



NOTRE DAME UNIVERSITY BANGLADESH

Computer Science and Engineering Department

Project Report

Course Title: Artificial Intelligence and Expert Systems Lab

Course Code: CSE-4104

Project Title: CR Voice Assistant

Submitted By:

Member's Name	ID No	Batch
1. Atika Islam	0692210005101004	CSE-18
2. Minhaj Morshed Chowdhury	0692210005101005	CSE-18
3. Sukla Topoti Saha Prome	0692210005101017	CSE-18

Submitted To:

Humayara Binte Rashid

Lecturer,

Department of CSE, NDUB

Date of Submission: 26.05.2025

Table of Contents:

1. Introduction-----	03
2. Objective-----	03
3. Motivation-----	04
4. Related Work -----	05
5. Methodology-----	06
6. Workflow-----	08
7. GUI Interface-----	08
8. Why Our Project is AI-----	09
9. Conclusion-----	09
10. References-----	10

List of Figures:

1. Fig. 4.1 Architecture Diagram-----	05
2. Fig. 4.2 Flowchart -----	05
3. Fig. 6.1 Workflow-----	08
4. Fig. 7.1 GUI Interface-----	08

Abstract:

This project introduces an advanced-level AI voice assistant for high-quality speech interaction between users and machines. Through the combination of speech-to-text (STT) and text-to-speech (TTS) technologies, the assistant is capable of taking commands via voice and can also reply through voice; thus, the conversation can be natural and hands-free. It's an incredibly versatile assistant that opens websites and system applications, opens chat, has hotword detection, makes WhatsApp calls/video calls, and plays something on YouTube. By combining Python and APIs, this voice assistant showcases the ability to govern logic flow and decision-making based on a set of rules to create the illusion of intelligence without ever using machine learning. The project focuses on accessibility, productivity, and flexibility to create a base for future progress in AI voice technologies.

1. Introduction:

In the last several years, voice has changed the way we interact with technology, as smart assistants such as Alexa, Siri, and Google Assistant are now ubiquitous in people's daily lives. Motivated by this ongoing development, we present in this work a custom AI-based voice assistant that seeks to perform reliable Speech-to-Text (STT) and Text-to-Speech (TTS) to support natural and conversational human-computer interactions. It should be able to not just understand and reply with voice commands but also be able to tie into other tools and services for real use, from managing your schedule, sending emails, running internet searches, and more.

Because it is constructed using Python and a wide range of other open-source libraries and APIs, the assistant provides a versatile, cost-effective, and easy-to-use platform that can easily be customized to individual users. It also explores the intersection of artificial intelligence, natural language processing, and automation, aiming to bridge the gap between theoretical knowledge and real-world applications. This project serves as a hands-on learning experience while demonstrating the feasibility of developing intelligent, voice-enabled systems with scalable and customizable features.

2. Objective:

The major objectives of this project will revolve around the following:

- **Development of the Interaction Voice-Based System:**

This is mainly developing an AI assistant with which users can interact mainly by voice. It implies that the system should be able to understand spoken commands and respond verbally.

- **Integrate Speech-to-Text (STT) and Text-to-Speech (TTS):**

This is one of the core technical objectives. The assistant will need to take the user's spoken language and convert it to text for processing and then convert its text-based responses into spoken language for communication.

- **Tool Integration for Task Fulfillment:**

The personal assistant should be more than just a mere conversational agent. A specific objective should be that such software be whatever it takes, within limits, to complete user requests -- that is, to interface them with other software (or services).

- **Enable Tool Integration for Task Fulfillment:**

The task handling will be handled by the assistant introduced in this project. It intends to fulfill user requests, including managing a WhatsApp call/video, opening the system, hotword detection, playing something on YouTube, and opening a website.

- **Web Search:**

Make the assistant enable us to do a web search. It is in that way that a feature would allow a user to request something online that the assistant would look up and get answers to by voice.

