

INDEX

1. Introduction

- 1.1. Overview
- 1.2. Problem Statement
- 1.3. Scope

2. Analysis

- 2.1. Objective of project
- 2.2. Requirement Gathering
- 2.3. HARDWARE AND SOFTWARE REQUIREMENTS
- 2.4. Working
- 2.5. Software Requirement Specification
- 2.6. Feasibility Study
- 2.7. Software Engineering Paradigm
- 2.8. Software Development Life Cycle

3. Design

- 3.1 SYSTEM DESIGN AND IMPLEMENTATION
- 3.2 DESIGN OF DATABASE
- 3.3 DATA FLOW DIAGRAM (DFD)
- 3.4 FLOW CHART

4. Tools Use

5. Result

6. Source Code

7. Testing

8. Conclusion

9. Reference

ABSTRACT

The Most famous application of iPhone is "SIRI" which helps the end user to communicate end user mobile with voice and it also responds to the voice commands of the user. Same kind of application is also developed by the Google that is "Google Voice Search" which is used for in Android Phones. But this Application mostly works with Internet Connections. But our Proposed System has capability to work with Internet Connectivity. It is named as "**voice Assistant with computer vision** " 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.

Here is a most important hidden feature called database management we can easily perform the sql query using voice command and show the data in the form of graphical interface.

In addition, this proposed system can change the way of interactions between end user and the computer. The system is being designed in such a way that all the services provided by the Computer are accessible by the end user on the user's voice commands.

Keywords: Jay (There is no need use key word)

1 . Introduction

1.1 OVERVEIW

Today, we can ask voice assistants like Apple's Siri, Google Now to perform simple tasks like, "What's the weather", etc. in our own natural language. The next evolution of natural language interaction with voice assistants is in the form of task automation such as "turn on the air conditioner whenever the temperature rises above 30 degrees Celsius", or "if there is motion on the security camera after 10pm, call Bob". A voice assistant is a digital assistant that uses voice recognition, speech synthesis and natural language processing (NLP) to provide a service through a particular application. Now everyone wants to have an assistant who listen our call, anticipates our needs and can take necessary action when needed. This luxury life is now available with the help of Artificial Intelligence based on voice assistant. Voice assistants come's in small packages and can perform a variety of actions after hearing our commands. They can launch apps, open web browser, answer basic informational queries, tell horoscope, calculate your BMI, answer our queries, play music, send email, set reminders, make lists, and do basic math calculations, etc.

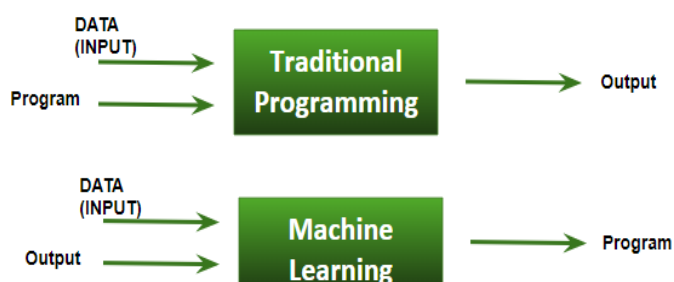
Example: Training of students during exam.

While preparing for the exam students don't actually cram the subject but try to learn it with complete understanding. Before the examination, they feed their machine(brain) with a good amount of high-quality data (questions and answers from different books or teachers notes or online video lectures).

Actually, they are training their brain with input as well as output i.e. what kind of approach or logic do they have to solve a different kind of questions. Each time they solve practice test papers and find the performance (accuracy /score) by comparing answers with answer key given, Gradually, the performance keeps on increasing, gaining more confidence with the adopted approach.

That's how actually models are built, train machine with data (both inputs and outputs are given to model) and when the time comes test on data (with input only) and achieves our model scores by comparing its answer with the actual output which has not been fed while training.

Researchers are working with assiduous efforts to improve algorithms, techniques so that these models perform even much better



Basic Difference in ML and Traditional Programming?

Traditional Programming

We feed in DATA (Input) + PROGRAM (logic), run it on machine and get output.

Machine Learning

We feed in DATA(Input) + Output, run it on machine during training and the machine creates its own program(logic), which can be evaluated while testing.

Advantages

Voice Assistant allows you to gain the perks of high-end technology and its functionalities. Our proposed application points to many advantages: 1. Our proposed application provides security to the user as it can authenticate the authorized user using Face Recognition technique. 2. The face recognition technology make the system secure and robust for the user as this does not required any input from the user through keyboard or mouse. 3. The application provides flexibility to the user as it can send email just listening the command given by the user. 4. Our proposed application stores personal information such as location data, reminders and contacts in the notebook. 5. This application includes the functions and services such as: opening system application, event handler, location services, music player service, checking weather, Google search, Wikipedia search, tell horoscope, general conversation and help menu

1.1 PROBLEM STATEMENT

Virtual Personal Assistant

The voice assistant is design to make the work easier of the user. As user can give command to them without making visual access to the screen. The biggest disadvantage of this system is that confidential data can be accessed by unauthorized user so the privacy can be breached. Due to this, the confidentiality, integrity and availability (CIA) of user data is affected. Looking to this problem the security features of "Face Recognition" is designed so that it can detect the authorized user face and take user command as input and provide response via a synthesis voice. Facial recognition technology (FRT) is one of the most controversial new tools. It was first developed in the 1960s. It has recently become accessible to the mass market-to both law enforcement and private consumers. Automatic face recognition involves:

KEY TAKEAWAYS

- A virtual assistant is a self-employed worker who specializes in offering administrative services to clients from a remote location, usually a home office.
- Typical tasks a virtual assistant might perform include scheduling appointments, making phone calls, making travel arrangements, and managing email accounts.
- Some virtual assistants specialize in offering graphic design, blog writing, bookkeeping, social media, and marketing services.

- For an employer, one advantage of hiring a virtual assistant is the flexibility to contract for just the services they need.

Face Recognition

Automatic face recognition (AFR) technologies have made many improvements in the changing world. Smart Attendance using Real-Time Face Recognition is a real-world solution which comes with day-to-day activities of handling student attendance system. Face recognition-based attendance system is a process of recognizing the students face for taking attendance by using face biometrics based on high - definition monitor video and other information technology. In my face recognition project, a computer system will be able to find and recognize human faces fast and precisely in images or videos that are being captured through a surveillance camera. Numerous algorithms and techniques have been developed for improving the performance of face recognition but the concept to be implemented here is Deep Learning. It helps in conversion of the frames of the video into images so that the face of the student can be easily recognized for their attendance so that the attendance database can be easily reflected automatically.

1.3 SCOPE

Future Scope of Machine Learning

FUTURE SCOPE Using this system as a framework, the system can be expanded to features security. Security is important these days so it can be combined with this system to give more advanced security features. In this, the voice authentication technology can be implemented for more security. More advancement are possible like operating on various tones or accents from different regions that mean it should be able to perform operations on various voice tones and accents.. Further modifications are possible like learning the answer of questions that are not known by the voice assistant and replying whenever next time the same question is put up by the user.

Robotics



Robotics is one of the fields that always gain the interest of researchers as well as the common. In 1954, George Devol invented the first robot that was programmable and it was named as **Animate** . After that, in the 21st century, Hanson Robotics created the first AI-robot, **Sophia**. These inventions were possible with the help of Machine Learning and Artificial Intelligence.

Researchers all over the world are still working on creating robots that mimic the human brain. They are using neural networks, AI, ML, computer vision, and many other technologies in this research. In the future, we may come across robots that would be capable of performing various tasks similar to a human.

Computer Vision

As the name suggests, [computer vision](#) gives a vision to a computer or a machine. Here comes into our minds what the Head of AI at Google, Jeff Dean, has once said, 'The progress we've made from 26% error in 2011 to 3% error in 2016 is hugely impactful. The way I like to think is, computers have now evolved eyes that work



Giving the ability to a machine to recognize and analyse images, videos, graphics, etc. is the goal of computer vision. The progress in the field of Artificial Intelligence and Machine Learning has made it possible to achieve the goal of computer vision faster.

2. Analysis

2.1 Objective of project

DB Assistant using machine learning

As we know that Python is a suitable language for scriptwriters and developers. Let's write a script for DB 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.

Computer Vision

The technology aims in imparting a tremendous knowledge oriented technical innovations these days. Deep Learning is one among the interesting domain that enables the machine to train itself by providing some datasets as input and provides an appropriate output during testing by applying different learning algorithms. Nowadays Attendance is considered as an important factor for both the student as well as the teacher of an educational organization. With the advancement of the deep learning technology the machine automatically detects the attendance performance of the students and maintains a record of those collected data. In general, the attendance system of the student can be maintained in two different forms namely.

- Manual Attendance System (MAS)
- Automated Attendance System (AAS).

Manual Student Attendance Management system is a process where a teacher concerned with the particular subject need to call the students name and mark the attendance manually. Manual attendance may be considered as a time-consuming process or sometimes it happens for the teacher to miss someone or students may answer multiple times on the absence of their friends. So, the problem arises when we think about the traditional process of taking attendance in the classroom. To solve all these issues we go with Automatic Attendance System (AAS). Automated Attendance System (AAS) is a process to automatically estimate the presence or the absence of the student in the classroom by using face recognition technology.

- Feature-based approach
- Brightness-based approach

2.2 How does voice assistant work?

The assistant needs a keyword in order to begin the conversation. The assistant jumps into action when you say a specific keyword. In general, keyword includes an assistant name along with greeting.

In the case of Jay voice assistant, it's simply '**Jay**'. In a similar fashion, these assistants need a keyword to begin listening.

The assistant then listens to what you say by the microphone (mic) in your laptop. Furthermore, the assistant then interprets or understand what you said by natural language processing. In most of the cases, voice assistants are connected to neural networks via the internet. Neural networks and artificial intelligence together help assistant to understand the context of human language.

2.3 Software requirement specification

A software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform. It is usually signed off at the end of requirements engineering phase. Software requirement is two types-

1. Functional Requirement.
2. Non Functional Requirement.

Functional requirement

- Face recognition starting
- Normal face detection with attendance system
- Advanced Motion detection for better security
- Searching on Wikipedia
- Searching on Google and playing YouTube video
- Find information online on weather
- Play your music
- Showtime, Calendar
- Open apps on your windows
- Open the website on your windows
- Real-time spoken assistant
- Find a place on google maps

- Write notes & show note
- Answer some normal conversation
- Play Audiobook
- Tell some jokes
- Voice-based Sending mail

Non Functional Requirement.

The software is designed to be light-weighted so that it doesn't be a burden on the machine running it. This system is being build keeping in mind the generally available hardware and software compatibility. Here are the minimum hardware and software

Requirement for virtual assistant.

