Final Project: Smart Home Assistant

Note : lines in orange are bonus

1. Introduction

The aim of this project is to design and implement an intelligent Smart Home Assistant capable of controlling home devices using natural language commands. The system integrates open-source Large Language Models (LLMs), a user-friendly interface, and voice capabilities to create an interactive and practical home automation experience.

2. Objectives

- Develop a smart assistant capable of understanding and executing commands to control home devices.
- Utilize function calling with an agent to execute device-specific actions. (You can use LangChain or other libraries to create agent but writing custom agent has bonus)
- Integrate open-source LLMs (LLaMA 3 via TogetherAI or Groq) for natural language understanding.
- Connect LLMs to dynamic data sources like weather, news, and current date and time.
- Build a simple, visually pleasing UI using NoCode platforms.
- Optionally add hardware integration using Raspberry Pi or Arduino for device controlling. (+20 point)
- Support multilingual interactions, especially Persian. (+30 point)
- Add speech interface using Whisper for STT, keyword-based VAD, and TTS for responses. (+30 point)

3. System Architecture

3.1 Components:

- **LLM Backend**: Uses open-source models from TogetherAI and Groq (LLaMA 3 70B).
- **Function Calling Agent**: A Python-based agent parses commands and executes mapped device functions.
- **Device Simulator/Controller**: Python scripts or microcontroller code for each device (lamps, ACs, TV).
- **Data Source Connectors**: APIs for real-time weather, news, date/time.
- **UI**: Built using lovable.dev or v0.dev to interact with users.
- **Voice Stack**:
    - **Whisper** for speech-to-text
    - **VAD** with a wake word ("Hey Assistant")
    - **TTS** using open-source models for audible replies

---

4. Home Device Details

- **Lamps**:
    - Kitchen
    - Bathroom
    - Room 1
    - Room 2
- **AC Units**:
    - Room 1
    - Kitchen
- **Television**:
    - Living Room

---

5. Technologies Used

5.1 Programming Language

- **Python**
- **Technology used for UI does not matter**

5.2 Language Models

- **LLaMA 3.3 70B** via TogetherAI and Groq

5.3 Frontend UI

- **V0.dev**, **Cursor**, or **lovable.dev** for low-code/no-code UI building

5.4 Data APIs

- **Weather API**
- **News API**
- **System date and time**

5.5 Voice Stack

- **STT**: OpenAI Whisper (offline or hosted) (+10 point)

- **VAD**: Silero VAD or Mozilla DeepSpeech with wake-word engine (+10 point)

- **TTS**: Coqui TTS or Piper (open-source) (+10 point)

5.6 Optional Hardware

- **Raspberry Pi** (+20 point)
- **Arduino** (+20 point)

6. Multilingual Support (Bonus)

- Extend prompts to include Persian (Farsi) translation capability
- Integrate bilingual LLM prompts or translation layer for Farsi commands

---

7. Voice Control Features (Bonus)

- **Wake Word Detection**: Listens for trigger phrase (e.g., "Hey Assistant") before activating STT pipeline
- **Real-time Feedback**: Spoken confirmations or responses via TTS

---

8. Conclusion

This project demonstrates the potential of combining LLMs, voice control, and home automation in an intuitive and open-source smart assistant. It brings together cutting-edge models and practical controls to deliver a seamless user experience in both English and Persian, with optional real-world hardware integration.

---

9. Scores

The entire project is worth 100 points, which correspond to completing the required sections. If you also implement the bonus part, you will earn additional points.

9. References

- https://www.together.ai/models/llama-3-3-70b-free
- https://console.groq.com/docs/model/llama-3.3-70b-versatile
- https://github.com/openai/whisper
- https://v0.dev/
- https://www.cursor.com/
- https://lovable.dev/