

# SAI'RA- A VIRTUAL ASSISTANT



*A project report is submitted to  
Devi Ahilya Vishwavidhyalaya, Indore  
in partial fulfilment for the award of  
Masters of Computer Applications*

**INTERNATIONAL INSTITUTE OF PROFESSIONAL STUDIES, INDORE**

**2020-21**

# SAI'RA- A VIRTUAL ASSISTANT



*A project report is submitted to  
Devi Ahilya Vishwavidhyalaya, Indore  
in partial fulfilment for the award of  
Masters of Computer Applications*

**Semester XI**  
**Session : Sept-Feb**

**UNDER GUIDANCE OF**  
PROF. JUGENDRA DONGRE

**SUBMITTED BY**  
ANUBHAV SINGH(IC 2K16 54)

**INTERNATIONAL INSTITUTE OF PROFESSIONAL STUDIES, INDORE**  
**2020-21**

# CERTIFICATE FROM GUIDE



## CERTIFICATE

This is certify that **ANUBHAV SINGH (IC 2K16 54** have completed their project work, titled “**SAI’RA- A VIRTUAL ASSISTANT**” as per the syllabus and have submitted a satisfactory report on this project as a part of fulfilment towards the degree of “**MCA**” from **DEVI AHILYA VISHWAVIDHYALAYA, INDORE.**

**PROGRAM INCHARGE**

.....

**PROJECT GUIDE**

.....

**DIRECTOR**

.....

# APPROVAL CERTIFICATE FROM EXAMINER



## CERTIFICATE

This is certify that **ANUBHAV SINGH (IC 2K16 54)** have completed their project work, titled “**SAI’RA- A VIRTUAL ASSISTANT**” as per the syllabus and have submitted a satisfactory report on this project as a part of fulfilment towards the degree of “**MCA**” from **DEVI AHILYA VISHWAVIDHYALAYA, INDORE.**

**INTERNAL EXAMINER**

.....

**EXTERNAL EXAMINER**

.....

# **ACKNOWLEDGEMENT**

Words are poor to substitute to express ones feelings adequately when deluged with emotion. First of all we would like to convey our honourable Director Dr. B.K. Tripathi and our Program In charge Dr. Ramesh Thakur for their valuable time and guidance.

We would like to convey our regards to Prof. Jugendra Dongre for his guidance, knowledge and support throughout the project.

We also obliged to our Batch Mentor Prof. Pradeep K. Jatav for his support and guidance.

**Anubhav Singh**

**IC 2K16 54**

**25-Feb-2021**

## TABLES OF CONTENTS

| Title                               | Page No. |
|-------------------------------------|----------|
| List of contents                    | a-b      |
| Abstract                            | c        |
| Chapter 1      INTRODUCTION         |          |
| 1.          Introduction            | 1        |
| 1.1        Domain                   | 2        |
| 1.2        Aim                      | 3        |
| Chapter 2      CURRENT SYSTEM       |          |
| 2.          Current system          | 4-6      |
| Chapter 3      PROJECT PROPOSING    |          |
| 3.          Project Proposal        | 7-8      |
| Chapter 4      PROJECT PLANNING     |          |
| 4.          Project Planning        | 9-11     |
| Chapter 5      TEAM STRUCTURE       |          |
| 5.          Structure of team       | 12-13    |
| Chapter 6      SCOPE OF THE PROJECT |          |
| 6.          Scope of the project    | 14-15    |
| 6.1        Objectives               | 15-16    |
| 6.2        Development plan         | 16       |
| Chapter 7      MODULES IN BREIF     |          |
| 7            Modules in brief       | 17-20    |
| Chapter 8      FEASIBILITY STUDY    |          |
| 8            Feasibility study      | 21-22    |
| Chapter 9      SYSTEM DESIGN        |          |
| 9.1        Block diagram            | 23-26    |

|            |                         |       |
|------------|-------------------------|-------|
| 9.2        | Control flow Diagrams   | 27    |
| Chapter 10 | METHODOLOGIES           |       |
| 10.        | Methodology             | 28-30 |
| Chapter 11 | APPLICATION DEVELOPMENT |       |
| 11.1       | N-Tier applications     | 31-34 |
| Chapter 12 | SYSTEM REQUIREMENTS     |       |
| 12.1       | Software requirements   | 35-36 |
| 12.2       | Hardware requirements   | 36    |
| Chapter 13 | TECHNOLOGY DESCRIPTION  |       |
| 13.1       | Python                  | 37-42 |
| Chapter 14 | TESTING                 |       |
| 14         | Testing Methodologies   | 43-51 |
| Chapter 15 | OUTPUT SCREENS          |       |
| 15         | Screens                 | 52-58 |
| Chapter 16 | SOURCE CODE             |       |
| 16         | Source code             | 59-70 |
| Chapter 17 | GANTT CHART             |       |
| 17         | Gantt Chart             | 71-72 |
| Chapter 18 | FUTURE ENHANCEMENTS     |       |
| 18         | Future enhancements     | 73-74 |
| Chapter 19 | CONCLUSIONS             |       |
| 19         | Conclusion              | 75-76 |
| Chapter 20 | BIBLIOGRAPHIES          |       |
| 20         | Bibliography            | 77-78 |

## **ABSTRACT**

As the technology advances, more and more systems are introduced which will look after the user's comfort. Few years before people were using a dedicated person to assist them but now-a-days virtual assistants are much popular in the market. They perform tasks very well with less time and great accuracy.

