

PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

PNR No.: 2018016402335721

Roll no: 331

1. Name of the Student

Tejas Sakunde

2. Title of the Project

voice assistant

3. Name of the Guide

Priya Maurya

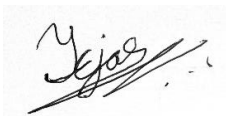
4. Teaching experience of the Guide

5. Is this your first submission?

Yes



No



Signature of the Student

Signature of the Guide

Date:

.....

Date:

Signature of the

Coordinator Date:

.....

Voice Assistant

A Project Report

Submitted in partial fulfilment of the

Requirements for the award of the Degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

By

Tejas Sakunde

Seat Number: 331

Under the esteemed guidance of

Mrs. Priya Maurya



DEPARTMENT OF INFORMATION TECHNOLOGY

**Shri Rajasthani Seva Sangh's SMT. PARMESHWARIDEVI
DURGADUTT TIBREWALA LIONS JUHU COLLEGE OF ARTS,
COMMERCE & SCIENCE**

(Affiliated to University of Mumbai)

J.B. NAGAR ANDHERI (EAST) MUMBAI, 400059

MAHARASHTRA2020-2021

**Shri Rajasthani Seva Sangh's SMT. PARMESHWARIDEVI
DURGADUTT TIBREWALA LIONS JUHU COLLEGE OF ARTS,
COMMERCE & SCIENCE**

(Affiliated to University of Mumbai)

J.B. NAGAR ANDHERI (EAST) MUMBAI-MAHARASHTRA-400059

DEPARTMENT OF INFORMATION TECHNOLOGY



CERTIFICATE

This is to certify that the project entitled, “ **VOICE ASSISTANT**”, is bonafied work of **Tejas Sakunde** bearing Seat.No: 331 submitted in partial fulfilment of the requirements for the award of degree of **BACHELOR OF SCIENCE** in **INFORMATION TECHNOLOGY** from **University of Mumbai**.

Internal Guide

Coordinator

External Examiner

Date:

College Seal

Abstract

Voice assistants are software agents that can interpret human speech and respond via synthesized voices. Apple's Siri, Amazon's Alexa, Microsoft's Cortana, and Google's Assistant are the most popular voice assistants and are embedded in smartphones or dedicated home speakers. 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. This column will explore the basic workings and common features of today's voice assistants. It will also discuss some of the privacy and security issues inherent to voice assistants and some potential future uses for these devices. As voice assistants become more widely used, librarians will want to be familiar with their operation and perhaps consider them as a means to deliver library services and materials.

ACKNOWLEDGEMENT

I, TEJAS SAKUNDE is student of Shri Rajasthani Seva Sangh's Smt. Parmeshwaridevi Durgadutt Tibrewala Lions Juhu College of Arts, Commerce & Science, J.B. Nagar Andheri (E), Mumbai 400059, studying in T.Y.B.Sc.Information Technology would like to thank the Principal Dr. (Mrs.) Trishla Mehta and the Coordinator Prof. (Mr.) Mukesh Sharma.

It has always been my sincere desire as a Information Technology student to get an opportunity to express our views, skills, attitude and talent in which we are proficient. A project is one such avenue through which a student who aspires to be a future developer does something creative. This project has given us the chance to get in touch with practical aspects of Information Technology.

We are extremely grateful to the University of Mumbai for having prescribed this project work to us as a part of the academic requirement in the Bachelor of Information Technology (B.Sc) Course.

We wish to appreciate to the Management and Staff of Smt. Parmeshwaridevi Durgadutt Tibrewala Lions Juhu College of Arts, Commerce & Science, Information Technology for providing the entire state of the art infrastructure and resources to enable the completion and enrichment of my project.

We wish to extend a special thanks to my project guide Prof. **Priya Akhilesh Maurya** without whose guidance the project may not have taken shape.

Finally, we than all my friends and family members who have directly or indirectly help me towards the execution of this project.

DECLARATION

I here by declare that the project entitled, “**Voice Assistant**” done at Shri RAJASTHANI SEVA SANGH’S SMT.PARMESHWARIDEVEI DURGADUTT TIBERWALA LIONS JUHU COLLEGE OF ARTS, COMMERCE & SCIENCE of Information Technology, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

**TEJAS SANTOSH
SAKUNDE**

A handwritten signature in black ink, appearing to read 'Tejas', with a horizontal line drawn through it.

**Name and Signature of the
Student**

INDEX

Contents

CHAPTER 1 :- INTRODUCTION	2
1.1 Background	2
1.2 Objectives.....	3
1.3 Scope and Purpose.....	3
1.3.1 Scope.....	3
1.3.2 Purpose	3
1.3.3 Scope.....	3
1.4 Applicability.....	4
CHAPTER 2:- SURVEY OF TECHNOLOGY	7
1.Python	7
2.Pip	7
2.1.Speech recognition engine/API support:.....	7
2.2.Wikipedia	8
2.3.Pygame.....	8
2.4.Pyttsx3.....	8
2.5.Random	8
3.MYSQL	8
4. StarUML	8
CHAPETR 3:- REQUIREMENTS AND ANALYSIS.....	10
3.1 Problem Definition	10
3.2 Requirements Specification	10
3.3 Planning and Scheduling	10
3.4 Software and Hardware Requirements	11
3.5 Preliminary Product Description	11
3.6 CONCEPTUAL MODELS.....	12
WATERFALL MODEL	12
Chapter 4 :- SYSTEM DESIGN	15
4.1 BASIC MODULES.....	15
4.2 Data Design	15
4.2.1 SCHEMA DESIGN	15
4.2.2 DATA ITEGRITY AND CONSTRAINTS	17

4.3 Procedural Design	18
4.3.1 Logic Diagrams	19
4.3.2 Data Structures	19
4.3.3 Algorithms Design	20
4.3.4 Flow Chart	20
4.3.5 Use Case Diagram	21
4.4 User interface design	22
4.5 SECURITY ISSUES	22
4.6 Test Cases Design	23
CHPATER 5 :- Implementation and Testing	26
5.1 Implementation Approaches	26
5.1.1 Star Uml	26
5.1.2 Windows 10	26
5.1.3 Vs Code	27
5.2 Coding Details and Code Efficiency	27
5.2.1 Coding Details	27
5.2.2 Code Efficiency	36
5.3 TESTING APPROACHES	40
5.3.1 Unit Testing	40
5.3.2 Integration Testing	41
5.33 Beta Testing	41
CHAPTER 6 :- Results and Discussion	44
6.1 Test Results	44
Main Window	44
Respond Window	45
Opening Google in Web Browser	45
Opening YouTube in Web Browser	46
Playing Music	46
Opening Google Chrome	47
Opening Gmail in Web Browser	47
Searching on Web for given Query	48
Saving Picture in database	48
Saving Picture from Database	49
Saving log Data in Database	50
6.2 USER DOCUMENTATION	50
Chapter 7:- Conclusions and REFERENCES	51

Chapter 7:- Conclusions and REFERENCES.....	52
7.1 Conclusions	52
7.2 FUTURE WORK	52
7.3 REFERENCES	52

CHAPTER 1 :- INTRODUCTION

CHAPTER 1 :- INTRODUCTION

An intelligent virtual assistant (IVA) or intelligent personal assistant (IPA) is a software agent that can perform tasks or services for an individual based on commands or questions. Sometimes the term "chatbot" is used to refer to virtual assistants generally or specifically accessed by online chat. In some cases, online chat programs are exclusively for entertainment purposes. Some 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 (spoken?) commands. A similar concept, however with differences, lays under the dialogue system.

For most of us, the ultimate luxury would be an assistant who always listens for your call, anticipates your every need, and takes action when necessary. That luxury is now available thanks to artificial intelligence assistants, aka voice assistants.

Voice assistants come in somewhat small packages and can perform a variety of actions after hearing a wake word or command. They can turn on lights, answer questions, play music, place online orders, etc.

Voice assistants are not to be confused with virtual assistants, which are people who work remotely and can therefore handle all kinds of tasks. Rather, voice assistants are technology based. As voice assistants become more robust, their utility in both the personal and business realms will grow as well.

1.1 Background

With the emergence of virtual assistant, the technology opens up the possibility of doing work with only voice command.

The virtual assistant help user to do work without even touching it or even going closer to it. User can do anything from sending email to lighting up the room with the voice command.

The virtual assistant has a password which can prevent any other user from doing malicious things and make sure no data can get to the not authorized person.

1.2 Objectives

The virtual assistant is a software that can work as your personal assistant that allow user to open application, play music, give weather forecast, tells latest news, search on Wikipedia, etc. The system can without internet leaving the task that requires internet like search something on google or playing you tube videos. The system has password due to which only the genuine person can use the system.

1.3 Scope and Purpose

1.3.1 Scope

- It take audio input from user and perform task
- It is time effective
- The user can connect it to the house and make it a smart home
- It is light weight software

1.3.2 Purpose

Voice assistants can make calls, send text messages, look things up online, provide directions, open apps, set appointments on our calendars, and initiate or complete many other tasks.

With the addition of separate apps on the phone, our voice can be a type of remote control for our lives. We can unlock cars and homes, turn on lights, adjust the thermostat, change the television channel, and much more.

1.3.3 Scope

Assistant for your home

Devices are connected to all other electronic devices and instruct them to do the work on your place. They instantly carry out the functions of the devices. For example if you command a virtual assistant to hear a song of your choice it will let it play.

Voice assistants in hospitality

Few of the functions which are carried out by voice assistants are

- Helps in managing hotels reservations and bookings.
- Works as a personal assistant in the room to the customer
- Carry out Itinerary, see local events, sight seeing ideas.
- Enable help in housekeeping, room service and other requests

Provide financial work solutions

- Make payments, transfer money as directed
- Give details to you for your savings required for a month.
- Pay your bills on time if directed.
- Provide you the information of best investment plans by comparing others.
- Make a saving plan for retirement.

1.4 Applicability

Virtual assistants may be integrated into many types of platforms or, like Amazon Alexa, across several of them:

- Into devices like smart speakers such as Amazon Echo, Google Home and Apple HomePod
- In instant messaging apps on both smartphones and via the Web, e.g. Facebook's M (virtual assistant) on both Facebook and Facebook Messenger apps or via the Web
- Built into a mobile operating system (OS), as are Apple's Siri on iOS devices and BlackBerry Assistant on BlackBerry 10 devices, or into a desktop OS such as Cortana on Microsoft Windows OS
- Built into a smartphone independent of the OS, as is Bixby on the Samsung Galaxy S8 and Note 8
- Within instant messaging platforms, assistants from specific organizations, such as Aeromexico's Aerobot on Facebook
- Within mobile apps from specific companies and other organizations, such as Dom from Domino's Pizza
- In appliances, cars, and wearable technology.¹

- Previous generations of virtual assistants often worked on websites, such as Alaska Airlines' Ask Jenn,^{[1](#)} or on interactive voice response (IVR) systems such as American Airlines' IVR by Nuance.