Hardware:

I3 processor or later.
RAM 4 GB or more.
Mic & speaker
Good internet connectivity

Software:

Windows 10(64-bit).
Python 3.or later
Chrome Driver
Selenium Web Automation
wolfram alpha API
Google weather API

Tools Description

Tools for Making projects:

Visual Studio Code
Python
MySQL
OpenCV

2.4 Feasibility Study

The more humanized the program is, more easier the user can use it. People should accept that even if developers constantly try to add more predefined commands, more responses to it, analyse and respond to the command more intelligently, the program will never be completely comprehensive and contain all the possible circumstances that the users meet. Nevertheless, the program will certainly be improved and be more user-friendly if there can be more readable commands, more humanized structure and more intelligent response.

2.5 Software Development Life Cycle

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process. The phases of SDLC is

- Planning and requirements.
- design.
- Implementation.
- Verification.
- Release and maintenance.

Life cycle phase

	Inception	Elaboration	construction	Training
Requirement	High	moderate	low	low
Design	Moderate	high	moderate	low
Implementation	Moderate	moderate	high	moderate
Assessment	Low	low	high	high
Deployment	Low	low	moderate	high

1. Requirement

requirements are statements that indicate what a system needs to do in order to provide a capability (i.e. utility or benefit.) Requirements are generally prepared during the early stages of a project's system development lifecycle (SDLC).

2. Design

The next stage of Software Development Life Cycle is the Design phase. During the design phase, developers and technical architects start the high-level design of the software and system to be able to deliver each requirement.

3.Implementation

During the implement phase developer start construct the software and data storage.

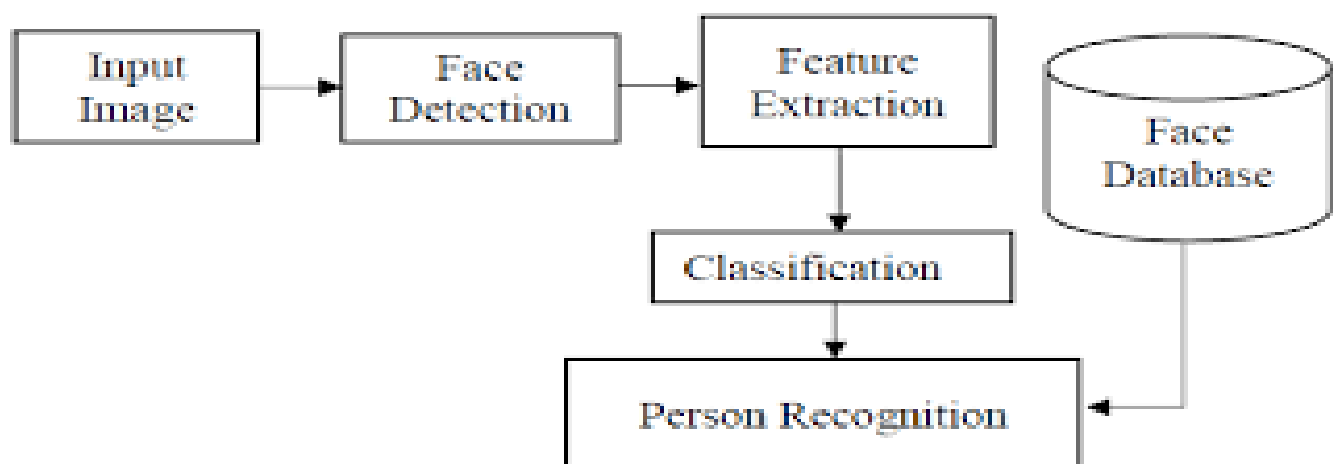
4.Verification

The next stage of software development life cycle in verification install and test Software.

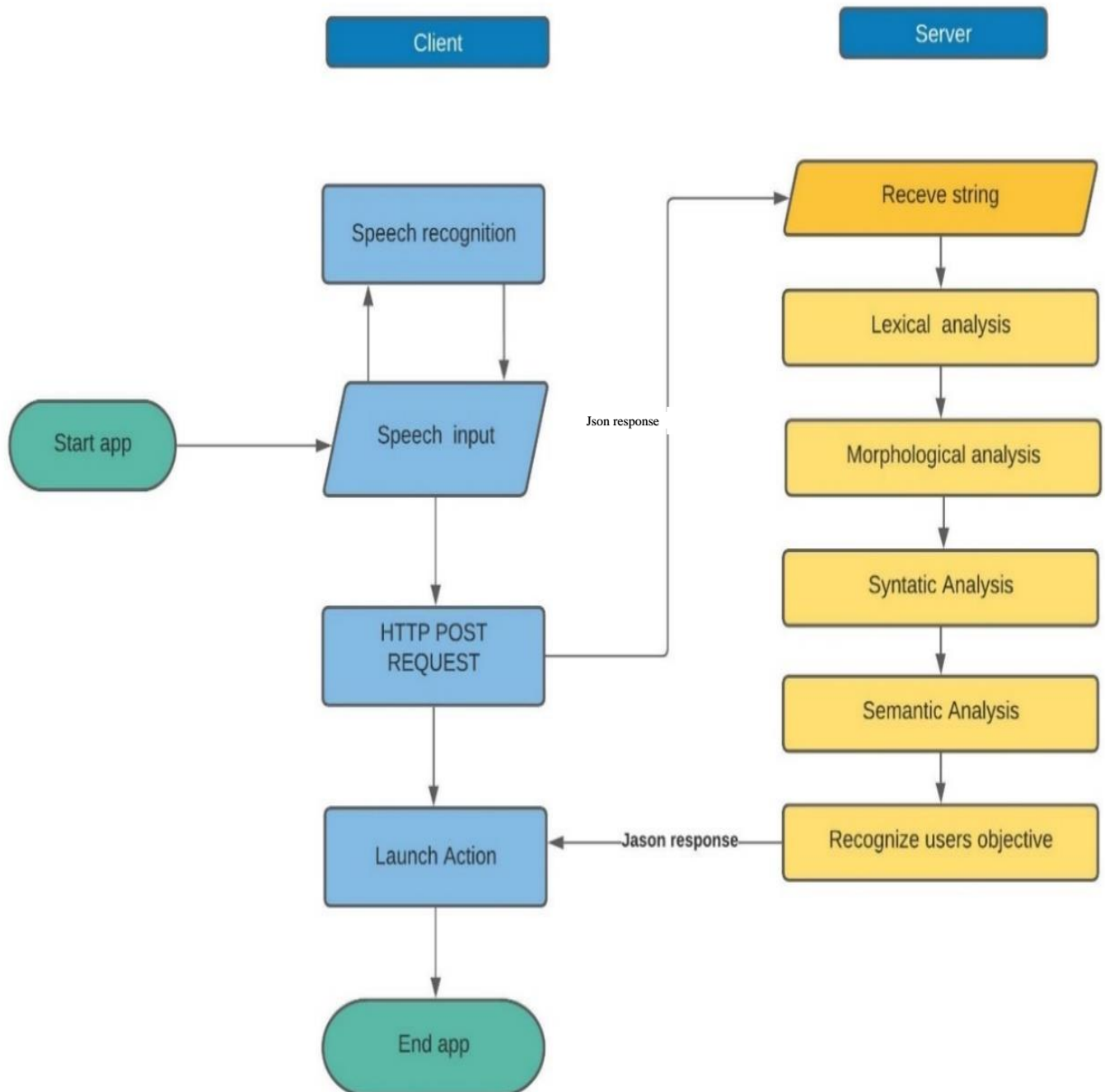
5.Maintenance after verification the phase of maintenance it is the check error and optimize capability.

OpenCV

OpenCV (Open-Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then It sees (which was later acquired by Intel). The library is cross platform and free for use under the open-source BSD license. Python is dynamically typed and garbage collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.



2.5 Working diagram of Jay personal assistant



3. Design

3.1 MODULAR DESCRIPTION

MODULE	DESCRIPTION
PYTTX3	a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline, and is compatible with both Python 2 and 3
SPEECHRECOGNITION	Library for performing speech recognition, with support for several engines and APIs, online and offline.
PYAUDIO	you can easily use Python to play and record audio on a variety of platforms.
PYPDF2	It is use for pdf reding
MYSQL.CONNECTOR	It is use for MySQL connectivity
TKINTER	Tkinter is the Python interface to the Tk GUI toolkit shipped with Python.
WIKIPEDIA	It is use for Wikipedia Search
WOLFRAMALPHA	It is simple search api
DATETIME	Python has a module named datetime to work with dates and time
OPENWEATHERMAP	It is use for finding weather
WEBBROWSER	The webbrowser module provides a high-level interface to allow displaying Web-based documents to users. Under most circumstances, simply calling the open() function from this module will do the right thing.
OS	The OS module in Python provides a way of using operating system dependent functionality.
OPEN CV	The repository contains only OpenCV-Python package build scripts, but not OpenCV itself. Python bindings for OpenCV are developed in official OpenCV
SMTPLIB	Python provides smtplib module , which defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon.

Tkinter:

Graphical User Interface (GUI) is nothing but a desktop application which helps you to interact with the computers. They are used to perform different tasks in the desktops, laptops and other electronic devices.

- GUI apps like **Text-Editors** are used to create, read, update and delete different types of files.
- GUI apps like Sudoku, Chess and Solitaire are games which you can play.
- GUI apps like **Google Chrome, Firefox and Microsoft Edge** are used to browse through the **Internet**.

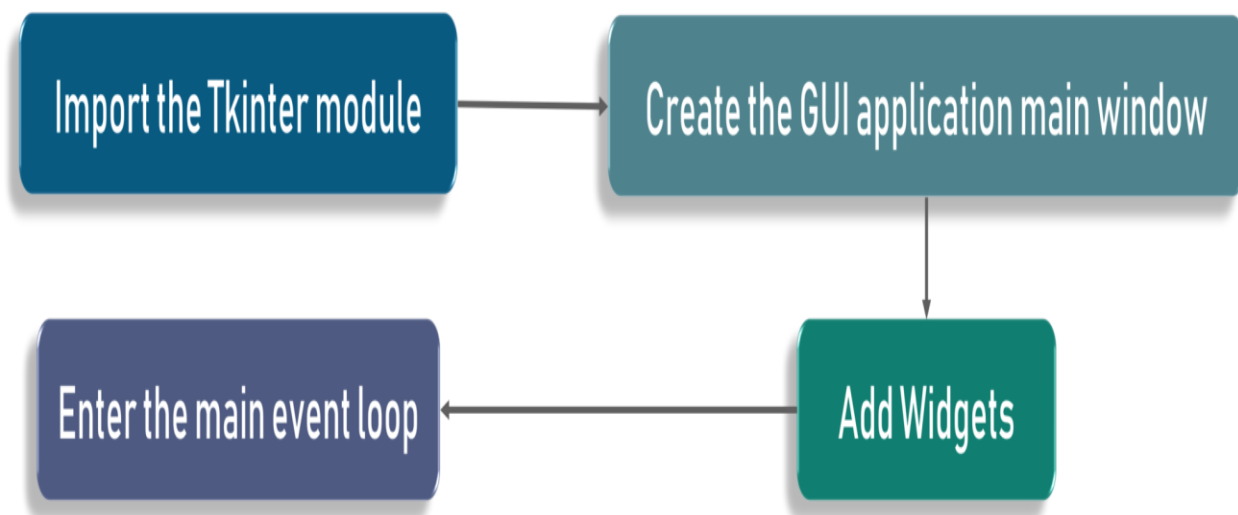
They are some different types of **GUI** apps which we daily use on the laptops or desktops. We are going to learn how to create those type of apps.

