Voice Chat Application

Web app

[Task 4]

Mohamad Abdulmoula,R. Rama

ramamohameed9@gmail.com

Department of Software Engineer

Under the Supervision Of:

Eng Asmaa Duramae

August 12, 2025

Table of Contents

Glossary of Terms

Term	Definition
Flutter	An open-source UI framework by Google for building cross-platform applications.
OpenAI GPT	A large language model capable of generating human-like responses to user inputs.
Speech-to-Text (STT)	A process that converts spoken words into text.
Text-to-Speech (TTS)	A process that converts text into spoken voice output.
Material 3	The latest version of Google's Material Design system for UI/UX design.
Provider	A state management solution in Flutter for efficient UI updates.
API Key	A security credential used to authenticate API requests.
Permission Handler	A Flutter package for handling runtime permissions on mobile devices.

Project Overview

The **Voice GPT Chat Application** is a Flutter-based mobile application that allows users to interact with an AI chatbot through both voice and text. It integrates **speech-to-text** for voice input, **text-to-speech** for voice output, and OpenAI's GPT model for intelligent conversation handling.
The app supports live speech recognition, manual text input, AI-powered responses, and performance metrics tracking. It is designed with **Material 3** design principles for a modern and accessible user interface.

Planning

A. Objective:

To develop an interactive mobile chatbot that supports both **voice input** and **voice output**, powered by **OpenAI's GPT**, allowing for seamless and intelligent real-time conversations.

B. Purpose:

This application aims to provide users with a **hands-free AI conversation experience**, useful for accessibility, personal assistance, and productivity. It merges advanced AI capabilities with modern mobile design, enhancing usability and engagement.

C. **Tools Used:**

a. **Programming Language:** Dart
b. **Framework:** Flutter
c. **State Management:** Provider
d. **AI Engine:** OpenAI GPT API
e. **Speech-to-Text:** speech_to_text package
f. **Text-to-Speech:** flutter_tts package
g. **Permission Handling:** permission_handler package
h. **Environment Variables:** flutter_dotenv package
i. **Version Control:** Git & GitHub

Requirements Analysis

A. Functional Requirements

ID	Requirement Description	Priority
FR-1	The system shall allow users to input text messages manually through a text field.	High
FR-2	The system shall allow users to record and transcribe speech using the microphone.	High
FR-3	The system shall process user input and send it to the GPT API for a response.	High
FR-4	The system shall display both user and AI messages in a chat interface.	High
FR-5	The system shall read AI responses aloud using text-to-speech.	Medium
FR-6	The system shall allow toggling TTS on/off via a UI button.	Medium
FR-7	The system shall track and display the number of requests and average response time.	Medium
FR-8	The system shall allow clearing the conversation and metrics.	Low
FR-9	The system shall handle errors gracefully and display appropriate messages.	High

B. Non-Functional Requirements

ID	Requirement Description	Priority
NFR-1	The application should process speech input and generate responses within 3 seconds for smooth interaction. (**Performance**)	High
NFR-2	The interface must follow Material 3 design principles and be accessible to users with different abilities. (**Usability**)	High
NFR-3	The application should handle network errors without crashing and provide fallback messages. (**Reliability**)	High
NFR-4	The application must run on both Android and iOS devices. (**Portability**)	High
NFR-5	The architecture should allow easy integration of additional AI features. (**Scalability**)	Medium
NFR-6	The API key should be stored securely using environment variables. (**Security**)	High

Implementation Phases

a. **Environment Setup**
   a) Installed Flutter SDK and dependencies.
   b) Configured .env file to store the OpenAI API key securely.
b. **UI Design**
   a) Created the main ChatScreen with AppBar, message list, and input controls.
   b) Implemented ChatBubble widget for user and assistant messages.
c. **Speech-to-Text Integration**
   a) Used speech_to_text to initialize microphone access and transcribe speech.
   b) Added real-time transcription updates to the input field.
d. **Text-to-Speech Integration**
   a) Used flutter_tts to vocalize AI responses.
   b) Added toggle button to enable/disable TTS output.
e. **GPT API Integration**
   a) Implemented GPTService for sending requests to OpenAI's GPT API.
   b) Processed API responses and updated UI dynamically.
f. **State Management**
   a) Used Provider (MetricsProvider) to store performance metrics such as request count and average response time.
g. **Testing & Debugging**
   a) Performed unit tests for GPT API calls.
   b) Tested speech recognition and TTS output on physical devices.

Conclusion

The Voice GPT Chat Application successfully combines voice recognition, AI-powered responses, and speech synthesis in a single Flutter app. It provides a hands-free conversational experience, is built with a modular architecture for scalability, and adheres to modern UI design principles. Future enhancements could include multi-language support, conversation history storage, and offline TTS/STT processing to improve performance and accessibility.

References

A. **Google Developers. (2023). *Flutter Documentation*. Retrieved from https://docs.flutter.dev**

B. **OpenAI. (2023). *OpenAI API Documentation*. Retrieved from https://platform.openai.com/docs**

C. **Flutter Community. (2023). *speech_to_text Flutter package*. Retrieved from https://pub.dev/packages/speech_to_text**

D. **Flutter Community. (2023). *flutter_tts Flutter package*. Retrieved from https://pub.dev/packages/flutter_tts**