CHAPTER 2:- SURVEY OF TECHNOLOGY

CHAPTER 2:- SURVEY OF TECHNOLOGY

1.Python

Python is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects

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.

Python was created in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system with reference counting

2.Pip

2.1.Speech recognition engine/API support:

- CMU Sphinx (works offline)
- Google Speech Recognition
- Google Cloud Speech API
- Wit.ai
- Microsoft Bing Voice Recognition
- Houndify API
- IBM Speech to Text
- Snowboy Hotword Detection (works offline)

2.2.Wikipedia

Wikipedia is a Python library that makes it easy to access and parse data from Wikipedia. Search Wikipedia, get article summaries, get data like links and images from a page, and more. Wikipedia wraps the MediaWiki API so you can focus on using Wikipedia data, not getting it.

2.3.Pygame

Pygame is a Python wrapper module for the SDL multimedia library. It contains python functions and classes that will allow you to use SDL's support for playing cdroms, audio and video output, and keyboard, mouse and joystick input.

2.4.Pyttsx3

Pytsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline, and is compatible with both Python 2 and 3.

2.5.Random

This package provides a Python 3 ported version of Python 2.7's random module. It has also been back-ported to work in Python 2.6.

In Python 3, the implementation of randrange() was changed, so that even with the same seed you get different sequences in Python 2 and 3. Note that several high-level functions such as randinit() and choice() use randrange().

In my testing code I heavily rely on stable random generator results and it makes porting code to Python 3 a lot harder, if all those tests have to be adjusted. This package fixes that.

3.MYSQL

MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use.

4. StarUML

StarUML is an open source software modelling tool that supports UML (Unified Modelling Language). It is based on UML version 1.4, provides eleven different types of diagram and it accepts UML 2.0 notation. It actively supports the MDA (Model Driven Architecture) approach by supporting the UML profile concept and allowing to generate code for multiple languages.

CHAPETR 3:- REQUIREMENTS AND ANALYSIS

CHAPETR 3:- REQUIREMENTS AND ANALYSIS

3.1 Problem Definition

The user cannot multi task while doing a specific thing for example:- If the user is writing a word document and wants to play music or send an e-mail he needs to leave the writing in word to do so but it with voice assistant he need to say a command and it will be done without even leaving what you are doing right now.

3.2 Requirements Specification

For the system to run smoothly your device need the following hardware and software requirements

HARDWARE REQUIREMENTS

HARDWARE	MINIMUM REQUIREMENTS
Processor	I3 based system or higher
MEMORY	4 GB RAM or Above
HARD DRIVE	4 GB
INTERNET	YES

SOFTWARE REQUIREMENTS

SOFTWARE	MINIMUM REQUIREMENTS
OPERATING SYSTEM	WINDOWS 7 OR HIGHER
BROWSER	ANY BROWSER CHROME RECOMMENDED
Database Management System	MySQL

3.3 Planning and Scheduling

The activities and considerations included in planning and scheduling a project are intended to provide the Project Manager and the project team members with a systematic approach to organizing, defining, scheduling, tracking and managing a project.

3.4 Software and Hardware Requirements

HARDWARE REQUIREMENTS

HARDWARE	MINIMUM REQUIREMENTS
Processor	I3 based system or higher
MEMORY	4 GB RAM or Above
HARD DRIVE	4 GB
INTERNET	YES

SOFTWARE REQUIREMENTS

SOFTWARE	MINIMUM REQUIREMENTS
OPERATING SYSTEM	WINDOWS 7 OR HIGHER
BROWSER	ANY BROWSER CHROME RECOMMENDED
Database Management System	MySQL

3.5 Preliminary Product Description

Preliminary study is problem solving activity that requires intensive communication between the system user and system developers. It does various feasibility studies. In these studies, a rough figure of system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken

Analysis is a detailed of the various operation performed by the system and the relationship exist between the system. In our project we analysed the relationship that we will be using in our project. In analysis, we also decided how many modules we will be including in our project.

Also, the brief overview of how our project will look like was done in the analyses. We also decided what type of connectivity we will be providing in our project was done. Also, we referred some of the books of python so that which will be useful for writing the code of the

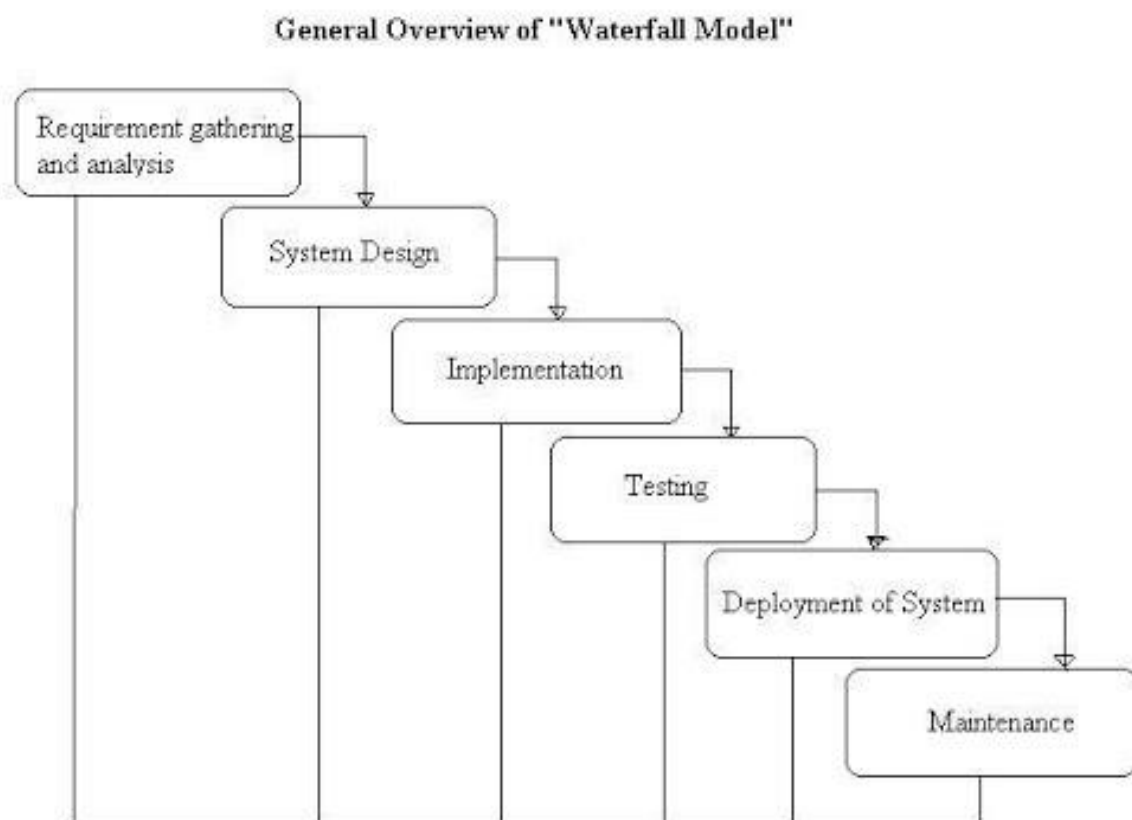
project. In analysis, also we analyse that in how many days we will be completing our project so that it will be submitted in correct time as per the given schedule.

3.6 CONCEPTUAL MODELS

WATERFALL MODEL

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially

The following illustration is a representation of the different phases of the Waterfall Model.



The sequential phases in Waterfall model are –

Requirement Gathering and analysis – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

System Design – The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

Implementation – With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

Integration and Testing – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures

Deployment of system – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

Maintenance – There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

Chapter 4 :- SYSTEM DESIGN

Chapter 4 :- SYSTEM DESIGN

4.1 BASIC MODULES

User Module:-

User can use voice command after giving the specific password.

4.2 Data Design

A database is a system intended to organize, store, and retrieve large amounts of data easily. It consists of an organized collection of data for one or more uses, typically in digital form.

Database design is the process of producing a detailed data model of a database

This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a Data Definition Language, which can then be used to create a database.

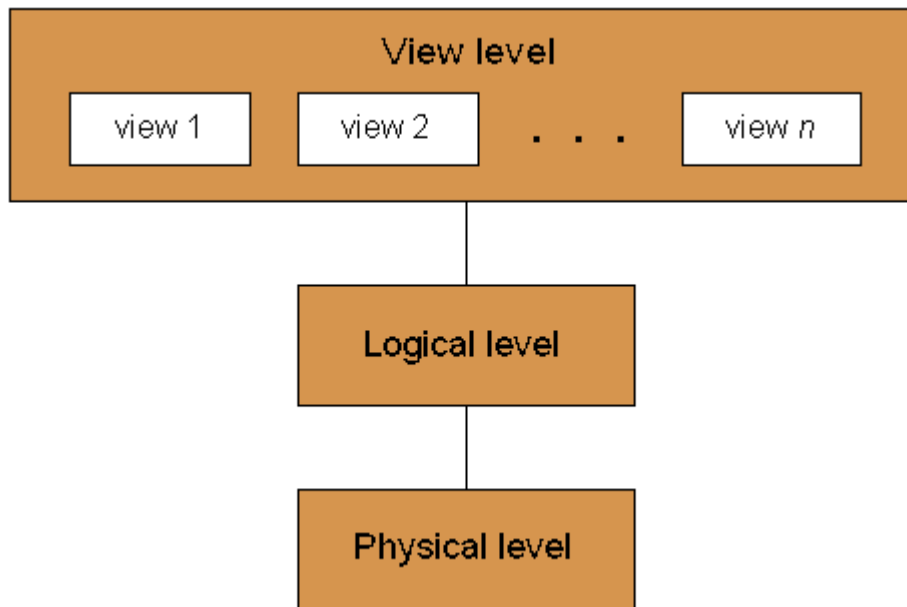
A fully attributed data model contains detailed attributes for each entity

4.2.1 SCHEMA DESIGN

A schema can be defined as the design of a database

The overall description of the database is called the database schema. It can be categorized into three parts. These are:

- Physical Schema
- Logical Schema
- View Schema



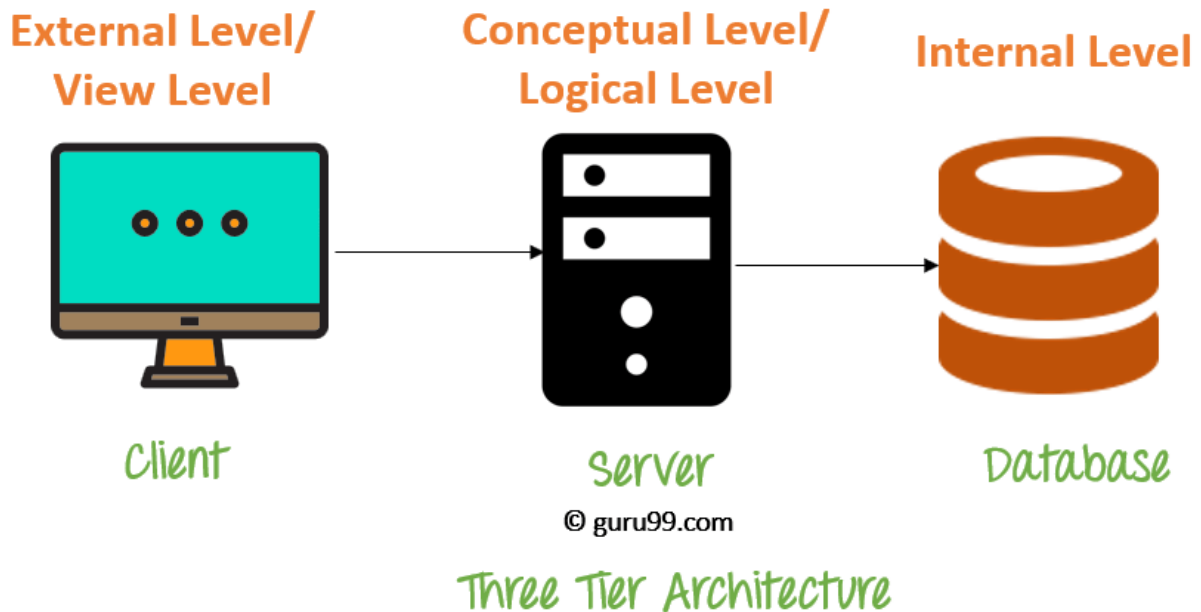
A **physical schema** can be defined as the design of a database at its physical level. In this level, it is expressed how data is stored in blocks of storage.

A **logical schema** can be defined as the design of the database at its logical level.

In this level, the programmers as well as the database administrator (DBA) work.

At this level, data can be described as certain types of data records which can be stored in the form of data structures. However, the internal details (such as an implementation of data structure) will be remaining hidden at this level.

View schema can be defined as the design of the database at view level which generally describes end-user interaction with database systems.



4.2.2 DATA INTEGRITY AND CONSTRAINTS

The Primary-Key Constraint: - The primary key constraint designates a column, or a combination of columns, as the primary for the table. This action enforces entity integrity, which requires each row to have a unique identifier, so that data modifications or queries always refer to a specific row without ambiguity. When you place a primary key constraint on a column, you're requiring each row in that column to have a unique value, which can't be NULL. A primary key constraint on multiple columns requires the combination of the values in those columns to be unique. Examples of primary key include invoice number, employee ID, purchase order number and item or part number.

The Foreign-Key Constraint: - The Foreign-key constraints defines the relationship between a column or combination of columns in the current table and a column or combination of columns in another table. In other words, it enforces referential integrity. This relationship might be one to one, such as in the case of an employee in the payroll table who must already exist in the employees table. Or it could be a many to one relationship. A typical example too many to one foreign-key relationship is the Customer ID in the voices table.

Table 1 :- Describe the Log Records

Column Name	Data Type	Description	Size	Allow Null Value
Query	varchar	User input	50	yes
Time	Varchar	Time of the user input	50	yes

Table 2:- Describe the Photo table

Column Name	Date Type	Description	Size	Allow Null Value
Id	int	Primary key for the Photo record	11	No
file_data	longblob	Photo store by the user	Default	Yes
Date	datetime	Date and time when photo is stored	Default	Yes

4.3 Procedural Design

Component level design also called procedural design occurs after data, architectural, and interface design have been established.

What is it? Info, architectural, and interface style should be translated into operational software. To accomplish this, the design must be represented at a level of abstraction that is close to code

Component level design establishes:

- The algorithmic detail required to manipulate data structures.
- Effect communication between software components via their interfaces, and
- Implement the processing algorithms allocated to each component.

Who does it? A software program engineer performs component level design.

Why is it important? You have to be able to determine whether the program will work before you build it. The component level style represents the software in a way that allows you to review the details of the design for correctness and regularity with earlier design representations. It provides a means intended for assessing whether data constructions, interface and algorithms will continue to work.

What are the steps? Design representations of the data, architecture, and interfaces form the foundation for component level style. The digesting narrative for each component is converted into a procedural design model using a set of structured programming constructs. Graphical, tabular or text-based notation is used to represent the look.

Approach is to represent the procedural style using some intermediate representation that can be translated easily into source code

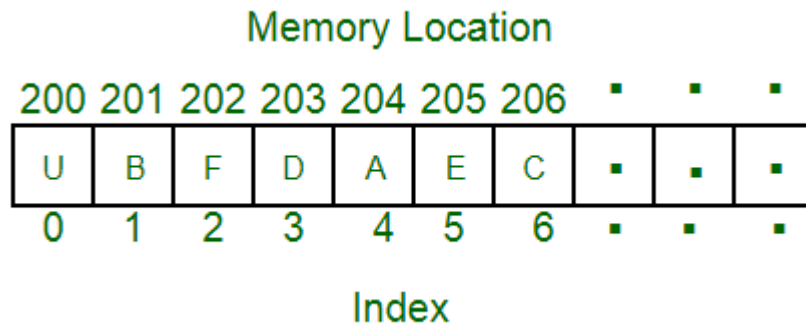
4.3.1 Logic Diagrams

Logic diagrams are presented for the design of pneumatic conveying systems based on the use of both mathematical models and conveying data. Logic diagrams are also presented for checking the performance of an existing system, or for a potential change of duty, again based on the use of both models and data. There is rarely a single solution to the specification of a pneumatic conveying system for a given duty. As a consequence the logic diagrams include numerous checks so that optimum solutions are achieved in terms of either obtaining the minimum power requirement for a given duty, or achieving a maximum material flow rate for the given conveying parameters. To help in this process several series of design curves are included to illustrate the potential influence of the major system variables such as conveying distance, pipeline bore, and air supply pressure, as well as the problematical issue of material type.

4.3.2 Data Structures

A **data structure** is a particular way of organizing data in a computer so that it can be used effectively.

For example, we can store a list of items having the same data-type using the *array* data structure.

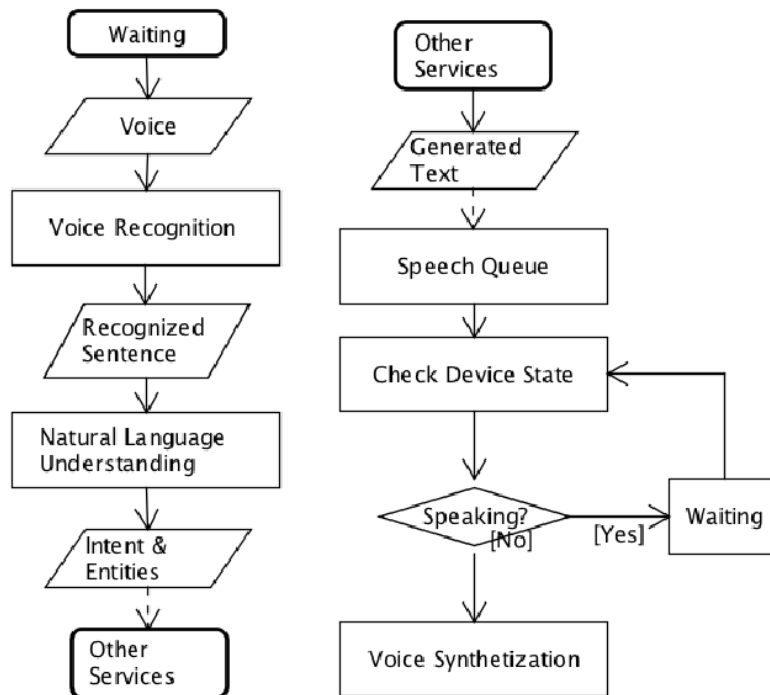


4.3.3 Algorithms Design

- Start the Application
- Click start button
 - If correct
 - Take the command
 - If valid command
 - ❖ Perform operation
 - If not
 - try again
 - If not
 - error

4.3.4 Flow Chart

A **flowchart** is a picture of the separate steps of a process in sequential order. It is a generic tool that can be adapted for a wide variety of purposes, and can be used to describe various processes, such as a manufacturing process, an administrative or service process, or a project plan.



4.3.5 Use Case Diagram

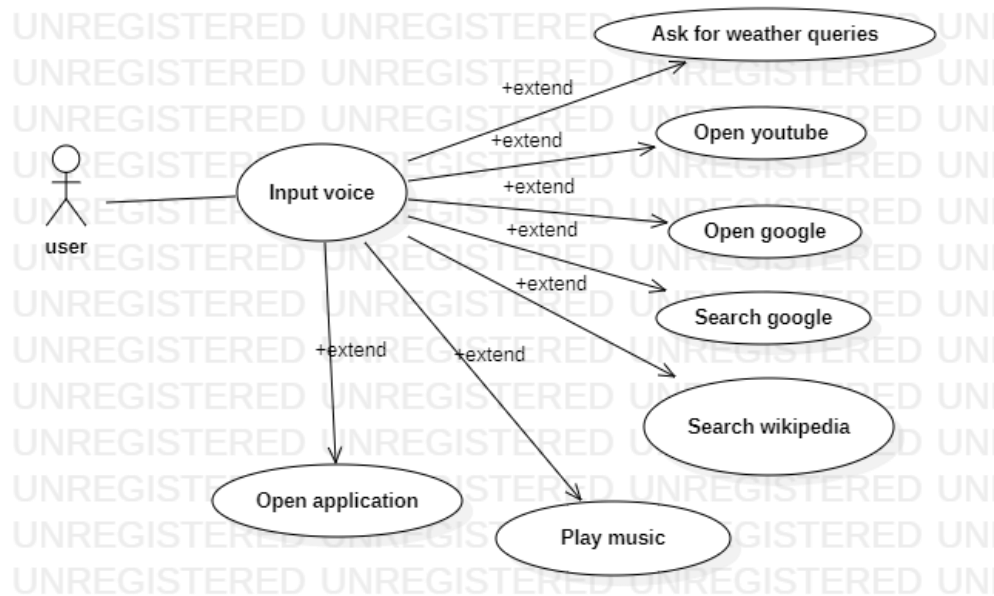
The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and State chart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

When the initial task is complete, use case diagrams are modelled to present the outside view.

