

# Smart Home Voice Assistant – Technical Overview

---

This document presents a comprehensive technical report for the Smart Home Voice Assistant project, designed and developed by Mohammad Amin Kiani and Radmehr Aghakhani as part of our university project. The system supports voice and text commands in both Persian and English, controlling multiple smart devices via natural language instructions.

## Authors

- Mohammad Amin Kiani (Student ID: 4003613052)
  - Radmehr Aghakhani (Student ID: 4003663002)
- University of Isfahan 403-403

## System Overview

The Smart Home Voice Assistant is a multimodal, bilingual system capable of recognizing user voice/text inputs, interpreting them using an LLM (Meta LLaMA via Together AI), and taking real-time actions via a Flask backend.

Supported device types include:

- TV (Living Room)
- Air Conditioner (Room 1, Kitchen)
- Lamps (Kitchen, Bathroom, Room 1, Room 2)

## Technologies Used

- Python (Flask, threading, os, uuid)
- Together AI API (LLM chat completion)
- OpenAI Whisper (for STT)
- TTS (Text-to-Speech, using Tacotron2 and VITS)
- Googletrans (for language detection and translation)
- HTML + CSS + JavaScript (for frontend interactions)
- Pyngrok (to expose the Flask app online)

## Key Features

- **Voice Command Support:** Full audio input support using microphone access and Whisper-based STT.
- **Device Control:** On/off control for TV, AC, and lamps in specified locations.
- **Bilingual UI & Responses:** Detects Persian vs English and responds appropriately in both text and voice.
- **Dynamic Button Rendering:** After successful command (e.g., “Turn on the TV”), a dynamic button appears to redirect to device-specific status pages (/tv, /ac, /lamp).
- **TTS Feedback:** Assistant speaks back the response using synthesized audio.

## Architecture

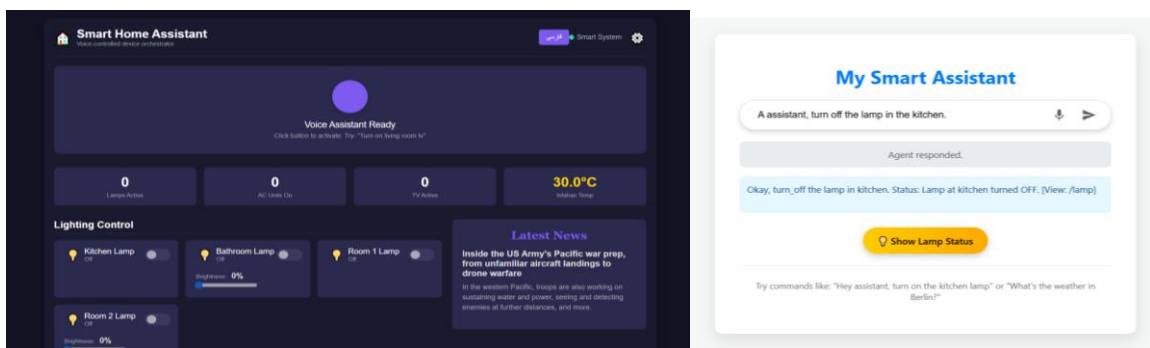
1. **UI Layer:** Built in HTML/JS, it handles text/voice input and response display. Dynamically renders redirection buttons post device command.
2. **Flask Backend:** Manages routes, processes voice/text commands, communicates with the LLM, and generates audio responses.
3. **Smart Agent:** A Python class wrapping LLM logic, translation, and JSON response interpretation.
4. **STT & TTS Layer:** Converts voice to text and back using OpenAI Whisper and TTS APIs.
5. **Public Hosting:** Via ngrok for remote access during testing.

## Usage Examples

- “Hey Assistant, turn on the kitchen lamp”
- “Hey Assistant, turn off the TV in the living room”
- “Hey Assistant, what’s the weather in Tehran?”

## Screenshots

Screenshots demonstrating real-time control, device response, and dynamic UI rendering are included in this documentation:



## Credits

Designed and implemented by Mohammad Amin Kiani and Radmehr Aghakhani under supervision of Dr. Kiani, Faculty of Engineering, University of Isfahan.