A Project Report

On

PERSONAL VOICE ASSISTANT

Submitted in partial fulfillment of the requirements for the award of the degree

Of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

BY

HASSAN ALI AL IDROOS	**160317733023**
HAMED MOHIUDDIN REHAN	**160317733043**
SULTAN MOHIUDDIN	**160317733049**

Under the guidance

Of

Mrs. Razia Begum

Assistant Professor

Department Of CSE, DCET

Department of Computer Science Engineering

Deccan College of Engineering and Technology

(Affiliated to Osmania University)

Hyderabad

2021

CERTIFICATE

This is to certify that the project report entitled **“PERSONAL VOICE ASSISTANT”** being submitted by **Hassan Ali Al Idroos (160317733023), Hamed Mohiuddin Rehan (160317733043), Sultan Mohiuddin (160317733049)** in partial fulfillment for the award of the Degree of **BACHELOR OF ENGINEERING IN COMPUTER SCIENCE** by **OSMANIA UNIVERSITY** is a record of bonafide work carried out by them under our guidance and supervision.

Internal Guide
Mrs. RAZIA BEGUM
Asst.Professor
Department of CSE
DCET, Hyderabad

External
Examiner

Head of the Department
DR. SYED RAZIUDDIN
Professor
Department of CSE
DCET, Hyderabad

DECLARATION

This is to certify that the work reported in the present project entitled "**PERSONAL VOICE ASSISTANT**" is a record of work done by us in the Department of Computer Science & Engineering, Deccan College of Engineering and Technology, Osmania University. The reports are based on the project work done by us and not copied from any other sources

The results presented in this dissertation have been verified and are found to be satisfactory. The results embodied in the dissertation has not been submitted to any other university for the award of any degree or diploma.

HASSAN ALI AL IDROOS
(160317733023)

HAMED MOHIUDDIN REHAN
(160317733043)

SULTAN MOHIUDDIN
(160317733049)

ACKNOWLEDGEMENT

We would like to express utmost respect and my deep sense of gratitude to **Dr .M.A. MALIK,** principal, who has given me an opportunity to undertake this major project on **“PERSONAL VOICE ASSISTANT”** and for providing us the conductive environment for carrying through our academic schedules and projects with an ease

We are thankful to Head of the department **DR.SYED RAZIUDDIN** for providing the necessary facilities during the execution of our project work.

We would like to express our sincere gratitude and indebtedness to our internal guide **Mrs. RAZIA BEGUM** for her valuable suggestions and interest throughout the course of this project.

This project would not have been a success without her. So, we would extend our deep sense of gratitude for the effort she took in guiding us in all the stages of completion of our project work.

We convey our heartfelt thanks to our Parents, Friends, Technical and Non-Technical staff of the college for their constant support in the successful completion of the project.

HASSAN ALI AL IDROOS	**160317733048**
HAMED MOHIUDDIN REHAN	**160317733043**
SULTAN MOHIUDDIN	**160317733049**

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The project aims to develop a voice-assistant for Windows-based systems. Jarvis draws its inspiration from virtual assistants like Cortana for Windows, and Siri for iOS.

It has been designed to provide a user-friendly interface for carrying out a variety of tasks by employing certain well-defined commands. Users can interact with the assistant through voice commands .

As a personal assistant, Jarvis assists the end-user with day-to-day activities like general human conversation, searching queries in google, bing or yahoo, searching for videos, retrieving images, live weather conditions, word meanings, searching for medicine details, health recommendations based on symptoms and reminding the user about the scheduled events and tasks.

The user statements/commands are analysed with the help of Python library modules such as speech recognition, pyaudio,pyttsx3 etc., to give an optimal solution.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

CHAPTER 1

1.1 INTRODUCTION

In the 21st century, human interaction is being replaced by automation very quickly. One of the main reasons for this change is performance. There's a drastic change in technology rather than advancement. In today's world, we train our machine's to do their tasks by themselves or to think like humans using technologies like Machine Learning, Neural Networks, etc. Now in the current era, we can talk to our machines with the help of virtual assistants. There are companies like Google, Apple, Microsoft, etc with virtual assistants like Google Now, Siri, Cortana, etc. which helps their users to control their machine by just giving input in the form of voice. These types of virtual assistants are very useful for old age, blind & physically challenged people, children, etc. by making sure that the interaction with the machine is not a challenge anymore for people. Even blind people who couldn't see the machine can interact with it using their voice only[1]. Some of the basic tasks that are supported by most of the virtual assistants are :

- Checking weather updates
- Sending and checking mails
- Search on Wikipedia
- Make and receive calls
- Stream music
- Open applications
- Text messages etc.

The voice assistant we have developed is a desktop-based built using python modules and libraries. This assistant is just a basic version that could perform all the basic tasks which have been mentioned above but current technology is although good in it is still to be merged with Machine Learning and Internet Of Things(IoT) for better enhancements. The understanding and executing commands are still to reach a new level like the virtual assistant of the iron man named Jarvis.This is although fictional yet this is what that can be achieved using virtual assistants. All you need to do is give a command to the assistant and the rest will be performed by the assistant. With the help of voice-activated virtual assistants, there will be no need to write long codes to perform a task, the system will do so for us. The machine will work in three modes- supervised, unsupervised or reinforcement learning depending upon the usage for which the assistant is developed. This is all possible with the help of machine learning.

It is a software agent that can perform tasks or services for an individual based on commands or questions. Virtual assistants are able to interpret human speech and respond via synthesized voices. Users can ask their assistants questions, control home automation devices and media playback via voice, and manage other basic tasks such as email, to-do lists, and calendars with verbal commands

Virtual Assistants help you ease your day to day tasks, such as showing weather report, creating reminders, making shopping lists etc. They can take commands via text (online chat bots) or by voice. Voice based intelligent assistants need an invoking word or wake word to activate the listener, followed by the command. We have so many virtual assistants, such as Apple's Siri, Amazon's Alexa and Microsoft's Cortana.

This system is designed to be used efficiently on desktops. Personal assistant software improves user productivity by managing routine tasks of the user and by providing information from online sources to the user Voice searches have dominated over text search. Web searches conducted via mobile devices have only just overtaken those carried out using a computer and the analysts are already predicting that 50% of searches will be via voice by 2020.Virtual assistants are turning out to be smarter than ever. Allow your intelligent assistant to make email work for you. Detect intent, pick out important information, automate processes, and deliver personalized responses.

This project was started on the premise that there is sufficient amount of openly available data and information on the web that can be utilized to build a virtual assistant that has access to making intelligent decisions for routine user activities.

1.2 Problem Statement

We are all well aware about Cortana, Siri, Google Assistant and many other virtual assistants which are designed to aid the tasks of users in Windows, Android and iOS platforms. But to our surprise, there's no such virtual assistant available for the paradise of Developers. This project aims at developing a personal assistant for Windows-based systems. The main purpose of the project is to perform the tasks of the user at certain commands, provided in the form of speech. It will ease most of the work of the user as a complete task can be done on a single command. Jarvis draws its inspiration from Virtual assistants like Cortana for Windows and Siri for iOS. Users can interact with the assistant either through voice commands or keyboard input.

Artificial Intelligence personal assistants have become plentiful over the last few years. Applications such as Siri, Bixby, Ok Google and Cortana make mobile device users' daily routines that much easier.

1.3 Objective

Currently, the project aims to provide the Users with a Virtual Assistant that would not only aid in their daily routine tasks like searching the web, extracting weather data, vocabulary help and many others but also help in automation of various activities. In the long run, we aim to develop a complete server assistant, by automating the entire server management process - deployment, backups, auto-scaling, logging, monitoring and make it smart enough to act as a replacement for a 6 general server administrator. As a personal assistant, Jarvis assists the end-user with day-to-day activities like general human conversation, searching queries in various search engines like Google, Bing or Yahoo, searching for videos, retrieving images, live weather conditions, word meanings, searching for medicine details, health recommendations based on symptoms and reminding the user about the scheduled events and tasks.

1.4 Scope

Presently, Jarvis is being developed as an automation tool and virtual assistant.

Among the Various roles played by Jarvis are:

1. Search Engine with voice interactions

2. Playing Music and Telling Jokes.

3. Reminder and To-Do application.

4. Calculation of complex mathematical expressions.

5. Weather Forecasting Application.

CHAPTER 2

LITERATURE SURVEY

CHAPTER 2

2.1 LITERATURE SURVEY

This research could be is a chunk of a bigger project concerning virtual voice assistant briefed by theories in human machine interaction. Moreover speech recognition has a brief history with numerous waves of innovations. Voice recognition for dictation, hunt and voice command has become vital feature on personal devices: like wearable devices and smartphone’s.

This system was developed as a humanoid application that confirms the necessity of language rework that sends messages and also use build-in application by processing the commands given by user to the system. Importantly smartphone gadget was way quicker followed by other wearable devices; so, many arrived to introduce in-voice virtual voice assistant with the importance of adopting and applying multiple smart technologies. This system has some basic features and most importantly mailing and secondly calendar, where user has the privilege to mail and able to create their required event by providing voice commands. For instance, if we use artificial intelligence we can are able to turn off the lights without the instruction given by the user.

Almost, Everyone has some knowledge about trending voice assistant like Cortana for windows, and Siri for apple users, this virtual voice assistants aren’t as brainy and intelligent as Ironman's Jarvis which appear in the superhero movie, but the intended actions are almost similar by virtual voice assistant. It’s like you need a ask question, and within a few fraction of seconds you will get an answer. It’s just give a command and get result. This field of virtual assistants having speech recognition has seen some major advancements or innovations. This is mainlybecause of its demand in devices like smartwatches or fitness bands, speakers, bluetooth earphones, mobile phones, laptop or desktop, television, etc. Almost all the digital devices which are coming nowadays are coming with voice assistants which help to control the device with speech recognition only.

A new set of techniques is being developed constantly to improve the performance of voice automated search. As the amount of data is increasing exponentially now known as Big Data the best way to improve the results of virtual assistants is to incorporate our assistants with machine learning and train our devices according to their uses. Other major techniques that are equally important are Artificial Intelligence, Internet Of Things, Big Data access and management, etc. With the use of voice assistants, we can automate the task easily, just give the input to the machine in the speech form and all the tasks will be done by it from converting your speech into text form to taking out keywords from that text and execute the query to give results to the user.

This has been one of the most helpful advancements in technology. Before AI we were the ones who were upgrading technology to do a task but now the machine is itself able to counter new tasks and solve it without need to involve the humans to evolve it. This has been helpful in day-to-day lifestyle. From mobile phones to personal desktops to mechanical industries these assistants are in very much demand for automating tasks and increasing efficiency.

2.2 Python

Python is an OOPs (Object Oriented Programming) based, high level, interpreted programming language. It is a robust, highly useful language focused on rapid application development (RAD). Python helps in easy writing and execution of codes. Python can implement the same logic with as much as 1/5th code as compared to other OOPs languages.

Python provides a huge list of benefits to all. The usage of Python is such that it cannot be limited to only one activity. Its growing popularity has allowed it to enter into some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), natural language processing, data science etc. Python has a lot of libraries for every need of this project. For JIA, libraries used are speech recognition to recognize voice, Pyttsx for text to speech, selenium for web automation etc.

If your Python code is not efficient enough, a general procedure to improve it is to find out what is taking most the time, and implement just that part more efficiently in some lower-level language. This will result in much less programming and more efficient code than writing everything in a low-level language.

CHAPTER 3

SYSTEM ANALYSIS

CHAPTER 3

SYSTEM ANALYSIS

3.1 Software Development Life Cycle

The software development life cycle (SDLC) is a process for planning, creating, testing and deploying an information system. The software development life cycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both. There are usually six stages in this cycle: Requirement analysis, Design, Development and Testing, Implementation, Deployment and Evaluation.

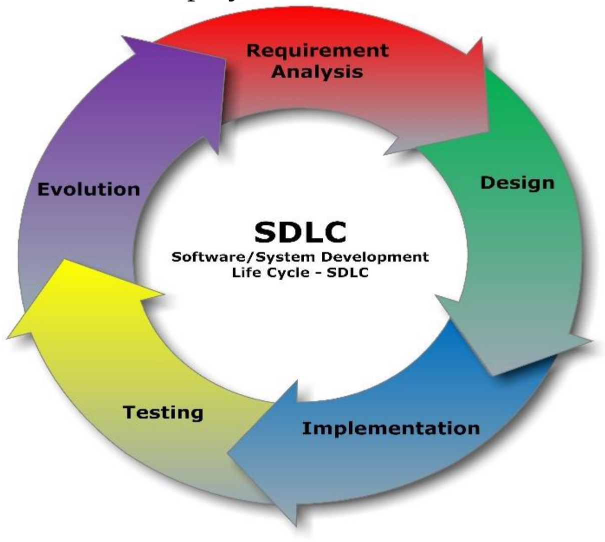


Fig:- 3.1 Software Development Life Cycle

Stage 1: Requirement gathering and analysis:
It is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with the inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product
Feasibility study in the economical, operational and technical areas.
After requirement gathering these requirements are analysed for their validity and the possibility of
Incorporating the requirements in the system to be development is also studied. Finally, a Requirement
Specification document is created which serves the purpose of guideline for the next phase of the model

Stage 2: Design
In this phase the system and software design is prepared from the requirement specifications which
Were studied in the first phase. System Design helps in specifying hardware and system requirements
And also helps in defining overall system architecture. The system design specifications serve as input
For the next phase of the model.

