Voice Bot Hooman Labs Assignment

# Overview

This project creates a voice bot that allows users to have conversations with an AI. The bot captures audio input from the user, converts it to text using Deepgram, generates a response using GPT-4o Mini, and then converts the text response back to speech using Amazon Polly.

# Prerequisites

Node.js and npm installed

AWS account with access to Amazon Polly

Deepgram API key

OpenAI GPT-4o Mini API key

# Setup

## 1. Install dependencies

Refer to requirement.txt

## 3. Environment Variables

Create a `.env` file in the root directory and add the following environment variables:

File should include this information.

GPT4O\_MINI\_API\_KEY=your\_openai\_api\_key\_here  
AWS\_ACCESS\_KEY\_ID=your\_aws\_access\_key\_id  
AWS\_SECRET\_ACCESS\_KEY=your\_aws\_secret\_access\_key  
AWS\_REGION=your\_aws\_region  
DEEPGRAM\_API\_KEY=your\_deepgram\_api\_key

## 4. Run the server

## Commond to run node server.js

# Project Structure

- `server.js`: The main server file that handles incoming requests, processes audio data, interacts with the Deepgram, GPT-4o Mini, and Amazon Polly APIs, and sends responses back to the client.

- `index.html`: The frontend HTML file that provides the user interface.

- `static/scripts.js`: The client-side JavaScript file that captures audio input, sends it to the server, and plays back the audio response.

- `requirements.txt`: Lists the dependencies required for the project.

- `README.md`: Documentation file (this file).

# How It Works

## Server-Side

1. The server captures audio data sent from the client.  
2. The audio data is processed by Deepgram to convert it to text.  
3. The text is sent to the GPT-4o Mini API to generate a response.  
4. The response text is converted to speech using Amazon Polly.  
5. The audio response is sent back to the client in base64 format.

## Client-Side

1. The user clicks a button to start recording.  
2. Audio is captured.  
3. The audio data is sent to the server.  
4. The server processes the audio and sends back a text and audio response.  
5. The audio response is played back to the user, and the conversation is displayed in text form on the UI.

# Dependencies

- `express`: A web framework for Node.js.

- `axios`: A promise-based HTTP client for the browser and Node.js.

- `dotenv`: A module to load environment variables from a `.env` file.

- `@aws-sdk/client-polly`: AWS SDK for interacting with Amazon Polly.

- `body-parser`: A middleware to parse incoming request bodies.

- `https`: A module for HTTPS request handling.