As this is an Introduction to GUI, make sure you stay tuned till the end as we will create a really simple and nice **GUI** app.

Well, it is a personal preference that I prefer GUI over command line. Not that there is something wrong with the command line but I prefer more intuitive and interactive applications with a lot of visuals.

Fundamentals of Tkinter

Consider the following diagram, it shows how an application actually executes in Tkinter:



To start out with, we first import the Tkinter model. Followed by that, we create the main window. It is in this window that we are performing operations and displaying visuals and everything basically. Later, we add the widgets and lastly we enter the main event loop.

If you noticed, there are 2 keywords here that you might not know at this point. These are the 2 keywords:

- Widgets
- Main Event Loop

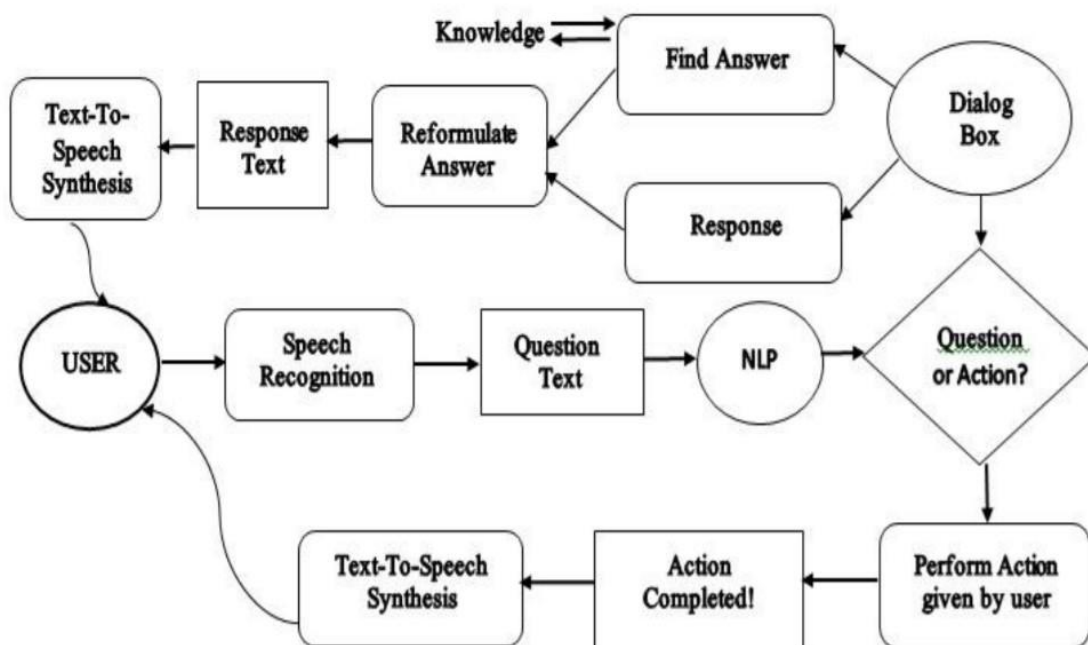
An event loop is basically telling the code to keep displaying the window until we manually close it. It runs in an infinite loop in the back-end.

For example, Microsoft Word is shown on the title tab when you open a word application, correct? Similarly here we call it GUI. We can call it anything we want based on the requirement.

Lastly, we have a label. A label is nothing is but what output needs to be shown on the window. In this case as you can already see, it is hello world

3.2 SYSTEM DESIGN AND IMPLEMENTATION

The proposed model of the voice assistant is as shown in the above figure 1. The model consists of user input through microphone to accept commands from the user. These commands are then go through Speech Recognition, it is the ability of a machine or program to identify words and phrases in spoken languages and convert them to a machine-readable format. On these input Natural Language Processing is applied, it is a field which is created by amalgamating computer science and artificial intelligence. Using NLP, we are concerned with interactions between computers and human natural languages. Then the BRAIN check whether it is a question or an action, if it is a action than the action is performed by the voice assistant and acknowledgment is given to the user via a synthesis voice or if it is a question than it is search in dialog box or knowledge base and then response via a synthesis voice to the user. Our Voice assistant uses google text-to-speech API to understand all the words spoken by the user, and based on certain conditions that satisfy being a command the voice assistant sends responses to the user.



3.3 DATA FLOW DIAGRAM DFD

DFD is a graphical representation which provides information flow between input and output data. It is also known as "Data Flow Chart or Bubble Chart". A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated.

Level 0 DFD:

The user gives the input in the form of voice; this voice command is recognized by the application. Then it will check whether it is the authorized user, then action is performed as per the command given by the user. Command given is compared as a form of action and question and response with the dialog box or search through the knowledge base.

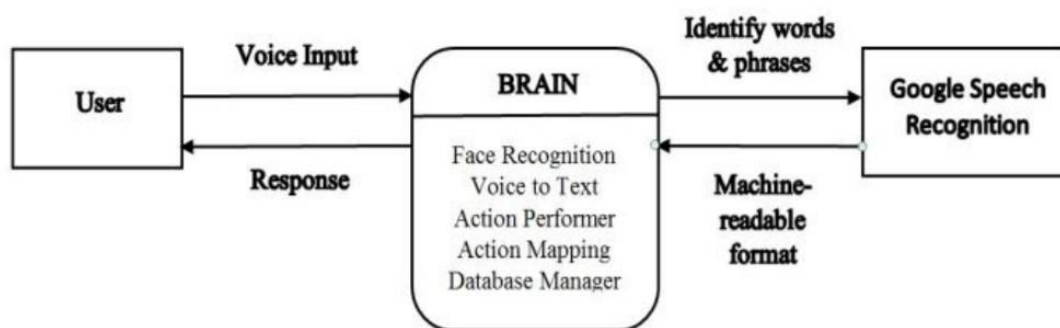
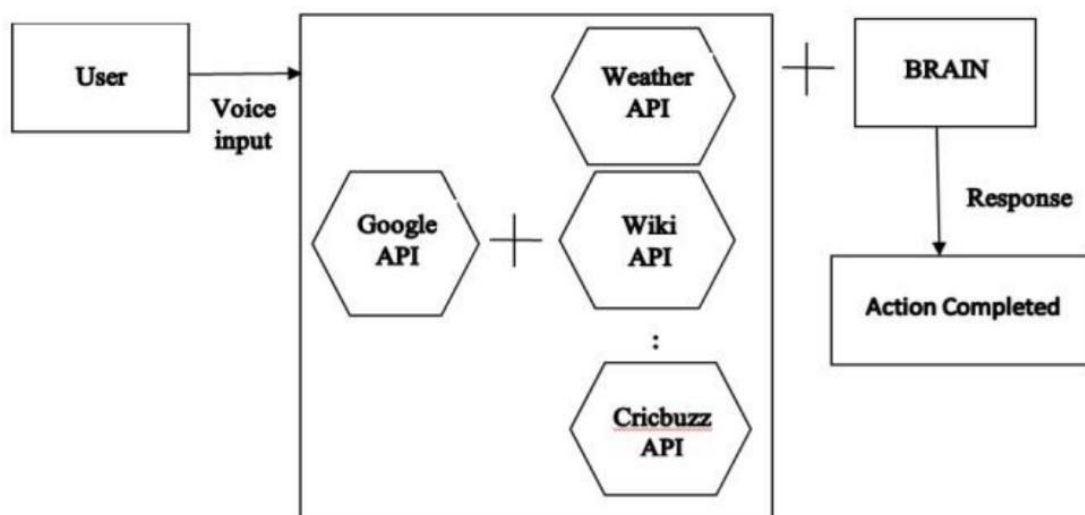


Fig -2: DFD Diagram Level 0

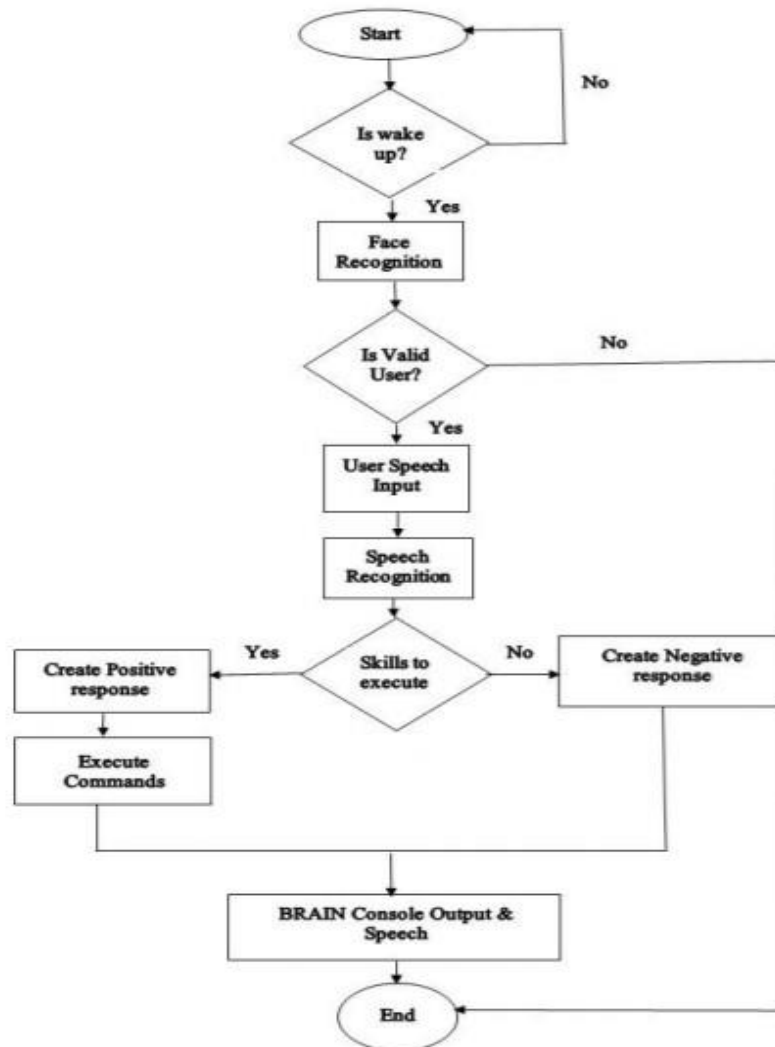
Level 1 DFD:

Input is given by user in the form of voice. [Google Voice API](#) will convert this voice data in text form and then the action is performed by the voice assistant according to the command given by the user by comparing with the dialog box and knowledge base.