In brief, the purposes of use case diagrams can be said to be as follows –

- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements are actors



4.4 User interface design

The **user interface** (UI) is the point of human-computer interaction and communication in a device. This can include display screens, keyboards, a mouse and the appearance of a desktop. It is also the way through which a **user** interacts with an application or a website.

4.5 SECURITY ISSUES

The protection of IT resources that includes hardware, software, data, procedure and people against unauthorized or natural use. The disaster is known as system security.

System security can be divided into four related topics:

- Security
- Integrity
- Privacy
- Confidentiality

SYSTEM SECURITY refers to technical innovations and procedures applied to hardware and operating systems to protect against intentional or accidental damage from a defined threat.

DATA SECURITY is the protection of data against loss, disclosure, modification and destruction.

SYSTEM INTEGRITY refers to the energetic operation of hardware and programs, adequate physical security and protection against external threats, such as illegal tapping and telephone tapping

PRIVACY defines the rights of the user or organization to determine what information they are willing to share or accept from others and how the organization can be protected from unwanted, unjust, or excessive disclosure of information about it.

CONFIDENTIALITY is a special status given to confidential information in a database to minimize any violation of privacy. It is an attribute of information that characterized your need of protection.

4.6 Test Cases Design

The importance of software testing and its impact can 'not be underestimated. Software Testing is a key component of software quality assurance and is a revision of specification, testing and coding. The increased visibility of the software system and the code associated with software failure are motivating factors for planning, through testing. It is not uncommon for a software organization to spend 40% of its efforts on experimentation.

Test of the white box :- The white box test is a test case design approach that employs the procedural design control architecture to produce test case. Using white box test approaches, software engineering can produce test cases.

- Ensure that all independent routes in a module have been exercised at least once.
- Exercise all logical decisions
- Execute all loops within their limits and their operational limits. Exercise internal data structures to maintain their validity.

Integration test :- Integration tests ensure that software and subsystem work together. Try the interface of all the modules to make sure the modules behave correctly when they are integration together.

System test :- It involves internal tests of the entire system before delivery to the user. Its purpose is to satisfy the user, the meets all the requirement of customer specifications.

Acceptance test :- It is a pre-delivery test in which the entire system is tested on the customer's site in real data to find errors.

Validation :- The system has been tested and implemented successfully, and therefore made sure that all the requirements listed in the specification of the software requirements are fully met. In case of incorrect entry, the corresponding error messages are displayed.

Compilation :- test It was a good idea to make our stress tests from beginning because it gave us time to solve some of the unexpected crashes and stability issues only occurred when the components were exposed to very high transaction volumes.

Execution proof :- This program has been loaded and executed successfully. Due to the good programming, no execution errors occurred.

CHAPTER 5 :- Implementation and Testing

CHAPTER 5 :- Implementation and Testing

5.1 Implementation Approaches

The technology used for development of the system 'VOICE ASSISTANT' are as follows: -

Tools and technology Tools and Techniques used:

- Star Uml
- Windows 10
- Vs Code

5.1.1 Star Uml

StarUml is a software modelling platform that supports UML (Unified Modelling Language). It is based on UML version 1.4 and provides eleven different types of diagram, and it accepts UML 2.0 notation. It actively supports the MDA (Model Driven Architecture) approach by supporting the UML profile concept. StarUml excels in customizability to the user's environment and has a high extensibility in its functionality. Using StarUml, one of the top leading software modelling tools, will guarantee to maximize the productivity and quality of your software projects

Features of star UML: -

- Supports most of the diagrams specified in UML 2.0.
- Very rich feature set and formatting options.
- Ability to generate source code from the UML diagram
- Reverse engineer the existing code into UML diagrams
- Supported languages: C, C# and Java.
- Fast load time/execution time compared with other UML tools.
- Familiar Visual Studio like user interface
- Supports exporting diagrams into JPG format

5.1.2 Windows 10

Windows 10 is a personal computer operating system developed and released by Microsoft as part of the Windows NT family of operating systems. It was officially unveiled in September 2014 following a brief demo at Build 2014. The first version of the operating system entered a public beta testing process in October 2014, leading up to its consumer release on July 29, 2015, and its release to volume licensing on August 1, 2015. Windows 10 introduces what Microsoft described as "universal apps"; expanding on Metro- style apps, these apps can be designed to run across multiple Microsoft product families with nearly identical code—including PCs, tablets, smartphones, embedded systems, One, Surface and Windows Holographic. The Windows user interface was revised to handle transitions between a mouse-oriented interface and a touchscreen-optimized interface based on available input devices

particularly on 2-in-1 PCs; both interfaces include an updated Start menu which incorporates elements of Windows 7's traditional Start menu with the tiles of Windows 8. The first release of Windows 10 also introduces a virtual desktop system, a window and desktop management feature called Task View, the Microsoft Edge web browser, support for fingerprint and face recognition login, new security features for enterprise environments, and DirectX 12 and WDDM 2.0 to improve the operating system's graphics capabilities for games. Windows 10 received mostly positive reviews upon its original release in July 2015; critics praised Microsoft's decision to downplay user-interface mechanics introduced by Windows 8 (including the full screen apps and Start screen) in non-touch environments to provide a desktop-oriented interface in line with previous versions of Windows, although Windows 10's touch-oriented user interface mode was panned for containing regressions upon the touch-oriented interface of Windows 8. Critics also praised the improvements to Windows 10's bundled software over 8.1, Xbox Live integration, as well as the functionality and capabilities of Cortana personal assistant and the replacement of Internet Explorer with Edge—although the browser was criticized for being a work in progress-- that was not yet feature complete.

5.1.3 Vs Code

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

5.2 Coding Details and Code Efficiency

5.2.1 Coding Details

```
from tkinter import *

from tkinter import filedialog,messagebox

from PIL import ImageTk, Image

import speech_recognition as sr

import pyttsx3, datetime, sys, wikipedia, wolframalpha, os, smtplib, random, webbrowser,
pygame, subprocess

import sqlite3

import time

import mysql.connector

con=sqlite3.connect('D:\\python_code\\project\\python project main\\main\\log.db')

mydb=mysql.connector.connect(host="localhost",user="root",password="tejas278",database
="photo",auth_plugin='mysql_native_password')

cursor=mydb.cursor()
```

```
time1=time.time()
time2=time.ctime(time1)
cursor=con.cursor()
con.commit()

client = wolframalpha.Client('AS45-ASD3-SD23-SW23')

engine = pyttsx3.init()
voices = engine.getProperty('voices')

def speak(audio):
    print('Karen:', audio)
    engine.setProperty('voice', voices[len(voices) - 1].id)
    engine.say(audio)
    engine.runAndWait()

def myCommand():

    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...")
        r.pause_threshold = 1
        audio = r.listen(source, timeout=5)
        print("cnverting...")
    try:
        query = r.recognize_google(audio, language='en-in')
```

```

    print('User: ' + query + '\n')

except sr.UnknownValueError:
    speak('Try again')
    pass

return query

def savedata():
    fn=filedialog.askopenfilename(title="Select file",filetypes=(("Image file","*.jpg"),("All files","*.*")))
    with open(fn, "rb") as f:
        data=f.read()
    sql="INSERT into files(id,file_data,date) values(NULL, %s, NOW())"
    cursor.execute(sql, (data,))
    mydb.commit()
    messagebox.showinfo("Success", "Your file has been saved")

def readdata():
    fn=filedialog.asksaveasfilename(initialdir=os.getcwd(),title="Select file",filetypes=(("Image file","*.jpg"),("All files","*.*")))
    sql="select file_data from files limit 2"
    cursor.execute(sql)
    r=cursor.fetchall()
    for i in r:
        data=i[0]
    with open(fn, "wb") as f:
        f.write(data)
    f.close()
    messagebox.showinfo("Success", "Your file has been saved")

```

```

def greetMe():
    speak('database connected')
    print('connected')
    speak('data inserted')
    print('inserted')
    speak('log data is saved')
    currentH = int(datetime.datetime.now().hour)
    if currentH >= 0 and currentH < 12:
        speak('Good Morning!')

    if currentH >= 12 and currentH < 18:
        speak('Good Afternoon!')

    if currentH >= 18 and currentH != 0:
        speak('Good Evening!')

class Widget:
    def __init__(self):
        root = Tk()
        root.title('EVA')
        root.config(background='Red')
        root.geometry('350x600')
        root.resizable(0, 0)
        root.iconbitmap(r'D:\python_code\project\python project main\main\eva1.ico')
        img = ImageTk.PhotoImage(Image.open(r"D:\python_code\project\python project
main\main\eva.jpg"))
        panel = Label(root, image = img)
        panel.pack(side = "bottom", fill = "both", expand = "no")

        self.compText = StringVar()