Virtual Assistant has no physical appearance. I have tried to make a virtual assistant look more interactive and usable with dynamic and adaptable. Thus, we see that the new technology always has more benefits and is more user-friendly. Here in virtual assistant SAI'RA user need not to get up to press or touch the keys of keyboard, rather they can simply speak while sitting.

**SAI'RA** means an **INTELLIGENT** program that can **SPEAK, ASK, RESPOND** and **ASSIST**.



# INTRODUCTION

## 1. INTRODUCTION

This project is basically a virtual assistant AI. It is capable of voice interactions, plays music, plays movies, provides weather, news, and distance from one place to another, shows location, open any third party websites or software etc. It is an example of weak AI that is it can only execute and perform quest designed by the user. Overall this project will be developed to help the user to do simple day-to-day online task in best way possible to reduce the wastage of time in searching different sites. The implemented voice assistant can perform the following task it can open YouTube, Gmail, and Google chrome and stack overflow. Predict current time, take a photo, search Wikipedia to abstract required data, predict weather in different cities, get top headline news from Times of India and can answer computational and geographical questions too.

### 1.1 Domain

A virtual assistant also called SAI'RA is an AI assistant or digital assistant is an application program that understands natural language voice commands and completes tasks for the user.

#### Why AI?

**Artificial Intelligence** enhances the speed, precision and effectiveness of human efforts. In financial institutions, **AI** techniques can be used to identify which transactions are likely to be fraudulent, adopt fast and accurate credit scoring, as well as automate manually intense data management tasks.

#### **Our virtual assistant will able to do the followings things-**

Weather forecasting, Launch Games, Launch Windows Applications, Open Websites, tells you about almost everything you ask, tells you date and time, greetings, news, etc. You can interact with your laptop's microphone/console. The response generated by the assistant will display on the console or as a speech via the speaker.

## 1.2 Aim

The aim of this project is to present a virtual assistant which can help user to perform day to day task with minimum efforts and less time consumption. SAIRA is a virtual assistant AI, essentially a digital voice that can recognize spoken commands and then talk back, meaning it can answer questions and perform certain tasks. With the help of python programming and having a little knowledge of Artificial Intelligence as a subject I've tried to build a personal AI assistant to perform tasks designed according to user.

# **CURRENT SYSTEM**

## 2. CURRENT SYSTEM

In current system, most of people use AI assistants proposed by Google (Google assistant), Amazon (Alexa), Microsoft (Cortana) etc. These assistances are Hi-tech yet you won't be able to perform your system based tasks. These assistants will work only on the device with was compatible with that particular company. Like

- Alexa on Amazon echo products.
- Google assistant on Google powered devices.
- Siri on iOS devices.
- Cortana on Microsoft devices.

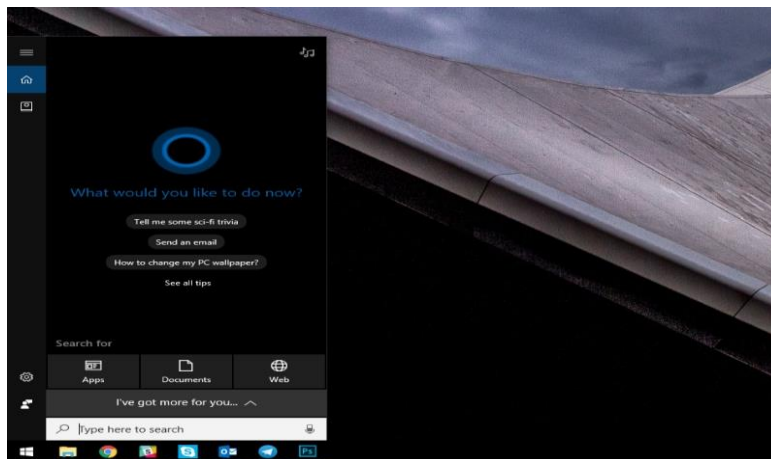
So, these devices are dependent on the companies associated with them. You need to pay a particular amount of that product to take services of such virtual assistants.

Another problem with these devices is they came with all time microphone enabled. So, there is a chance that the company associated with that product may use your personal data for their benefit.

Amazon has confirmed that employees sometimes eavesdrop on people as they interact with its **Alexa** digital assistant — but there's a way to block them from listening in ... “We take the security and privacy of our customers' personal **information** seriously,” Amazon said in a statement released to McClatchy.

Here are some currently used assistants:

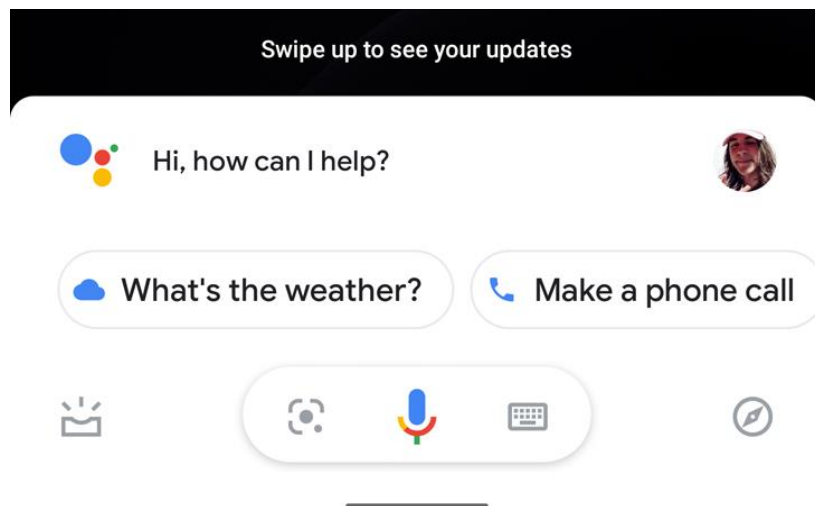
### 1. MICROSOFT CORTANA



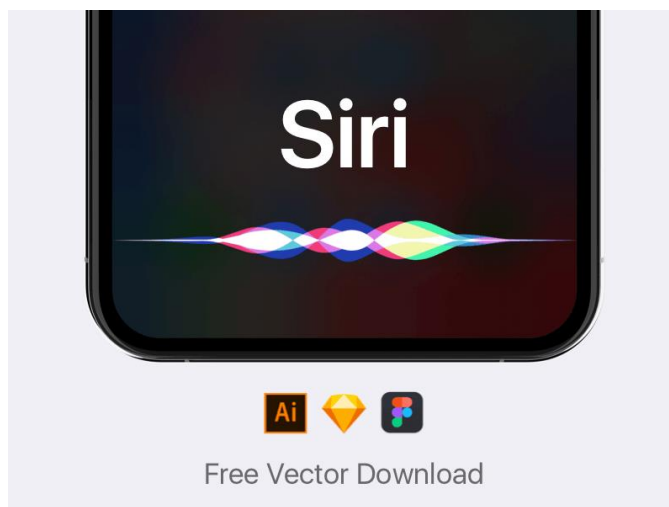
## 2. AMAZON ALEXA



## 3. GOOGLE ASSISTANT



## 4. APPLE SIRI



# **PROJECT PROPOSAL**

### **3. PROJECT PROPOSAL**

The current system has all the required course related books on a single platform and it is much easier for the student to search, read or download book. . The availability and updating of books is insured by authorized admin .This project provides authorization of every student on the database of IIPS. This project will be deployed on IIPS official site further. Since for starting phase this is for IIPS only, but this problem of book is faced by many collages. So, we can manage the books of different collages too.

The following is the Proposal for the project submitted to the project stakeholders.

**SAI'RA A VIRTUAL ASSISTANT**

**PROJECT PROPOSAL**

**DATE: 24-02-2021**

**PREPARED BY:** Anubhav Singh

(IIPS, DAVV, Indore)



# PROJECT PLANNING

## 4. Project Planning

Initially, the project scope is defined and the appropriate methods for completing the project are determined. Following this step, the durations for the various tasks necessary to complete the work are listed and grouped into a work breakdown structure. Project planning is often used to organize different areas of a project, including project plans, work loads and the management of teams and individuals. The logical dependencies between tasks are defined using an activity network diagram that enables identification of the critical path. Therefore the duration of the tasks is often estimated through a weighted average of optimistic, normal, and pessimistic cases. The critical chain method adds "buffers" in the planning to anticipate potential delays in project execution. Float or slack time in the schedule can be calculated using project management software. Then the necessary resources can be estimated and costs for each activity can be allocated to each resource, giving the total project cost. At this stage, the project schedule may be optimized to achieve the appropriate balance between resource usage and project duration to comply with the project objectives. Once established and agreed, the project schedule becomes what is known as the baseline schedule. Progress will be measured against the baseline schedule throughout the life of the project. Analyzing progress compared to the baseline schedule is known as earned value management.

### **The basic processes of project planning are:**

- **Scope planning** – Specifying the in-scope requirements for the project to facilitate creating the work breakdown structure.
- **Preparation of the work breakdown structure** – Spelling out the breakdown of the project into tasks and sub-tasks.
- **Project schedule development** – Listing the entire schedule of the activities and detailing their sequence of implementation.
- **Resource planning** – Indicating who will do what work, at which time, and if any special skills are needed to accomplish the project tasks.

- **Budget planning** – Specifying the budgeted cost to be incurred at the completion of the project.
- **Procurement planning** – Focusing on vendors outside your company and subcontracting.
- **Risk management** – Planning for possible risks and considering optional contingency plans and mitigation strategies.
- **Quality planning** – Assessing quality criteria to be used for the project.
- **Communication planning**– Designing the communication strategy with all project stakeholders.

**The purpose of the project planning phase is to:**

- Establish basic requirements.
- Establish cost, schedule, list of deliverables, and delivery dates.
- Establish resources plans.
- Obtain management approval and proceed to the next phase.

# TEAM STRUCTURE

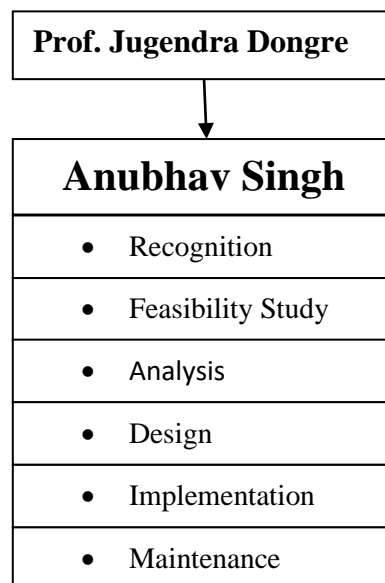
## 5. TEAM STRUCTURE

This project is made under the supervision of **Prof. Jugendra Dongre Sir**, who has guided me regarding the successful implementation of this project.

The following are the students working under this project with their respective roles:

### **Anubhav Singh**

- Recognition: Gather and validate the system.
- Feasibility Study: Define the problem and scope of existing system. Overview the new system and determine its objective and produce the project schedule.
- Analysis: System study and organizational knowledge. Problem identification, problem analysis and problem solving.
- Design: Includes the design of application, network, databases, user interfaces and system interfaces. Prepare a design document which will be used during next phases.
- Implementation: Implement the design into source code through coding and combine all the modules together.
- Maintenance: Implement the changes that software might undergo over a period of time and implement the new requirements.