3.4 FLOW CHART

Flow chart is the graphical representation of algorithms. Different symbols are used to represent flow chart. As the system is started, it first authenticates the authorized user, then voice assistant is on running in the background listening for available voice commands; once the user gives a command, based on the conditions provided to the voice assistant, the voice assistant gives the necessary output. This output is sent to the Speech Recognition which is convert the speech into machine-readable form. Based on the input received the personal voice assistant then performs the desired task.



4 Implementation

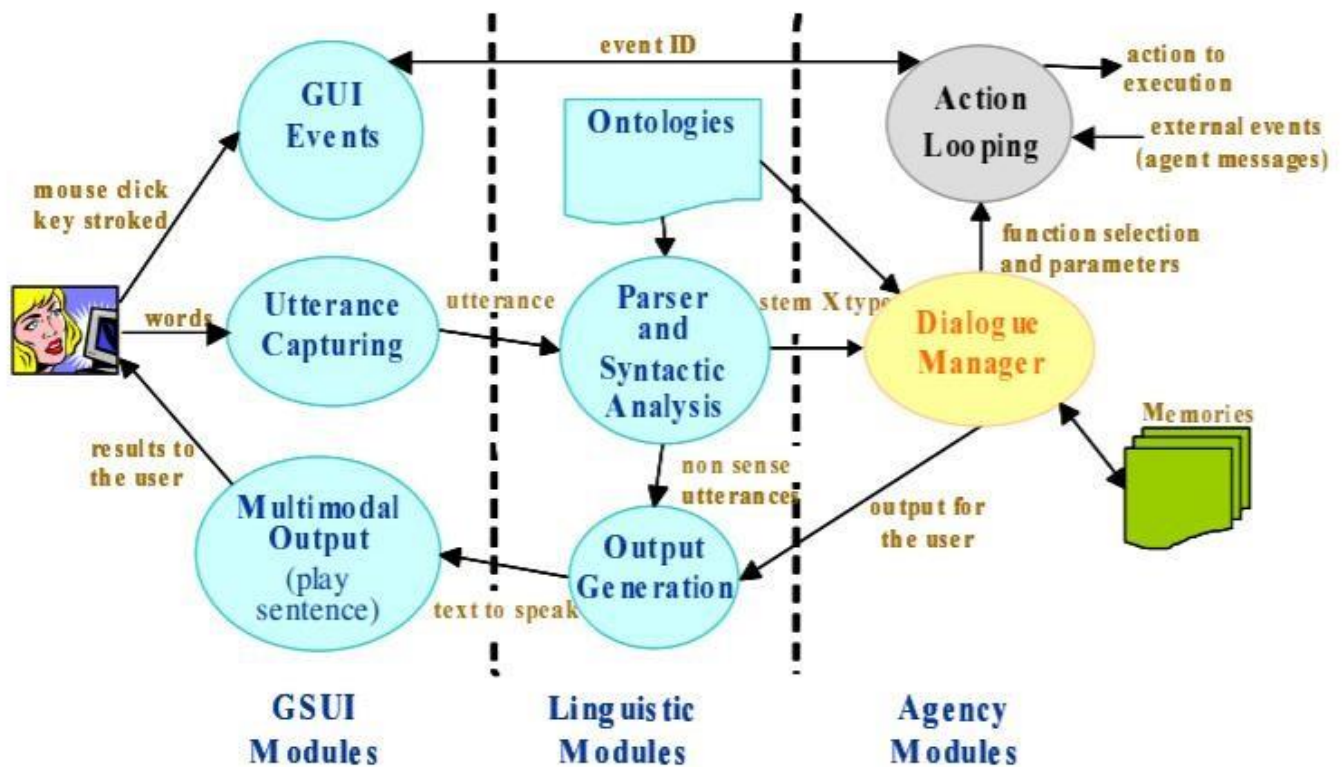
Tools Description

In this project we use tkinter tool and we use system HP Window 10 and software IDLE Python 3.3

Tkinter is actually an inbuilt **Python** module used to create simple **GUI** apps. It is the most commonly used module for **GUI** apps in the **Python**.

You don't need to worry about installation of the **Tkinter** module as it comes with **Python** default.

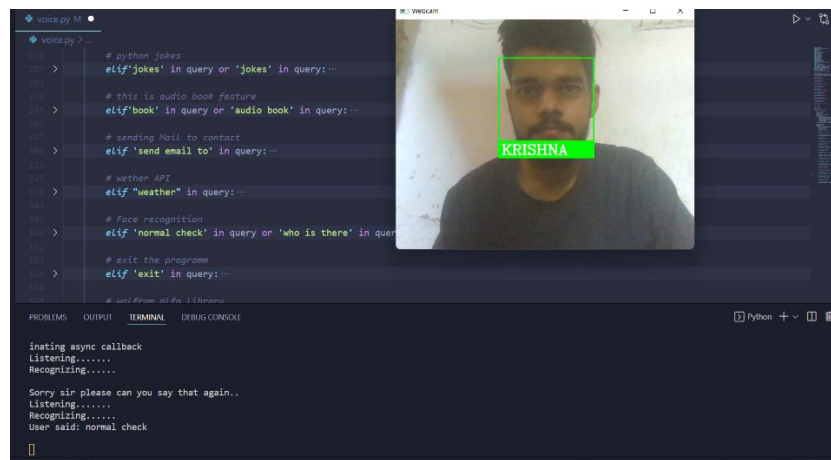
Process Logic of modules



5 Result

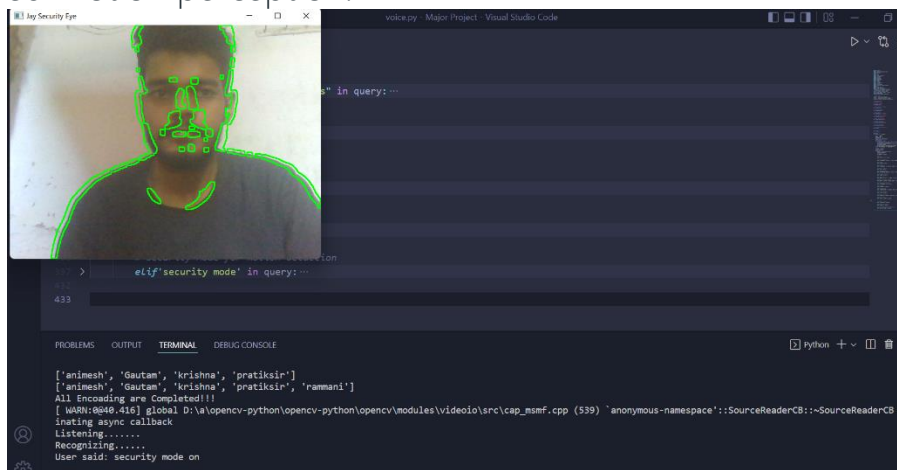
Face recognition Using OpenCV

A facial recognition system is a technology capable of matching a human face from a digital image or a video frame against a database of faces, typically employed to authenticate users through ID verification services, works by pinpointing and measuring facial features from a given image



Motion Detection using OpenCV for Better Security

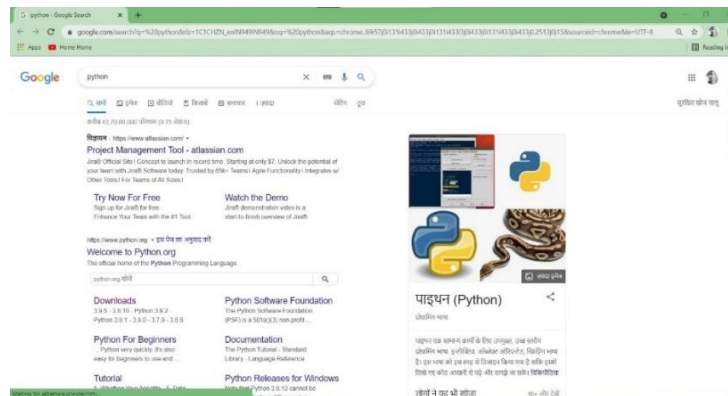
Motion detection is the process of detecting a change in the position of an object relative to its surroundings or a change in the surroundings relative to an object. It can be achieved by either mechanical or electronic methods.[1] When it is done by natural organisms, it is called motion perception.



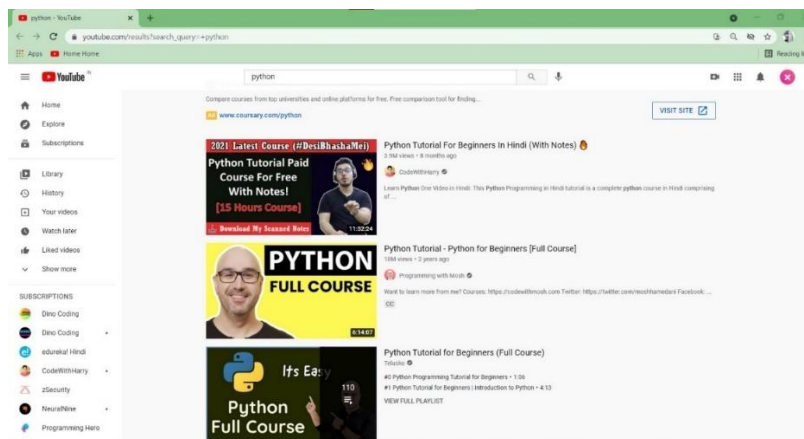
Launch a Web Browser using web browser module

In Python, **web browser module** provides a high-level interface which allows displaying Web-based documents to users. The web browser module can be used to launch a browser in a platform-independent manner as shown below:

Search Python in google



Play Python in google



Wikipedia module in Python

The Internet is the single largest source of information, and therefore it is important to know how to fetch data from various sources. And with Wikipedia being one of the largest and most popular sources for information on the Internet.

Wikipedia is a multilingual online encyclopaedia created and maintained as an open collaboration project by a community of volunteer editors using a wiki-based editing system.