```



```
self.userText = StringVar()
```

```
self.userText.set('Click \'Start Listening\' to Give commands')
```

```
userFrame = LabelFrame(root, text="USER", font=('Black ops one', 10, 'bold'))
```

```
userFrame.pack(fill="both", expand="yes")
```

```
left2 = Message(userFrame, textvariable=self.userText, bg='dodgerBlue', fg='white')
```

```
left2.config(font=("Comic Sans MS", 10, 'bold'))
```

```
left2.pack(fill='both', expand='yes')
```

```
compFrame = LabelFrame(root, text="EVA", font=('Black ops one', 10, 'bold'))
```

```
compFrame.pack(fill="both", expand="yes")
```

```
left1 = Message(compFrame, textvariable=self.compText, bg='Red', fg='white')
```

```
left1.config(font=("Comic Sans MS", 10, 'bold'))
```

```
left1.pack(fill='both', expand='yes')
```

```
btn = Button(root, text='Start Listening!', font=('Black ops one', 10, 'bold'),
bg='deepSkyBlue', fg='white', command=self.clicked).pack(fill='x', expand='no')
```

```
btn2 = Button(root, text='Close!', font=('Black Ops One', 10, 'bold'), bg='deepSkyBlue',
fg='white', command=root.destroy).pack(fill='x', expand='no')
```

```
btn3=Button(root,text="save to database",font=('Black ops one', 10, 'bold'),
bg='deepSkyBlue', fg='white',command=savedata).pack(fill='x', expand='no')
```

```
btn4=Button(root,text="read to database",font=('Black ops one', 10, 'bold'),
bg='deepSkyBlue', fg='white',command=readdata).pack(fill='x', expand='no')
```

```
speak('Hello TJ, I am EVA! What should I do for You?')
```

```
self.compText.set('Hello TJ, I am EVA! What should I do for You?')
```

```
root.bind("<Return>", self.clicked)
```

```
root.mainloop()
```

```
def clicked(self):
```

```
    print('Working')
```

```
    query = myCommand()
```

```
    self.userText.set('Listening...')
```

```
    self.userText.set(query)
```

```
    query = query.lower()
```

```
    sql=("""insert into logrecord(query,time)
```

```
        values('{}','{}');""".format(query,time2))
```

```
    cursor=con.cursor()
```

```
    cursor.execute(sql)
```

```
    con.commit()
```

```
    print('log data inserted')
```

```
    if 'open ccleaner' in query:
```

```
        self.compText.set('okay')
```

```
        speak('okay')
```

```
        subprocess.call(r'C:\Program Files\CCleaner\CCleaner.exe')
```

```
    elif 'open google chrome' in query:
```

```
        self.compText.set('okay')
```

```
        speak('okay')
```

```
        subprocess.call(r'C:\Program Files (x86)\Google\Chrome\Application\chrome.exe')
```

```
    elif 'open powerpoint' in query:
```

```
        self.compText.set('okay')
```

```
        speak('okay')
```

```
        subprocess.call(r'C:\Program Files\Microsoft Office\Office14\POWERPNT.EXE')
```

elif 'open youtube' in query:

```
self.compText.set('okay')  
speak('okay')  
webbrowser.open('www.youtube.com')
```

elif 'open google' in query:

```
self.compText.set('okay')  
speak('okay')  
webbrowser.open('www.google.co.in')
```

elif 'open gmail' in query:

```
self.compText.set('okay')  
speak('okay')  
webbrowser.open('www.gmail.com')
```

elif 'shutdown' in query:

```
self.compText.set('okay')  
speak('okay')  
os.system('shutdown -s')
```

elif "what's up" in query or 'how are you' in query:

```
stMsgs = ['Just doing my thing!', 'I am fine!', 'Nice!', 'I am nice and full of energy']  
self.compText.set(random.choice(stMsgs))  
speak(random.choice(stMsgs))
```

elif 'email' in query:

```
self.compText.set('Who is the recipient? ')  
speak('Who is the recipient? ')  
recipient = myCommand()
```

```
self.userText.set(recipient)
recipient = recipient.lower()

if 'me' in recipient:
    try:
        self.compText.set('What should I say? ')
        speak('What should I say? ')
        content = myCommand()
        self.userText.set(content)

        server = smtplib.SMTP('smtp.gmail.com', 587)
        server.ehlo()
        server.starttls()
        server.login("Your_Username", 'Your_Username')
        server.sendmail('Your_Username', "Recipient_Username", content)
        server.close()
        self.compText.set('Email sent!')
        speak('Email sent!')

    except:
        self.compText.set('Email sent!')
        speak('Sorry ' + 'Sir' + '!, I am unable to send your message at this moment!')

elif 'nothing' in query or 'abort' in query or 'stop' in query:
    self.compText.set('Okay')
    speak('okay')
    self.compText.set('Bye Sir, have a good day.')
    speak('Bye Sir, have a good day.')

elif 'hello' in query:
```

```
self.compText.set('Hello Sir')
speak('Hello Sir')

elif 'log data' in query:
    speak('This is your log data')
    for r in cursor:
        logdata=time2
        print(logdata)
        self.compText.set(logdata)

elif 'bye' in query:
    self.compText.set('Bye ' + 'Sir' + ', have a good day.')
    speak('Bye ' + 'Sir' + ', have a good day.')

elif 'play music' in query:
    music_folder = 'D://python_code//project//python project main//main//music'
    music = ['NEFFEX']
    random_music = music_folder + random.choice(music) + '.mp3'
    os.system(random_music)

    self.compText.set('Okay, here is your music! Enjoy!')
    speak('Okay, here is your music! Enjoy!')

else:
    try:
        try:
            res = client.query(query)
            results = next(res.results).text
            self.compText.set(results)
            speak(results)
```

```
except:
    results = wikipedia.summary(query, sentences=2)
    self.compText.set(results)
    speak(results)
```

```
except:
    speak('I don\'t know Sir! Google is smarter than me!')
    self.compText.set('I don\'t know Sir! Google is smarter than me!')
    webbrowser.open('www.google.com')
```

```
if __name__ == '__main__':
    greetMe()
    widget = Widget()
```

Functionality of Code :-

The user use start button to give input for performing operations.

The operation perform are :-

- Open Google
- Open YouTube
- Open Google Chrome
- Play Music
- Search Google

5.2.2 Code Efficiency

Code efficiency is a broad term used to depict the reliability, speed and programming methodology used in developing codes for an application. Code efficiency is directly linked with algorithmic efficiency and the speed of runtime execution for software. It is the key element in ensuring high performance.

Code efficiency plays a significant role in applications in a high-execution-speed environment where performance and scalability are paramount. One of the recommended best practices in coding is to ensure good code efficiency. Well-developed programming codes should be able to handle complex algorithms.

Code Optimization: -

In computer science, program optimization or software optimization is the process of modifying a software system to make some aspect of it work more efficiently or use fewer resources. In general, a computer program may be optimized so that it executes more rapidly,

or to make it capable of operating with less memory storage or other resources, or draw less power. Optimization can occur at a number of levels

Design level: -

At the highest level, the design may be optimized to make best use of the available resources, given goals, constraints, and expected use/load.

Algorithms and data structures: -

Given an overall design, a good choice of efficient algorithms and data structures, and efficient implementation of these algorithms and data structures comes next.

Source code level: -

Beyond general algorithms and their implementation on an abstract machine, concrete source code level choices can make a significant difference.

Build level: -

Between the source and compile level, directives and build flags can be used to tune performance options in the source code and compiler respectively, such as using pre-process or defines to disable unneeded software features, optimizing for specific processor models or hardware capabilities, or predicting branching, for instance.

Compile level: -

Use of an optimizing compiler tends to ensure that the executable program is optimized at least as much as the compiler can predict.

Assembly level: -

At the lowest level, writing code using an assembly language, designed for a particular hardware platform can produce the most efficient and compact code if the programmer takes advantage of the full repertoire of machine instructions.

Reasons for the selection of technology: -

Python :-

Python is one of the top 10 popular programming languages of 2017. Python is a general purpose and high level programming language. You can use Python for developing desktop GUI applications, websites and web applications. Also, Python, as a high level programming language, allows you to focus on core functionality of the application by taking care of common programming tasks. The simple syntax rules of the programming language further makes it easier for you to keep the code base readable and application maintainable. There are also a number of reasons why you should prefer Python to other programming languages.

1) Readable and Maintainable Code :- While writing a software application, you must focus on the quality of its source code to simplify maintenance and updates. The syntax rules of Python allow you to express concepts without writing additional code. At the same time, Python, unlike

other programming languages, emphasizes on code readability, and allows you to use English keywords instead of punctuations. Hence, you can use Python to build custom applications without writing additional code. The readable and clean code base will help you to maintain and update the software without putting extra time and effort.

2) Multiple Programming Paradigms :-Like other modern programming languages, Python also supports several programming paradigm. It supports object oriented and structured programming fully. Also, its language features support various concepts in functional and aspect-oriented programming. At the same time, Python also features a dynamic type system and automatic memory management. The programming paradigms and language features help you to use Python for developing large and complex software applications.

3) Compatible with Major Platforms and Systems :-At present, Python is supports many operating systems. You can even use Python interpreters to run the code on specific platforms and tools. Also, Python is an interpreted programming language. It allows you to you to run the same code on multiple platforms without recompilation. Hence, you are not required to recompile the code after making any alteration. You can run the modified application code without recompiling and check the impact of changes made to the code immediately. The feature makes it easier for you to make changes to the code without increasing development time.

4) Robust Standard Library :-Its large and robust standard library makes Python score over other programming languages. The standard library allows you to choose from a wide range of modules according to your precise needs. Each module further enables you to add functionality to the Python application without writing additional code. For instance, while writing a web application in Python, you can use specific modules to implement web services, perform string operations, manage operating system interface or work with internet protocols. You can even gather information about various modules by browsing through the Python Standard Library documentation.

5) Many Open Source Frameworks and Tools :-As an open source programming language, Python helps you to curtail software development cost significantly. You can even use several open source Python frameworks, libraries and development tools to curtail development time without increasing development cost. You even have option to choose from a wide range of open source Python frameworks and development tools according to your precise needs. For instance, you can simplify and speedup web application development by using robust Python web frameworks like Django, Flask, Pyramid, Bottle and Cherrypy. Likewise, you can accelerate desktop GUI application development using Python GUI frameworks and toolkits like PyQt, PyJs, PyGUI, Kivy, PyGTK and WxPython.

6) Simplify Complex Software Development :-Python is a general purpose programming language. Hence, you can use the programming language for developing both desktop and web applications. Also, you can use Python for developing complex scientific and numeric applications. Python is designed with features to facilitate data analysis and visualization. You can take advantage of the data analysis features of Python to create custom big data solutions without putting extra time and effort. At the same time, the data visualization libraries and APIs provided by Python help you to visualize and present data in a more appealing and effective way. Many Python developers even use Python to accomplish artificial intelligence (AI) and natural language processing tasks.

7) Adopt Test Driven Development :-You can use Python to create prototype of the software application rapidly. Also, you can build the software application directly from the prototype simply by refactoring the Python code. Python even makes it easier for you to perform coding and testing simultaneously by adopting test driven development (TDD) approach. You can easily write the required tests before writing code and use the tests to assess the application code continuously. The tests can also be used for checking if the application meets predefined requirements based on its source code.

However, Python, like other programming languages, has its own shortcomings. It lacks some of the built-in features provided by other modern programming language. Hence, you have to use Python libraries, modules, and frameworks to accelerate custom software development. Also, several studies have shown that Python is slower than several widely used programming languages including Java and C++. You have to speed up the Python application by making changes to the application code or using custom runtime. But you can always use Python to speed up software development and simplify software maintenance.