(Team Structure)

# **SCOPE OF PROJECT**

## 6. SCOPE OF PROJECT

Everyone needs an assistant to perform some of simple but important tasks. By hiring a **virtual assistant**, you're able to delegate tasks that'll help you get rid of time-wasting tasks. For example, this may include sending emails, answering calls, give date and time, open websites, and search everything on Wikipedia etc. This project will be dynamic, user friendly and error free. The scope of Virtual Assistant is not only about product-development but also up to the implementation and deployment. This includes the training of the various end-users and maintenance of the products.

### 6.1 OBJECTIVES OF THE PROJECT

The main objectives of the project are as follows:

- ***Reduced Risk:***

Having human as an assistant can sometime be harmful because it can harm your social media handles and can go into your system to explore and finally steal important documents. So, having a Virtual Assistant to operate your system can minimise such risk as you can regulate it's limit.

- ***Optimized Processing Time:***

As we all know mistakes are often done by humans and it can lead to too much time to undone those mistakes. But machines always do what they are meant to with full efficiency and in minimum time.

- ***Mobility of Users:***

Once setting up this assistant user can access it anywhere with their desktop. The computer program is independent to do their desired taks.

- ***Reduced Lending Cost:***

The decrease in processing time and making it a paperless solution reduces labour cost, paper and printing cost, etc.

- ***Customer Satisfaction:***

Having own Virtual assistant with minimum cost and system based can give satisfaction to each and every customer. Customers dealing with guaranteed quick quotes, faster processing time, better customer service, easier to apply as a repeat customer and can be guaranteed of better one to one experience.

## **6.2 DEVELOPMENT PLAN**

The design and implementation of this project has been carried out in a completely step-by-step manner that are:

- Design
- Software Requirement
- Coding
- Test
- Project Report
- Implementation



# **MODULES IN BREIF**

## 7. MODULES IN BRIEF

Briefly the module of used can be classified in following points:

### 1. **tkinter**

The tkinter package (“Tk interface”) is the standard Python interface to the Tk GUI toolkit. Both Tk and tkinter are available on most Unix platforms, as well as on Windows systems.

### 2. **Cv2**

OpenCV-Python is a library of Python bindings designed to solve computer vision problems. **cv2**. imread() method loads an image from the specified file. If the image cannot be read (because of missing file, improper permissions, unsupported or invalid format) then this method returns an empty matrix.

### 3. **Pytttsx3**

Pytttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline, and is compatible with both Python 2 and 3.

### 4. **datetime**

**datetime module** supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and **datetime** are an object in **Python**, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

### 5. **speech\_recognition**

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

### 6. **ecapture**

This module is used to capture images from your camera.

**7. wikipedia**

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

**8. os**

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

**9. request**

The request module is used to send all types of HTTP request. It accepts URL as parameters and gives access to the given URL'S.

**10. wolframalpha**

Wolfram Alpha is an API which can compute expert-level answers using Wolfram's algorithms, knowledge base and AI technology. It is made possible by the Wolfram Language.

**11. 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.

**12. random**

The random module is a built-in module to generate the pseudo-random variables. It can be used perform some action randomly such as to get a random number, selecting a random elements from a list, shuffle elements randomly, etc.

**13. winshell**

The **winshell module** is a light wrapper around the Windows shell functionality. It includes convenience functions for accessing special folders, for using the shells file copy, rename & delete functionality, and a certain amount of support for structured storage.

**14. pywhatkit**

We are using pywhatkit for webscrapping of youtube and to search directly on youtube.

**15. pyaudio**

**pyAudio** is a set of **Python** bindings for PortAudio, a cross-platform C++ library interfacing with audio drivers.

**16. pywin32**

Python has the “Python for Windows Extensions” package known as pywin32 that allows us to easily access Window’s Component Object Model (COM) and control Microsoft applications via python. This article will cover some basic use cases for this type of automation and how to get up and running with some useful scripts.

# **FEASIBILITY STUDY**

## **8. Feasibility Study**

Making a multifunctional input device using a web camera and image processing techniques. We plan to make a keyboard that is dynamic and responsive to the Applications being run.

The proposed system would have an Applications frontend which would help initialize the keyboard to the new environment.

A feasibility study was an evaluation of a proposal designed to determine the difficulty in carrying out a designated task. Generally, a feasibility study precedes technical development and project implementation.

### **8.1 Economic Feasibility**

To develop the proposed system, it needs no extra facilities and devices. All dependencies are satisfied from the open source projects. All tools used are free and open source hence its development is economically.

### **8.2 Technical Feasibility**

Proposed system is technically feasible because the proposed system requires only those H/W and S/W tools that are available in the system. It requires the installation of Python Shell which can be done for free. Moreover, expandability will be maintained in the new system. New modules can be added later on the Applications, if required in the future. Additionally, the Applications will have User friendly Screen.

### **8.3 Behavioural Feasibility**

Behavioural feasibility determines how much effort will go in the proposed information system, and in educating and training the users on the new system. Since the user interface is very simple and easily understandable, no training is required for using this system.

# **SYSTEM DESIGN**

## 9. SYSTEM DESIGN

### BLOCK DIAGRAM



#### **Speech Signal:**

Speech signal is given by microphone to the assistant. It will take speech input because of speech recognition python module.

#### **Speech Signal Pre Processing:**

Speech taken as input will be pre processed and check whether the given input is clear and within the threshold set by programmer.

