech to text, processes the question using a Large Language Model (LLM), and responds with a professional, interview■ready answer in real time.

2. Purpose and Use Case

- Practice technical and HR interviews in real time.
- Simulate real interviewer–candidate interaction.
- Improve clarity, confidence, and answer structure.
- Useful for students, job seekers, and developers preparing for interviews.

3. System Architecture (Working Model)

The system follows a client–server architecture using WebSockets for real■time communication.

- Frontend (React + Tailwind CSS): Captures microphone input and displays questions and answers.
- Backend (FastAPI): Handles WebSocket connections and AI inference.
- Speech Recognition (Browser API): Converts live speech into text.
- LLM (Hugging Face – FLAN■T5): Generates interview■ready answers.

4. Technology Stack

- Frontend: React.js, Tailwind CSS, Web Speech API
- Backend: FastAPI, WebSockets, Python
- AI Model: Google FLAN■T5 (via Hugging Face Transformers)
- Communication: WebSocket (real■time bidirectional data)
- Deployment Ready: GitHub, Localhost / Cloud VM

5. Data Flow – Step by Step

- User clicks ‘Start Interview’ on frontend.
- Browser microphone captures live speech.
- SpeechRecognition API converts speech to text.
- Question text is sent to backend via WebSocket.

- Backend sends the question to the LLM with interview■style prompt.
- LLM generates a structured, professional answer.
- Answer is sent back to frontend and displayed.

6. Key Features

- Live speech■to■text interview questions.
- Professional, structured AI answers.
- Clear separation of interviewer question and AI answer.
- Modern, professional UI suitable for production.

7. How to Run the Project (Local Setup)

Backend Setup:

1. Create virtual environment 2. Install dependencies using requirements.txt 3. Run: python -m uvicorn main:app --reload

Frontend Setup:

1. Navigate to frontend folder 2. Install dependencies: npm install 3. Run: npm run dev 4. Open browser at http://localhost:5173

8. Future Enhancements

- Answer scoring and feedback.
- Follow■up interview questions.
- Role■based interviews (ML, Backend, HR).
- Cloud deployment with authentication.

9. Conclusion

AI Interview Copilot demonstrates how Generative AI, speech recognition, and real■time web technologies can be combined to create a powerful interview practice platform. This project is ideal for learning full■stack AI development and can be extended to production■level systems.