5.3 TESTING APPROACHES

5.3.1 Unit Testing

Unit Testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing

Unit Testing Benefits: -

- Unit testing increases confidence in changing/ maintaining code. If good unit tests are written and if they are run every time any code is changed, we will be able to promptly catch any defects introduced due to the change. Also, if codes are already made less interdependent to make unit testing possible, the unintended impact of changes to any code is less.
- Codes are more reusable. In order to make unit testing possible, codes need to be modular. This means that codes are easier to reuse.
- Development is faster. How? If you do not have unit testing in place, you write your code and perform that fuzzy ‘developer test’ (You set some breakpoints, fire up the GUI, provide a few inputs that hopefully hit your code and hope that you are all set.) But, if you have unit testing in place, you write the test, write the code and run the test. Writing tests takes time but the time is compensated by the less amount of time it takes to run the tests; You need not fire up the GUI and provide all those inputs. And, of course, unit tests are more reliable than ‘developer tests’. Development is faster in the long run too. How? The effort required to find and fix defects found during unit testing is very less in comparison to the effort required to fix defects found during system testing or acceptance testing.

- The cost of fixing a defect detected during unit testing is lesser in comparison to that of defects detected at higher levels. Compare the cost (time, effort, destruction, humiliation) of a defect detected during acceptance testing or when the software is live.
- Debugging is easy. When a test fails, only the latest changes need to be debugged. With testing at higher levels, changes made over the span of several days/weeks/months need to be scanned.

5.3.2 Integration Testing

Integration Testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

- Top Down is an approach to Integration Testing where top-level units are tested first and lower level units are tested step by step after that. This approach is taken when top-down development approach is followed. Test Stubs are needed to simulate lower level units which may not be available during the initial phases.
- Bottom Up is an approach to Integration Testing where bottom level units are tested first and upper-level units' step by step after that. This approach is taken when bottom-up development approach is followed. Test Drivers are needed to simulate higher level units which may not be available during the initial phases.
- Big Bang is an approach to Integration Testing where all or most of the units are combined together and tested at one go. This approach is taken when the testing team receives the entire software in a bundle.
- Hybrid is an approach to Integration Testing which is a combination of Top Down and Bottom Up approaches.

5.33 Beta Testing

Beta Testing is one of the Acceptance Testing types, which adds value to the product as the end user (intended real user) validates the product for functionality, usability, reliability, and compatibility. Inputs provided by the end-users helps in enhancing the quality of the product further and leads to its success. This also helps in decision making to invest further in the future products or the same product for improvisation. Since Beta Testing happens at the end user's side, it cannot be the controlled activity.

Beta Testing is one of the Customer Validation methodologies to evaluate the level of customer satisfaction with the product by letting it to be validated by the end users, who actually use it, for over a period of time. Product experience gained by the end users are asked for feedback on design, functionality, and usability and this helps in assessing the quality of the product.

Purpose of beta testing: -

The points mentioned below can even be considered as the objectives for Beta Test and are very much required to produce far better results for a product-

1. Beta Test provides a complete overview of the true experience gained by the end users while experiencing the product.
2. It is performed by a wide range of users and the reasons for which the product is being used varies highly. Marketing managers focus on target market's opinion on each and

every feature, while a usability engineer / common real user focus on product usage and easiness, technical users focus on installation and uninstallation experience, etc.

3. Real world compatibility for a product can be ensured to a greater extent through this testing, as a great combination of real platforms is used here for testing on a wide range of devices, OS, Browsers, etc.
4. As a wide range of platforms which the end users are actually using, might not be available to the internal testing team during the QA, this testing also helps to uncover the hidden bugs and gaps in the final product.
5. Few specific platforms will cause the product to fail with showstopper bug which was not covered during QA. And this helps in improvising/fixing the product to be a compatible one with all possible platforms