#### **Feature Extraction:**

Now speech will check given command within script to check a dedicated task to that input. If it is found then it will give output to language modelling otherwise go back to search for speech signal.

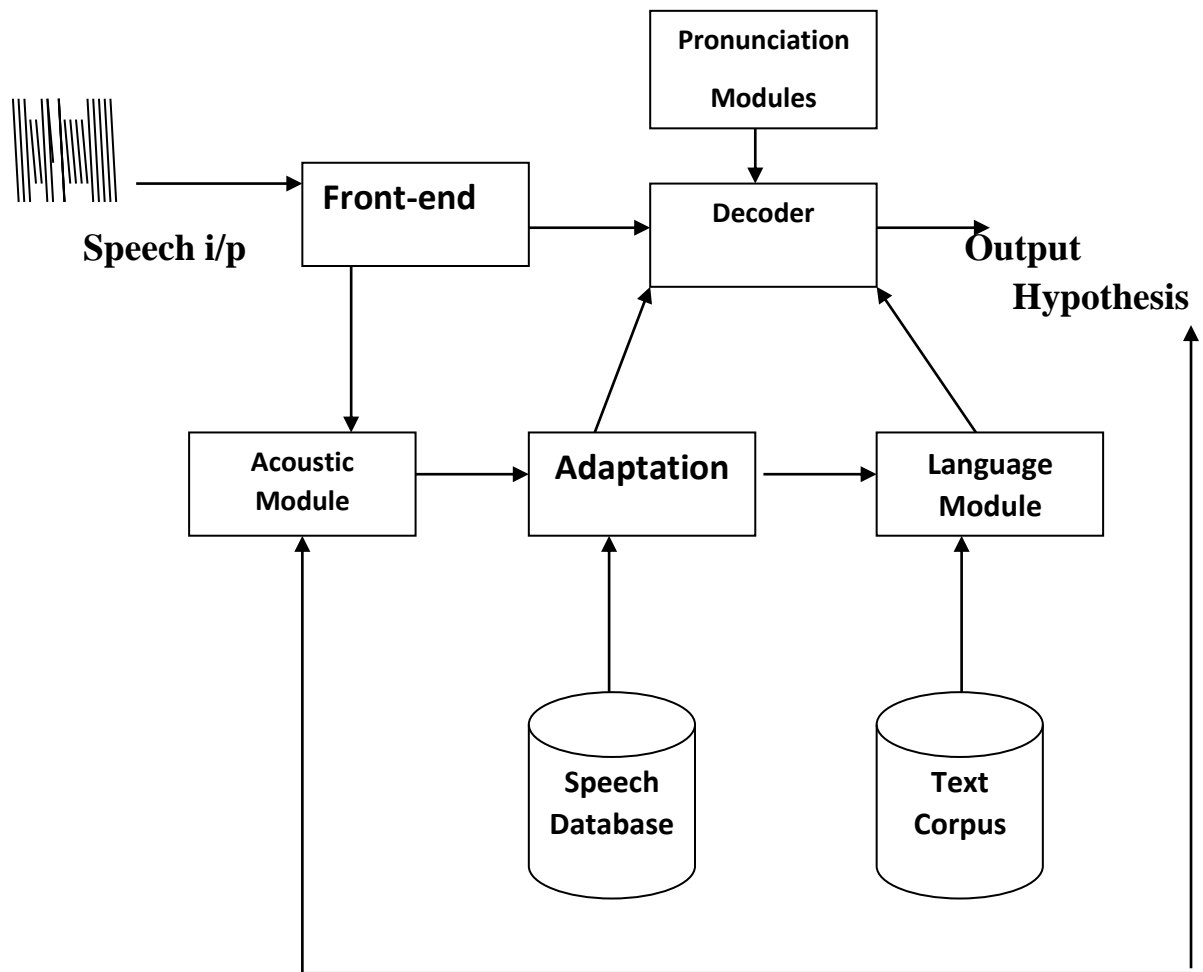
#### **Language modelling:**

After extracting features the output will be converted into preferred language of user.

