VOICE ASSISTANT

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology/Master of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

Jenil Padshala (AP21110010064)

Under the Guidance of

(Mrs. Karnena Kavitha Rani)

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

[December 2022]

Certificate

This is to certify that the work present in this Project entitled "**VOICE ASSISTANT**" has been carried out by **Jenil Padshala** under my supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in **School of Engineering and Sciences**.

Supervisor

(Signature)

Prof. / Dr. [Name]

Designation,

Affiliation.

Co-supervisor

(Signature)

Prof. / Dr. [Name]

Designation,

Affiliation.

Table of Contents

Abstract

The project aims to build an interactive voice assistant using Python language.

It should be able to perform the following tasks:

1. Greet the user based on the current time of the day ("Good Morning" in morning, "Good Evening" during evening time and so on).
2. Inform current time and date to the user when asked.
3. Search a song on Spotify
4. Perform a Google search.
5. Search on YouTube.
6. Remind the user about "Today's Events" from his Google Calendar.
7. Tell the user some jokes.

1. Introduction

This assistant is capable of using device's microphone to receive voice requests and processing on it to perform particular task.

This program is developed using Python language and its different modules and libraries.

For this project we have used modules necessary for voice recognition and analysis, accessing API and creating GUI.

1.1 Modules Required:

- **SpeechRecognition:** Library for performing speech recognition
- **GTTS:** Google-Text-To-Speech, to convert text into audio file.
- **Playsound:** Single function module to play audio file
- **Datetime:** Used to access current date and time.
- **OS:** To create/delete file directory.
- **Calendar:** To perform operations related to date.
- **Webbrowser:** to display web-based documents to user.
- **Pyjokes:** Library of one-line jokes for programmers
- **threading:** To allow separate flow of execution
- **customtkinter:** For Graphical User Interface (GUI)
- **warnings:** to deal with warnings and prevent program from terminating
- (Few other modules have been installed for accessing google account for Calendar API)
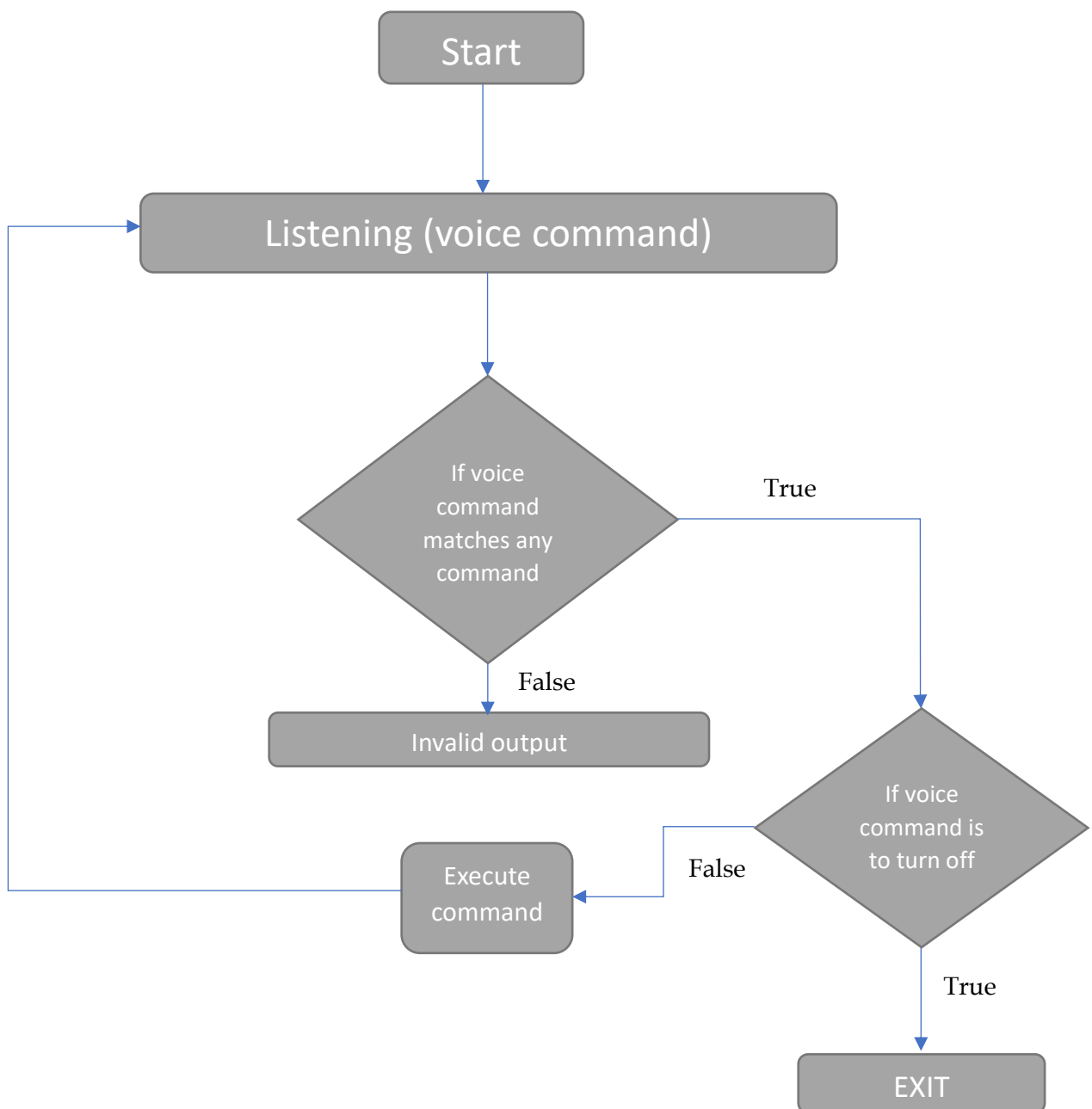
1.2 Working Of the Program:

1. First, the user clicks the "START" button to call the assistant.
2. To assign a task to the assistant, the user must call it using the wake word "assistant".
3. The program identifies the task and will display the desired output if it can perform the task.
4. To exit the program command 'see you later', the assistant will exit the program on its own.

2. Project Implementation:

The program is implemented using a few important functions and if-elif-else statements.

2.1 Flowchart:

```
                    ┌─────────────┐
                    │    Start    │
                    └─────────────┘
                           │
                           ▼
        ┌──────────────────────────────────────┐
        │      Listening (voice command)        │
        └──────────────────────────────────────┘
                           │
                           ▼
                    ╱──────────────╲
                   ╱   If voice      ╲        True
                  ╱    command        ╲──────────────┐
                  ╲    matches any     ╱             │
                   ╲   command        ╱              │
                    ╲────────────────╱               │
                           │                         ▼
                        False                  ╱──────────────╲
                           ▼                  ╱   If voice      ╲
                 ┌──────────────────┐        ╱    command is     ╲    True
                 │  Invalid output  │   False ╲   to turn off    ╱────────┐
                 └──────────────────┘    ┌─────╲────────────────╱         │
                                         │                                 ▼
                              ┌──────────────────┐              ┌─────────────┐
                              │     Execute       │              │    EXIT     │
                              │    command        │              └─────────────┘
                              └──────────────────┘
```

2

2.2 Functions:

1. **greet (): Function to greet the user**
    - if current hr<12, respond "Good Morning"
    - if current hr is between 12 and 16, respond "Good Afternoon"
    - if current hr>16, respond "Good Evening"

2. **today (): Function to know today's day and date**
    - using datetime module, identify and the date, week,month, day and year
    - return a string with the above data (Eg. Today is Monday, 12th December 2022)

3. **currentTime(): Function to know the current time**
    - using datetime module, identify the current hr and min.
    - if hr>12, print 'pm' after the time
    - else, print 'am' after the time
    - return hr, min and postfix in string format

4. **rec_audio(): Function to recognize and record audio**
    - Using speech recognition module, record the user's audio.
    - Convert the audio into string.
    - Return the string.

5. **response (): Function to output audio**
    - Using gtts module, the string value is converted into audio file and stored in the folder.
    - Using playsound module, the audio is played.
   - Immediately, the audio file is deleted using os.remove()

6. **google_calendar (): Function to retrieve today's google calendar tasks**
    - Using the Google Calendar API, access today's tasks
    - Using response () function, output the list of tasks retrieved.

3

2.3 Graphical User Interface (GUI):

The graphical user interface of the program has been implemented using customized version of tkinter module (customtkinter).

