LOVELY PROFESSIONAL UNIVERSITY

Transforming Education Transforming India

REPORT FOR DESKTOP VOICE ASSISTANT

As a project work for Course

PYTHON PROGRAMMING (INT 213)

Name: Ashish Kumar

Registration Number: 12004718

Name: Mansi Srivastav

Registration Number: 12008464

Program: B.Tech (CSE)

Semester: Third

School: School of Computer Science and Engineering

Date of Submission: 20th November 2021

Table of Contents

ABSTRACT:-

As the technology is developing day by day people are becoming more dependent on it, one of the mostly used platform is computer. We all want to make the use of these computers more comfortable, traditional way to give a command to the computer is through keyboard but a more convenient way is to input the command through voice. Giving input through voice is not only beneficial for the normal people but also for those who are visually impaired who are not able to give the input by using a keyboard. For this purpose, there is a need of a voice assistant which can not only take command through voice but also execute the desired instructions and give output either in the form of voice or any other means.

INTRODUCTION: -

An AI personal assistant is a piece of software that understands verbal or written commands and completes task assigned by the client. It is an example of weak AI that is it can only execute and perform quest designed by the user.

SKILLS: -

The implemented voice assistant can perform the following task it can open YouTube, Gmail, Google chrome and GeeksforGeeks. Predict current time and date, take a note, search Wikipedia to abstract required data, can play music get top headline news from times of India and you can do some normal talk as well.

TEAM MEMBERS: -

Ashish Kumar: -

Contributions: -

1.Coding(joined)

2.GUI

GITHUB LINK:

https://github.com/Ashish-Kumar1437/ProjectAI

Mansi Srivastav: -

Contributions: -

1.Coding(joined)

2.Report

GITHUB LINK:

https://github.com/mansi23-01/projectAI

LIBRARIES: -

1)Speech recognition:

Speech recognition is an important feature used in house automation and in artificial intelligence devices. The main function of this library is it tries to understand whatever the humans speak and converts the speech to text.

2)pyttxs3:

pyttxs3 is a text to speech conversion library in python. This package supports text to speech engines on Mac os x, Windows and on Linux.

3)Wikipedia:

Wikipedia is a multilingual online encyclopedia used by many people from academic community ranging from freshmen to students to professors who wants to gain information over a particular topic. This package in python extracts data's required from Wikipedia.

4)Datetime:

This is an inbuilt module in python and it works on date and time.

5)os:

This module is a standard library in python and it provides the function to interact with operating system.

6)Time:

The time module helps us to display time.

7)Web browser:

This is an in-built package in python. It extracts data from the web.

8)Tkinter:

Tkinter is the de facto way in Python to create Graphical User Interfaces (GUIs) and is included in all standard Python Distributions. In fact, it's the only framework built into the Python standard library.

9)Pillow:

This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities.

10)Pyjokes:

Pyjokes is a python library for one-line jokes for programmers (jokes as a services). You can get funny one-liner, mostly related to programming.

11)Pygame:

It consists of computer graphics and sound libraries designed to be used.

12)PyAutoGUI:

PyAutoGUI is a Python automation library used to click, drag, scroll, move, etc. It can be used to click at an exact position.

13)Random:

Python Random module is an in-built module of Python which is used to generate random numbers. These are pseudo-random numbers means these are not truly random. This module can be used to perform random actions such as generating random numbers, print random a value for a list or string, etc.

IMPLEMENTATION: -

Import the following libraries:

```
1   from tkinter import *
2   from tkinter.font import BOLD, ITALIC, names
3   from PIL import Image,ImageTk
4   import pyttsx3
5   import speech_recognition as sr
6   import datetime
7   import wikipedia
8   import webbrowser
9   import os
10  import random
11  import pyautogui
12  from pygame import mixer
13  import pyjokes
14  from time import sleep,strftime
```

Setting up the speech engine:

The **pyttsx3** module is stored in a variable name engine.

Sapi5 is a Microsoft Text to speech engine used for voice recognition.

The voice Id can be set as either 0 or 1,

0 indicates Male voice

1 indicates Female voice

```
20  t=0
21  engine =pyttsx3.init('sapi5')
22  voices=engine.getProperty('voices')
23  engine.setProperty('voice',voices[t].id)
```

SPEAK FUNCTION:

Speak function converts text to speech. The speak function takes the text as its argument, further initialize the engine.