Stage 3: Implementation/Coding
On receiving system design documents, the work is divided in modules/units and actual coding is
Started. Since, in this phase the code is produced so it is the main focus for the developer. This is the
Longest phase of the software development life cycle.

Stage 4: Testing
After the code is developed it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. During this phase unit testing, integration testing, system testing, acceptance testing is done.

Stage 5: Deployment
After successful testing the product is delivered/ deployed to the customer.

Stage 6: Maintenance
Once when the customers starts using the developed system then the actual problems comes up and needs to be solved from time to time. This process where the care is taken for the developed product is known as maintenance.

3.2 Existing System

Normally, if a user wants to send an email, search for a topic etc., while performing some other task they have to pause and do so.This is usually time consuming.

For example, to make a note of something while browsing we have to open notepad and type the note. Sometimes, this may take longer. As we know Python is a suitable language for scriptwriters and developers.

Let’s write a script for Voice Assistant using Python. The query for the assistant can be manipulated as per the user’s need. Speech recognition is the process of converting audio into text. This is commonly used in voice assistants like Alexa, Siri, etc. Python provides an API called Speech Recognition to allow us to convert audio into text for further processing.

In this article, we will look at converting large or long audio files into text using the speech Recognition API in python. Our virtual assistant will able to do the followings things-

- Weather forecasting, Launch Games, Launch Windows Applications, Open Websites, tells you about almost everything you ask,
- Tells you date and time, greetings, news, etc.
- You can interact with your laptop’s microphone/console.

The response generated by the assistant will display on the console or as a speech via the speaker.

3.3 Proposed System

Most of the Personal voice-assistants work with Internet Connections. But our Proposed System has capability to work with and without Internet Connectivity. It is named as Personal Assistant with Voice Recognition Intelligence, which takes the user input in form of voice or text and process it and returns the output in various forms like action to be performed or the search result is dictated to the end user. In addition, this proposed system can change the way of interactions between end user and the mobile devices. The system is being designed in such a way that all the services provided by the mobile devices are accessible by the end user on the user's voice commands.

The main purpose of an intelligent virtual assistant is to answer questions that users may have. Purpose of voice assistant is to being capable of voice interaction, music playback, making to-do lists, setting alarms, streaming podcasts, playing audiobooks, and providing weather, traffic, sports, and other real-time information, such as news. Virtual assistants enable users to speak natural language voice commands in order to operate the device and its apps.

Virtual assistants can tremendously save you time. We spend hours in online research and then making the report in our terms of understanding. Our voice assistant can do that for you. Provide a topic for research and continue with your tasks while it does the research

CHAPTER 4

SYSTEM SPECIFICATION

CHAPTER 4

SYSTEM SPECIFICATION

4.1 Software Requirements

The basic software necessities to develop such a project will include-

1. Python 3.5+.
2. Windows 7 or 10
3. Working Microphone
4. Microsoft Visual Studio Code with Python package of 3.0 version or above with modules described below and they can be downloaded using pip command in the VS Code

4.2 Hardware Requirements

The project will probably use all of the following stated hardware requirements-

1. Minimum of 1 GB of RAM
2. Intel Pentium Dual-Core 4.0 GHz
3. Keyboard
4. Mouse
5. Monitor/Display

4.3 Other Requirements

1. pyttsx3
2. Speech recognition
3. Wikipedia
4. SMTPLib
5. pyautogui
6. psutil
7. wolframalpha
8. webbrowser
9. operator
10. date and time

4.4 System Architecture

The total design consists of these phases:

1) Collection of data which is in speech format.

2) Analyse the voice and convert it to text.

3) Storing the data and processing it.

4) Speech generation from the text output that is processed.

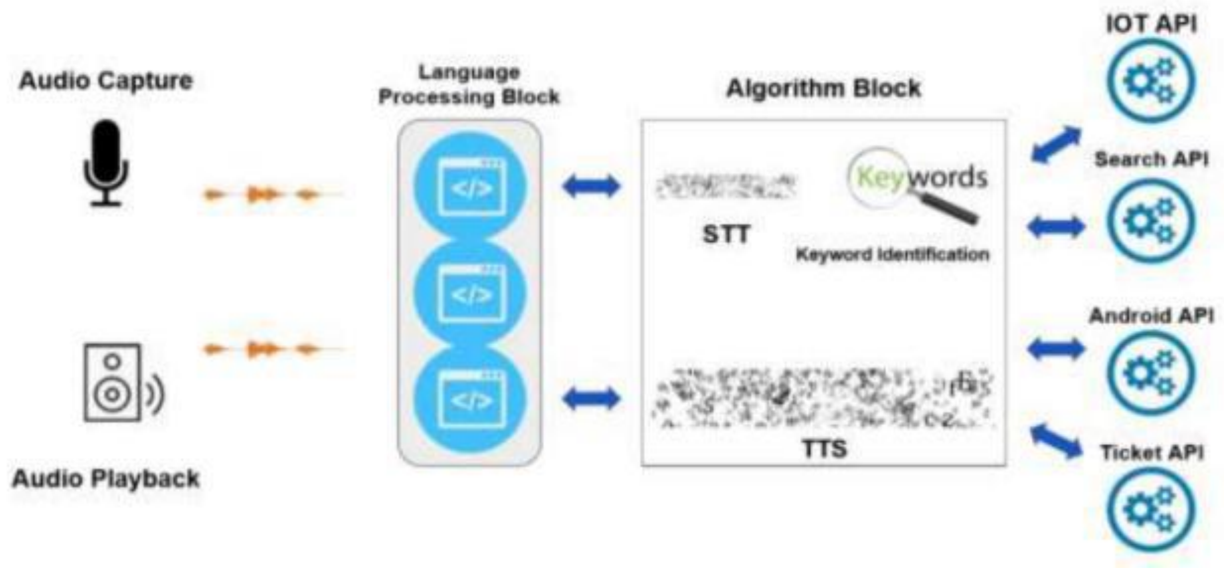


Fig 4.4:- System Architecture

Phases Summary

The data that is collected in the speech form is stored and used as input for next phase of the process. In next phase, the input which is given in the form of voice is processed continuously and is converted into text by using STT (Speech-to-text). In third phase, the text which is converted, is analysed by Python Script which processes it and identifies the action to be taken for the command. In the last phase, after the action to be taken is identified, output will be obtained from text to speech conversion using TTS (text-to-speech).