CHAPTER 6 :- Results and Discussion

CHAPTER 6 ;- Results and Discussion

6.1 Test Results

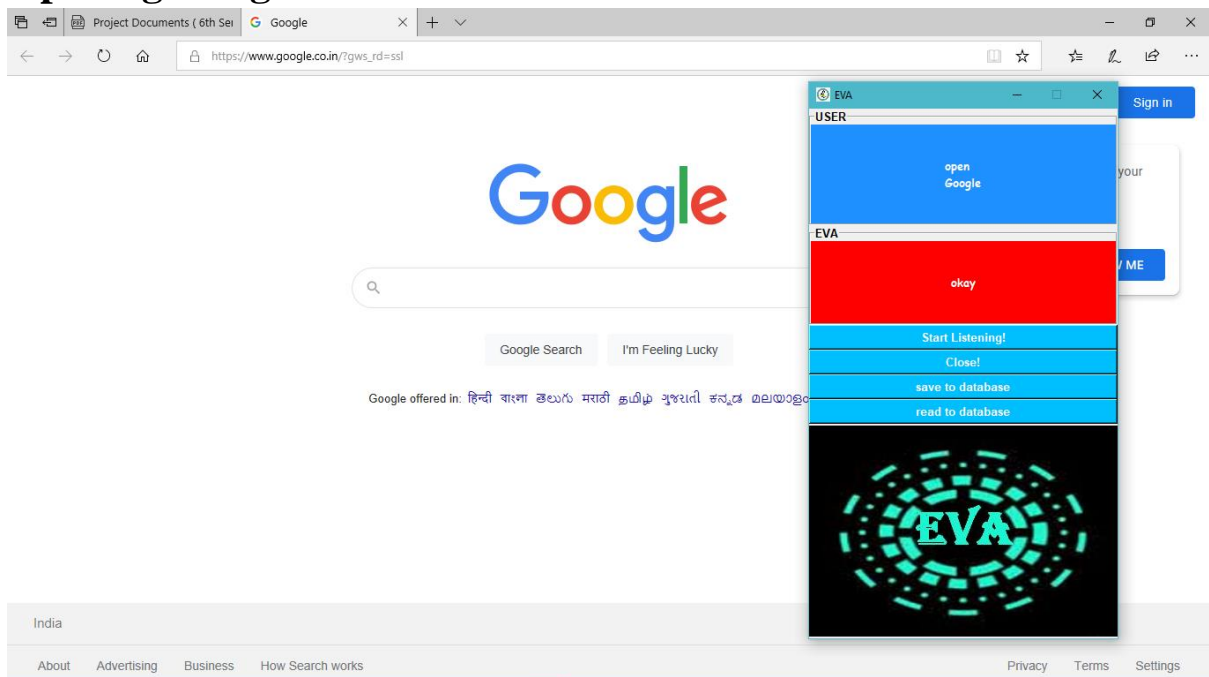
Main Window



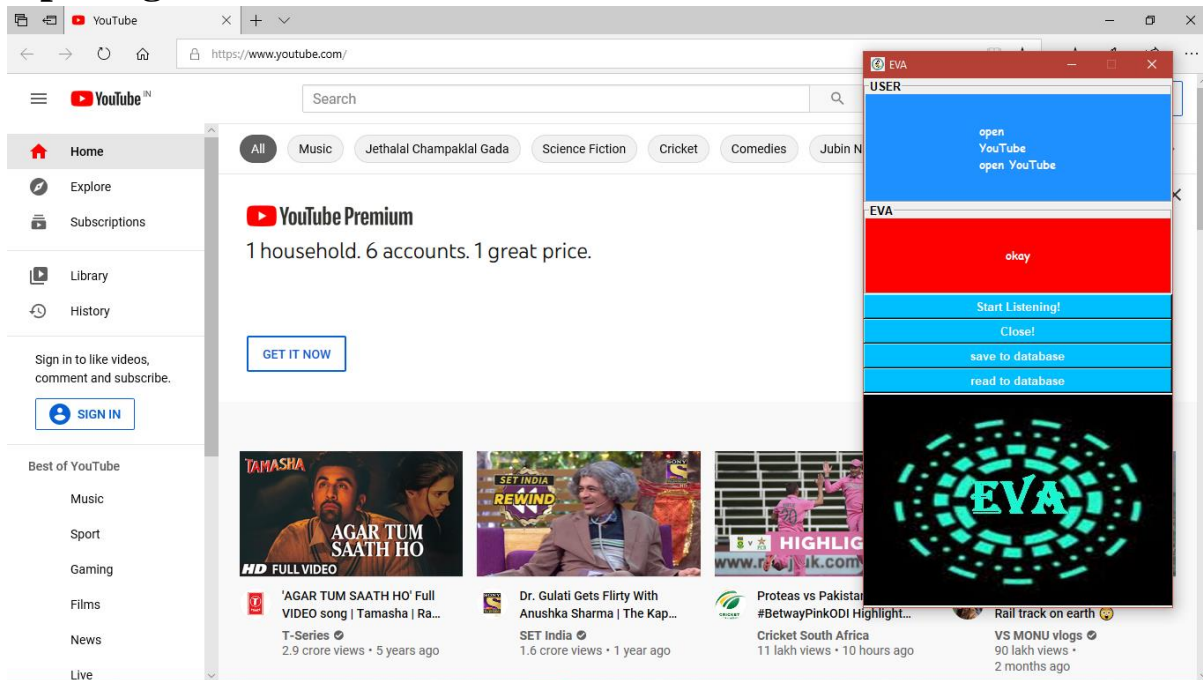
Respond Window



Opening Google in Web Browser



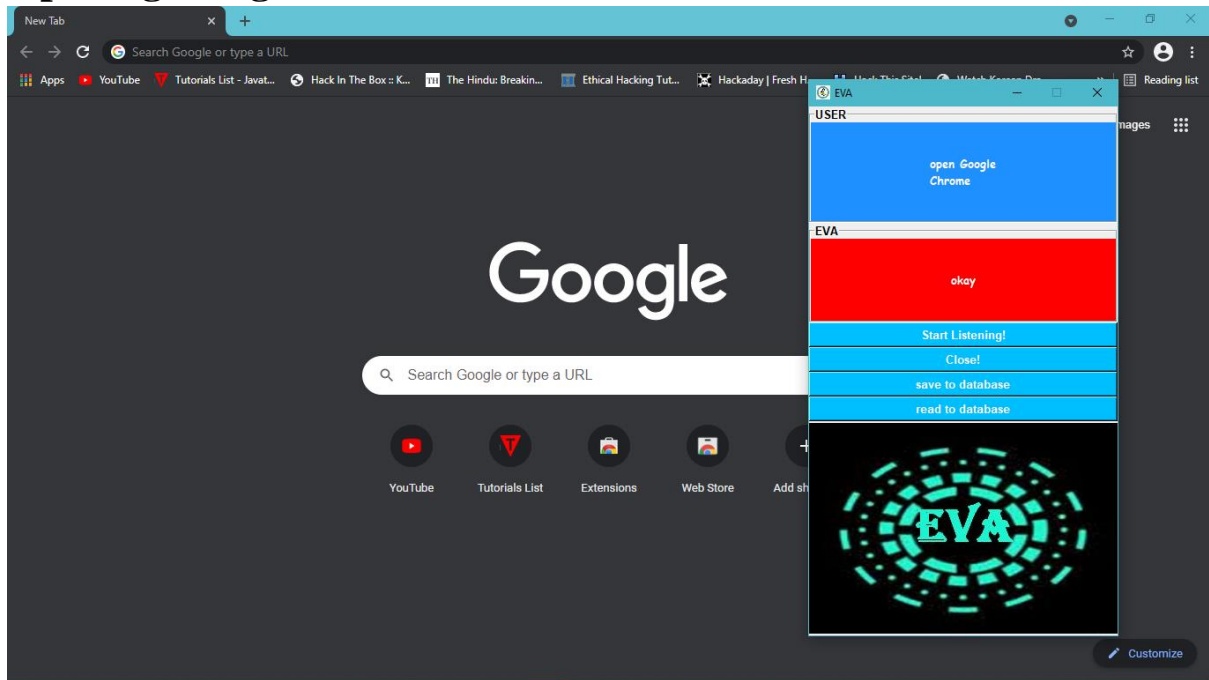
Opening YouTube in Web Browser



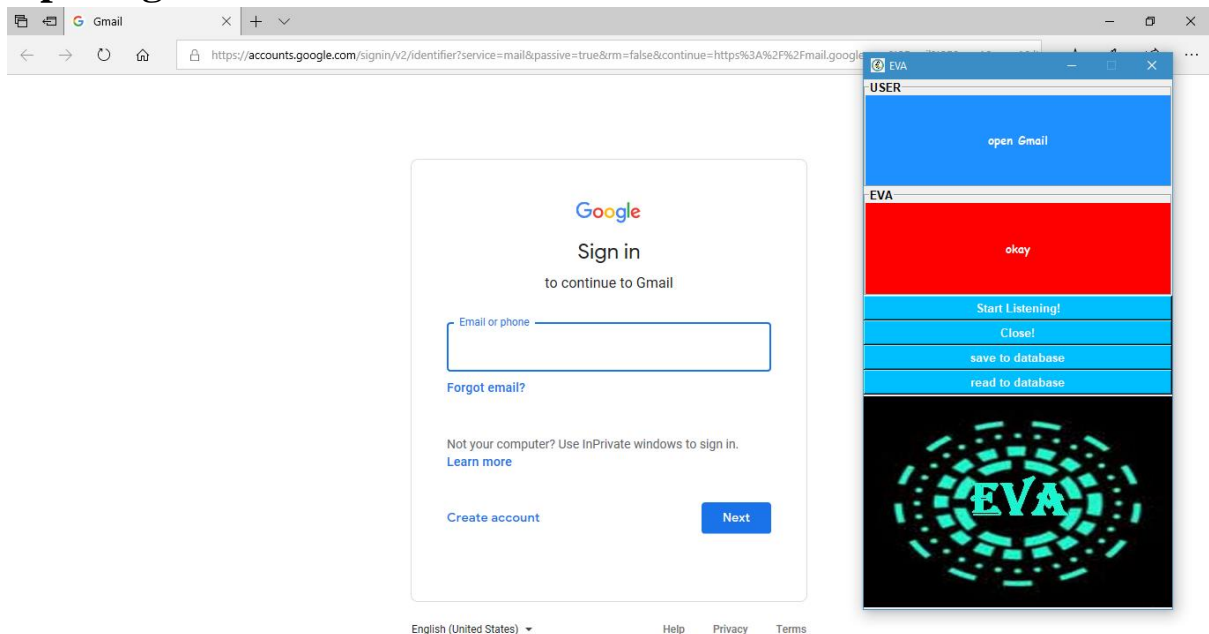
Playing Music



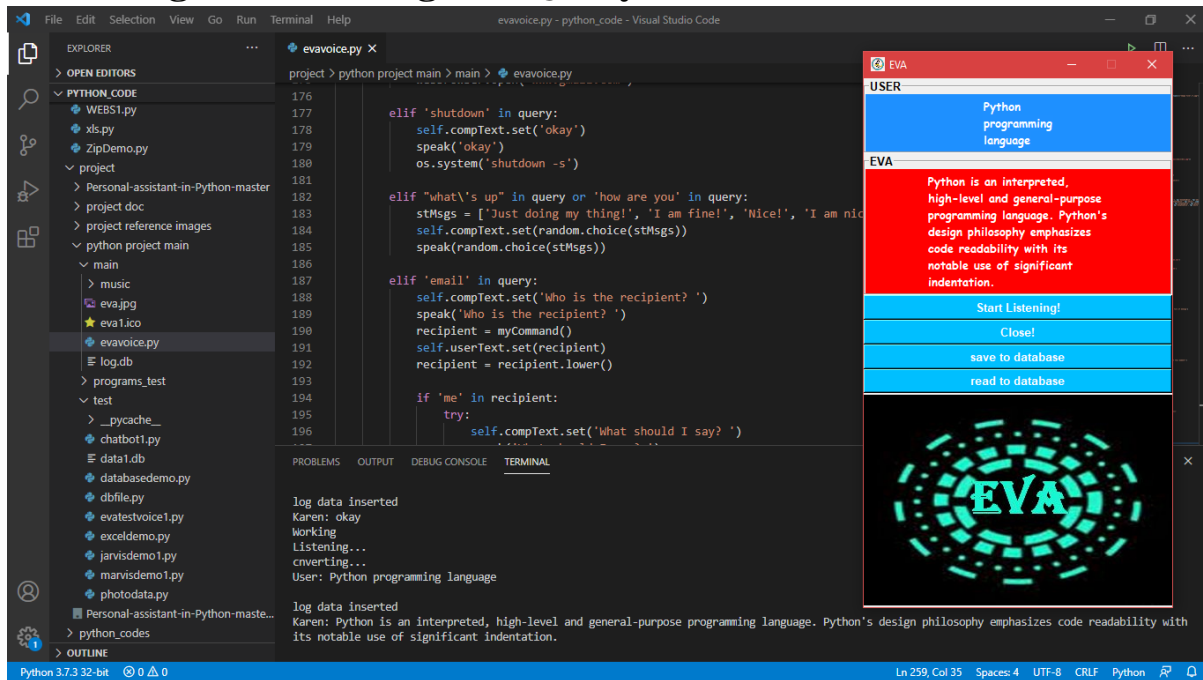
Opening Google Chrome



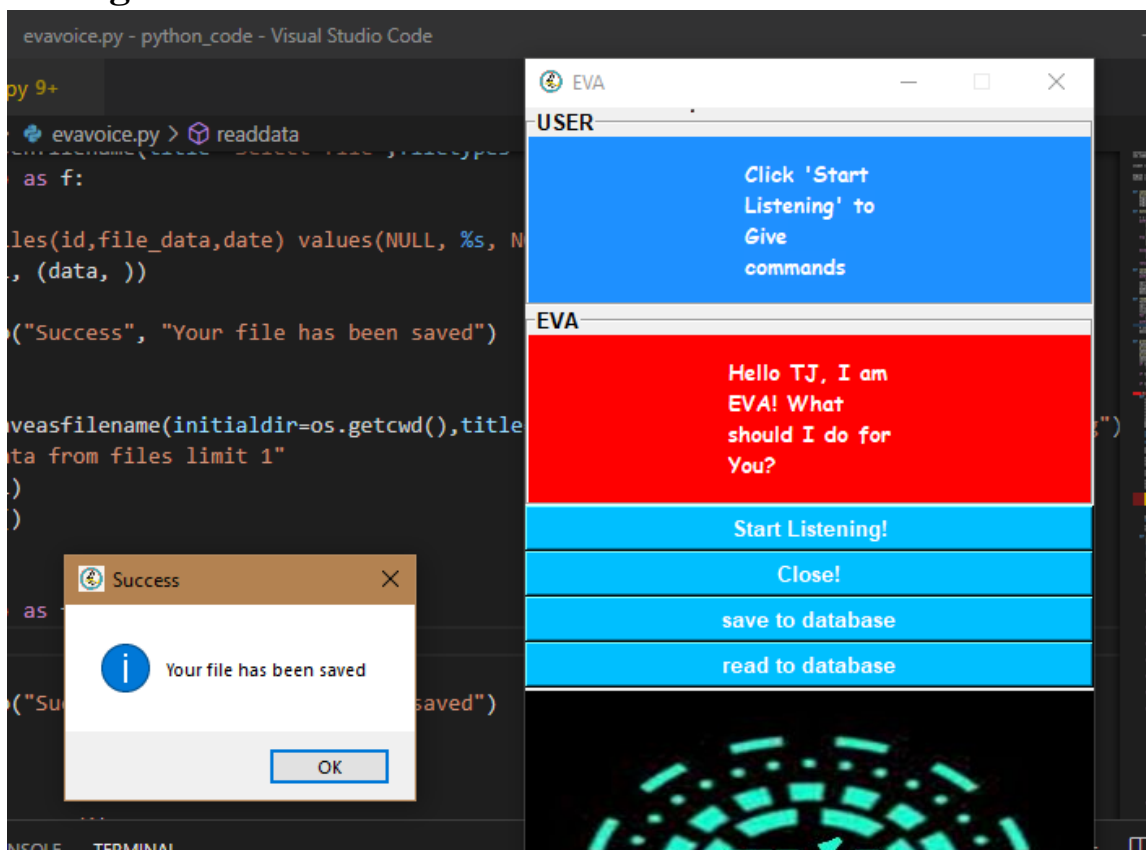
Opening Gmail in Web Browser



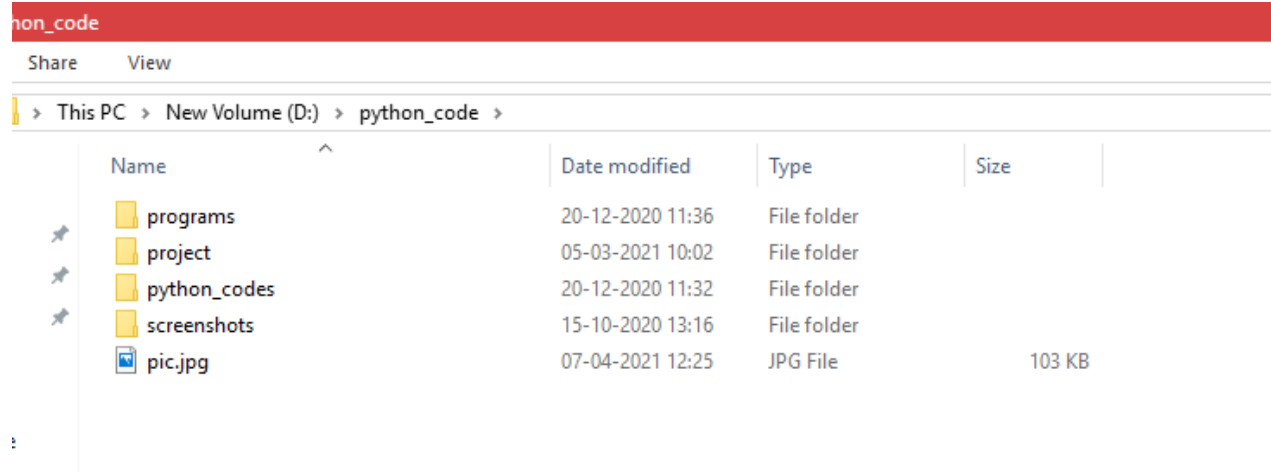
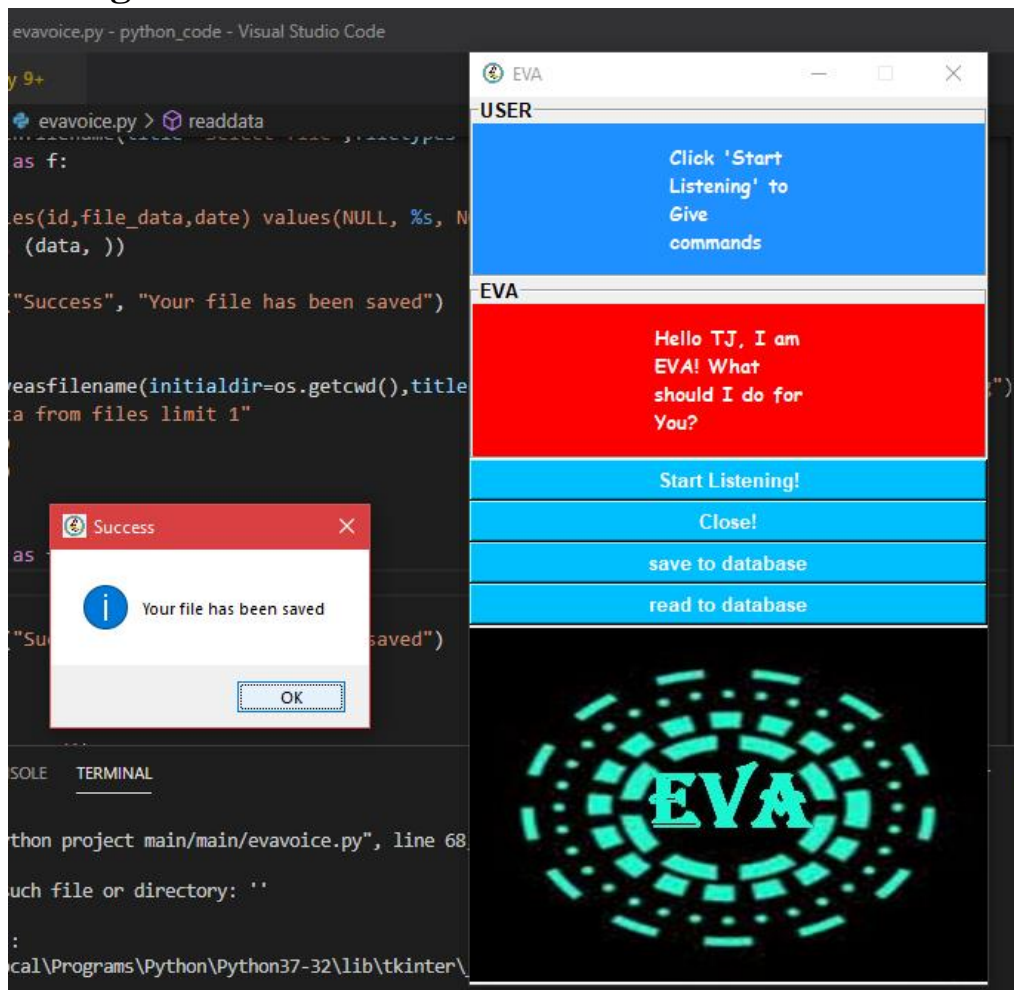
Searching on Web for given Query



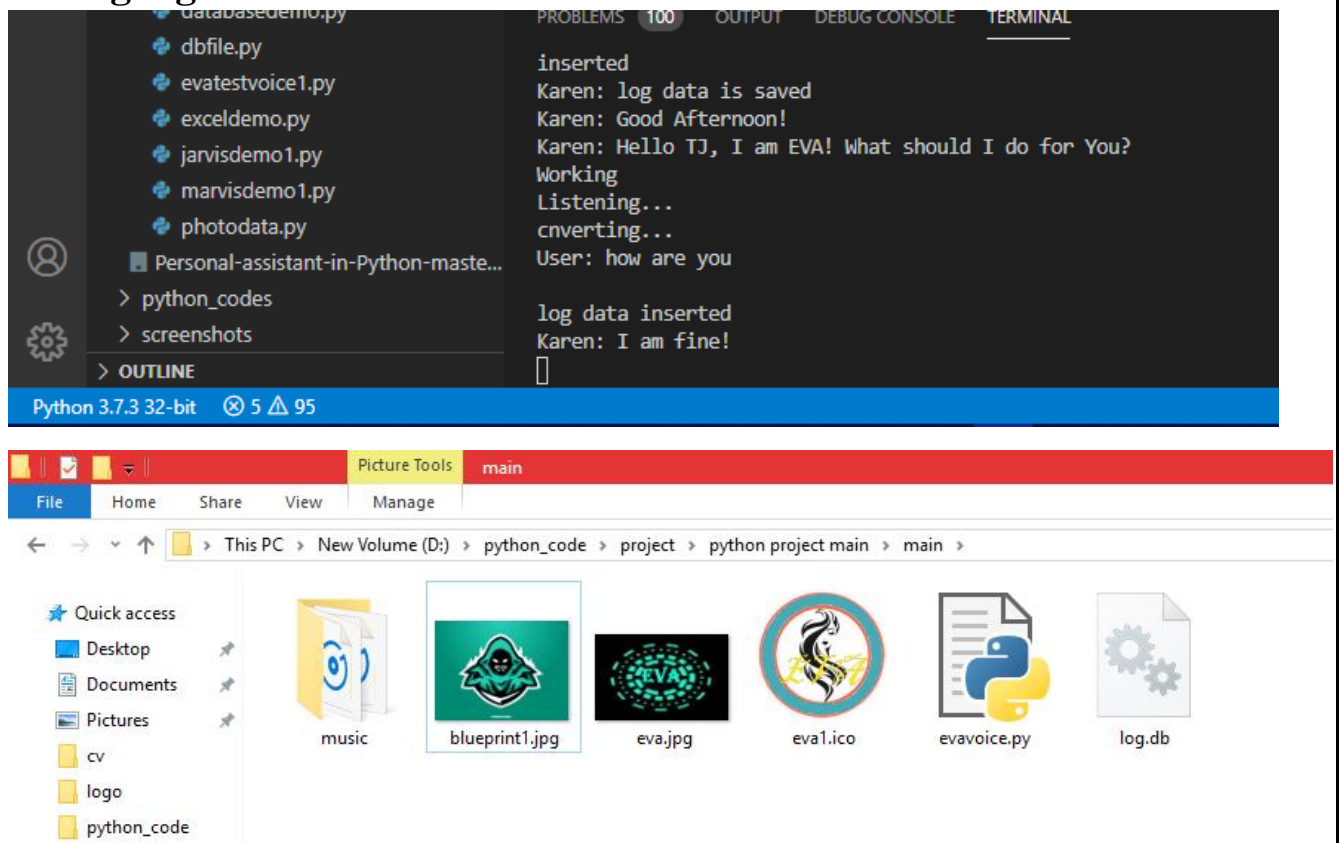
Saving Picture in database



Saving Picture from Database



Saving log Data in Database



6.2 USER DOCUMENTATION

User documentation refers to the documentation for a product or service provided to the end users. The user documentation is designed to assist end users to use the product or service. This is often referred to as user assistance. The user documentation is a part of the overall product delivered to the customer.

User documentation is important because it provides an avenue for users to learn: -

1. how to use your software
2. features of your software
3. tips and tricks of your software
4. how to resolve common problems with your software

User Documentation part consist of the description about all the forms that the application contains, they are as follows: -

1. Use start button to give command.
2. If you don't give command in 5 seconds it will stop listening to the command and will say try again.
3. You can use "stop", "abort", "quit" to stop the application and you can also use close button given below start button.
4. Your log data is saved in log.db file

Chapter 7:- Conclusions and REFERENCES

Chapter 7:- Conclusions and REFERENCES

7.1 Conclusions