RunAndWait: This function Blocks while processing all currently queued commands. It Invokes callbacks for engine notifications appropriately and returns back when all commands queued before this call are emptied from the queue.

```
25    def speak(text):
26        write(f'{assistant_name} : {text}')
27        root.update()
28        engine.say(text)
29        engine.runAndWait()
```

TAKECOMMAND FUNCTION:

Takecommand function is used to understand and to accept human language. The microphone captures the human speech and the recognizer recognizes the speech to give a response.

The exception handling is used to handle the exception during the run time error and, the **recognize_google** function uses google audio to recognize speech.

```
32  def takeCommand():
33      r = sr.Recognizer()
34      with sr.Microphone() as source:
35
36          print("Listening...")
37          audio = r.listen(source,phrase_time_limit = 3)
38      try:
39          print("Recognizing...")
40          query = r.recognize_google(audio, language='en-in')
41          write(f"You : {query}")
42          root.update()
43      except Exception as e:
44          print(e)
45          print("Unable to Recognize your voice.")
46          return 'none'
47      return query
```

WISH ME FUNCTION:

WishMe function is used **to greet the user.**

The **now().hour** function abstract's the hour from the current time.

If the hour is greater than zero and less than 12, the voice assistant wishes you with the message "Good Morning".

If the hour is greater than 12 and less than 18, the voice assistant wishes you with the following message "Good Afternoon".

Else it voices out the message "Good evening".

```
49    def wishme():
50        hour=datetime.datetime.now().hour
51        if hour >= 0 and hour < 12:
52
53            speak("Hello,Good Morning")
54
55        elif hour >= 12 and hour < 18:
56
57            speak("Hello,Good Afternoon")
58
59        else:
60
61            speak("Hello,Good Evening")
```

START_AI FUNCTION:

The main function starts from here, the commands given by the humans is stored in the variable **statement**.

```
63  def Start_AI():
64      menuframe.pack_forget()
65      clock.pack_forget()
66      labelf.pack_forget()
67      start_b.pack_forget()
68      workscreen()
69      root.update()
70      run=1
71      wishme()
72      while run==1:
73
74          statement=takeCommand().lower()
75          if 'none' in statement:
76              continue
```

Skill 1 -Fetching data from Wikipedia:

The following commands helps to open Wikipedia and extract information from Wikipedia. The **webbrowser.open_new_tab()** function takes one arguments URL of the site to be opened.
The **wikipedia.summary()** function takes two arguments, the statement given by the user and how many sentences from Wikipedia is needed to be extracted is stored in a variable **result.**

```
77          if 'wikipedia' in statement:
78              if 'open wikipedia' in statement:
79                  webbrowser.open_new_tab('http://www.wikipedia.com')
80              else:
81                  statement.replace('wikipedia' , '')
82                  result=wikipedia.summary(statement,3)
83                  speak('According to wikipedia baba..')
84                  print(result)
85                  speak(result)
```

Skill 2 -Accessing the Web Browsers — Google chrome, G-Mail, GeeksForGeeks and YouTube:

The web browser extracts data from web. The **open_new_tab** function accepts **URL** as a parameter that needs to be accessed.

```
87          elif 'open google' in statement:
88              speak('Google rolling in..')
89              webbrowser.open_new_tab('https://www.google.com')
90
91          elif 'open youtube' in statement:
92              speak('Opening Youtube')
93              webbrowser.open_new_tab('http://www.youtube.com')
94
95          elif 'open gmail' in statement:
96              webbrowser.open_new_tab("gmail.com")
97              speak("opening Google Mail")
98
99          elif 'open geeksforgeeks' in statement or 'open geeks'in statement:
100             speak('Opening GeeksForGeeks')
101             webbrowser.open_new_tab('http://www.geeksforgeeks.org')
```

Skill 3 – Play and Pause Music:

The **os.listdir()** is used to get the list of all files and directories in the specified directory. The **os.path.join()** method combines one or more

path names into a single path. The **os.startfile()** method allows us to start a file with its associated program.