CHAPTER 5

DESIGN

CHAPTER 5

DESIGN

UML DIAGRAMS

5.1 Use Case Diagram

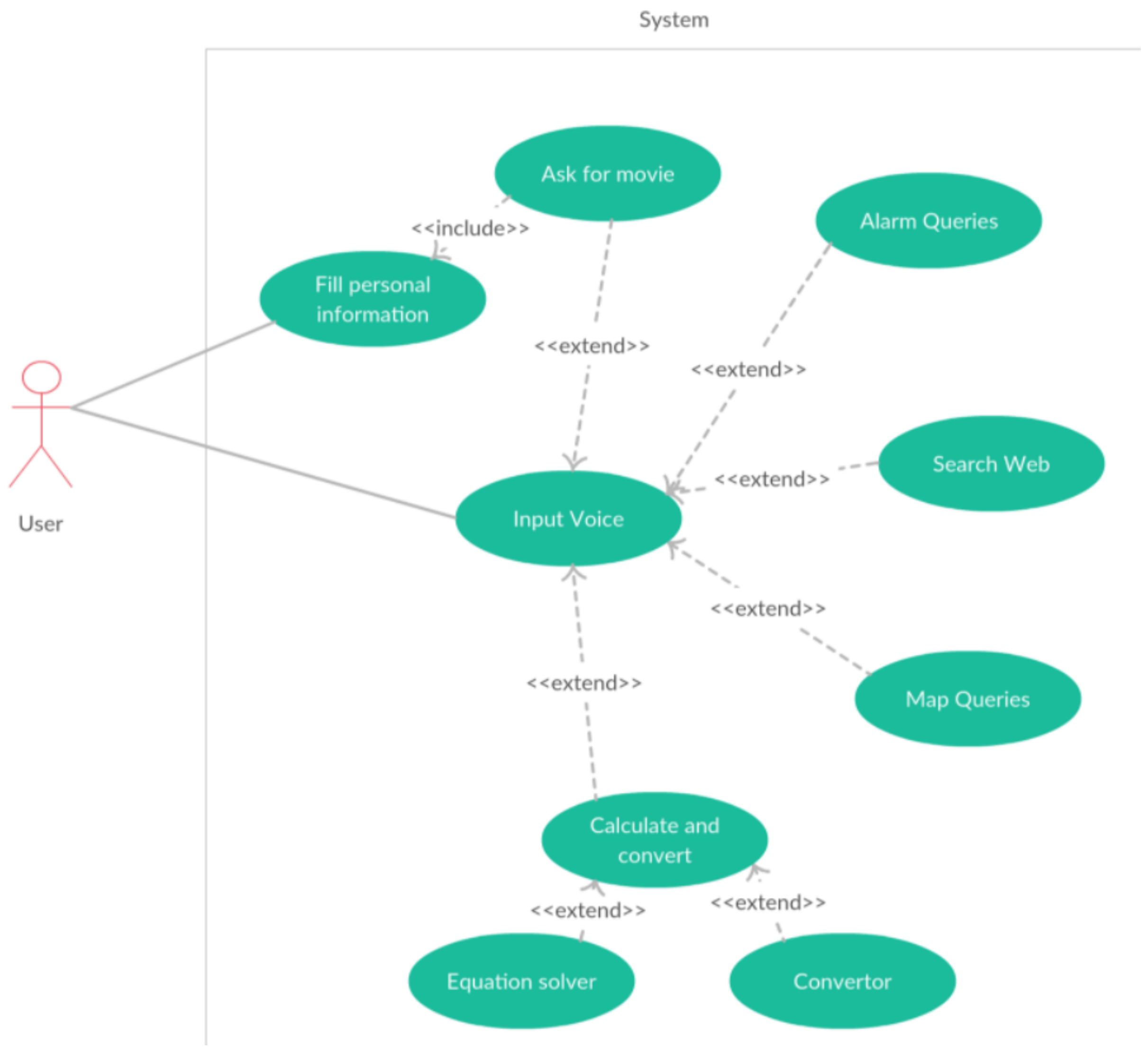


Fig 5.1:- Use-Case Diagram

5.2 Activity Diagram

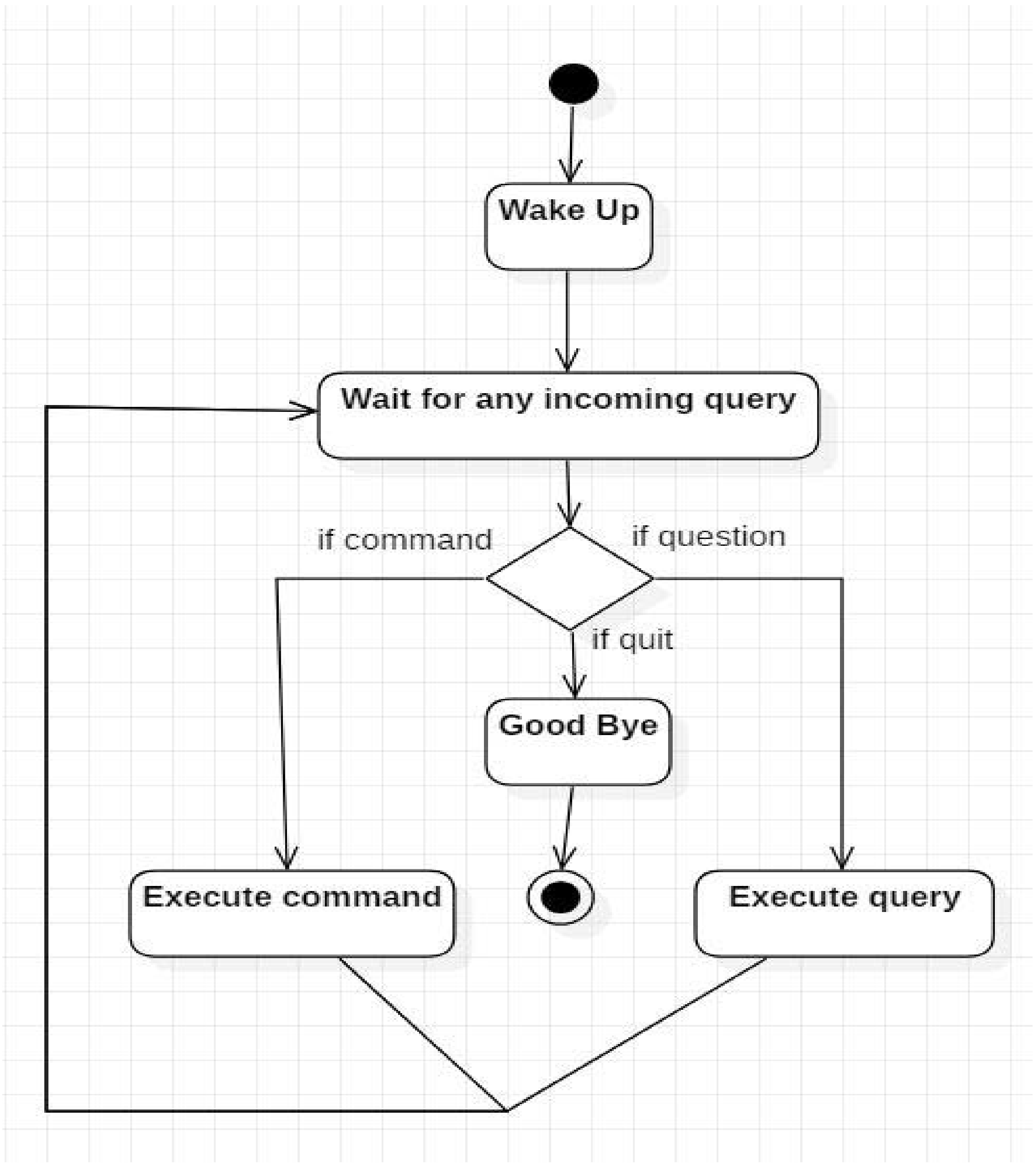


Fig 5.1 Activity Diagram

5.3 Class Diagram

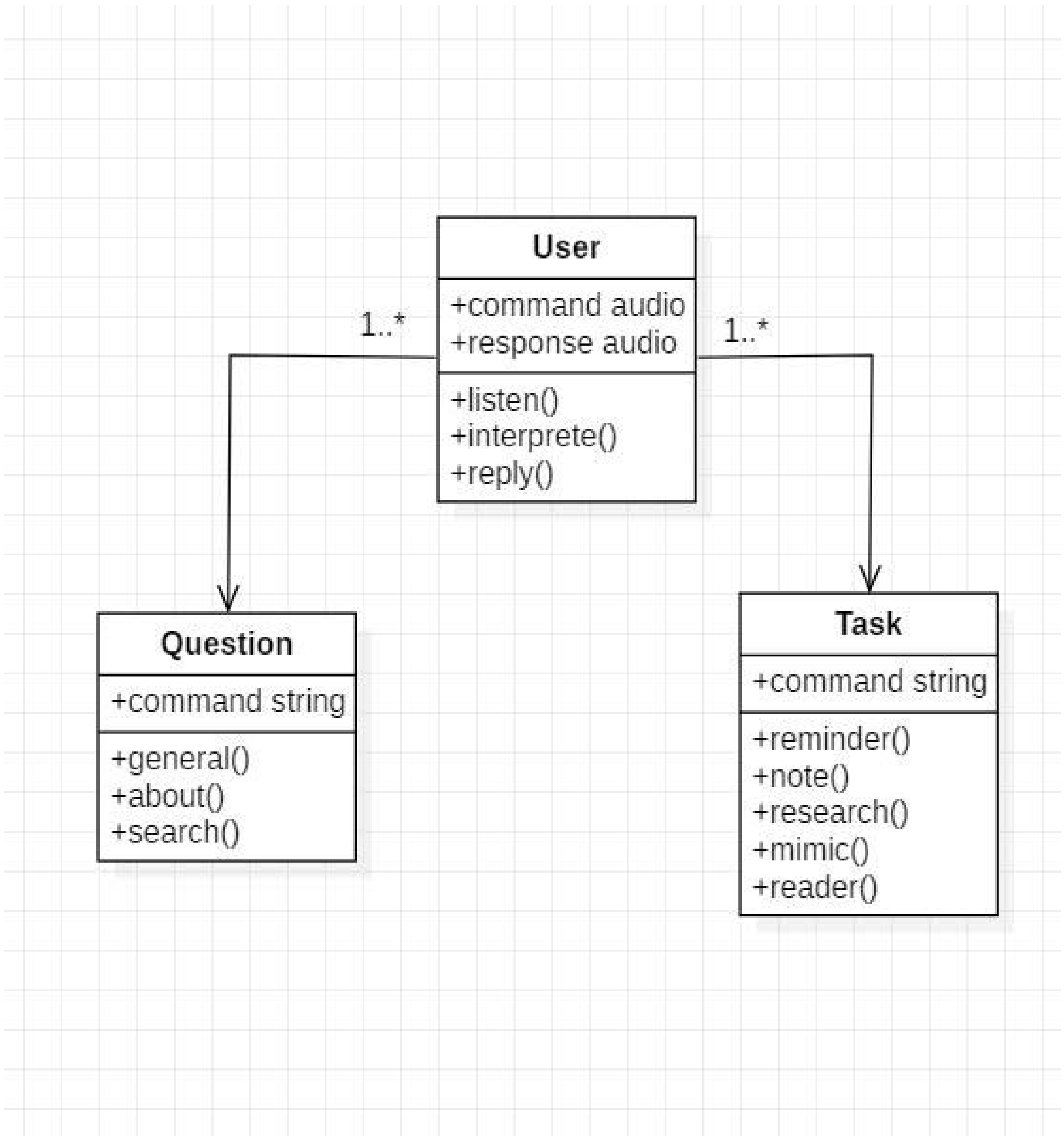


Fig 5.3:- Class Diagram

5.4 Sequence Diagram

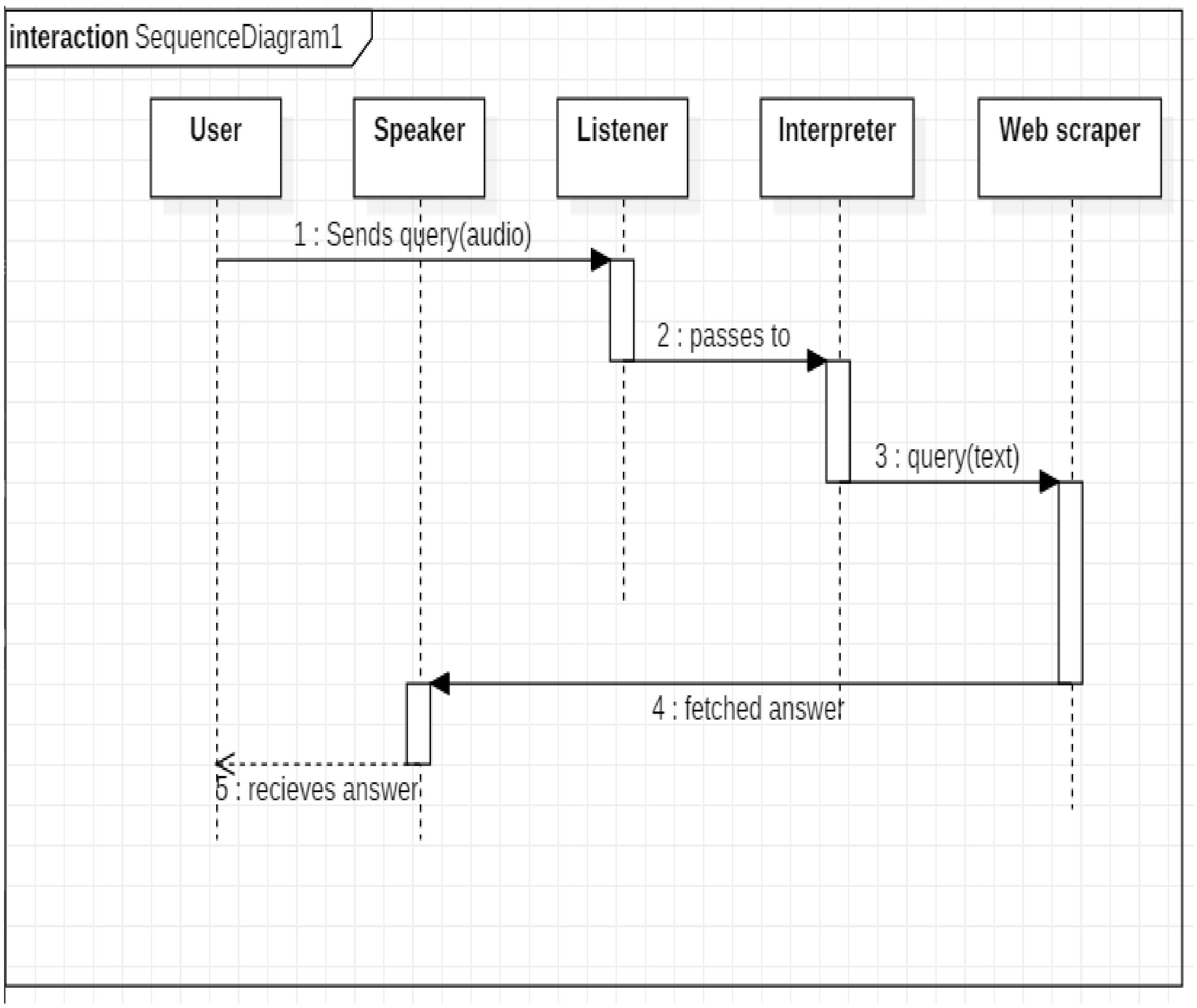


Fig 5.4:- Sequence Diagram

CHAPTER 6

MODULE DESCRIPTION

CHAPTER 6

MODULE DESCRIPTION

6.1 PYTTSX3

The pyttsx3 is an offline module that is used for text to speech conversion in Python and it is supported by both Python 2 & 3. The run and wait functionality is also in this module only. It determines how much time the system will wait for another input or in other words the time interval between inputs. This is a free module available in the python community which can be installed using the pip command just like other modules.

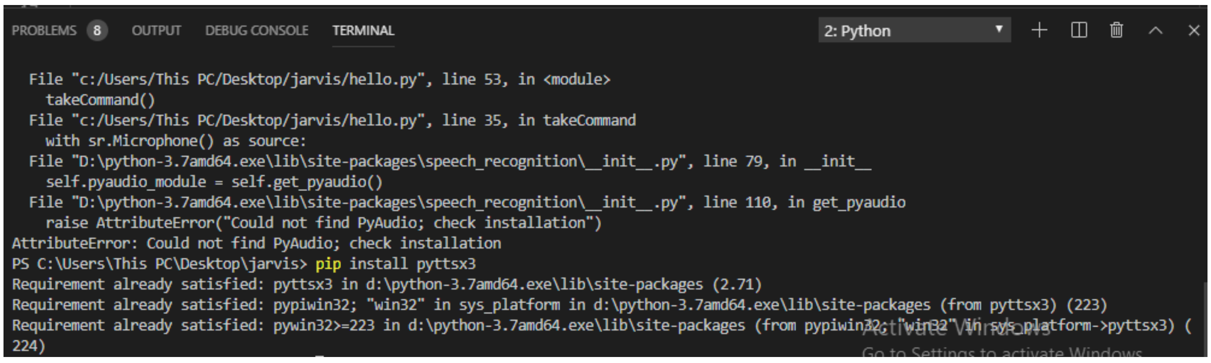
The pyttsx3 module supports two voices first is female and the second is male which is provided by "sapi5" for windows.

It supports three TTS engines:

- *sapi5* – SAPI5 on Windows
- *nsss* – NSSpeechSynthesizer on Mac OS X
- *espeak* – eSpeak on every other platform

Installataion

To install the pyttsx3 module, first of all, you have to open the terminal and write pip install pyttsx3



```
  File "c:/Users/This PC/Desktop/jarvis/hello.py", line 53, in <module>
    takeCommand()
  File "c:/Users/This PC/Desktop/jarvis/hello.py", line 35, in takeCommand
    with sr.Microphone() as source:
  File "D:\python-3.7amd64.exe\lib\site-packages\speech_recognition\__init__.py", line 79, in __init__
    self.pyaudio_module = self.get_pyaudio()
  File "D:\python-3.7amd64.exe\lib\site-packages\speech_recognition\__init__.py", line 110, in get_pyaudio
    raise AttributeError("Could not find PyAudio; check installation")
AttributeError: Could not find PyAudio; check installation
PS C:\Users\This PC\Desktop\jarvis> pip install pyttsx3
Requirement already satisfied: pyttsx3 in d:\python-3.7amd64.exe\lib\site-packages (2.71)
Requirement already satisfied: pypiwin32; "win32" in sys_platform in d:\python-3.7amd64.exe\lib\site-packages (from pyttsx3) (223)
Requirement already satisfied: pywin32>=223 in d:\python-3.7amd64.exe\lib\site-packages (from pypiwin32; "win32" in sys_platform->pyttsx3) (
224)
```

Fig 6.1 Installation of pyttsx3 module

6.2 DATETIME

The DateTime module is imported to support the functionality of the date and time. For example, the user wants to know the current date and time or the user wants to schedule a task at a certain time. In short this module supports classes to manipulatedate and time and perform operations according to it only. This is an essential module, especially in tasks where we want to keep a track of time. This module is very small in size and helps to control the size of our program. If the modules are too large or heavy then the system will lag and give slow responses.

The datetime classes are categorize into 6 main classes –

- date – An idealized naive date, assuming the current Gregorian calendar always was, and always will be, in effect. Its attributes are year, month and day.
- time – An idealized time, independent of any particular day, assuming that every day has exactly 24*60*60 seconds. Its attributes are hour, minute, second, microsecond, and tzinfo.
- datetime – Its a combination of date and time along with the attributes year, month, day, hour, minute, second, microsecond, and tzinfo.
- timedelta – A duration expressing the difference between two date, time, or datetime instances to microsecond resolution.
- tzinfo – It provides time zone information objects.
- timezone – A class that implements the tzinfo abstract base class as a fixed offset from the UTC (New in version 3.2).