1. First, we created 500X500 px window frame.
2. Next, we add a logo that changes colour when the assistant speaks.
3. Below the logo, we have a "START" button to call the assistant using the concept of threading.
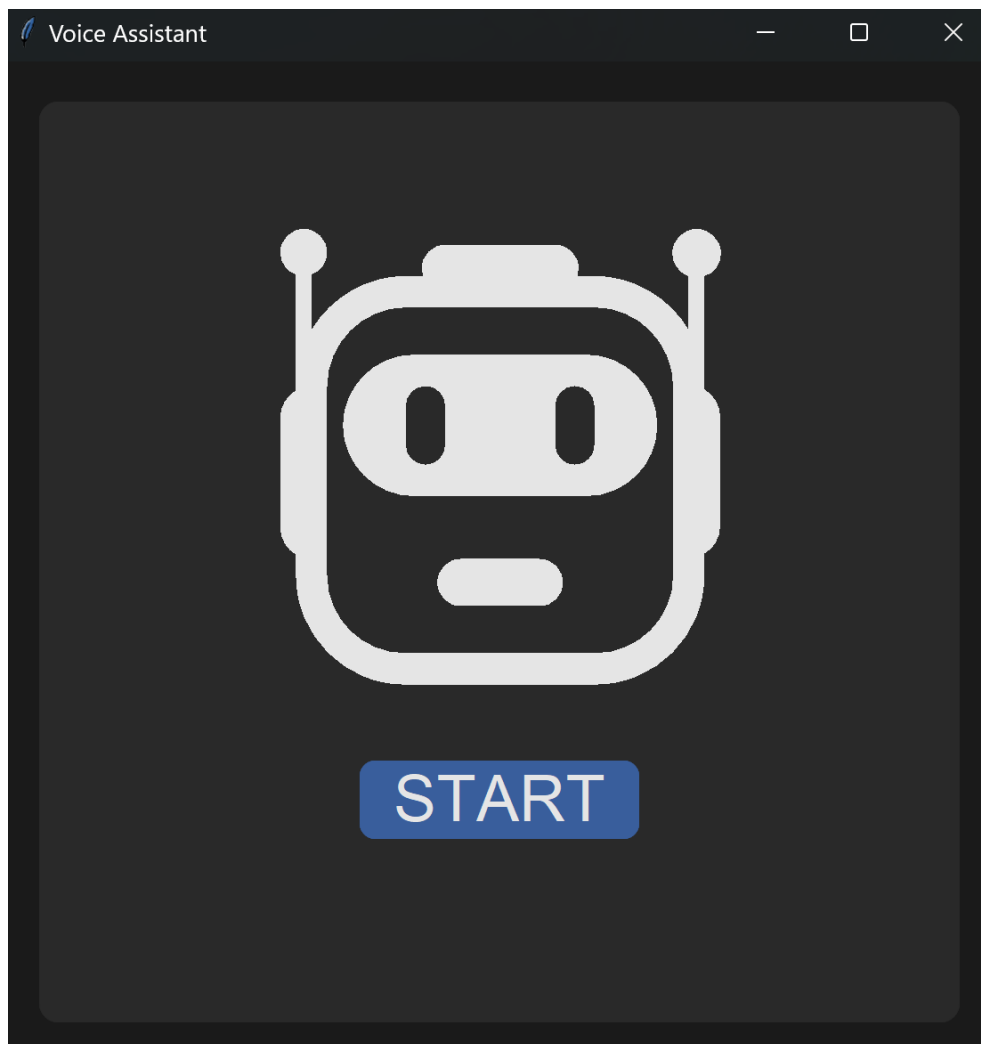
(Fig. Voice Assistant GUI)

3. Hardware And Software Requirements

3.1 Hardware:
1. Microphones
2. Processor – Intel i5 & above or AMD Ryzen 4 & above

3.2 Software:
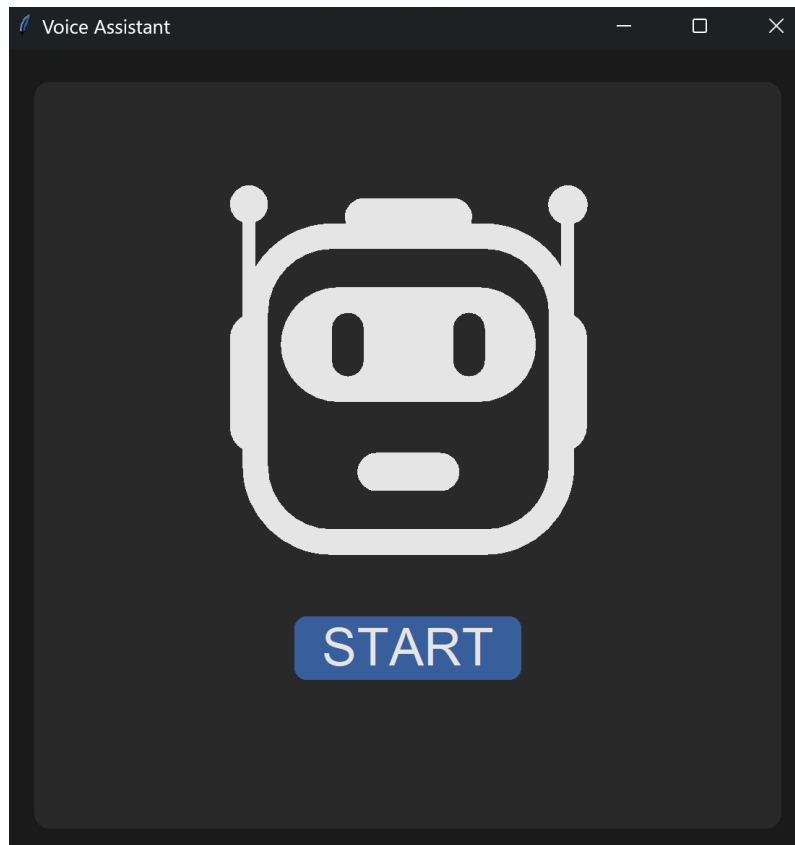1. Python 3.10 or above
2. Compatible IDE: VS Code or PyCharm

4. Results

The program displays a window for the assistant.

Rest of the process occurs in background.

GUI Window for the Voice Assistant:

(Fig. Voice Assistant GUI)

Background process (displayed in IDE Terminal):

```
Good Morning
Listening...
You said: Assistant Google Python Language
Searching ['Python', 'Language'] on Google
Listening...
Could you please repeat
Listening...
You said: Assistant see you later
Good bye
```

******