Project Brief: AI Speech-to-Text App (Including Voice Commands)

Project Title: AI Speech-to-Text App (Including Voice Commands)

Project Overview:

The **AI Speech-to-Text App** is a dynamic voice assistant–style application designed to enable **real-time transcription, summarisation, and interactive voice-based control**. Users will be able to engage in **spoken conversations with the app**, transcribe live meetings, livestreams, and recorded video/audio content (up to 1 hour), and receive **AI-generated summaries** of what was discussed. The app will also include a **custom voice command system** and **optional voice training module** that improves recognition for individual users, creating a smarter and more personal interaction experience.

Objectives:

1. Develop a voice-interactive application that transcribes real-time and pre-recorded audio up to 60 minutes in length.
2. Enable natural conversational interactions, allowing users to speak directly to the app and receive spoken or textual responses.
3. Implement voice command recognition for performing specific in-app functions through speech.
4. Integrate AI-generated summaries that concisely reflect the contents of transcribed audio or conversations.
5. Offer a voice training feature that allows users to record samples and improve recognition of their individual voice profiles.
6. Ensure a responsive and accessible user interface for uploading content and viewing live interactions or summary outputs.

Key Features:

1. **Two-Way Voice Assistant Functionality:** Supports natural conversations between the user and AI via microphone input.
2. **Transcription Engine:** Converts audio from meetings, livestreams, or uploaded recordings into accurate text.
3. **AI Summarisation:** Uses large language models (e.g. Gemini or Whisper + summarisation layer) to deliver **contextual summaries** of conversations.
4. **Voice Training Module:** Allows users to upload or record voice samples to improve speech recognition accuracy over time.
5. **Multi-Modal Input:** Accepts **microphone input, file uploads**, or **live audio streams** as sources.
6. **Copy and Export Options:** Transcripts and summaries can be copied to clipboard or downloaded in .txt or .pdf format.

Technical Specifications:

1. **Programming Language:** Python
2. **Frameworks and Libraries:**
   - Flask/Django: Web application framework.
   - SpeechRecognition, PyAudio, whisperx or Vosk for STT (speech-to-text)
   - webrtcvad for voice activity detection
   - NLP libraries (e.g., SpaCy or NLTK): for language understanding and intent recognition.
3. **Database:** SQLlite for storing user-defined configurations and logs.
4. **Deployment:** Docker containerization and cloud service deployment (e.g., AWS or Azure).
5. **Version Control:** Git for source code management.

Expected Outcomes:

1. A fully functional AI-powered voice assistant and transcription tool with real-time command capabilities.
2. Enhanced speech recognition accuracy through user-specific voice training.
3. Streamlined meeting and media analysis via automatic transcription and smart summarisation.
4. A responsive UI offering live feedback, searchable transcripts, and multi-format exports.

Risks and Mitigations:

1. **Recognition Errors:** Include fallback text input and voice re-activation prompts; use robust models with adaptive training.
2. **Latency or Lag:** Optimise streaming and model inference for real-time responsiveness.
3. **Privacy & Security:** Encrypt all user voice data and provide clear options to manage and delete stored recordings.
4. **Audio Input Challenges:** Support a wide range of file formats and stream configurations with input validation and cleanup handling.