When an object of this class is instantiated, it represents a date in the format YYYY-MM-DD. Constructor of this class needs three mandatory arguments year, month and date.
Constructor syntax:
class datetime.date(year, month, day)
The arguments must be in the following range –

- MINYEAR <= year <= MAXYEAR
- 1 <= month <= 12
- 1 <= day <= number of days in the given month and year

Note – If the argument is not an integer it will raise a TypeError and if it is outside the range a ValueError will be raised.

6.4 WEBBROWSER

This module allows the system to display web-based information to users. For example, the user wants to open any website and he gives input as “Open Google”. The input is processed using the web browser module and the user gets a browser with google opened in it. The browser which will be used is the default set web browser.

Example:-

```
import webbrowser
webbrowser.open('http://www.python.org')
```

Output:-

```
True
```

6.4 WIKIPEDIA

Wikipedia is a library in python which it possible for the virtual assistant to process the queries regarding Wikipedia and display the results to users. This is an online library and needs an internet connection to fetch the results. The no. of lines that the user wants to get as a result can be set manually.

Installation

In order to extract data from Wikipedia, we must first install the Python Wikipedia library, which wraps the official Wikipedia API. This can be done by entering the command below in your command prompt or terminal:

```
pip install wikipedia
```

6.5 OS MODULE

OS Module provides an operating system dependent functionalities. If we want to perform operations on files like reading, writing, or manipulate paths, all these types of functionalities are available in an OS module. All the operations available raise an error “OSError” in case of any error like invalid names, paths, or arguments which may be incorrect or correct but just no accepted by the operating system.

6.6 SMTPLIB

Python has this module for in the standard library for working with emails & email servers. The SMTPLIB defines an object known as “SMTP client session object” which is used to send mails by the user. There are 3 steps involved - initialize, sendmail(), quit. When the optional parameters which are host and port, are provided connect method is called with these arguments during the first step which is initialization. Here is a simple syntax to create one SMTP object, which can later be used to send an e-mail –

```
import smtplib

smtpObj = smtplib.SMTP( [host [, port [, local_hostname]]] )
```

Here is the detail of the parameters −

- **host** − This is the host running your SMTP server. You can specify IP address of the host or a domain name like tutorialspoint.com. This is optional argument.
- **port** − If you are providing *host* argument, then you need to specify a port, where SMTP server is listening. Usually this port would be 25.
- **local_hostname** − If your SMTP server is running on your local machine, then you can specify just *localhost* as of this option.

An SMTP object has an instance method called **sendmail**, which is typically used to do the work of mailing a message. It takes three parameters −

- The *sender* − A string with the address of the sender.
- The *receivers* − A list of strings, one for each recipient.
- The *message* − A message as a string formatted as specified in the various RFCs.

CHAPTER 7

CODING AND OUTPUT

7.2 UI CODE

```
from tkinter import Label,Entry,Text,Button,messagebox,filedialog,Tk,Menu,INSERT,DISABLED,FALSE, END, simpledialog

from PIL import Image, ImageTk

from threading import Thread, Timer, main_thread

import socket

import time

import sys

import pyttsx3 #pip install pyttsx3 , pip install -U pyttsx3==2.71 (For Speak)

import datetime

import speech_recognition as sr #pip install SpeechRecognition

import wikipedia #pip install wikipedia

import smtplib

import webbrowser as wb

import os

import pyautogui #pip install pyautogui (For Screenshot)

import psutil #pip install psutil

import pyjokes #pip install pyjokes

import random

import string

import operator

import json

import wolframalpha
import time

from urllib.request import urlopen

import requests

import winshell


# few paths and APIs

videoFilesPath = r"C:\Users\SULTAN SIDDIQ M\Downloads\Video"

AudioFilesPath = r"C:\Users\SULTAN SIDDIQ M\Downloads\Music"

screeshotSavePath = r"C:\Users\{}\Downloads\{}.png".format(os.getenv('username'
), ''.join(random.choices(string.ascii_letters + string.digits, k=20)))

wordAppLocation = r"C:\ProgramData\Microsoft\Windows\Start Menu\Programs\
Word 2016.lnk"


toiNewsapi = '3f2fc248c9af419e847a4cf2f8529e25' # https://newsapi.org/

weatherApi = 'cc523fe571c72f039b5d67907f1a2518'  # https://openweathermap.org/
api

wolframAlphaApi = 'QALVTK-
JW28J7RQ9T' # https://products.wolframalpha.com/simple-api/documentation/


gmailLoginData = ('jarvis201098@gmail.com', 'jarvis1998')


# voice code for output starts

engine = pyttsx3.init("sapi5")

voices = engine.getProperty('voices')

engine.setProperty('voice', voices[0].id)
engine.setProperty('rate', 170)

engine.setProperty('volume', 1)


def speak(audio):  #speak method defined

engine.say(audio)

engine.runAndWait()


# Code to check internet

def Isconnect():      #method to check internet

print("Checking Internet.....")

try:

socket.create_connection(("www.google.com", 80))

neticon.config(image=connlogo)

nettext.config(text="Connected :)", fg="green")

return True

except OSError:

pass

neticon.config(image=notconn)

nettext.config(text="No Internet ;( ", fg="red")

return False


defcolor = "blue"

# wishe me code here

def wisheme():       #method to wisheme at the beginning
```

```
update("Welcome back Batch 13!", "green")

speak("Welcome back Batch 13!")

time_()

date()

hour = int(datetime.datetime.now().hour)

if hour >=6 and hour<12:

update("Good Morning Sir","green")

speak("Good Morning Sir")

elif hour >=12 and hour<18:

update("Good Afternoon Sir!","green")

speak("Good Afternoon Sir!")

elif hour >=18 and hour <24:

update("Good Evening Sir!","green")

speak("Good Evening Sir!")

else:

update("Good Night Sir!","green")

speak("Good Night Sir!")


update("Jarvis at your service. Please tell me how can I help you?","yellow")

speak("Jarvis at your service. Please tell me how can I help you?")


def update(message,color):          #method which update the message box

msg.configure(state='normal')
msg.delete('1.0', END)

msg.insert('end', message)

msg.configure(state='disabled')


# msg.config(text=message,fg=color)


def TakeCommand():        #method which recives audio input

r = sr.Recognizer()

with sr.Microphone() as source:

print("Listening....")

update("Listening....","green")

speak("Listening.....")

r.pause_threshold = 0.8


r.energy_threshold = 500

audio = r.listen(source,phrase_time_limit=5)

try:

update("Recognising....","orange")

Isinternet = Isconnect()

print("Connected" if Isinternet else "Not Connected")

if (Isinternet):

query = r.recognize_google(audio, language="en-IN")

else:
```

```
update("You are not connected to the internet.","yellow")
speak("You are not connected to the internet.")
return None

print("You Said", query)
update(f"You Said: {query} ","green")
time.sleep(1)

except Exception as e:
print(e)
print("say that again please...")
update("Say That Again Please !!","red")
speak("say that again please...")
return None

return query

def time_():
strTime = datetime.datetime.now().strftime("%H:%M:%S")
Time12=datetime.datetime.now().strftime("%I:%M:%S")

update(f"the current time is {strTime}  or   {Time12}","red")
speak(f"the current time is {strTime}  or   {Time12}")
```

```
def date():

year = (datetime.datetime.now().year)

month = (datetime.datetime.now().month)

date = (datetime.datetime.now().day)

speak("the current date is")

update(f"Date is {date}-{month}-{year}","red")

speak(date)

speak(month)

speak(year)


def sendEmail(to, content):

sender = gmailLoginData

server = smtplib.SMTP('smtp.gmail.com', 587)

server.ehlo()

server.starttls()

# Enable low security in gmail

server.login(sender[0], sender[1])

server.sendmail(sender[0], to, content)

server.close()

def screenshot():

img = pyautogui.screenshot()

img.save(screeshotSavePath)

def cpu():

usage = str(psutil.cpu_percent())
```

```
battery = psutil.sensors_battery()

update('CPU is at '+ str(usage) + "\nBattery is at " + str(battery.percent), 'yellow')

speak('CPU is at'+ usage)

speak("Battery is at")

speak(battery.percent)

def jokes():

update(pyjokes.get_joke(), 'yellow')

speak(pyjokes.get_joke())


def Introduction():

update("I am JARVIS 1.0 , Personal AI assistant ,\nI am created by Batch 13 , \nI ca
n help you in various regards , \nI can search for you on the Internet , \nI can also gr
ab definitions for you from wikipedia , \nIn layman terms , I can try to make your lif
e a bed of roses , \nWhere you just have to command me , and I will do it for you ",'
yellow')


speak("I am JARVIS 1.0 , Personal AI assistant , "

"I am created by Batch 13 , "

"I can help you in various regards , "

"I can search for you on the Internet , "

"I can also grab definitions for you from wikipedia , "

"In layman terms , I can try to make your life a bed of roses , "

"Where you just have to command me , and I will do it for you , ")

def Creator():
```

```
update("Batch 13 is an extra-
ordinary person ,\nHe has a passion for Robotics, Artificial Intelligence and Machin
e Learning ,\nHe is very co-
operative ,\nIf you are facing any problem regarding the 'Jarvis', He will be glad to h
elp you ",'yellow')

speak("Batch 13 is an extra-ordinary person ,"

"He has a passion for Robotics, Artificial Intelligence and Machine Learning ,"

"He is very co-operative ,"

"If you are facing any problem regarding the 'Jarvis', He will be glad to help you ")


flag = 1

message = "OFF"

terminate = 0


session=1

def runnow():       #method which starts execution on started method

global session

if session==1:

wisheme()

session=0


count = 0


while 1:

if main_thread().is_alive():
```

```
pass
else:
break
if terminate == 1:
break
query = TakeCommand()

if query != None:
query = query.lower()
speak(f"You Said :{query} ")
else:
continue

# All the commands said by user will be
# stored here in 'query' and will be
# converted to lower case for easily
# recognition of command

if 'time' in query:
time_()
elif 'date' in query:
date()
elif 'how are you' in query:
update("I am fine, Sir Thanks for asking \nHow are you?", "yellow")
```

```
speak("I am fine, Sir Thanks for asking")

speak("How are you Sir?")

query = TakeCommand()

if query == None:

continue

elif 'fine' in query or "good" in query:

speak("It's good to know that your fine")

elif query != None:

speak("I hope you get well soon.")

elif 'wikipedia' in query:

update("Searching...",'yellow')

speak("Searching...")

query = query.replace("wikipedia","")

result = wikipedia.summary(query, sentences=2)

speak("According to Wikipedia")

print(result)

update(result,'yellow')

speak(result)

elif 'open youtube' in query:

update("What should I search?",'yellow')

speak("What should I search?")

Search_term = TakeCommand()

update("Here we go to Youtube\n",'yellow')
```

```
speak("Here we go to Youtube\n")

wb.open("https://www.youtube.com/results?search_query="+Search_term)

time.sleep(5)


elif 'search google' in query:

update("What should I search?",'yellow')

speak("What should I search?")

Search_term = TakeCommand().lower()

wb.open('https://www.google.com/search?q='+Search_term)


#elif 'search' in query:

#query = query.replace("query","")

#wb.open(query)


elif "who am i" in query:

update("If you can talk, then definitely you are a human",'yellow')

speak("If you can talk, then definitely you are a human")

elif "why you came to this world" in query:

update("Thanks to Batch 13. further it is a secret",'yellow')

speak("Thanks to Batch 13. further it is a secret")

elif 'word' in query:

update("opening MS Word",'yellow')

speak("opening MS Word")

word = wordAppLocation
```

```
os.startfile(word)


elif 'what is love' and 'tell me about love' in query:

update("It is 7th sense that destroy all other senses \nAnd I think it is just a mere illu
sion \nIt is waste of time",'yellow')

speak("It is 7th sense that destroy all other senses , "

"And I think it is just a mere illusion , "

"It is waste of time")


elif 'empty recycle bin' in query:

winshell.recycle_bin().empty(confirm = False, show_progress = False, sound = True
)

update("Recycle Bin Recycled",'yellow')

speak("Recycle Bin Recycled")


elif 'send email' in query:

try:

update("What should I say?",'yellow')

speak("What should I say?")

content = TakeCommand()

update("Who is the Reciever?",'yellow')

speak("Who is the Reciever?")


# new win

newWin = Tk()
```

```
newWin.withdraw()

retVal = simpledialog.askstring("Enter Value","Enter recieptant's email",parent=new
Win)

newWin.destroy()


print(retVal)

to = (retVal)

sendEmail(to,content)

update(content,'yellow')

speak(content)

update("Email has been sent.",'yellow')

speak("Email has been sent.")

except Exception as e:

print(e)

update("Unable to send the email.",'yellow')

speak("Unable to send the email.")


elif 'search in chrome' in query:

update("What should I search ?",'yellow')

speak("What should I search ?")

chromepath = r'C:/Program Files (x86)/Google/Chrome/Application/chrome.exe %s'

search = TakeCommand()

wb.get(chromepath).open_new_tab(search+'.com')


elif 'log out' in query:
```

```
os.system("shutdown -l")
elif 'restart' in query:
os.system("shutdown /r /t 1")
elif 'shutdown' in query:
os.system("shutdown /s /t 1")

elif 'play songs' in query:
video = videoFilesPath
audio = AudioFilesPath
update("What songs should i play? Audio or Video",'yellow')
speak("What songs should i play? Audio or Video")
ans = (TakeCommand().lower())
while(ans != 'audio' and ans != 'video'):
update("I could not understand you. Please Try again.",'yellow')
speak("I could not understand you. Please Try again.")
ans = (TakeCommand().lower())

if 'audio' in ans:
songs_dir = audio
songs = os.listdir(songs_dir)
print(songs)
elif 'video' in ans:
songs_dir = video
songs = os.listdir(songs_dir)
print(songs)
```

```
update("select a random number",'yellow')

speak("select a random number")

rand = TakeCommand()

if rand == None:

continue

rand = rand.lower()

while('number' not in rand and rand != 'random'):                    #used while loop to k
eep the jarvis on the speak command untill req. command is given.

speak("I could not understand you. Please Try again.")          #first used 'rand' before
while then again after, so that rand is already defind, and Input is taken and then che
cked if it is according to reuirement or not. And if it is not which means while loop i
s true, then commands under 'while loop' will execute untill desired approach.As it w
ill again ask the user for input in the same block.

rand = (TakeCommand().lower())


if 'number' in rand:

rand = int(rand.replace("number ",""))

os.startfile(os.path.join(songs_dir,songs[rand]))

continue                                             #'continue' is used, so that after executing t
he commands in 'if' or 'elif' block, it will move to the next part of execution (or code)
. but in this case as this is the last execution of related function then it will move to t
he next function (i.e. in this code, it will be TakeCommand() )

elif 'random' in rand:

rand = random.randint(1,219)

os.startfile(os.path.join(songs_dir,songs[rand]))

continue


elif 'remember that' in query:
```

```
update("What should I remember ?",'yellow')

speak("What should I remember ?")

memory = TakeCommand()

update("You asked me to remember that"+memory, 'yellow')

speak("You asked me to remember that"+memory)

remember = open('memory.txt','w')

remember.write(memory)

remember.close()


elif 'do you remember anything' in query:

remember =open('memory.txt', 'r')

data = remember.read()

update("You asked me to remeber that {}".format(data),'yellow')

speak("You asked me to remeber that {}".format(data))


elif "write a note" in query:

update("What should i write, sir",'yellow')

speak("What should i write, sir")

note = TakeCommand()

file = open('note.txt', 'w')

update("Sir, Should i include date and time",'yellow')

speak("Sir, Should i include date and time")

dt = TakeCommand()

if 'yes' in dt or 'sure' in dt:

strTime = datetime.datetime.now().strftime("%H:%M:%S")
```

```
file.write(strTime)

file.write(" :- ")

file.write(note)

update('done', 'yellow')

speak('done')

else:

file.write(note)


elif "show note" in query:

update("Showing Notes",'yellow')

speak("Showing Notes")

file = open("note.txt", "r")

data = file.read()

update(data,'yellow')

speak(data)


elif "weather" in query:


# Google Open weather website

# to get API of Open weather

api_key = weatherApi

city_name = query.replace("weather","")

complete_url = 'http://api.openweathermap.org/data/2.5/weather?q={}&appid={}'.fo
rmat(city_name,api_key)
```

```
response = requests.get(complete_url)

x = response.json()


if x["cod"] != "404":

y = x["main"]

current_temperature = y["temp"]

current_pressure = y["pressure"]

current_humidiy = y["humidity"]

z = x["weather"]

weather_description = z[0]["description"]

update(" Temperature (in kelvin unit) = " +str(current_temperature)+"\n atmospheric
 pressure (in hPa unit) ="+str(current_pressure) +"\n humidity (in percentage) = " +st
r(current_humidiy) +"\n description = " +str(weather_description), 'yellow')

print(" Temperature (in kelvin unit) = " +str(current_temperature)+"\n atmospheric p
ressure (in hPa unit) ="+str(current_pressure) +"\n humidity (in percentage) = " +str(
current_humidiy) +"\n description = " +str(weather_description))

speak(" Temperature of " + str(city_name) + " is " + str(current_temperature) + 'in k
elvin unit')

time.sleep(5)


else:

update(" City Not Found ", 'yellow')

speak(" City Not Found ")


elif 'news' in query:

try:

# https://newsapi.org/ : get api from this link
```

```
jsonObj = urlopen('https://newsapi.org/v2/everything?q=tesla&from=2021-05-
16&sortBy=publishedAt&apiKey={}'.format(toiNewsapi))

data = json.load(jsonObj)

i = 1

speak('here are some top news from the times of india')

for item in data['articles']:

if i== 5:

break

update('''=============== TOP HEADLINES ============'''+ '\n' + str(i) + '.
 ' + item['title'] + '\n' + item['description'] + '\n','yellow')

speak(str(i) + '. ' + item['title'] + '\n')

i += 1


except Exception as e:

print(str(e))


elif 'take a screenshot' in query:

screenshot()

update("Done!",'yellow') #Download folder

speak("Done!") #Download folder

elif 'cpu' in query:

cpu()

elif 'joke' in query:

jokes()

elif 'tell me about yourself' and 'who are you' in query:

Introduction()
```

```
elif 'tell me about mac' and 'creator' in query:

Creator()


#show location on map

elif "where is" in query:

query = query.replace("where is", "")

location = query

update("User asked to Locate : " + str(location),'yellow')

speak("User asked to Locate")

speak(location)

wb.open("https://www.google.com/maps/place/" + location + "")


# most asked question from google Assistant

elif "will you be my gf" in query or "will you be my bf" in query:

update("I'm not sure about, may be you should give me some time",'yellow')

speak("I'm not sure about, may be you should give me some time")


elif "i love you" in query:

update("It's hard to understand, I am still trying to figure this out.",'yellow')

speak("It's hard to understand, I am still trying to figure this out.")



#calculation

elif "calculate" in query:

app_id = wolframAlphaApi
```

```
client = wolframalpha.Client(app_id)
indx = query.lower().split().index('calculate')
query = query.split()[indx + 1:]
res = client.query(' '.join(query))
answer = next(res.results).text
update("The answer is " + str(answer), 'yellow')
speak("The answer is " + str(answer))

#General Questions
elif "what is" in query or "who is" in query:

# Use the same API key
# that we have generated earlier
client = wolframalpha.Client(wolframAlphaApi)
res = client.query(query)

try:
update(str(next(res.results).text),'yellow')
speak (next(res.results).text)
except StopIteration:
update("No results",'yellow')
speak("No results")

#sleep-time
elif "don't listen" in query or "stop listening" in query:
```

```
update("for how much seconds you want me to stop listening commands",'yellow')

speak("for how much seconds you want me to stop listening commands")

a = int(TakeCommand())

time.sleep(a)

print(a)


#quit

elif 'offline' in query:

update("going Offline",'yellow')

speak("going Offline")

quit()


def started():     #method which start on click of microphone icon

global terminate,flag

if flag==1:

l1.config(image=stop)


try:

terminate=0

t1 = Thread(target=runnow)

t1.start()

update("I am Jarvis!! .How May I help You","white")

flag=0

except RuntimeError:

update(RuntimeError,"red")
```