In this article, we will see how to use Python's Wikipedia module to fetch a variety of information from the Wikipedia website.

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

Source Control (Ctrl+Shift+G)
PS D:\python project> & "C:/Program Files/Python39/python.exe" "d:/python project/Jay.py"
Listening.....
Recognizing.....
User said: Samsung Wikipedia

In the Hebrew Bible, Samson (; Hebrew: שמשון, Šimšōn, "man of the sun") was the last of the judges of the ancient Israelites mentioned in the Book of Judges (chapters 13 to 16) and one of the last leaders who "judged" Israel before the institution of the monarchy. He is sometimes considered to be an Israelite version of the popular Near Eastern folk hero also embodied by the Sumerian Enkidu and the Greek Heracles. The biblical account states that Samson was a Nazirite, and that he was given immense strength to aid him against his enemies and allow him to perform superhuman feats, including slaying a lion with his bare hands and massacring an entire army of Philistines using only the jawbone of a donkey.
```

Send mail from your Gmail account using Python

Here, we are going to learn how to send a simple basic mail using Python code. Python, being a powerful language don't need any external library to import and offers a native library to send emails- "SMTP lib". "smtplib" creates a Simple Mail Transfer Protocol client session object which is used to send emails to any valid email id on the internet. Different websites use different port numbers.

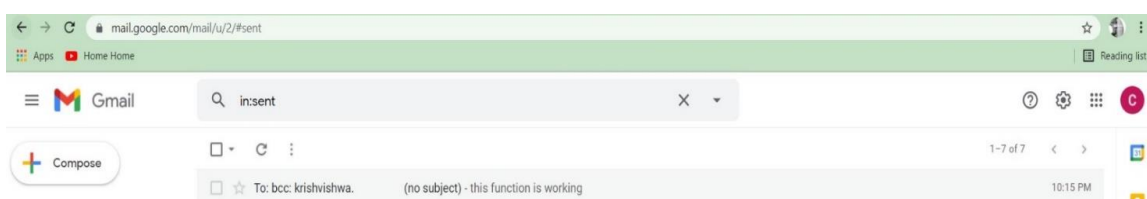
In this article, we are using a Gmail account to send a mail. Port number used here is '587'. And if you want to send mail using a website other than Gmail, you need to get the corresponding information

```
Listening.....
Recognizing.....
User said: send email

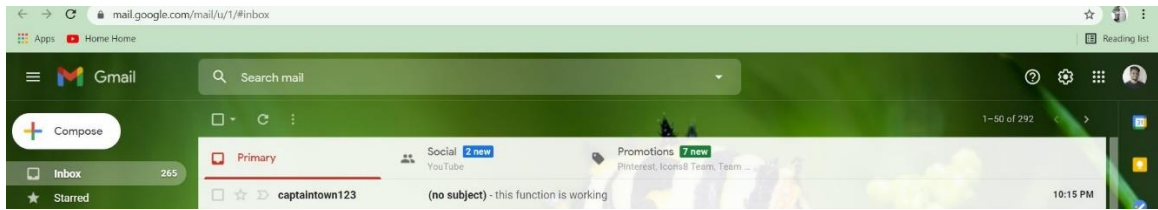
Listening.....
Recognizing.....
User said: this function is working

Email has been send
```

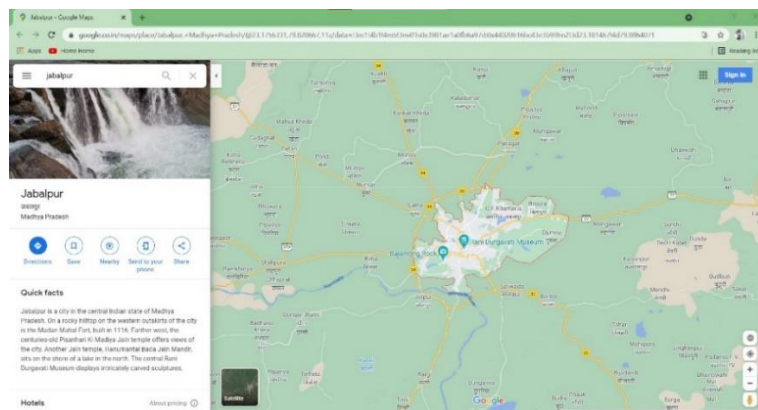
Sender side



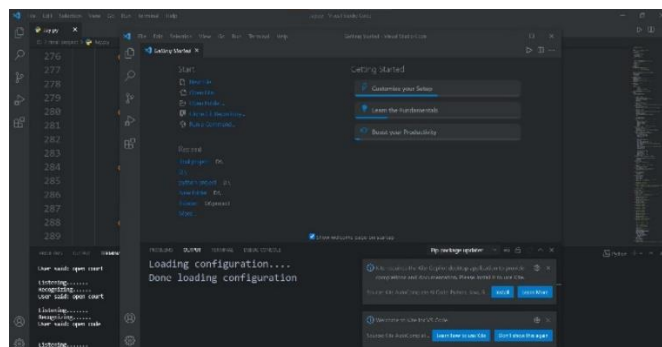
Receiver side



Python script to open a Google Map location on chrome



Open System Software using Os Module



Assistant using Wolfram Alpha API.

The Wolfram Alpha Webservice API provides a web-based API allowing the computational and presentation capabilities of Wolfram Alpha to be integrated into web, mobile, desktop, and enterprise applications. Wolfram Alpha is an API which can compute expert-level answers using Wolfram's algorithms, knowledgebase and AI technology. It is made possible by the Wolfram Language. This article tells how to create a simple assistant application in Python which can answer simple questions like the ones listed below.


```
Listening.....
Recognizing.....
User said: what is the capital of India

New Delhi, Delhi, India
█
```

```
Listening.....
Recognizing.....
User said: who is the Prime Minister of India

Narendra Modi (from 26/05/2014 to present)
█
```

Write a note & show note

```
Listening.....
Recognizing.....
User said: write a note

Listening.....
Recognizing.....
User said: my name is Krishna Vishwakarma

Listening.....
Recognizing.....
User said: show note
```

Find current weather of any city using OpenWeathermap API in Python

openweathermap is a service that provides weather data, including current weather data, forecasts, and historical data to the developers of web services and mobile applications.

It provides an API with JSON, XML and HTML endpoints and a limited free usage tier. Making more than 60 calls per minute requires a paid subscription starting at *USD 40* per month. Access to historical data requires a subscription starting at *150 USD* per month. Users can request current weather information, extended forecasts and graphical maps (showing cloud cover, wind speed, pressure and precipitation)

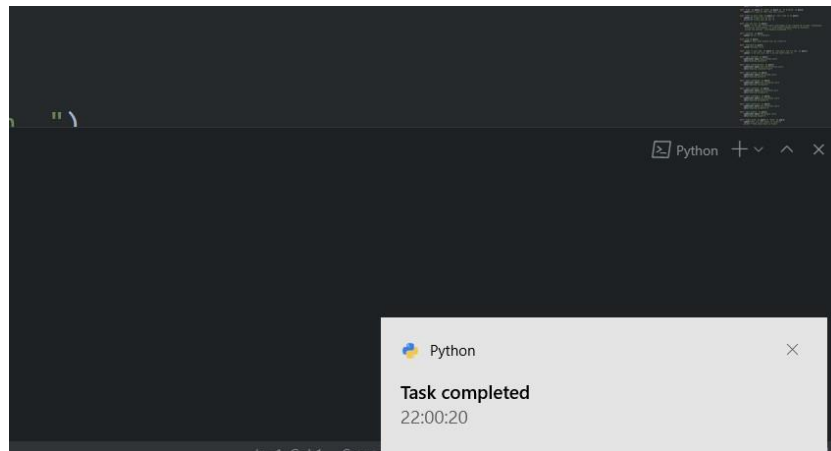
```
Listening.....
Recognizing.....
User said: Jabalpur weather

