## 1. Introduction

Rapid Response is a Flutter-based mobile application integrated with a .NET Core backend. The application allows users to upload audio files, which are then sent to the backend for processing using OpenAI’s API. The API transcribes the audio to text and performs entity recognition to extract key details.

## 2. Technologies Used

Frontend: Flutter (Dart)

Backend: .NET Core (C#)

External API: OpenAI Whisper API for speech-to-text conversion

Networking: HTTP requests using RESTful APIs

Database: Used In-Memory storage for now will proceed to rigid databases once completely integrated with required features.

## 3. System Architecture

The system follows a client-server architecture where the Flutter application acts as the client and the .NET Core application serves as the backend.

## 4. Frontend Implementation

The Flutter application consists of the following components:

• FileUploadPage: Handles file selection and upload.

• ResponseScreen: Displays server responses.

• MainApp: Entry point for the Flutter application.

## 5. Backend Implementation

The .NET Core backend consists of an API controller (OpenAIController) that handles incoming audio files, sends them to OpenAI for transcription, and processes the response for entity recognition.

## 6. API Endpoints

1. POST /api/OpenAI/audio-to-text - Accepts an audio file and transcribes it.

2. POST /api/OpenAI/CompleteSentence - Processes the transcribed text for entity recognition.

## 7. Workflow

1. The user selects and uploads an audio file from the Flutter app.  
2. The file is sent to the .NET backend via an HTTP request.  
3. The backend forwards the file to OpenAI Whisper for transcription.  
4. The transcribed text is processed for entity recognition.  
5. The extracted details are sent back to the frontend and displayed to the user.

## 8. Error Handling

• Handles missing or empty audio files with appropriate HTTP responses.

• Logs errors in case of API failures.

• Provides user-friendly messages on failed requests.

## 9. Conclusion

Rapid Response provides a seamless experience for converting emergency-related audio recordings into structured data. The application ensures accurate speech-to-text conversion and enhances emergency reporting efficiency.

John Emmanuel