The **pyautogui.press()** is used to press any key which it take as an attribute.

```
102         elif 'play music' in statement or 'play song' in statement:
103             speak('Here you go with Music')
104             dir='song'
105             song_list=os.listdir(dir)
106             os.startfile(os.path.join(dir,random.choice(song_list)))
107 
108         elif 'stop song' in statement or 'stop music' in statement:
109             pyautogui.press('playpause')
110             sleep(0.5)
```

Skill 4 – To fetch latest news:

The voice assistant fetches top headline news from Time of India by using the web browser function.

```
112         elif 'news' in statement:
113             news = webbrowser.open_new_tab("https://timesofindia.indiatimes.com/home/headlines")
114             speak('Here are some headlines from the Times of India,Happy reading')
115 
```

Skill 5 – Searching data from web:

From the **web browser** you can **search** required data by passing the user statement (command) to the **open_new_tab()** function

```
116         elif 'search'  in statement:
117             statement = statement.replace("search", "")
118             webbrowser.open_new_tab(statement)
```

Skill 6 – Tell Date and Time:

The current time is abstracted from **datetime.now()** function which displays the hour, minute and second and is stored in a variable name **curtime.** The **datetime.today()** is used abstract today date.

```
120         elif 'time' in statement:
121             curtime=datetime.datetime.now().strftime('%H:%M:%S')
122             hour=int(curtime[0:2])
123             min=int(curtime[3:5])
124             str='AM'
125             if hour>12 :
126                 hour=hour%12
127                 str='PM'
128             print(hour," ",min)
129             speak(f'Time is {hour}{min}{str}')
130 
131         elif 'the date' in statement:
132             date=datetime.datetime.today().strftime(r"%d %m %y")
```

Skill 7 –Switch Window:

The short cut for switch window is “alt+tab”. **pyautogui.keyDown()** is used to hold down any key and **pyautogui.keyUp()** to release the key.

```
134         elif 'switch window' in statement or 'next window' in statement:
135             pyautogui.keyDown('altleft')
136             pyautogui.press('tab')
137             sleep(0.5)
138             pyautogui.keyUp('altleft')
```

Skill 7 –Joke:

The **get_joke()** function abstract a one-line joke.

```
140         elif 'joke' in statement:
141             joke=pyjokes.get_joke()
142             print(joke)
143             speak(joke)
```

Skill 8 –Make Note:

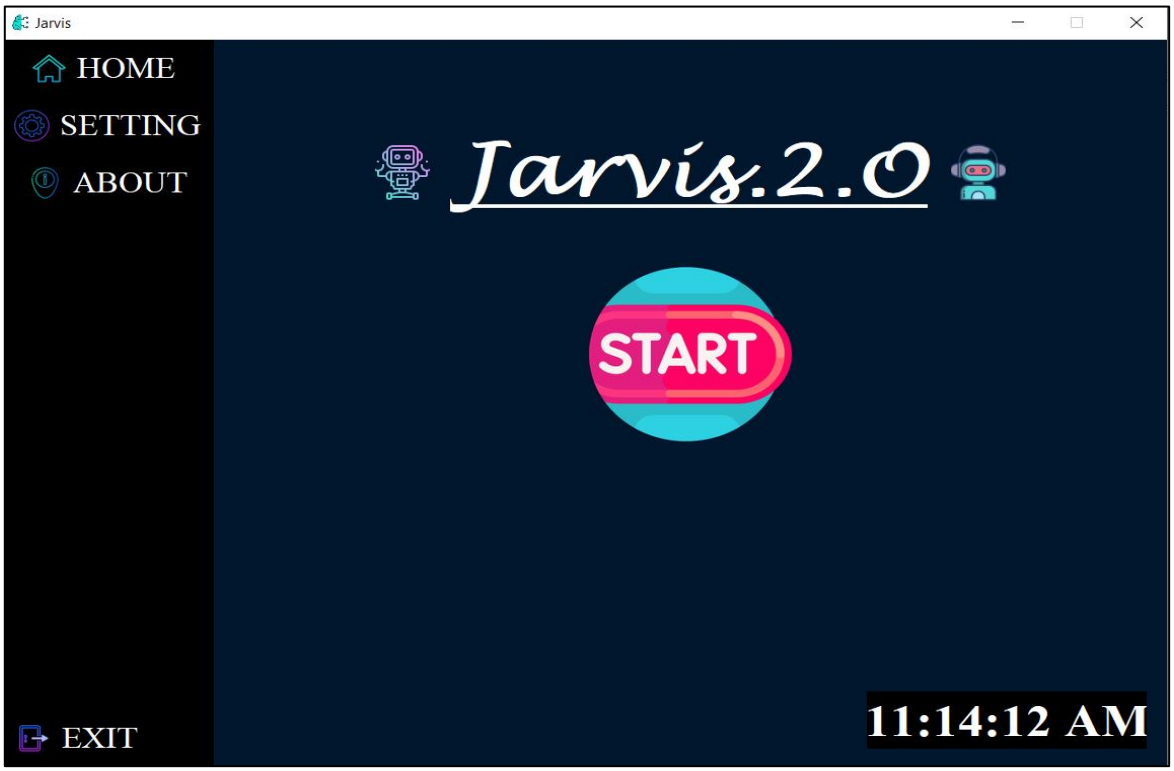
```
145         elif 'make a note' in statement or 'write this down' in statement:
146             speak('What would you like me to write down?')
147             start=1
148             note=''
149             while start==1:
150                 statement=takeCommand().lower()
151                 if 'save this file' not in statement:
152                     if statement == 'none':
153                         speak('Can You repeat this statement again')
154                     else:
155                         note=note+" "+statement
156                 else:
157                     start=2
158             speak('what should be the fie name?')
159             file=takeCommand().title()
160             f=open(os.path.join(f'{os.getcwd()}/Documents',f'{file}.txt'),'w')
161             f.write(note)
162             f.close()
163             speak('File Saved')
```