Temperature is 28°C
Time: Saturday 10:02 pm
Sky Description: Partly cloudy
e
█
```

Get Current Date and Time using Python

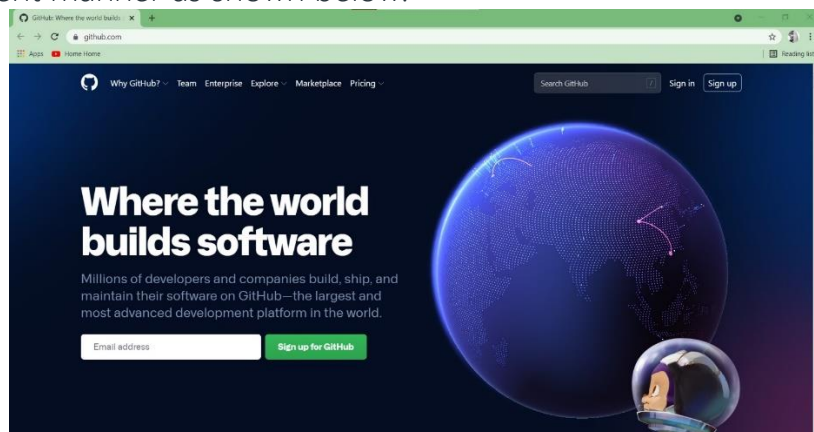
In Python, date and time are not a data type of its own, but a module named `datetime` can be imported to work with the date as well as time. Datetime module comes built into Python, so there is no need to install it externally.

To get both current date and time `datetime.now()` function of `datetime` module is used. This function returns the current local date and time.

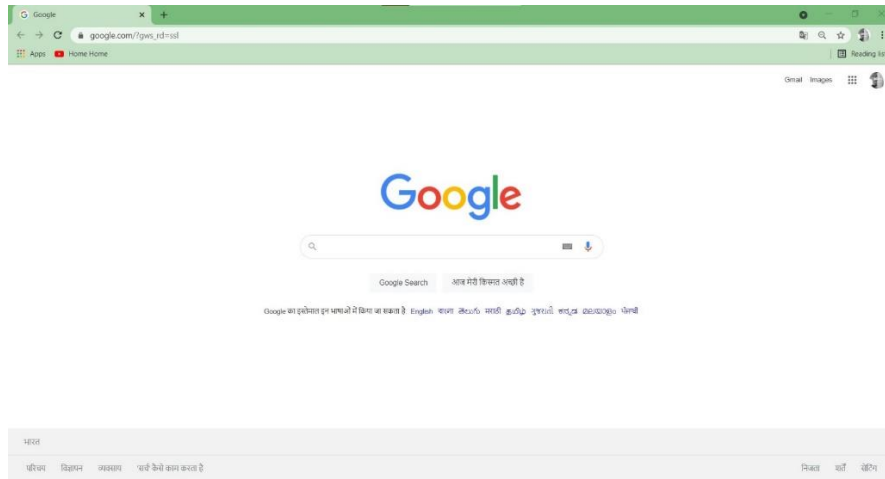


Launch a Web Browser using web browser module

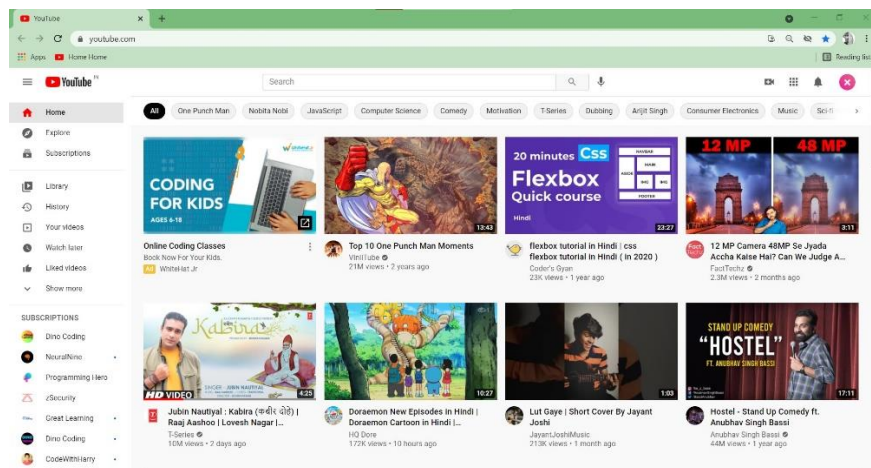
In Python, **web browser module** provides a high-level interface which allows displaying Web-based documents to users. The web browser module can be used to launch a browser in a platform-independent manner as shown below:



Open google



Open You-tube



Python Script to create random jokes using pyjokes

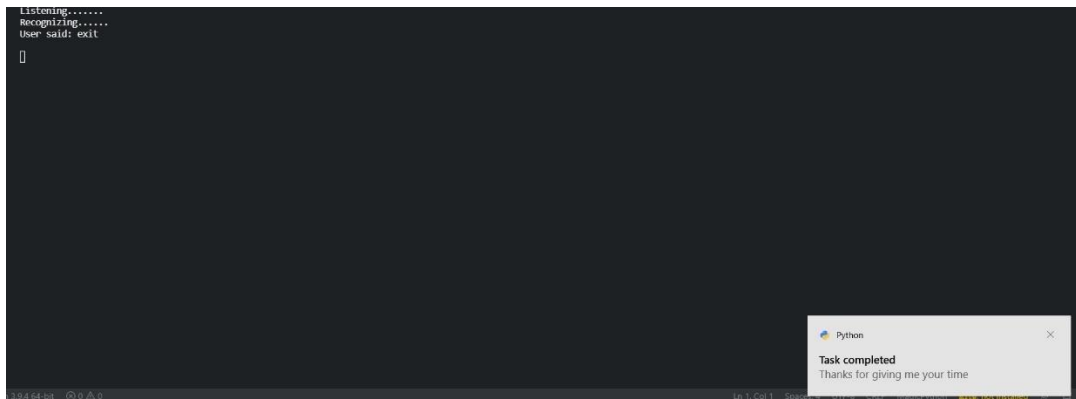
Python supports creation of random jokes using one of its libraries. Let us explore it a little more, **Pyjokes** is a python library that is used to create one-line jokes for programmers. Informally, it can also be referred as a fun python library which is pretty simple to use. Let us see how you can actually use it to perform the required task,

```
Listening.....
Recognizing.....
User said: joke
```

```
Complaining about the lack of smoking shelters, the nicotine addicted Python programmers said there ought to be 'spaces for tabs'.
```

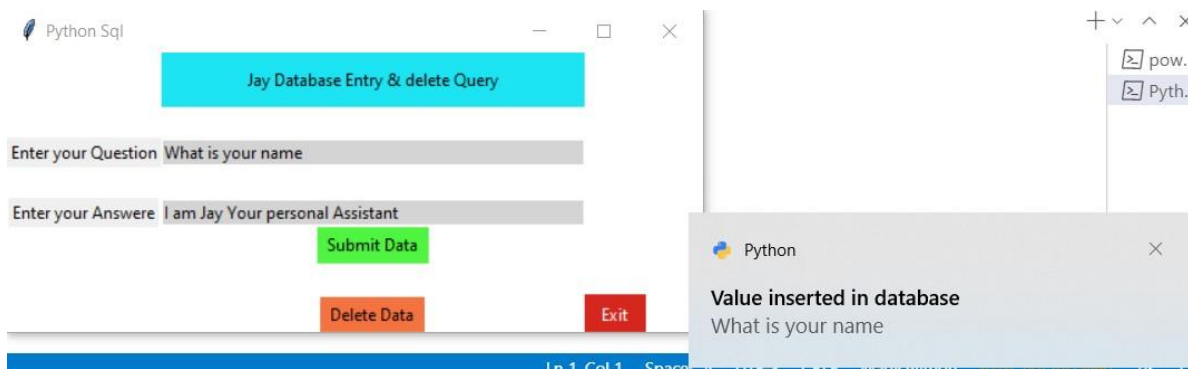
Exit Command

This Command Use for Exit python Program

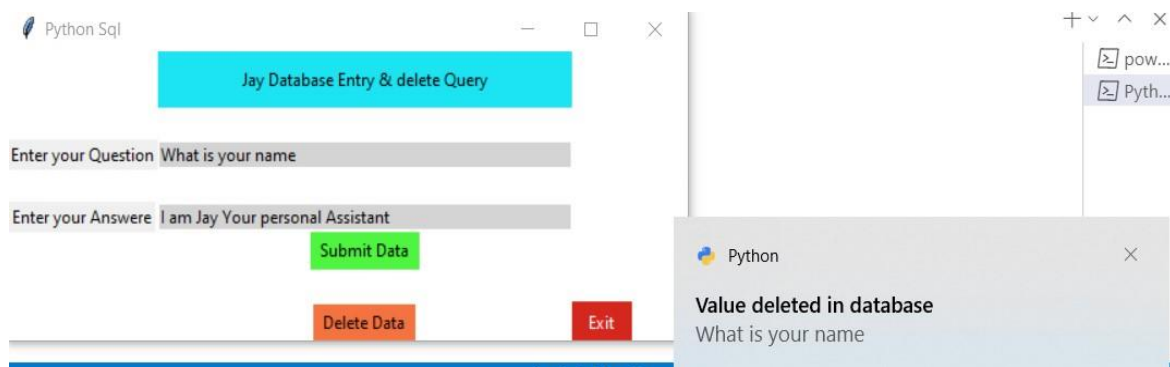


For Database Entry and Delete Python code with Gui

Entry data



Delete Data



Future Enhancement

- Adding A.I. in Face Recognition
- Advanced API Connection
- Making Powerful GUI
- Making security

6. Source Code

```
from Jarvis import JarvisAssistant
import datetime
import sys
from PyQt5 import QtWidgets, QtCore, QtGui
from PyQt5.QtCore import QTimer, QTime, QDate, Qt
from PyQt5.QtGui import QMovie
from PyQt5.QtCore import *
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
from PyQt5.uic import loadUiType
from Jarvis.features.gui import Ui_MainWindow
from Jarvis.config import config
from httplib2 import RelativeURIError
import pyttsx3
import speech_recognition as sr
import os
import mysql.connector
import wikipedia
import pyjokes
import pyautogui
import webbrowser
import smtplib
import string as s
import requests
import wolframalpha
import win32com.client as wincl
import cv2
import numpy as np
import face_recognition
from pynotifier import Notification
from requests.api import request
from bs4 import BeautifulSoup
from urllib.request import urlopen
from random import *
from playsound import playsound
from PIL import Image

engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].id)

def speak(audio):
    engine.say(audio)
    engine.runAndWait()

def wishMe(name):
    hour = int(datetime.datetime.now().hour)
    if hour >= 0 and hour < 12:
        speak(f"Good Morning
```

```

        {name} !""")
elif hour >= 12 and hour < 18:
    speak(f""""Good Afternoon
        {name} !""")
else:
    speak(f""""Good Evening
        {name} !""")
    speak("I am Jay , your personal Assistant \t what can i help you")

def noti(notifi):
    Notification(title='Task completed', description=notifi,duration=5,
urgency='normal').send()

def takeCommand():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening.....")
        r.pause_threshold = 1
        audio = r.listen(source)
    try:
        print("Recognizing.....")
        query = r.recognize_google(audio, language='en-in')
        print(f"User said: {query}\n")
    except Exception as e:
        print(e)
        print("Sorry sir please can you say that again..")
        return 'None'
    return query

# email sending
def sendEmail(to, content):
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    server.login('jayvoiceassistant@gmail.com', 'JayVoiceAssistant123')
    server.sendmail('jayvoiceassistant@gmail.com', to, content)
    server.close()

# searching data on google
def speakquery(answe):
    con = mysql.connector.connect(
        host="localhost", user="root", passwd="", database="voice")
    mycursor = con.cursor()
    sql = "select * from `voicequery` where `query` like '" + answe + "';"
    mycursor.execute(sql)
    myresult = mycursor.fetchall()
    for x in myresult:
        print(x[2])
        speak(x[2])

# simple Google search
def googlesearch(query):
    speak(f'searching {query} on google')

```

```

webbrowser.open("https://www.google.com/search?q="+query+"&rlz=1C1CHZN_enIN949IN949&oq="+query +
                "&aqs=chrome..69i57j0i131i433j0i433j0i131i433l3j0i433j0i131i433j0i433j0.2513j0j15&sourceid=chrome&ie=UTF-8")

# encoding the faces
def faceEncodings(images):
    encodeList = []
    for img in images:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        encodeList.append(encode)
    return encodeList

# save the user Data
def attendance1(name):
    with open('Attendance.csv', 'r+') as f:
        myDataList = f.readlines()
        nameList = []
        for line in myDataList:
            entry = line.split(',')
            nameList.append(entry[0])
        if name not in nameList:
            time_now = datetime.datetime.now()
            tStr = time_now.strftime('%H:%M:%S')
            dStr = time_now.strftime('%d/%m/%Y')
            speak(f"Welcome {name}")
            f.writelines(f'\n{name},{tStr},{dStr}')

def check1():
    cap = cv2.VideoCapture(0)
    while True:
        ret, frame = cap.read()
        faces = cv2.resize(frame, (0, 0), None, 0.25, 0.25)
        faces = cv2.cvtColor(faces, cv2.COLOR_BGR2RGB)

        facesCurrentFrame = face_recognition.face_locations(faces)
        encodesCurrentFrame = face_recognition.face_encodings(
            faces, facesCurrentFrame)

        for encodeFace, faceLoc in zip(encodesCurrentFrame, facesCurrentFrame):
            matches = face_recognition.compare_faces(encodeListKnown, encodeFace)
            faceDis = face_recognition.face_distance(encodeListKnown, encodeFace)
            # print(faceDis)
            matchIndex = np.argmin(faceDis)
            if matches[matchIndex] != True:
                break
            if matches[matchIndex]:
                name = personNames[matchIndex].upper()
                # print(name)
                y1, x2, y2, x1 = faceLoc
                y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4
                cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
                cv2.rectangle(frame, (x1, y2 - 35), (x2, y2),

```