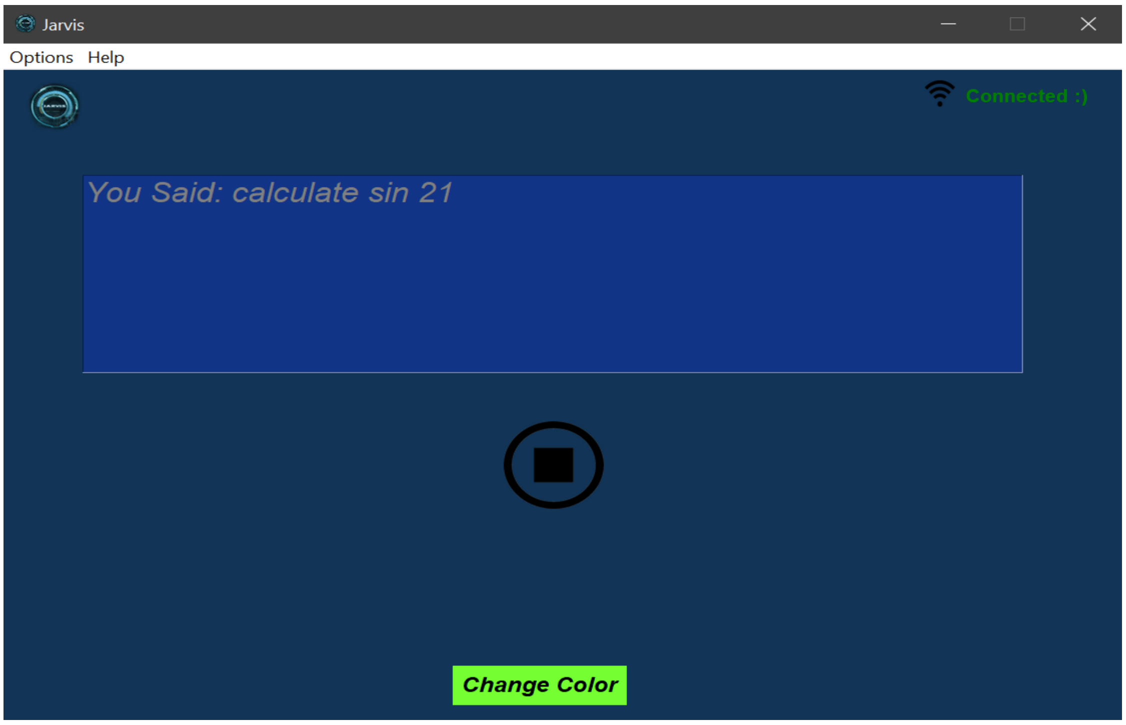
```
else:

update("Stoped","red")

l1.config(image=play)

flag = 1

terminate = 1


def colorchange():          #method which change color of the backgroud of the windo
w randomly


color1="#123456"

color2="#75ff33"

color3="#33ffbd"

color4="#33ff57"

color5="#eef9bf"


colors=[color1,color2,color3,color4,color5,"#a7e9af","#6a8caf","#fd5e53","#f9fcfb"
,"#b0eacd","#21bf73","#be8abf","#ea9abb","#fea5ad","#f8c3af"]

color=random.choice(colors)

msgcolor=random.choices(colors)

root.config(background=color)

l1.config(background=color)

neticon.config(background=color)

nettext.config(background=color)

logolab.config(background=color)

msg.config(background=msgcolor)

ccb.config(background=random.choice(colors),fg=msgcolor)
```

```
def helpwindow():       #method which open the help window

helproot =Tk()

# helproot.geometry("500x400")

textarea=Text(helproot,height="30",width="80")

textarea.pack(padx=20,pady=20)

with open('helpfile.txt','r') as target:

textarea.insert(INSERT,target.read())


textarea.config(state=DISABLED)

helproot.mainloop()


def Exit():             #method to exit

sys.exit(1)


if __name__ == "__main__":

root = Tk()

root.iconbitmap("images/icon.ico")

netlogo = Image.open("images/defaulticon.png")

netlogo = netlogo.resize((30, 30), Image.ANTIALIAS)

netlogo = ImageTk.PhotoImage(netlogo)


connlogo = Image.open("images/connected.png")

connlogo = connlogo.resize((30, 30), Image.ANTIALIAS)

connlogo = ImageTk.PhotoImage(connlogo)


notconn = Image.open("images/notconnected.png")
notconn = notconn.resize((30, 30), Image.ANTIALIAS)
notconn = ImageTk.PhotoImage(notconn)

logo = Image.open("images/icon.ico")
logo = logo.resize((60, 60), Image.ANTIALIAS)
logo = ImageTk.PhotoImage(logo)

root.geometry("1120x775")
root.resizable(FALSE, FALSE)
root.title("Jarvis")

play = Image.open("images/microphone.png")
play = play.resize((100, 120), Image.ANTIALIAS)
play = ImageTk.PhotoImage(play)
stop = Image.open("images/stop.png")
stop = stop.resize((100, 100), Image.ANTIALIAS)
stop = ImageTk.PhotoImage(stop)
# menu code started......
menu=Menu(root)
root.config(menu=menu)
submenu=Menu(menu,tearoff=False)
menu.add_cascade(label="Options",menu=submenu)
submenu.add_command(label="About us")
submenu.add_separator()
submenu.add_command(label="Exit",command=Exit)
```