3. Motivation:

There are some reasons to get motivated for this project. These are:

- **Growing Demand for Voice Interfaces:**

With the increasing popularity of smart assistants like Alexa, Siri, and Google Assistant, voice interaction has become a preferred and convenient way for people to engage with technology. This project aims to explore how such systems are built and make them more accessible using Python.

- **Enhancing Accessibility:**

Voice assistants empower people with disabilities by providing hands-free control over devices and applications. Creating a custom assistant can address specific accessibility needs, making technology inclusive.

- **Bridging AI and Daily Life:**

This project highlights the intersection of AI, NLP, and user-friendly interfaces, demonstrating how AI can improve productivity and simplify everyday tasks.

- **Customization Possibilities:**

Unlike commercial assistants, this project explores how to create a personalized voice assistant tailored to specific tasks, workflows, or niche applications, such as helping students, automating workplace tasks, or managing a smart home.

- **Learning and Skill Development:**

Developing a voice assistant allows you to dive into:

- a. Speech Recognition: Understanding how machines interpret human speech.
- b. Natural Language Processing: Building conversational systems.

- c. Text-to-Speech: Synthesizing natural-sounding responses. This hands-on learning helps bridge the gap between theory and real-world AI applications.

4. Related Work:

Artificial intelligence voice assistant and home automation [1]

Contributions:

1. Integration of Voice Assistance with Home Automation
2. Integrated APIs to provide real-time weather updates, news, and search functionality.
3. Leveraged AI to optimize energy consumption in smart homes by controlling appliances intelligently.
4. Enhanced accessibility for users with disabilities or limited mobility through voice control features.

Tools: Python, Pyttsx3, Selenium, Speech-Recognition Library

Limitations:

1. The system struggles with accuracy in noisy environments.
2. The system has some very bad security risks.
3. Limited ability to handle complex or multi-step queries.
4. Context retention for follow-up commands is minimal.

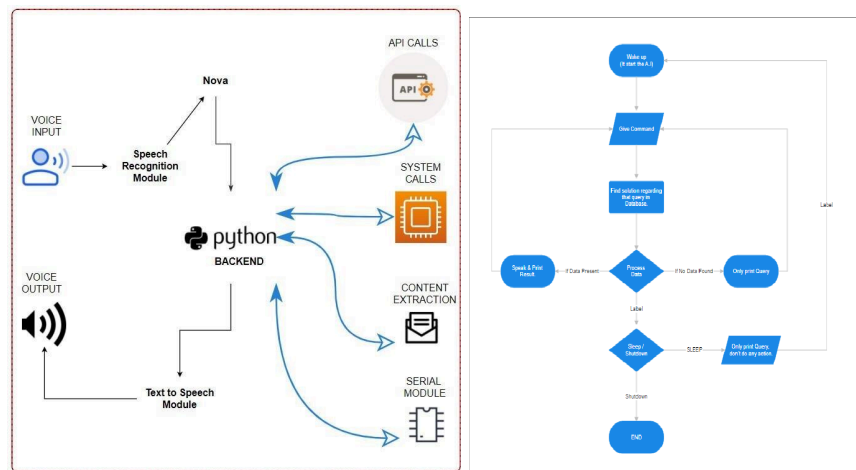


Fig. 4.1 & 4.2: Architecture Diagram and Flow-Chart.

● **AI Voice Assistant: Your Intelligent Conversational Companion [2]**

Features:

1. Speech-to-Text (STT): Convert spoken language into written text.
2. Text-to-Speech (TTS): Generate vocal responses from text input.
3. Vocal Interaction: Engage in natural conversations with the AI assistant.

4. Tool Integration: Utilize built-in tools for calendar management, contact handling, email composition, web searching, and personal knowledge base access.

Tools:

Python 3.9+, Google API credentials (for Calendar, Contacts, and Gmail access), Tavily API key (for web search), Groq API key (for Llama3), Google Gemini API key (for using the Gemini model), Deepgram API key (for voice processing).