```

        (0, 255, 0), cv2.FILLED)
    cv2.putText(frame, name, (x1 + 6, y2 - 6),
                 cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 2)
    attendance1(name)
    cv2.imshow('Webcam', frame)
    if cv2.waitKey(1) == 13:
        pyautogui.press("enter")
        break
cap.release()
cv2.destroyAllWindows()

```

```

class MainThread(QThread):

```

```

    def run(self):
        self.TaskExecution()

```

```

    def TaskExecution(self):
        path = 'images'
        images = []
        personNames = []
        myList = os.listdir(path)
        print(myList)
        for cu_img in myList:
            current_Img = cv2.imread(f'{path}/{cu_img}')
            images.append(current_Img)
            personNames.append(os.path.splitext(cu_img)[0])
            print(personNames)
            encodeListKnown = faceEncodings(images)
        print('All Encoding are Completed!!!')

```

```

        wishMe("Krishna")

```

```

        while True:
            query = takeCommand().lower()
            speakquery(query)

            if 'where is' in query:
                query = query . replace('where is', '')
                speak(f'serching {query} on maps ')
                webbrowser.open("https://www.google.co.in/maps/place/" + query+'')
                noti(f"Serching {query}")

```

```

            elif 'search' in query:
                query = query . replace('search', '')
                googlesearch(query)
                noti("task completed")

```

```

        #Taking screen shot

```

```

        elif 'capture' in query:
            im1 = pyautogui.screenshot("hello.jpg")
            noti("Image are capture")

```

```

#show image
    elif 'show image' in query:

        # open method used to open different extension image file
        im = Image.open(r"hello.jpg")
        # This method will show image in any image viewer
        im.show()

# Maps
    elif 'where is' in query:
        query = query . replace('where is', '')
        speak(f'Serching {query} on maps ')
        webbrowser.open("https://www.google.co.in/maps/place/" + query+'')
        noti(f"Serching {query}")

# serch on wikipedia
    elif 'wikipedia' in query or 'tell me about' in query:
        query = query . replace("wikipedia", "")
        query = query . replace("tell me about", "")
        speak('Searching Wikipedia...')
        noti(f'{query} Searching')
        results = wikipedia.summary(query, sentences=2)
        speak("According to Wikipedia")
        print(results)
        speak(results)

# show time
    elif 'time' in query:
        strTime = datetime.datetime.now().strftime("%H:%M:%S")
        noti(strTime)
        speak(f"Sir, the time is {strTime}")

# play music
    elif 'play music' in query or 'music' in query:
        noti("Playing music")
        music_dir = 'D:\\Major Project\\music'
        songs = os.listdir(music_dir)
        os.startfile(os.path.join(music_dir, songs[0]))

# showing IP address
    elif "ip" in query:
        ip = requests.get('https://api.ipify.org').text
        print(ip)
        speak(f"Your ip address is {ip}")

# switch the window
    elif "switch the window" in query or "switch window" in query:
        speak("Okay sir, Switching the window")
        pyautogui.keyDown("alt")
        pyautogui.press("tab")
        pyautogui.keyUp("alt")

# serching on youtube

```



```

elif 'play' in query:
    query = query . replace("play", "")
    speak("serching on youtube")
    webbrowser.open("https://www.youtube.com/results?search_query="+query)

# python jokes
elif 'jokes' in query or 'jokes' in query:
    a = pyjokes.get_joke()
    print(a)
    speak(a)

# sending Mail to contact
elif 'send email to' in query:
    # query=query.replace('send email to ', '')
    # name=query
    # receiver=email_list[name]
    try:
        speak("What should I say?")
        content = takeCommand()
        jay = 'This E-mail has been genrated by JayVoice Assistant'
        receiver = 'krishvishwa888@gmail.com'
        contactJay = (f'{jay}\n \t{content}\n')
        sendEmail(receiver, contactJay)
        noti("Email has been send")
        speak("Email has been sent!")
    except Exception as e:
        print(e)
        speak("Sorry sir . I am not able to send this email")

# wether API
elif 'weather' in query:
    query = query . replace("weather", "")
    city = query
    url = "https://www.google.com/search?q="+ "weather"+city
    html = requests.get(url).content
    soup = BeautifulSoup(html, 'html.parser')
    temp = soup.find(
        'div', attrs={'class': 'BNeawe iBp4i AP7Wnd'}).text
    str = soup.find('div', attrs={'class': 'BNeawe tAd8D AP7Wnd'}).text
    data = str.split('\n')
    time = data[0]
    sky = data[1]
    final_response = f""""sir currently {city} wether is {sky} with the
temperature of {temp} """
    print(final_response)
    speak(final_response)
    noti(f"{city} Current Temprature is {temp}")

# exit the programm
elif 'exit' in query:
    noti("Thanks for giving me your time")
    speak("Thanks for giving me your time")
    exit()

```

```

# wolfram alfa library
elif 'what is' in query or "who is" in query:
    client = wolframalpha.Client("VYGWWR-X682V98YHT")
    res = client.query(query)
    try:
        print(next(res.results).text)
        speak(next(res.results).text)
    except StopIteration:
        print("No results")

# write a notes
elif 'write a note' in query:
    speak("What should i write, sir")
    note = takeCommand()
    file = open('voice.txt', 'w')
    file.write(note)

# showing the notes
elif "show note" in query:
    speak("Showing Notes")
    file = open("voice.txt", "r")
    print(file.read())
    speak(file.read(6))

# opening apps using db
elif 'open ' in query:
    query = query . replace('open ', '')
    con = mysql.connector.connect(host="localhost", user="root", passwd="",
database="voice")
    mycursor = con.cursor()
    sql2 = "select * from `voiceweb` where `search` like '"+query+"';"
    mycursor.execute(sql2)
    myresult2 = mycursor.fetchall()
    for x in myresult2:
        print(x[2])
        webbrowser.open_new(x[2])

# security Mode for motion detaction
elif 'security mode' in query:
    cam = cv2.VideoCapture(1)
    while cam.isOpened():
        ret, frame1 = cam.read()
        ret, frame2 = cam.read()
        diff = cv2.absdiff(frame1, frame2)
        gray = cv2.cvtColor(diff, cv2.COLOR_RGB2GRAY)
        blur = cv2.GaussianBlur(gray, (5, 5), 0)
        _, thresh = cv2.threshold(blur, 20, 255, cv2.THRESH_BINARY)
        dilated = cv2.dilate(thresh, None, iterations=3)
        contours, _ = cv2.findContours(
            dilated, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
        cv2.drawContours(frame1, contours, -1, (0, 255, 0), 2)
        for c in contours:
            if cv2.contourArea(c) < 5000:
                continue

```

```

        x, y, w, h = cv2.boundingRect(c)
        cv2.rectangle(frame1, (x, y), (x+w, y+h), (200, 255, 0), 2)
        speak("Security alert Some one is found on camera")
        try:
            receiver = 'krishvishwa888@gmail.com'
            contactJay = 'This email has been generated automatically \n
Security alert Some one is found on camera'
            sendEmail(receiver, contactJay)
        except Exception as e:
            print("Email Has some problem")
        # playsound('D:\\final year\\voiceassistant\\alert.wav')
        if cv2.waitKey(10) == ord('q'):
            break
        cv2.imshow('Jay Security Eye', frame1)

    elif 'normal check' in query or 'who is there' in query or 'check' in query:
        check1()

startExecution = MainThread()

# Gui code is here
class Main(QMainWindow):
    def __init__(self):
        super().__init__()
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)
        self.ui.pushButton.clicked.connect(self.startTask)
        self.ui.pushButton_2.clicked.connect(self.close)

    def __del__(self):
        sys.stdout = sys.__stdout__

    # def run(self):
    #     self.TaskExection
    def startTask(self):
        self.ui.movie = QtGui.QMovie("Jarvis/utils/images/live_wallpaper.gif")
        self.ui.label.setMovie(self.ui.movie)
        self.ui.movie.start()
        self.ui.movie.start()
        startExecution.start()