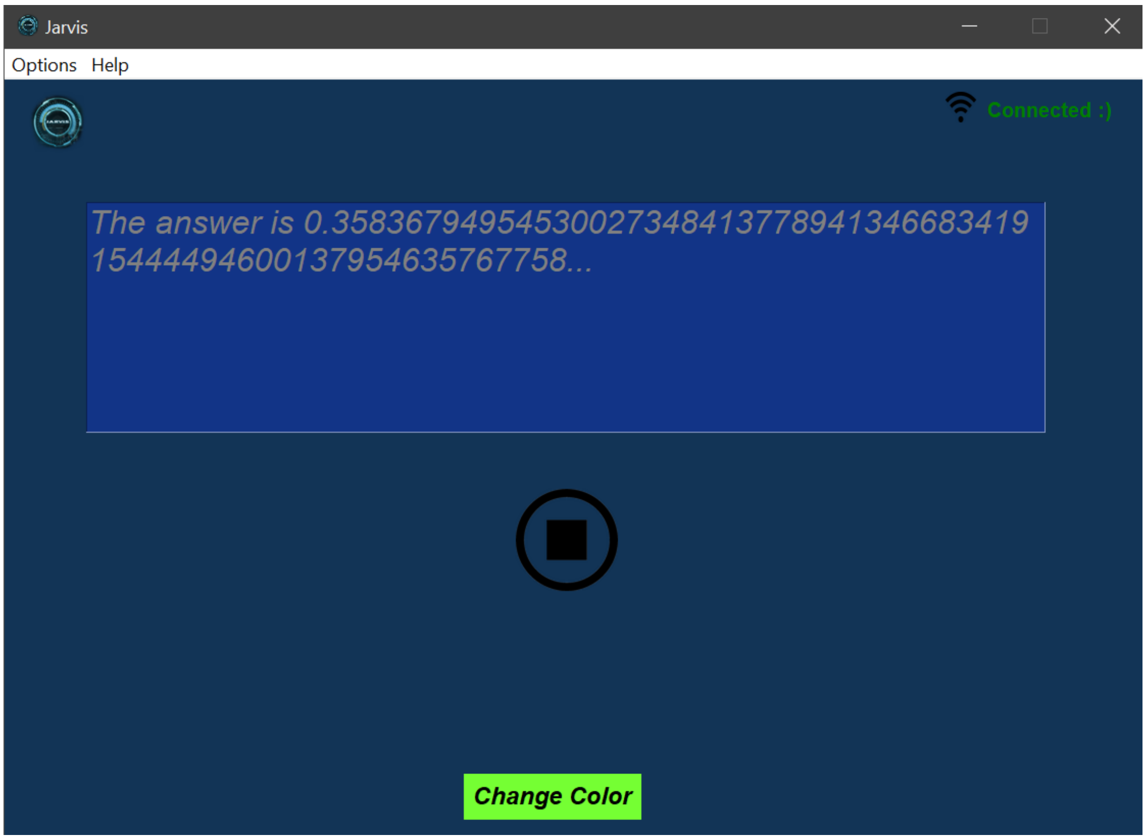
```
helpmenu=Menu(menu,tearoff=False,bg="white")

menu.add_cascade(label="Help",menu=helpmenu)

helpmenu.add_command(label="help",command=helpwindow)


# menu code ends.......

root.config(background="#123456")

l1 = Button(root, image=play, bg="#123456", command=started, borderwidth=0)

l1.place(x=500, y=400)

msg = text = Text(root, state='disabled', width=52, height=6, font="bell 16 italic", bg
="#123487", fg="gray")

# msg = Label(root, text=message, height="2",width="50", font="bell 16 italic", bg=
"#123487", fg="Blue")

msg.place(x=80, y=120)

neticon = Label(root, image=netlogo, bg="#123456")

neticon.place(x=920, y=10)

nettext = Label(root, text="Checking...", bg="#123456", fg="Yellow", font="bell 10
 bold")

nettext.place(x=960, y=15)

logolab=Label(root,image=logo,bg="#123456")

logolab.place(x=20,y=10)

Timer(1.0, Isconnect).start()

ccb=Button(root,text="Change Color",font="comic 12 bold italic",bg="#75ff33",bor
der=0.3,command=colorchange)

ccb.place(x=450,y=680)

root.mainloop()
```

7.2 Outputs

Calculation

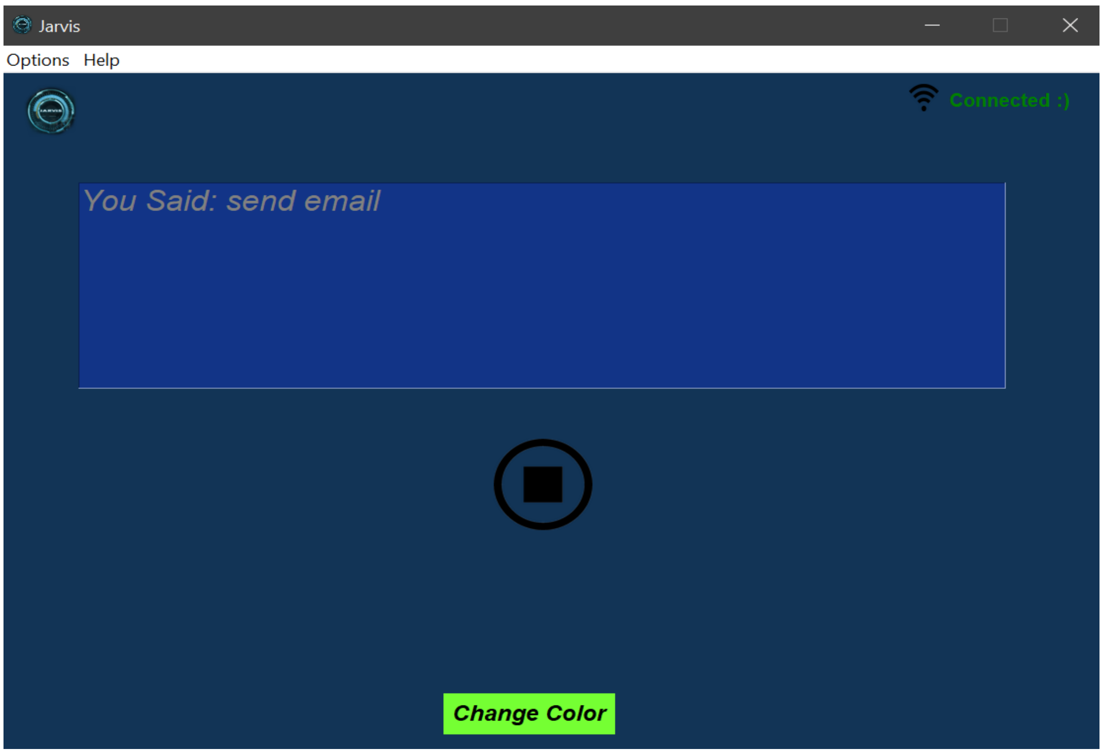


You Said: calculate sin 21

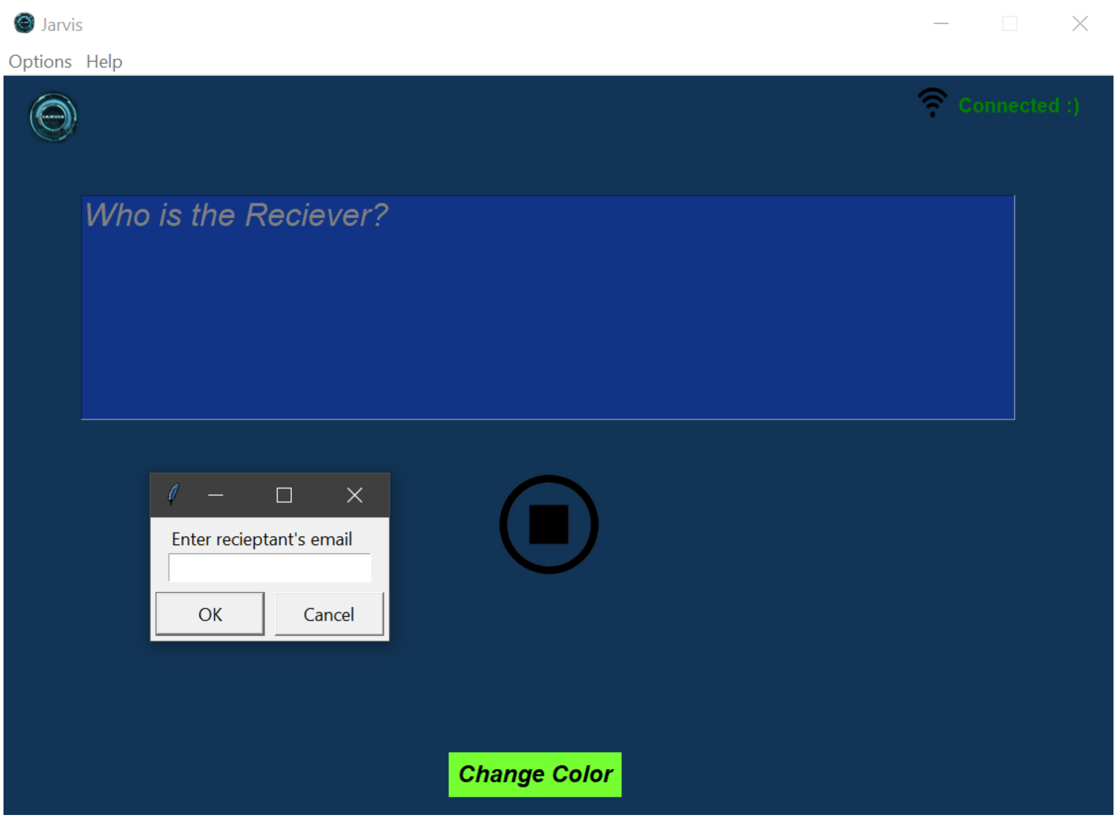


The answer is 0.358367949545300273484137789413466834191544449460013795463576775 8...

Email

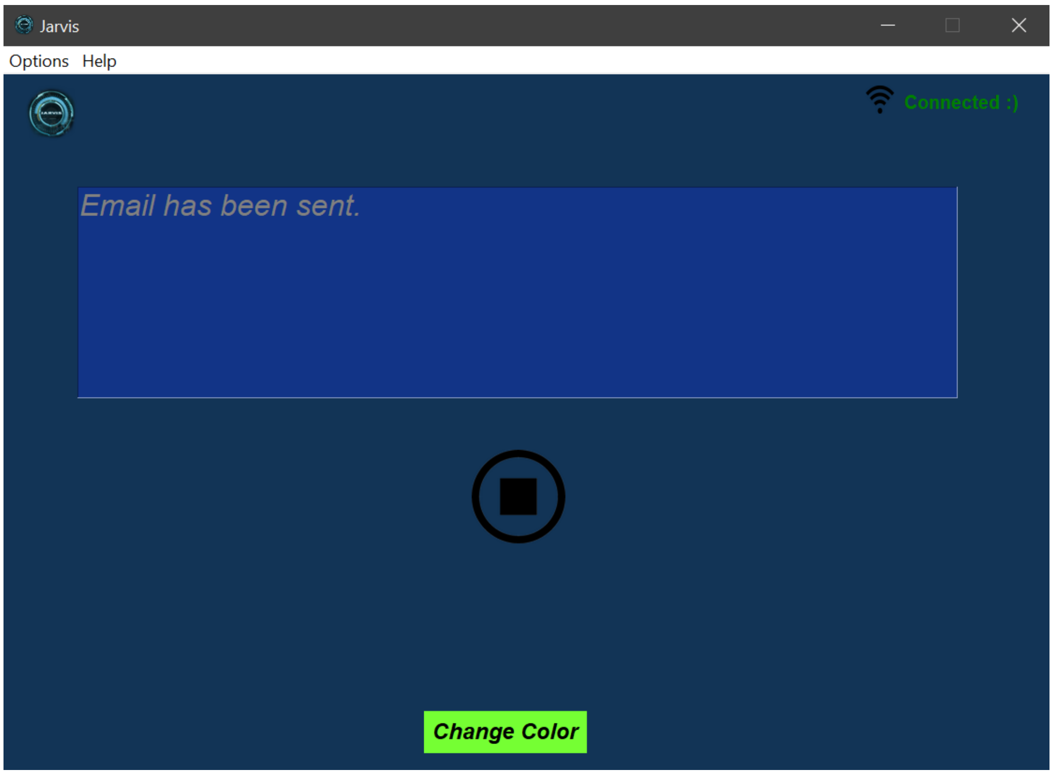


You Said: send email



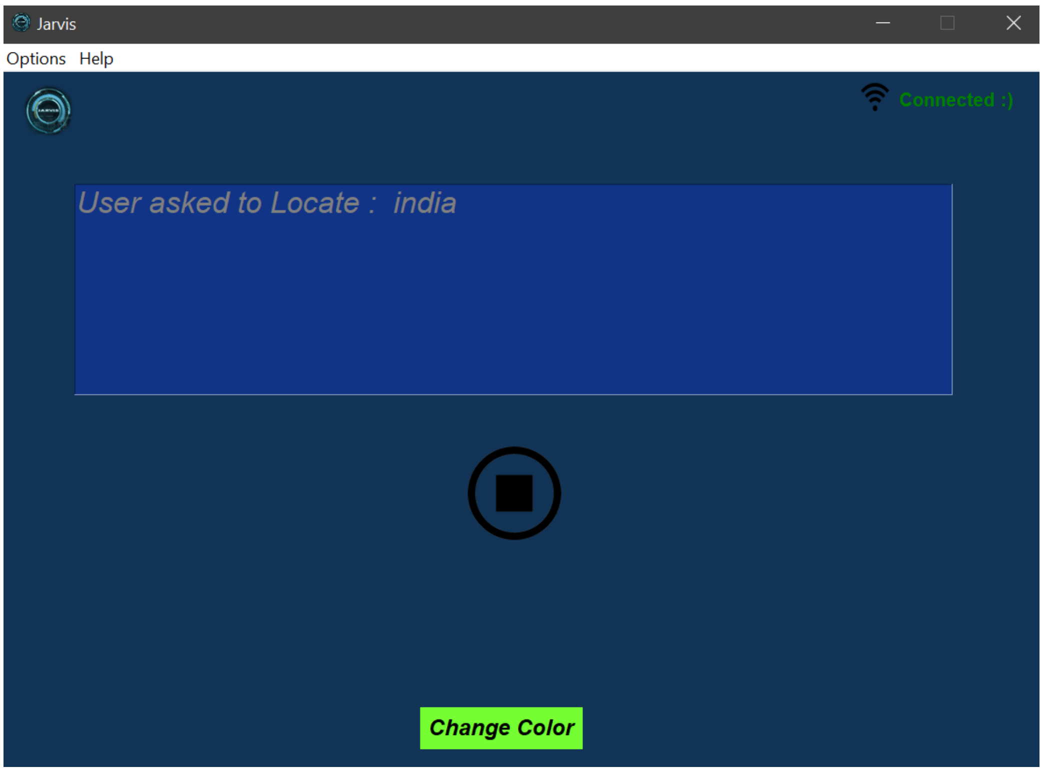
Who is the Reciever?