5. Methodology:

5.1 Project Setup & Environment Configuration:

Tools & Libraries:

Install Python (3.13 recommended) and set up a virtual environment.

Install dependencies:

- Speech recognition (for voice input),
- Pyttsx3 (text-to-speech synthesis),
- Pywhatkit (for queries and automation),
- requests (API interactions),
- web browser (system-level tasks).

5.2 Speech-To-Text (STT) Processing:

Speech Recognition:

- Use speech_recognition to capture microphone input.
- Implement error handling for background noise or unclear speech.
- Convert audio input to text using Google's Speech-to-Text API.

Wake Word Detection:

Configure a trigger phrase (e.g., "Jarvis/Alexa") to activate the assistant.

5.3 Text-to-Speech (TTS) Response:

Voice Output:

- Use pyttsx3 to convert text responses to audible speech.
- Customize voice parameters (speed, gender, accent) for user preference.

5.4 Core Functionality Implementation:

Basic Commands:

- Open websites (YouTube, Google) with a web browser.
- Set reminders or alarms.

Advanced Features:

- Web searches via pywhatkit.
- Use WhatsApp.
- Hotword Detection.

API Integration:

- Fetch real-time data (e.g., weather, news) by connecting to third-party APIs.
- Securely store API keys using environment variables.

5.5 Logic & Workflow Design:

Command Parsing:

Use if-elif-else logic or NLP techniques (e.g., regex) to map user commands to functions.

Example:

python

if "play music" in command:

 play_on_youtube(query)

elif "send email" in command:

 send_email()

Multithreading: Implement background processes for uninterrupted listening and task execution.

5.6 Testing & Debugging:

Iterative Testing:

- Test individual modules (e.g., speech recognition, email function).
- Validate API responses and error handling (e.g., network issues).

User Feedback:

Improve accuracy by refining voice-input thresholds and command phrases.

5.7 Customization & Scalability:

Add New Features:

Integrate machine learning models (e.g., ChatGPT API) for dynamic responses.

UI Integration:

Build a GUI using ‘eel’ for visual interaction.

