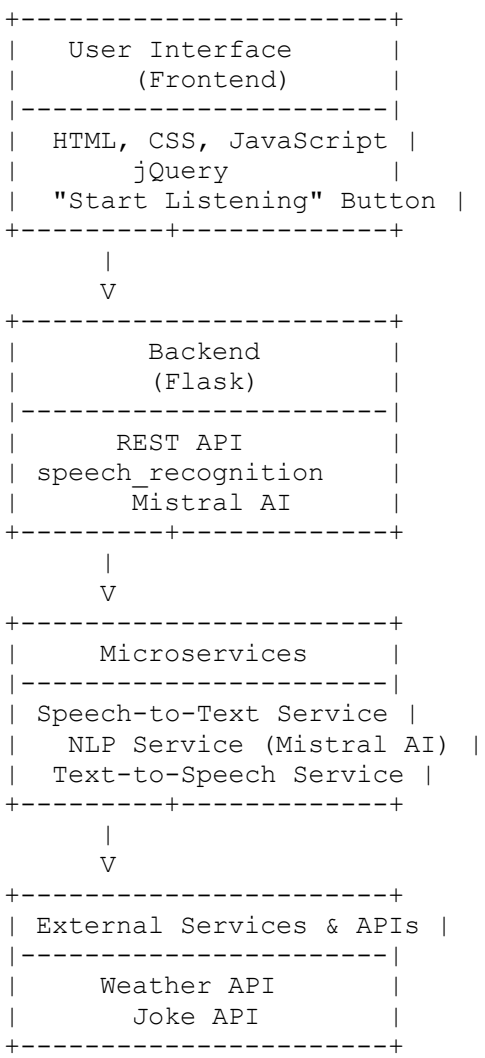
CHECKPOINT 2

Presentation of the Technology Stack

1. **Backend Technologies:**
   - **Flask:** It's a lightweight and flexible Python web framework that handles routing and logic.
   - **Mistral AI:** A large language model (LLM) is utilized for natural language processing and understanding user commands.
2. **Frontend Technologies:**
   - **HTML, CSS, JavaScript:** We will use these core web technologies for creating the user interface.
3. **Speech Recognition and Processing:**
   - **SpeechRecognition Library:** It enables the conversion of voice input into text using Google's Speech Recognition API in Python.
   - **Pygame:** This is a multimedia library used to play the generated speech output in real-time.
4. **Task Execution and Automation:**
   - **Webbrowser Library:** It allows the assistant to open websites and execute web-based tasks.
   - **Custom Commands:** It supports opening web pages, adjusting settings like voice speed and gender, and retrieving useful information based on user queries.
5. **Project Deployment and Testing**
   - **Postman:** Used for API endpoint testing to ensure reliable backend performance.

Architecture Diagram

1. **User Interface (Frontend):**
   - It's a web interface developed using HTML, CSS, JavaScript, jQuery.
   - The user can record the audio with a "Start Listening" button.
2. **Backend (Flask):**
   - A Flask-based REST API processes the requests from the user.
   - The incoming audio is converted to text with the speech_recognition library.
   - Natural language processing (NLP) is performed using Mistral AI.
3. **Microservices:**
   - Speech-to-Text Service converts the user's voice to text.
   - NLP Service (Mistral AI) determines the user's intent by analyzing the text.
   - Text-to-Speech Service converts the response to audio and plays it on the user's device.
4. **External Services & APIs:**
   - Weather API processes the user's weather requests.
   - Joke API can be used to create funny responses.



General Concept

This project aims to develop a voice-activated virtual assistant that understands and responds to user commands using a microservices-based architecture. This system will process speech input, analyze intent, execute relevant tasks, and provide voice-based responses.

Future Enhancements

Contextual Awareness: Remembering previous conversations.

Smart Home Integration: Controlling IoT devices.

Personalization: Allowing users to customize voice preferences.

Offline Mode: Implementing local processing for basic commands.