Enter recieptant's email

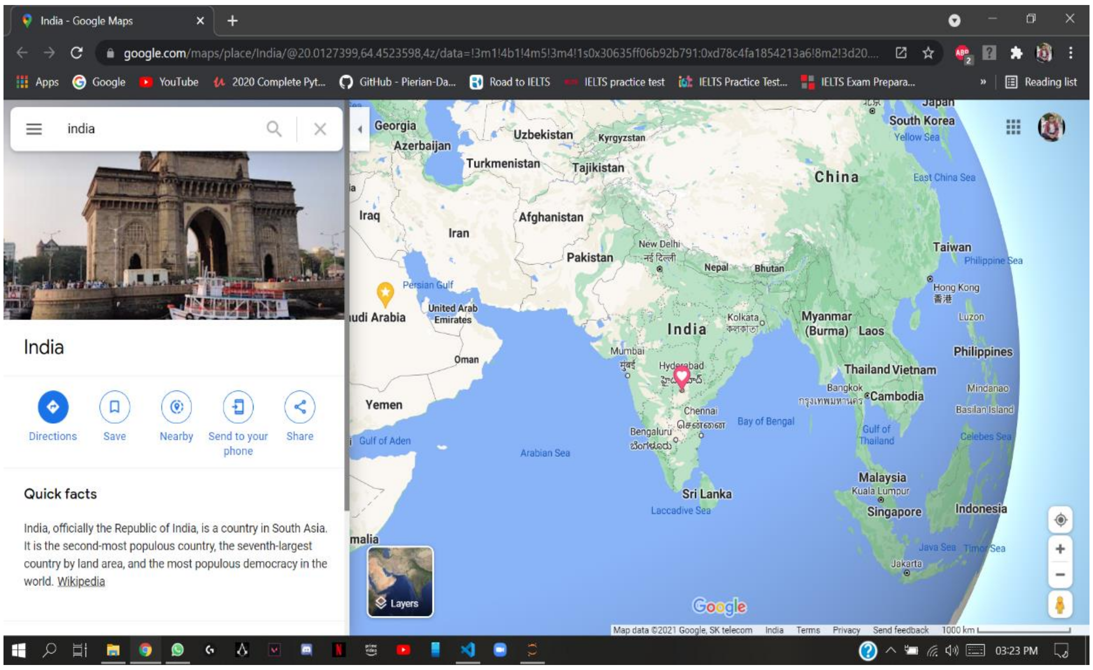


Email has been sent.

Locate Place on Map

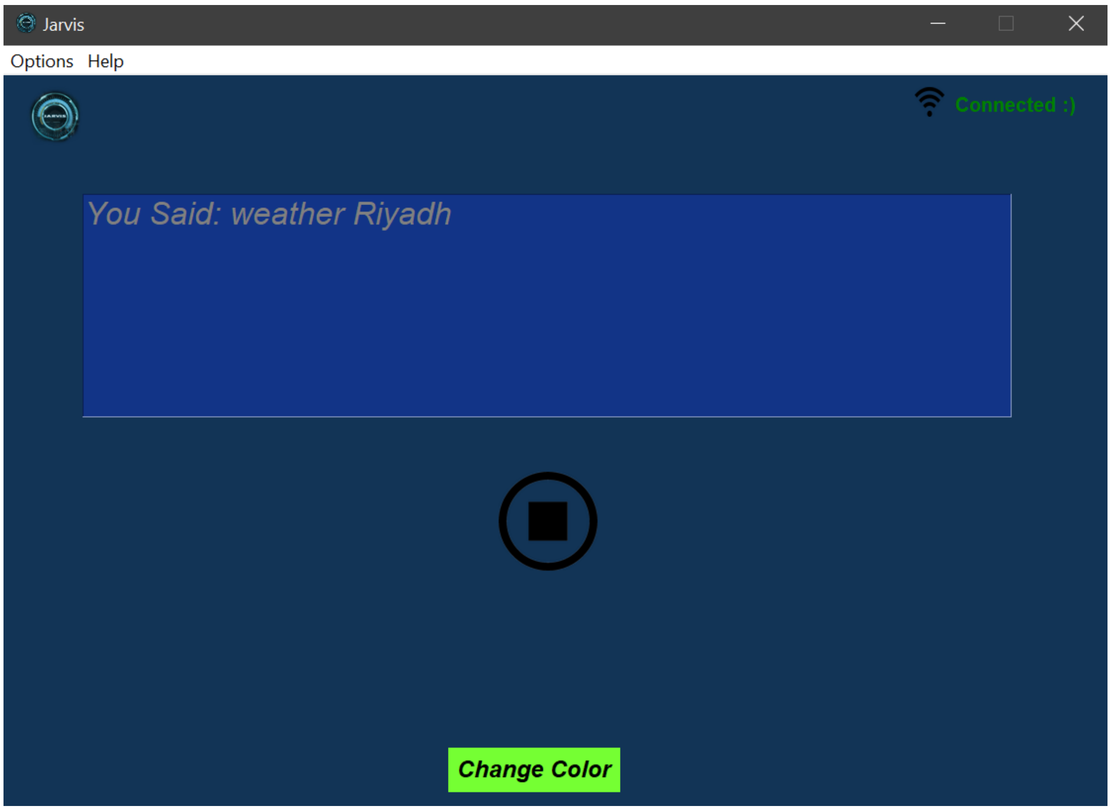


User asked to Locate : india

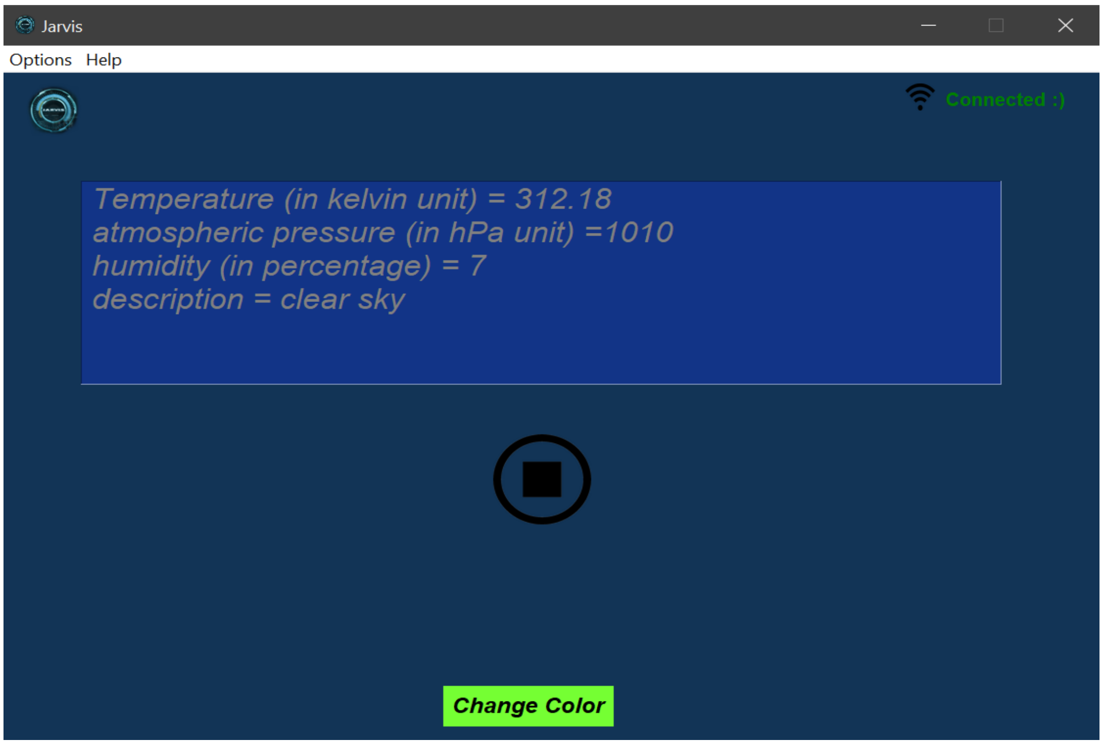


India, officially the Republic of India, is a country in South Asia. It is the second-most populous country, the seventh-largest country by land area, and the most populous democracy in the world. Wikipedia

Weather

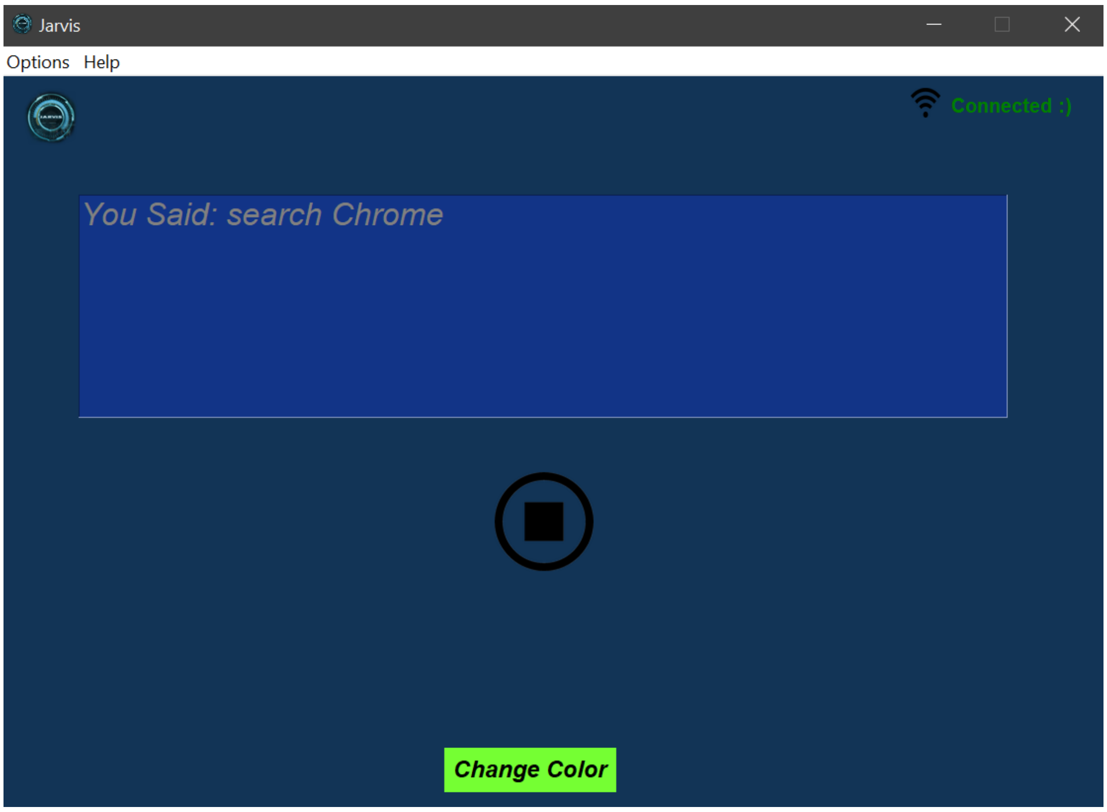


You Said: weather Riyadh

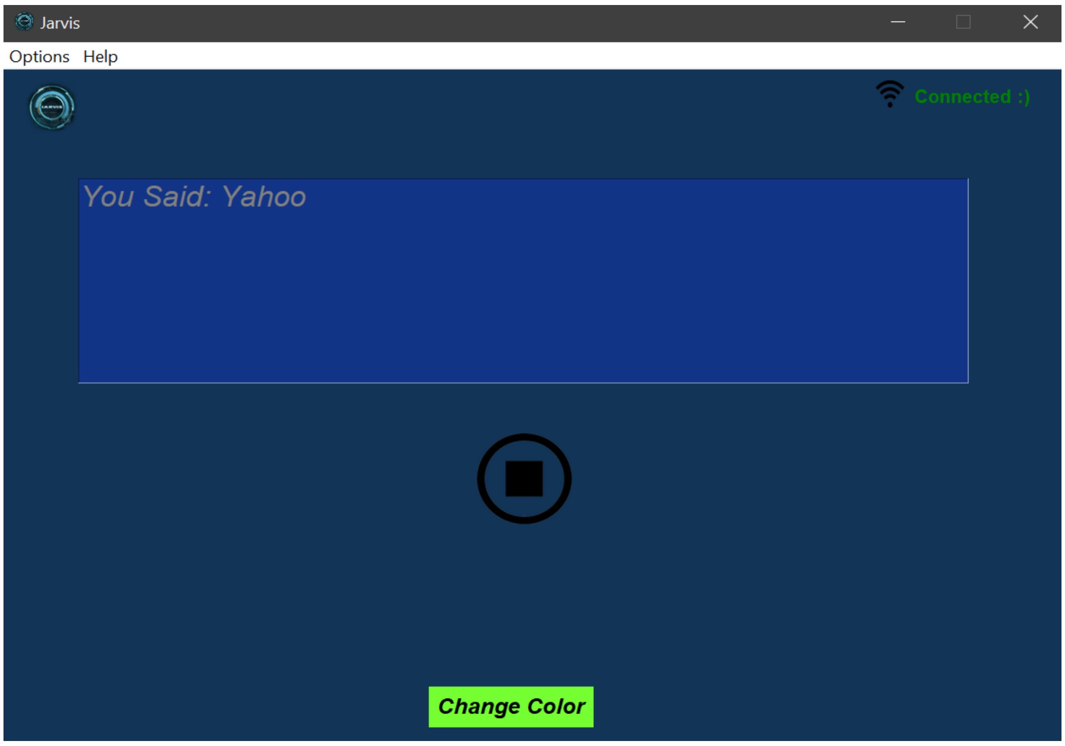


Temperature (in kelvin unit) = 312.18
atmospheric pressure (in hPa unit) =1010
humidity (in percentage) = 7
description = clear sky