Skill 9 – Some Common Talks:

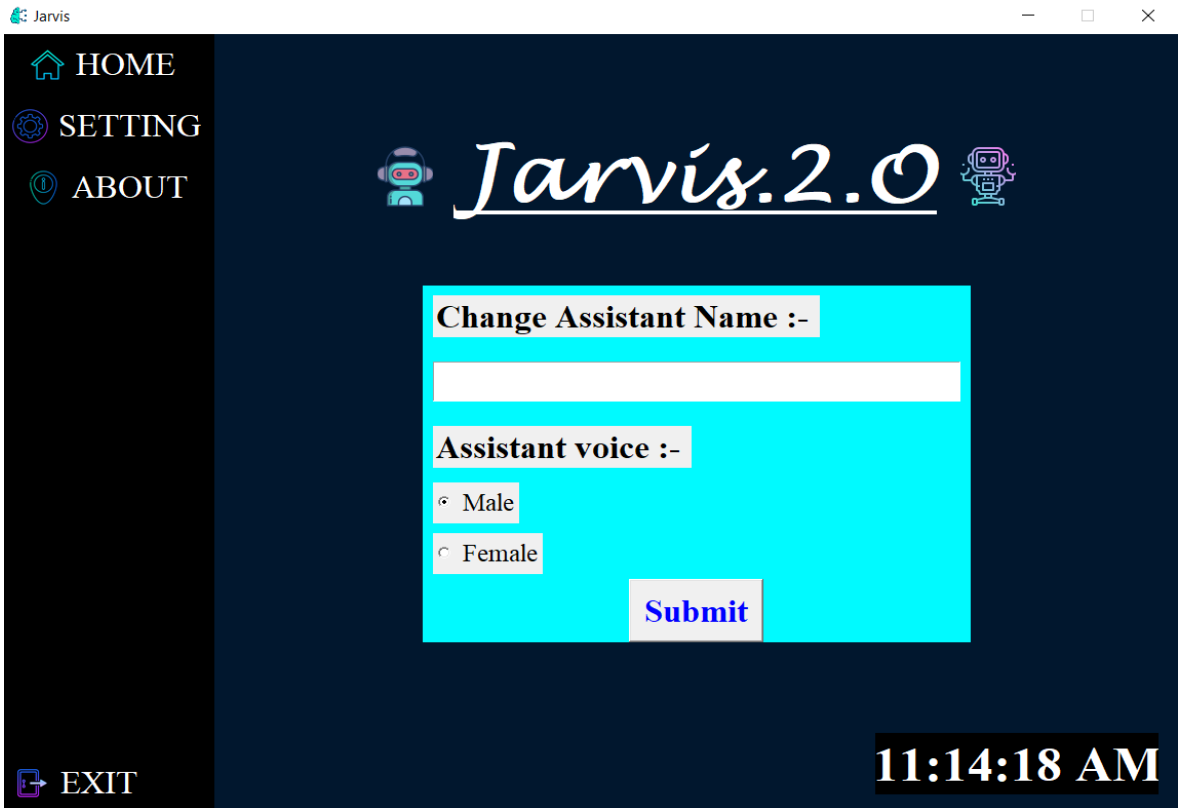
```
165     elif "hello" in statement:
166         wishme()
167
168     elif "thank you" in statement or "thanks" in statement:
169         speak("welcome sir")
170
171     elif "can you do for me" in statement:
172         speak("I can do multiple tasks for you. tell me whatever you want to perform")
173
174     elif "old are you" in statement:
175         speak("I am a little baby sir")
176
177     elif 'who are you' in statement or 'what can you do' in statement:
178         speak(f"I am {assistant_name} version 2 point 0 your personal assistant. I am programmed to minor tasks like opening youtube,google chrome, gmail and
            Geeksforgeeks ,tell date and time, search wikipedia, get top headline news from times of india and you can do some normal talk as well !")
179
180
181     elif "who made you" in statement or "who created you" in statement or "who discovered you" in statement:
182         speak("My Creators are Ashish and Mansi")
183
184     elif "goodbye" in statement or "offline" in statement or "bye" in statement:
185             speak("Alright sir, going offline. It was nice working with you")
186             run=0
```