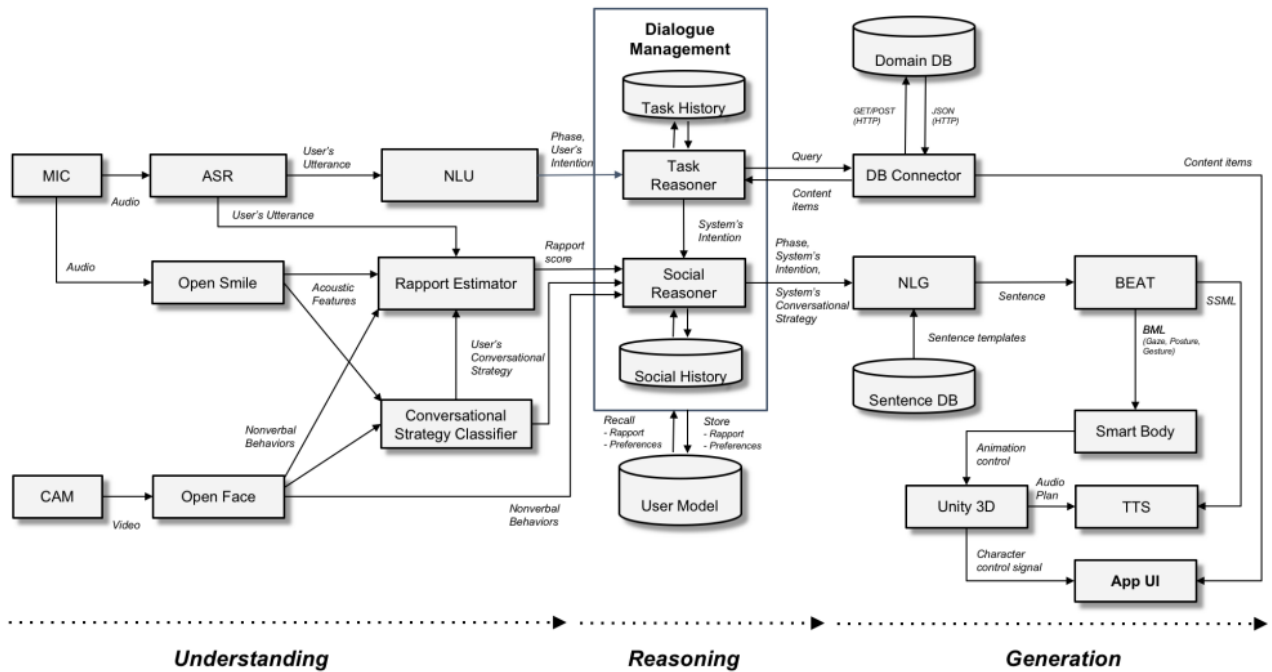
#### **Digital text:**

The output will be shown on screen in digital text format set by programmer or preferred by the user.