6. Workflow:

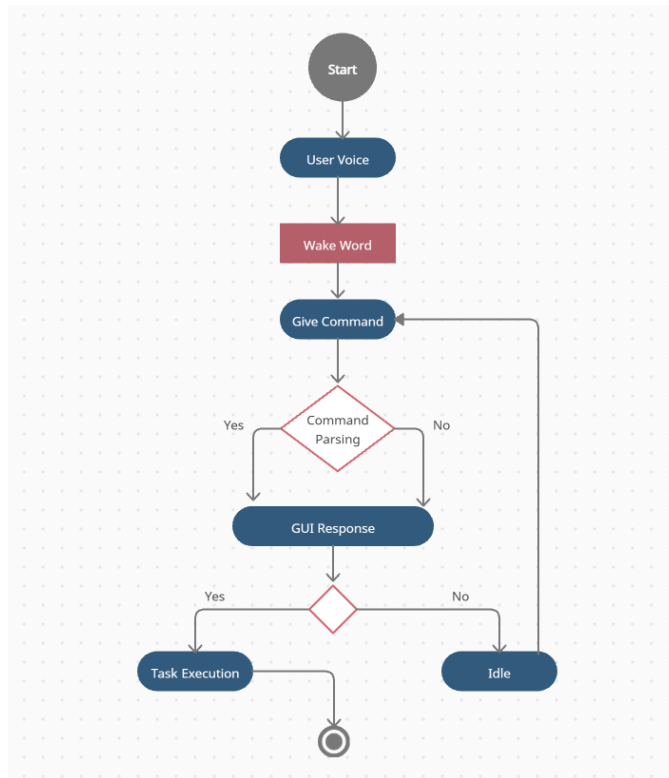


Fig. 6.1: Workflow

7. GUI Interface:

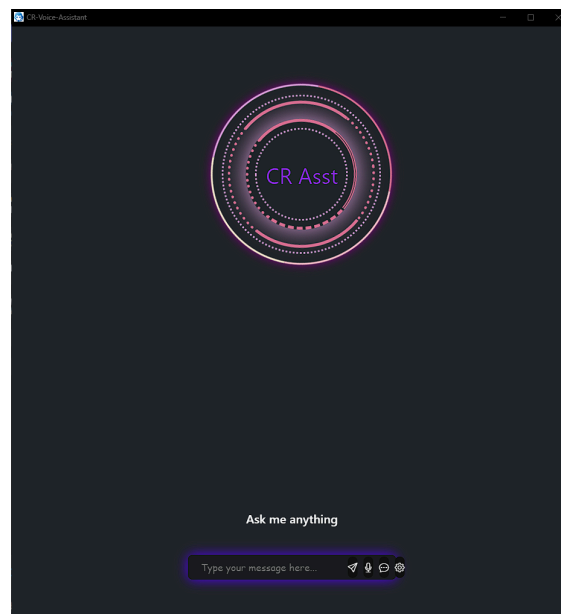


Fig. 7.1: GUI Interface of CR Voice Assistant.

8. Why our project is AI:

- Natural Language Processing (NLP):

A voice assistant uses rule-based or deterministic algorithms to process human language.

- Speech Recognition

Translating spoken words into text requires some level of AI, even if based on deterministic logic.

- Decision-Making

AI enables the voice assistant to decide what action to perform based on user input.

- Context Awareness

The assistant interacts with other systems to perform tasks like sending emails, setting reminders, or turning off lights. This involves decision-making and integrating various rules to automate actions.

- Automation

AI allows the assistant to handle contextual queries using pre-programmed logic.

- Simulating Intelligence

While the voice assistant may not "learn" (no ML), it mimics intelligence by responding logically to queries, often relying on:

1. Rule-based systems (if-else conditions).
2. Predefined responses and patterns.
3. Decision trees for handling complex tasks.

9. Conclusion:

This project successfully demonstrates the design and implementation of an AI-powered voice assistant capable of facilitating natural human-computer interaction through speech. By integrating core technologies such as Speech-to-Text (STT), Text-to-Speech (TTS), and various external APIs, the assistant can understand user commands, execute tasks, and deliver vocal responses in real time. From managing WhatsApp tasks and performing web searches to handling personal tasks, the assistant exemplifies how artificial intelligence can be practically applied to enhance productivity, accessibility, and user experience.

While the assistant currently relies on rule-based logic and predefined workflows, it lays a strong foundation for future enhancements, such as incorporating machine learning for improved contextual understanding and personalization. The project not only bridges theoretical AI concepts with hands-on application but also highlights the potential for customization and scalability in voice-driven systems. As voice interfaces continue to grow in relevance, this work

contributes meaningfully to the development of accessible and intelligent digital assistants tailored to everyday needs.

Future works of our project is to integrate APIs to provide real-time weather updates, news, and search functionality. And enhance accessibility for users with disabilities or limited mobility through voice control features.

10.Reference:

- [1]. Singh, N. S., Panwar, N. S. S., Dahiya, N. H., & Khushboo, N. (2024). Artificial intelligence voice assistant and home automation. *International Journal of Science and Research Archive*, 12(1), 2006–2017. <https://doi.org/10.30574/ijrsra.2024.12.1.0954>
- [2]. kaymen99. “*GitHub - Kaymen99/AI-Voice-Assistant: AI Voice Assistant: Talk to an AI Agent That Helps You with Event Scheduling, Contact Management, Accessing Your Knowledge Base, and Web Searches Using Simple Voice Commands.*” GitHub, 2024, github.com/kaymen99/AI-Voice-assistant/. Accessed 25 May 2025.
- [3]. ProjectsWithDigambar. “*Projectswithdigambar/Jarvis.*” GitHub, 8 July 2024, github.com/projectswithdigambar/jarvis. Accessed 5 May 2025.