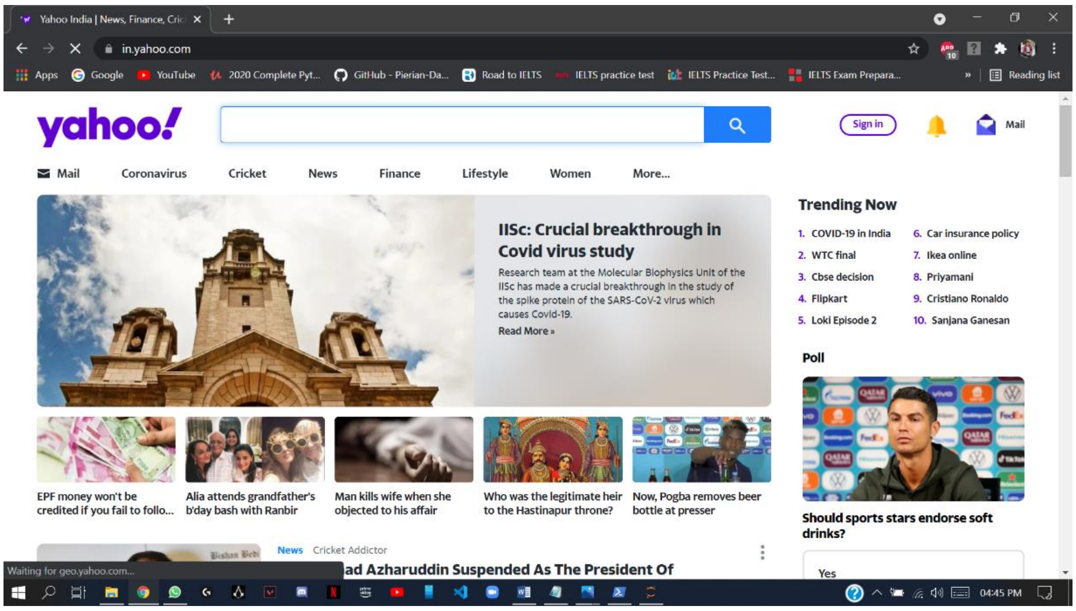
Chrome Search



You Said: search Chrome



You Said: Yahoo


Yahoo India | News, Finance, Cric
in.yahoo.com
Apps Google YouTube 2020 Complete Pyt... GitHub - Pierian-Da... Road to IELTS IELTS practice test IELTS Practice Test... IELTS Exam Prepara... Reading list
yahoo!
Sign in Mail
Mail Coronavirus Cricket News Finance Lifestyle Women More...
Waiting for geo.yahoo.com...
04:45 PM

IISc: Crucial breakthrough in Covid virus study

Research team at the Molecular Biophysics Unit of the IISc has made a crucial breakthrough in the study of the spike protein of the SARS-CoV-2 virus which causes Covid-19.

Read More »

EPF money won't be credited if you fail to follo...

Alia attends grandfather's b'day bash with Ranbir

Man kills wife when she objected to his affair

Who was the legitimate heir to the Hastinapur throne?

Now, Pogba removes beer bottle at presser

News Cricket Addictor

ad Azharuddin Suspended As The President Of

Trending Now

1. COVID-19 in India
2. WTC final
3. Cbse decision
4. Flipkart
5. Loki Episode 2
6. Car insurance policy
7. Ikea online
8. Priyamani
9. Cristiano Ronaldo
10. Sanjana Ganesan

Poll

Should sports stars endorse soft drinks?

Yes

CHAPTER 8

TESTING STRATEGIES

CHAPTER 8

TESTING STRATEGIES

A **test strategy** is an outline that describes the testing approach of the software development cycle. The purpose of a test strategy is to provide a rational deduction from organizational, high-level objectives to actual test activities to meet those objectives from a quality assurance perspective

8.1 Introduction to Automated and Manual Testing

Automated Testing is a software testing technique that performs using special automated testing software tools to execute a test case suite. On the contrary, Manual Testing is performed by a human sitting in front of a computer carefully executing the test steps.

The automation testing software can also enter test data into the System Under Test, compare expected and actual results and generate detailed test reports. Software Test Automation demands considerable investments of money and resources.

Successive development cycles will require execution of same test suite repeatedly. Using a test automation tool, it's possible to record this test suite and re-play it as required. Once the test suite is automated, no human intervention is required. This improved ROI of Test Automation. The goal of Automation is to reduce the number of test cases to be run manually and not to eliminate Manual Testing altogether.

Manual Testing is a type of software testing in which test cases are executed manually by a tester without using any automated tools. The purpose of Manual Testing is to identify the bugs, issues, and defects in the software application. Manual software testing is the most primitive technique of all testing types and it helps to find critical bugs in the software application.

Any new application must be manually tested before its testing can be automated. Manual Software Testing requires more effort but is necessary to check automation feasibility. Manual Testing concepts does not require knowledge of any testing tool. One of the Software Testing Fundamental is "**100% Automation is not possible**". This makes Manual Testing imperative.

8.2 TEST CASES

The following cases were carried out and the respective results were encountered for each test case:

8.2.1 Test Case for Speech Recognition

Description	**Expected Output**	**Actual Output**	**Remark Pass/Fail**	**Date**	**Name of Creator**
Wikipedia Function	It loads the UI Panel.	Speak again	Fail	13th Jan 2021	Rehan
Description	**Expected Output**	**Actual Output**	**Remark Pass/Fail**	**Date**	**Name of Creator**
Wikipedia Function	It loads the UI Panel.	Gives the information about the word spoken.	Pass	13th Jan 2021	Rehan

8.2.2 Test Case for Date and Time

Description	Expected Output	Actual Output	Remark Pass/Fail	Date	Name of Creator
Identifying the date and time	It loads the UI Panel.	Speak again	Fail	13th Jan 2021	Hassan
Description	**Expected Output**	**Actual Output**	**Remark** **Pass/Fail**	**Date**	**Name of Creator**
Identifying the date and time	It loads the UI Panel.	Gives the current date and time.	Pass	13th Jan 2021	Hassan

8.2.3 Test Case for CPU and Battery

Description	Expected Output	Actual Output	Remark Pass/Fail	Date	Name of Creator
Analysing the CPU optimization and Battery percentage.	It loads the UI Panel.	Speak again	Fail	13th Jan 2021	Sultan
Description	**Expected Output**	**Actual Output**	**Remark** **Pass/Fail**	**Date**	**Name of Creator**
Analysing the CPU optimization and Battery percentage.	It loads the UI Panel.	Tells about the CPU and Battery percent.	Pass	13th Jan 2021	Sultan

8.2.4 Test Case for Making a Note

Description	Expected Output	Actual Output	Remark Pass/Fail	Date	Name of Creator
Checking the ability to make a note	It loads the UI Panel.	Speak again	Fail	13th Jan 2021	Hassan
Description	**Expected Output**	**Actual Output**	**Remark Pass/Fail**	**Date**	**Name of Creator**
Checking the ability to make a note	It loads the UI Panel.	Makes a note which the user commands it to do so.	Pass	13th Jan 2021	Hassan

8.2.5 Test Case for Taking a Screenshot

Description	Expected Output	Actual Output	Remark Pass/Fail	Date	Name of Creator
Using the pyautogui to take the screenshot.	It loads the UI Panel.	Speak again	Fail	13th Jan 2021	Rehan
Description	**Expected Output**	**Actual Output**	**Remark Pass/Fail**	**Date**	**Name of Creator**
Using the pyautogui to take the screenshot.	It loads the UI Panel.	Takes the Screenshot	Pass	13th Jan 2021	Rehan

8.2.6 Test Case for Location on Map

Description	**Expected Output**	**Actual Output**	**Remark Pass/Fail**	**Date**	**Name of Creator**
Locating a place.	It loads the UI Panel.	Speak again	Fail	13th Jan 2021	Sultan
Description	**Expected Output**	**Actual Output**	**Remark Pass/Fail**	**Date**	**Name of Creator**
Locating a place.	It loads the UI Panel.	Displays the location of the place on map.	Pass	13th Jan 2021	Sultan

8.2.7 Test Case for News Report

Description	**Expected Output**	**Actual Output**	**Remark Pass/Fail**	**Date**	**Name of Creator**
Obtaining the News.	It loads the UI Panel.	Speak again	Fail	13th Jan 2021	Sultan
Description	**Expected Output**	**Actual Output**	**Remark Pass/Fail**	**Date**	**Name of Creator**
Obtaining the News.	It loads the UI Panel.	Displays the headlines to the user.	Pass	13th Jan 2021	Sultan

8.2.8 Test Case for Calculations

Description	**Expected Output**	**Actual Output**	**Remark Pass/Fail**	**Date**	**Name of Creator**
Solving the expressions.	It loads the UI Panel.	Speak again	Fail	13th Jan 2021	Rehan
Description	**Expected Output**	**Actual Output**	**Remark Pass/Fail**	**Date**	**Name of Creator**
Solving the expressions.	It loads the UI Panel.	Displays the solution to user using wolframalpha.	Pass	13th Jan 2021	Rehan

CHAPTER 9

CONCLUSION

CHAPTER 9

CONCLUSION

The main aim of the project was to develop an Desktop Assistant that will be used to identify answers related to user submitted questions. To provide with sufficient information that is required by the user. A background research took place, which included an overview of the conversation procedure and any relevant desktop Assistant available. A desktop Assistant already in user were excellent service that is provided. The system is made on python programming language to be more specific Python 3.8.

Different libraries where used such as Speech Recognition, Text to Speech convertor, Short Mail Transferring Protocols (SMTP). It provides information regarding the weather, News, it can play music, it can search for topics on Wikipedia, can setup an alarm, Display the current date and time.

User can collect information through this application. It reduces both man power and time. Due to support of NLP user can ask queries in very formal way. No need ask queries in very strict and specific way. The user should aware of general rules of English Language. The goal is to provide people a quick and easy way to have their questions answered.

CHAPTER 10

FUTURE ENHANCEMENTS

CHAPTER 10

FUTURE ENHANCEMENTS

The virtual assistants which are currently available are fast and responsive but we still have to go a long way. The understanding and reliability of the current systems need to be improved a lot. The assistants available nowadays are still not reliable in critical scenarios. The future of these assistants will have the virtual assistants incorporated with Artificial Intelligence which includes Machine Learning, Neural Networks, etc. and IoT. With the incorporation of these technologies, we will be able to achieve new heights. What the virtual assistants can achieve is much beyond what we have achieved till now. Most of us have seen Jarvis, that is a virtual assistant developed by iron man which is although fictional but this has set new standards of what we can achieve using
voice-activated virtual assistants.

REFERENCES

1. www.geeksforgeeks.org

2. www.udemy.com
3. Kei Hashimoto, Junichi Yamagishi, William Byrne, Simon King, Keiichi Tokuda, “An analysis of machine translation and speech synthesis in speech-to-speech translation system” proceedings of 5108978-1-4577-0539- 7/11/$26.00 ©2011 IEEE.

4. Nil Goksel-Canbek Mehmet Emin Mutlu, “On the track of Artificial Intelligence: Learning with Intelligent Personal Assistant” International Journal of Human Sciences.

5. H. Phatnani, Mr. J. Patra and Ankit Sharma’ “CHATBOT ASSISTING: SIRI” Proceedings of BITCON-2015 Innovations For National Development National Conference on Research and Development in Computer Science and Applications, E-ISSN2249–8974.

6. Sutar Shekhar, P. Sameer, Kamad Neha, Prof. Devkate Laxman, " An Intelligent Voice Assistant Using Android Platform", March 2015, IJARCSMS, ISSN: 232 7782

7. VINAY SAGAR, KUSUMA SM, "Home Automation Using Internet of Things", June-2015, IRJET, e-ISSN: 2395 -0056.

8. “Speech recognition with flat direct models,” IEEE Journal of Selected Topics in Signal Processing, 2010.

9. Rishabh Shah, Siddhant Lahoti, Prof. Lavanya. K, “An Intelligent Chatbot using Natural Language Processing”. International Journal of Engineering Research, Vol. 6, pp. 281-286, 1 May 2017.
High Technology Letters
Volume 26, Issue 6, 2020
ISSN NO : 1006-6748
514