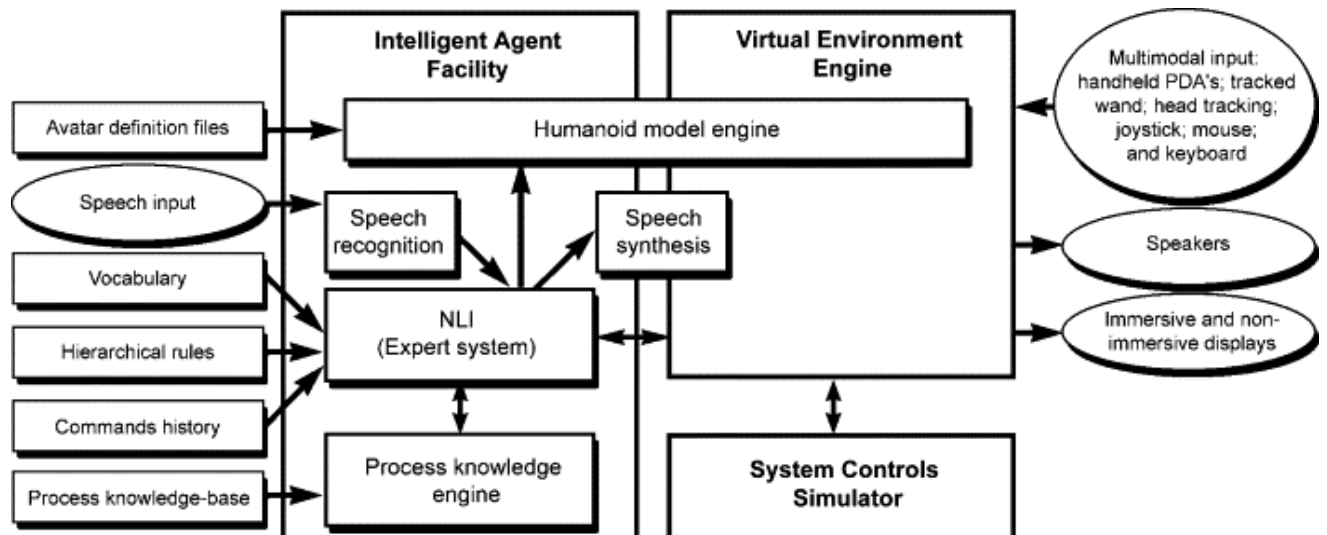




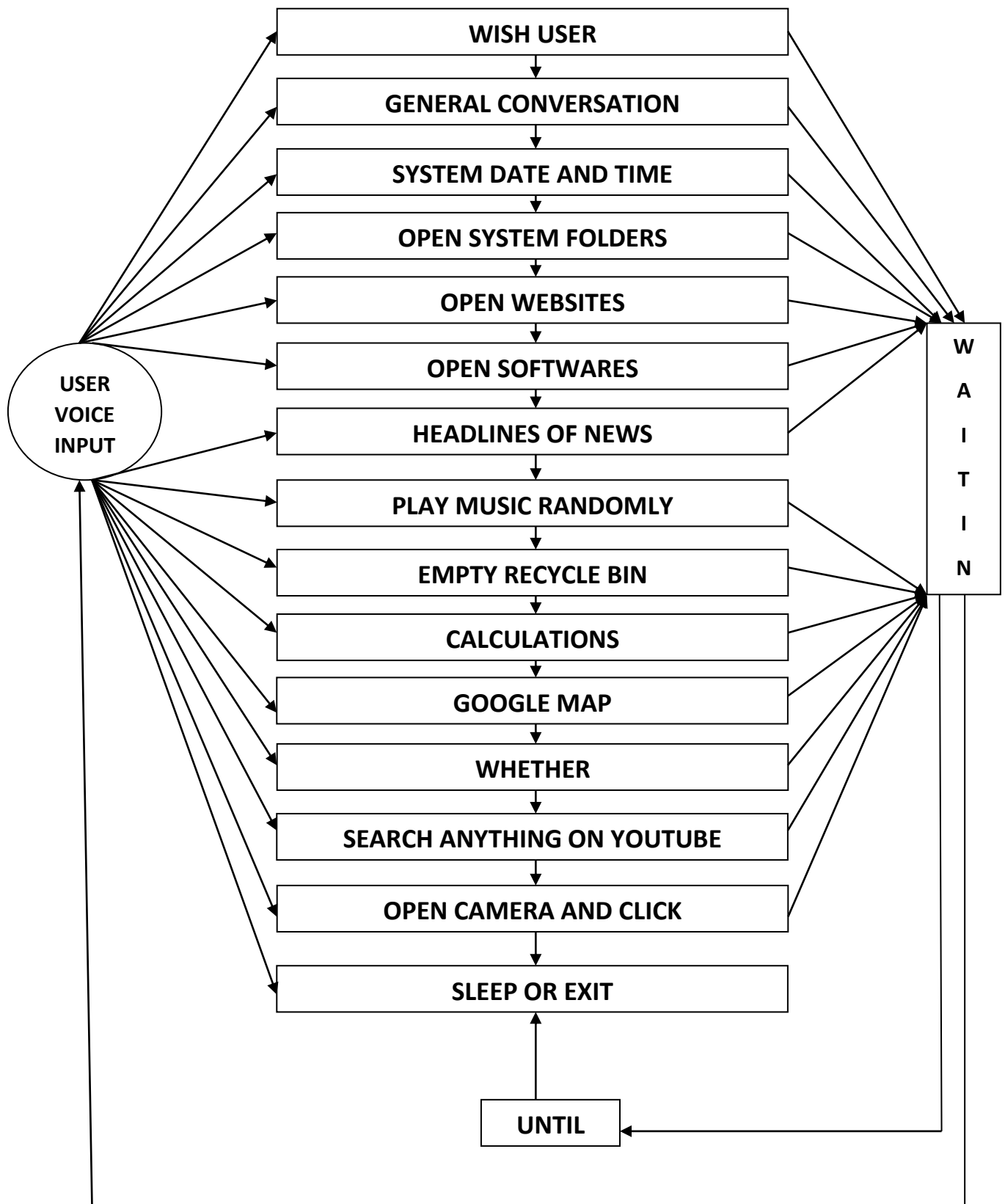
**OUTLINE OF SPEECH RECOGNITION SYSTEM**



## INNER SYSTEM WORKING



## FLOW OF SAI'RA

**CONTROL FLOW OF SAIRA****IF FAILS**

# **METHODOLOGIES**

## 10. METHODOLOGY

The implementation of the whole system is proposed to be done in a completely step-by-step manner. The entire project is divided in the following phases:

### **Phase 1: Analysis Phase:**

In this phase the requirements are first gathered from the customer and the requirement analysis starts. Web sites providing related sites are visited. Some existing systems are also compared and the feasibility of the requirements is checked. A plan to the interaction of the end user with the proposed system is made which tells all the basic requirements of the end user, their informational needs along with their problems. Study of the complete functional system is done. Proposed system is formed as the problems in the existing system found. Study of the concepts of database design and develop this system is done.

### **Phase 2: Project Planning Phase:**

This phase involves plan for the project that we have is prepared, which describes all the section of project management. Plan is prepared with the help of Project Plan, Schedule, Assigning Roles and Responsibilities, Communication Plan, Quality Assurance Plan, Project Cost Estimate and Risk Plan.

### **Phase 3: Conceptual Design Phase:**

In this phase of the project system is shown in the modular form.

- Identification of Entities and their Relationships:  
After completion of the analysis phase modelling phase starts. This is based on the analysis done. In modelling with the concepts of DBMS high level entities are found and the relationship between the entities is shown.
- Development of ER and EER model for the proposed system:  
Entity Relationship diagram and extended ER diagram is made to show relationship between the entities. It makes easy to learn the system.
- Identification of classes and their relationship:  
Classes represent entities with common characteristics or features. These features include attributes, operations and associations.

- Development of the class Diagram:  
Class diagram show the class of the system, their interrelationship (including inheritance, aggression and association), and the operations and attributes of the classes.

**Phase 4: Logical Design Phrase:**

Logical design phase is for preparing for the structure of the database, interface forms. This phase is the conversion of the conceptual design of the modelling phase. This design is just before the implementation phase.

**Phase 5: Implementation and Testing:**

This phase is for the implementation of the design development in the logical design phase. In this phase programs are written, physical database designing is done and after implementing all the function of user requirements test of implemented functions with various test cases is done.

**Phase 6: User Interface:**

User interface is very essential part of the any product through which all type interacts thus user interface must be user friendly. User friendly environment enables user to get information efficiently without any problem in interacting with the system.

# **APPLICATION DEVELOPMENT**

## 11. APPLICATION DEVELOPMENT

### **N-Tier Applications:**

N-Tier Applications can easily implement the concepts of Distributed Application Design and Architecture. The N-Tier Application provide strategic benefits to Enterprise Solutions. While 2-Tier, client-server can help us create quick and easy solution and may be used for Rapid Prototyping, they can easily become maintenance and security night mare.

