PROJECT NAME: IOT DEVICE

MOTIVATION FOR OUR PROJECT:

Building an IOT device similar to Alexa offers a range of benefits that can drive both the personal and academic growth. There are mainly a few reasons for undertaking this project:

- Hands-on learning experience: By building this project we can gain practical experience with hardware components, programming languages, and communication protocols.
- Skill-development: We can improve our skills in areas like embedded systems, networking and cloud services.
- Cutting-edge technology exposure: For voice recognition features we can gain exposure to newest technologies like speech-to-text(STT) and text-to-speech(TTS) engines. We can also learn how to integrate various IOT devices and services into a cohesive system.
- Problem-solving and innovation: Through this project we can develop innovative solutions to real-world problems, such as home automation and smart assistants. This can also help us overcome technical challenges, which can be highly rewarding and intellectually stimulating.

OVERVIEW

Objective:
To design and develop a voice-controlled IOT device capable of recognizing voice commands, processing them, and performing corresponding actions, similar to how Alexa operates.

Features:

1. Voice recognition: The device will listen for voice commands and convert them from speech to text.
2. Command processing: It will interpret the text commands and execute predefined functions.
3. Text-to-Speech: The device will provide verbal responses using a text-to-speech engine.
4. IOT-integration: The device will communicate with other IOT devices and services using MQTT or HTTP.
5. User-interaction: The device will handle various user requests as voice queries and supply them with adequate information in real-time.

Components:

1. Hardware:
    - ●Microcontroller: Raspberry Pi or Arduino
    - ●Microphone: USB Microphone for voice input

●Speaker: USB or 3.5mm speaker for audio output
●Optional: Sensors and automators for additional functionalities

2. Software:
    - ●Programming Languages: Python for high-level operations, C/C++ for low-level operations
    - ●Libraries: SpeechRecognition, gTTS, Paho MQTT, etc.
    - ●Operating System: Raspbian OS for Raspberry Pi

3. Network and Communication:
    - ●Protocols: MQTT for IOT communication, HTTP for web APIs.
    - ●Broker: Mosquitto or a cloud-based MQTT broker.

Implementation Steps:

1. Set up hardware:
   - ●Connect and configure the microphone and speaker.
   - ●Test the hardware components to ensure proper functionality.
2. Develop voice recognition:
   - ●Implement speech-to-text functionality using libraries like SpeechRecognition.
   - ●Handle voice input and convert it to text for further processing.
3. Command processing:
   - ●Use the OpenAI API to send voice queries (converted to text) to ChatGPT.
   - ●Retrieve and process the responses from ChatGPT.
4. Text-to-Speech:
   - ●Use gTTS(Google Text-to-Speech) to convert text responses to speech.
   - ●Ensure the device can audibly respond to user commands.
5. API Integration:
   - ●Set up the ChatGPT API key and configure the device to make HTTP requests.
   - ●Implement logic to handle API responses and convert them into spoken output.
6. Enhancements and Testing:
   - ●Test the device to ensure it can handle various voice queries and provide accurate responses.
   - ●Optimize the speech recognition and text-to-speech components for better performance.

GLIMPSE OF THE FEATURES IN THE FINAL PROJECT:

1. Voice Recognition:

    - Continuous Listening: The device continuously listens for user commands.
    - Speech-to-Text Conversion: Captures spoken words and converts them into text using the SpeechRecognition library.

2. Natural Language Processing:
    - ChatGPT Integration: Sends text queries to the ChatGPT API for processing and retrieves informative responses.
    - Dynamic Responses: Generates responses based on a wide range of topics and queries.
3. Text-to-Speech:
    - Audio Feedback: Converts text responses from ChatGPT into spoken words using the gTTS library.
    - Clear Output: Plays the audio responses through a connected speaker, ensuring clear and understandable feedback.
4. Customization:
    - Voice Settings: Allows customization of the assistant’s voice (e.g. male/female, different accents).
    - Response Personalization: Tailors responses based on user preferences and past interactions.

REFERENCES:

- https://blog.arduino.cc/2019/11/12/create-a-voice-controlled-device-with-alexa-and-arduino-iot-cloud-in-7-minutes/
- https://www.instructables.com/Arduino-IoT-Cloud-Smart-Home-With-Alexa-Using-Node/
- https://medium.com/@arthurltonelli/building-an-iot-device-with-alexa-aws-python-and-raspberry-pi-274d941ef3c3
- https://towardsdatascience.com/building-an-intelligent-voice-assistant-from-scratch-3d5749f4af07
-