GUI SCREENSHOT:-

Home Screen:

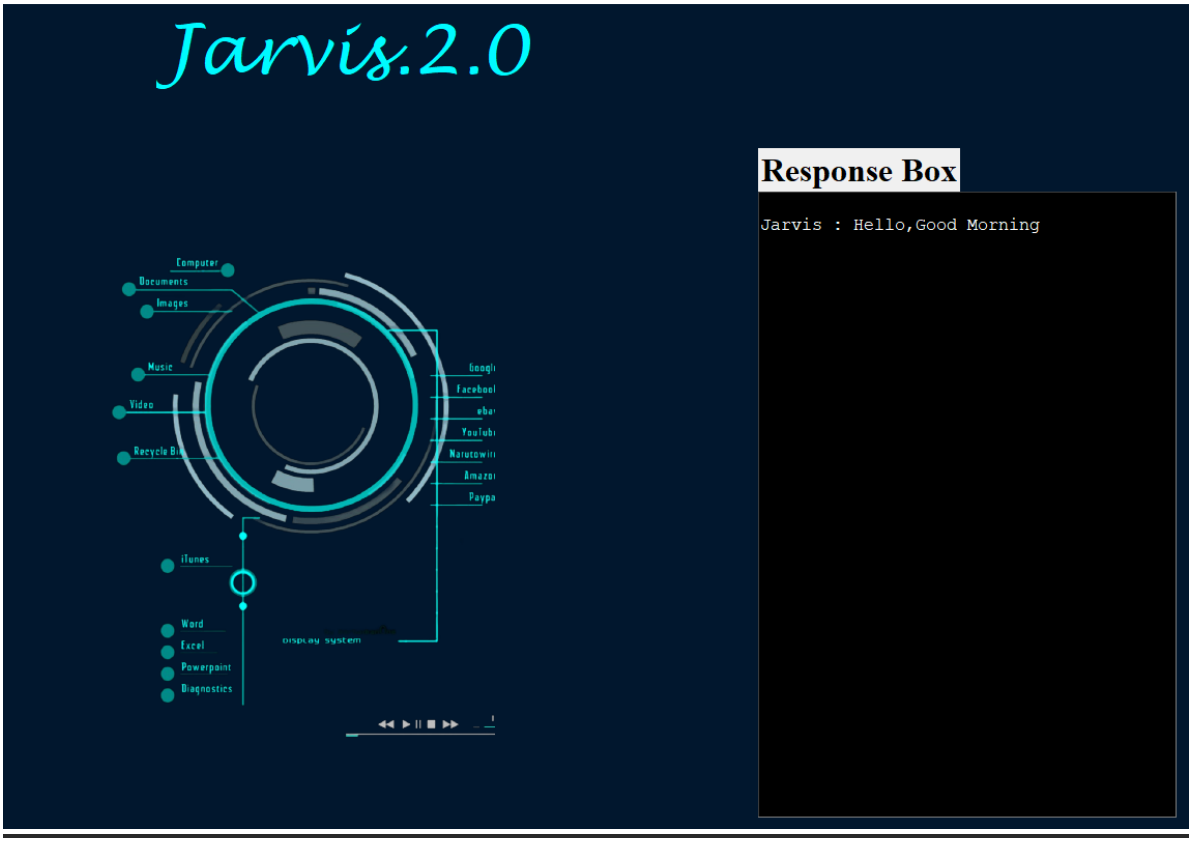


Setting Screen:

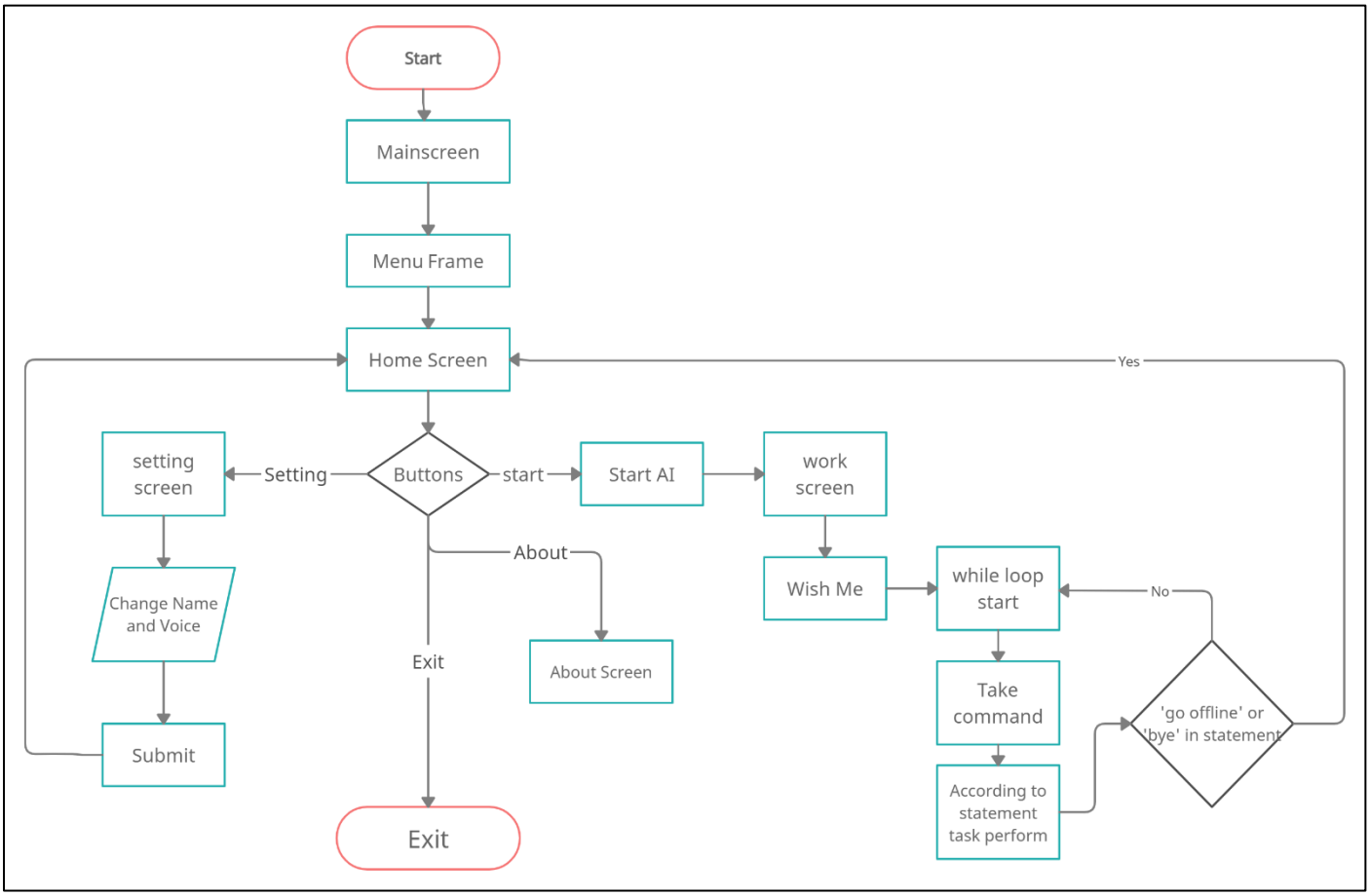


About Screen:

Work Screen:



Work Flow of AI:



TESTING:-

Testing is a process by which one can find problems or error in there project.

Test Cases:

S.No	Test	Description	Result
1.	GUI Window	Running all GUI windows one by one and their respective functionality	Passed
2.	Task	Running each task one by one	Passed
3.	Complete Program	Running Complete Algo	Passed

CONCLUSION:-

This is a Simple Desktop assistant which can perform some simple tasks like opening webpages, telling date and time, search something and play songs. We will definitely add some more tasks.

REFRENCES:-

Tkinter -> https://docs.python.org/3/library/tkinter.html

Speech Recognition -> https://www.geeksforgeeks.org/speech-recognition-in-python-using-google-speech-api/

Pyautogui -> https://pyautogui.readthedocs.io/en/latest/quickstart.html