app = QApplication(sys.argv)
jarvis = Main()
jarvis.show()
exit(app.exec_())

```

7 Testing

Software testing is a process, to evaluate the functionality of a software application with an intent to find whether the developed software met the specified requirements or not and to identify the defects to ensure that the product is defect free in order to produce the quality product.

Software Testing Types:

Manual Testing: Manual testing is the process of testing software by hand to learn more about it, to find what is and isn't working. This usually includes verifying all the features specified in requirements documents, but often also includes the testers trying the software with the perspective of their end user's in mind. Manual test plans vary from fully scripted test cases, giving testers detailed steps and expected results, through to high-level guides that steer exploratory testing sessions. There are lots of sophisticated tools on the market to help with manual testing, but if you want a simple and flexible plan to start, take a look at Test pad

Automation Testing: Automation testing is the process of testing the software using an automation tool to find the defects. In this process, testers execute the test scripts and generate the test results automatically by using automation tools. Some of the famous automation testing tools for functional testing are QTP/UFT and Selenium.

6.1 Testing Methods

Static Testing

Dynamic Testing

Static Testing: It is also known as Verification in Software Testing. Verification is a static method of checking documents and files. Verification is the process, to ensure that whether we are building the product right i.e., to verify the requirements which we have and to verify whether we are developing the product accordingly or not.

Dynamic Testing: It is also known as Validation in Software Testing. Validation is a dynamic process of testing the real product. Validation is the process, whether we are building the right product i.e., to validate the product which we have developed is right or not.

Testing Approaches:

There are three types of software testing approaches.

White Box Testing

Black Box Testing

Grey Box Testing

White Box Testing: It is also called as Glass Box, Clear Box, Structural Testing. White Box Testing is based on applications internal code structure. In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases. This testing is usually done at the unit level.

Black Box Testing: It is also called as Behavioural/Specification-Based/Input-Output Testing. Black Box Testing is a software testing method in which testers evaluate the functionality of the software under test without looking at the internal code structure.

Grey Box Testing: Grey box is the combination of both White Box and Black Box Testing. The tester who works on this type of testing needs to have access to design documents. This helps to create better test cases in this process.

Testing Levels

1. Unit Testing
2. Integration Testing
3. System Testing
4. Acceptance Testing

Unit Testing: Unit Testing is done to check whether the individual modules of the source code are working properly. i.e. testing each and every unit of the application separately by the developer in the developer's environment. It is AKA Module Testing or Component Testing

Integration Testing: Integration Testing is the process of testing the connectivity or data transfer between a couple of unit tested modules. It is AKA I&T Testing or String Testing. It is subdivided into Top-Down Approach, Bottom-Up Approach and Sandwich Approach (Combination of Top Down and Bottom Up).

System Testing (end to end testing): It's a black box testing. Testing the fully integrated application this is also called as end to end scenario testing. To ensure that the software works in all intended target systems. Verify thorough testing of every input in the application to check for desired outputs. Testing of the users experiences with the application.

Acceptance Testing: To obtain customer sign-off so that software can be delivered and payments received. Types of Acceptance Testing are Alpha, Beta & Gamma Testing.

Principles of software testing

Software testing is a process of executing a program with the aim of finding the error. To make our software perform well it should be error free. If testing is done successfully it will remove all the errors from the software.

There are seven principles in software testing:

1. Testing shows presence of defects
2. Exhaustive testing is not possible
3. Early testing
4. Defect clustering
5. Pesticide paradox
6. Testing is context dependent
7. Absence of errors fallacy

Testing shows presence of defects: The goal of software testing is to make the software fail. Software testing reduces the presence of defects. Software testing talks about the presence of defects and doesn't talk about the absence of defects. Software testing can ensure that defects are present but it can not prove that software is defects free. Even multiple testing can never ensure that software is 100% bug-free. Testing can reduce the number of defects but not removes all defects.

Exhaustive testing is not possible: It is the process of testing the functionality of a software in all possible inputs (valid or invalid) and pre-conditions is known as exhaustive testing. Exhaustive testing is impossible means the software can never test at every test cases. It can test only some test cases and assume that software is correct and it will produce the correct output in every test cases. If the software will test every test cases then it will take more cost, effort, etc. and which is impractical.

Early Testing: To find the defect in the software, early test activity shall be started. The defect detected in early phases of SDLC will very less expensive. For better performance of software, software testing will start at initial phase i.e. testing will perform at the requirement analysis phase.

Defect clustering: In a project, a small number of the module can contain most of the defects. Pareto Principle to software testing state that 80% of software defect comes from 20% of modules.

Pesticide paradox: Repeating the same test cases again and again will not find new bugs. So it is necessary to review the test cases and add or update test cases to find new bugs.

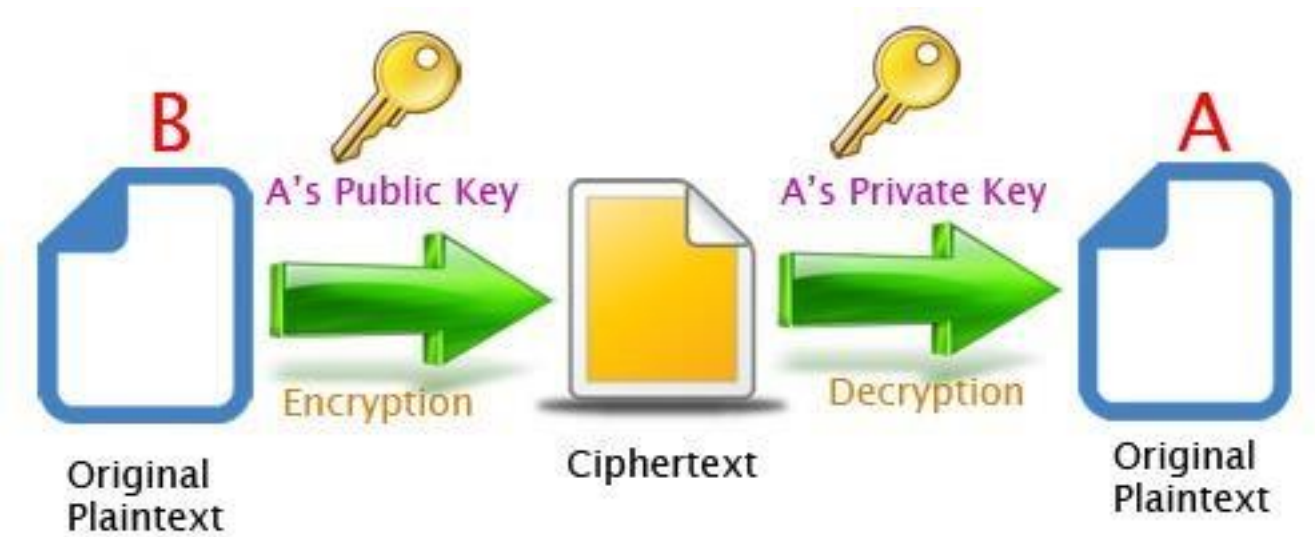
Testing is context dependent: Testing approach depends on context of software developed. Different types of software need to perform different types of testing. For example, The testing of the e-commerce site is different from the testing of the Android application.

Absence of errors fallacy: If a built software is 99% bug-free but it does not follow the user requirement then it is unusable. It is not only necessary that software is 99% bug-free but it also mandatory to fulfill all the customer requirements.

6.2 SAMPLE TEST CASE

Introduction to encryption

Encryption is basically the method of disguising plain or clear text in such a way as to hide its contents from anyone for whom it is not intended.



A cryptographic algorithm, or cipher, is a mathematical function used during encryption and decryption and works in combination with a specific key. Different keys will encrypt the same plaintext to a different cipher text. Public Key Encryption

Both PGP and GPG use public-key cryptography. In a public-key system, each user has a pair of keys consisting of a private key and a public key. The private key is to be kept secret to the user and should never be revealed and the public key can be given to anyone with whom the user wants to communicate. The public key can be thought of as an open safe. When you encrypt a document using a public key it is like putting it in the safe, shutting it and spinning the combination lock several times. The corresponding private key is the combination to open the safe. Therefore only the person who holds the private key can recover the document encrypted using the associated public key.

GPG vs PGP

PGP (short for Pretty Good Privacy) is a public key encryption program designed to deliver automated encryption services to a number integrated applications. GPG is basically a free tool that provides the same functionality as PGP. Keys created in one can be used in the other and similarly files encrypted using one can

be decrypted using the other. GPG is, by default, a command line tool. There are graphical interfaces available for it but it is not necessary to install them for this test. For the purpose of this test we are going to use PGP and GPG separately.

Passphrases

In order to unlock your private key you must create a passphrase. There is no limit on the length of a passphrase, and it should be carefully

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test.[1] Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

meets the requirements that guided its design and development, responds correctly to all kinds of inputs, performs its functions within an acceptable time, it is sufficiently usable, can be installed and run in its intended environments, and achieves the general result its stake

Software Testing Methods

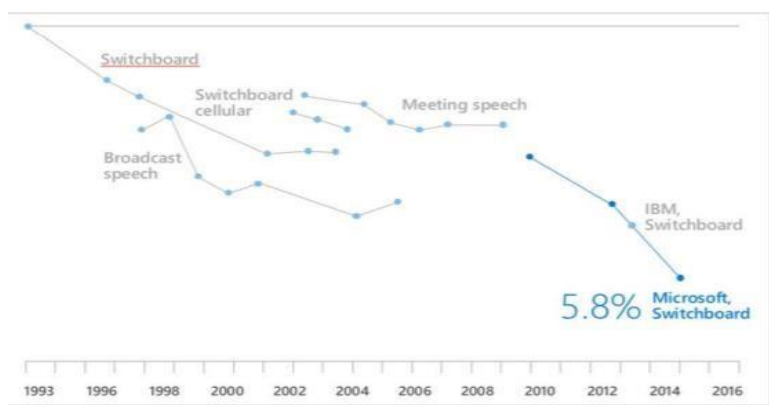
Black box and white box testing are the two fundamental methods for judging product behaviour and **performance**, but there are other methods as well.

6.3 Common Errors Found

Definition

Every programmer encounters errors, both those who are just beginning, and those who have been programming for years. Encountering errors and exceptions can be very frustrating at times, and can make coding feel like a hopeless endeavour. However, understanding what the different types of errors are and when you are likely to encounter them can help a lot. Once you know why you get certain types of errors, they become much easier to fix.

Error rate for speech recognition system



This particular traceback has two levels. You can determine the number of levels by looking for the number of arrows on the left hand side. In this case:

The first shows code from the cell above, with an arrow pointing to Line 8 (which is `favorite_ice_cream()`).

The second shows some code in the function `favorite_ice_cream`, with an arrow pointing to Line 6 (which is `print(ice_creams[3])`).

The last level is the actual place where the error occurred. The other level(s) show what function the program executed to get to the next level down. So, in this case, the program first performed a function call to the function `favorite_ice_cream`. Inside this function, the program encountered an error on Line 6, when it tried to run the code `print(ice_creams[3])`.

Long Tracebacks

Sometimes, you might see a traceback that is very long – sometimes they might even be 20 levels deep! This can make it seem like something horrible happened, but the length of the error message does not reflect severity, rather, it indicates that your program called many functions before it encountered the error. Most of the time, the actual place where the error occurred is at the bottom-most level, so you can skip down the traceback to the bottom.

So what error did the program actually encounter? In the last line of the traceback, Python helpfully tells us the category or type of error (in this case, it is an `Index Error`) and a more detailed error message (in this case, it says “list index out of range”).

If you encounter an error and don't know what it means, it is still important to read the traceback closely. That way, if you fix the error, but encounter a new one, you can tell that the error changed. Additionally, sometimes knowing where the error occurred is enough to fix it, even if you don't entirely understand the message.

8. Conclusion

Conclusion

Thus we conclude by saying that "PERSONAL ASSISTENT" provides new facility to the user and also helps them to save time and money which can be more improved in the future.

The more a person interacts with voice-activated devices, the more trends, and patterns the system identifies based on the information it receives. Then, this data can be utilized to determine user preferences and tastes, which is a long-term selling point for making a home smarter. Google and Amazon are looking to integrate voice-enabled artificial intelligence capable of analyzing and responding to human emotion.

Shivi is fully working Virtual Personal Assistance, which can perform task in offline condition as we given her local modules to her. In online condition Shivi gets more resources to work with. Also, any peripheral which is connected with the raspberry pi is can be control with the Virtual Personal Assistance, just by giving the command. The local modules can be added or removed by user as user sees fit. Also, there is simple option for conversation

9. REFERENCE

Websites referred

- www.stackoverflow.com
- www.google.com
- www.geeksforgeeks.com
- www.python.org
- www.w3school.com
- <https://opencv.org>

YouTube Channels referred

- Codewithharry
- Great learning