This Project main task is to use voice command to perform operations and to make work of the user more effective.

I also conclude that this project has helped us gain more knowledge about the topic that we are indulged ourselves into “Python”. I would be glad to enhance and promote this project if given chance and help ourselves and society in the near future.

The developed application is tested with sample inputs and outputs obtained in according to the

requirement. Even though I have tried our level best to make it a dream project. Due to time constraints I could not add more facilities to it.

The efficiency of the developed system can be enhanced with some minor modifications. Future

development can be made in proposed system by integration more services like:

- New effective modules can be added time to time

7.2 FUTURE WORK

Machine learning and artificial intelligence are making huge impacts on almost every industry. The e-commerce trend, in particular, has been dramatically transformed by AI-powered voice assistants. Voice interfacing is also advancing at a fast rate in the banking and healthcare industries to keep up with the demands of modern consumers.

Millennial consumers are fueling the shift towards voice assistants powered by artificial intelligence. Significant AI adoption is driving the move to voice applications. Additionally, IoT devices such as thermostats, speakers, and smart appliances are making voice assistants ever more useful in the lives of everyday users.

7.3 REFERENCES

1) For Python installation:

<https://www.python.org/>

2) For Visual studio code installation:

<https://code.visualstudio.com/download>

3) For Python pip installation:

<https://pypi.org/project/pip/>

4) For Mysql Installation:

<https://dev.mysql.com/downloads/>

5) <https://www.google.com/>

6) <https://www.youtube.com/>