The N-Tier Application provide specific advantage that are vital to the business continuity to the enterprise. Typical features of a real life N-Tier may include the following:

- Security
- Availability and Scalability
- Management
- Easy Maintenance
- Data Abstraction

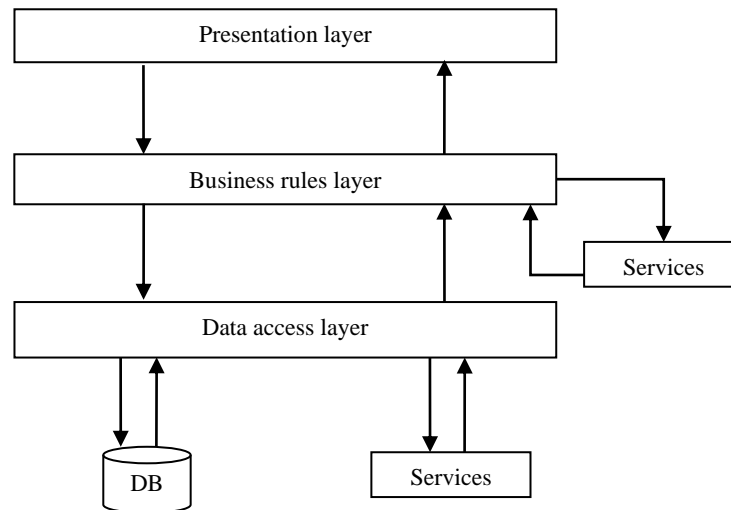
The above mentioned points are some of the key design goals of a successful N-Tier application that intends to provide a good business solution. An N-Tier application helps us distribute the overall functionality into various tiers or layers:

- Presentation Layer
- Business Rules Layer
- Data Access Layer
- Database/Data Store

Each layer can be development independently of the other provided that it adheres to the standards and communicates with the other layers as per the specifications.

This is the one of the biggest advantages of the N-Tier application. Each layer can potentially treat the other layer as a 'Block-Box'. In other words, each layer does not care how other layer processes the data as long as it sends the right data in a correct format.





Data flown through different layers

- **PRESENTATION LAYER**

This layer is also called as the Client layer comprises of components that are dedicated to presenting the data to the user. For example: Windows/WEB forms and bottoms, Edit boxes, Text boxes, Labels, Grids, etc.

- **BUSINESS RULES LAYER**

This layer encapsulates the business rules or the business logic of the encapsulations. To have a separate layer for business logic is of a great advantage. This is because any changes in business rules can be easily handled in this layer. As long as, the interface between the layers remains the same, any changes to the functionality/processing logic in this layer can be made without impacting the others. A lot of client-server apps failed to implement successfully as changing the business logic was a painful process.

- **DATA ACCESS LAYER**

This layer comprises of components that help in accessing the database. If used in the right way, this layer provides a level of abstraction for the database structures. Simply put changes made to the databases, tables, etc. do not affect the rest of the application because of the data access layer. The different application layers sends the data requests to this layer and receive the response from this layer.

- **DATABASE LAYER**

This layer comprises of the Database Components such as DB Files, Tables, Views, etc. The Actual database could be created using SQL Server, Oracle, Flat files, etc. In an N-Tier application, the entire application can be implemented in such a way that it is independent of the actual database. For instance, you could change the Database Location with minimal changes to Data Access Layer. The rest of the Application should remain unaffected.

## **PERFORMANCE REQUIREMENTS**

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

- The system should be able to interface with the existing system.
- The system should be accurate.
- The system should be better than the existing system.

The existing system is completely dependent on the user to perform all the duties.

# **SOFTWARE AND HARDWARE REQUIREMENTS**

## 12. SOFTWARE AND HARDWARE REQUIREMENT

### SOFTWARE REQUIREMENTS

- Operating System : Windows 10 pro
- User Interface : tkinter
- Programming Language : Python
- IDE/Workbench : Pycharm/ VS Code
- Language Distribution : Anaconda
- Libraries :  
pyttsx3  
cv2  
datetime  
speechrecognition  
wikipedia  
webbrowser  
random  
wolframalpha  
winshell  
pywhatkit  
request  
os  
tkinter

### HARDWARE REQUIREMENTS

- Processor : Pentium i3
- Hard Disk : 40 GB or more
- RAM : 1 GB or more
- System Type : 32 Bit OS

# TECHNOLOGY DESCRIPTION

## 13. TECHNOLOGY DESCRIPTION

### Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

### WHY PYTHON IS BETTER FOR WRITING AN AI SCRIPT

Python is often compared to other interpreted languages such as Java, JavaScript, Perl, TCL, or Smalltalk. Comparisons to C++, Common Lisp and Scheme can also be enlightening. In this section I will briefly compare Python to each of these languages. These comparisons concentrate on language issues only. In practice, the choice of a programming language is often dictated by other real-world constraints such as cost, availability, training, and prior investment, or even emotional attachment. Since these aspects are highly variable, it seems a waste of time to consider them much for this comparison.

## Java

Python programs are generally expected to run slower than Java programs, but they also take much less time to develop. Python programs are typically 3-5 times shorter than equivalent Java programs. This difference can be attributed to Python's built-in high-level data types and its dynamic typing. For example, a Python programmer wastes no time declaring the types of arguments or variables, and Python's powerful polymorphic list and dictionary types, for which rich syntactic support is built straight into the language, find a use in almost every Python program. Because of the run-time typing, Python's run time must work harder than Java's. For example, when evaluating the expression  $a+b$ , it must first inspect the objects  $a$  and  $b$  to find out their type, which is not known at compile time. It then invokes the appropriate addition operation, which may be an overloaded user-defined method. Java, on the other hand, can perform an efficient integer or floating point addition, but requires variable declarations for  $a$  and  $b$ , and does not allow overloading of the  $+$  operator for instances of user-defined classes.

For these reasons, Python is much better suited as a "glue" language, while Java is better characterized as a low-level implementation language. In fact, the two together make an excellent combination. Components can be developed in Java and combined to form applications in Python; Python can also be used to prototype components until their design can be "hardened" in a Java implementation. To support this type of development, a Python implementation written in Java is under development, which allows calling Python code from Java and vice versa. In this implementation, Python source code is translated to Java bytecode (with help from a run-time library to support Python's dynamic semantics).

## Javascript

Python's "object-based" subset is roughly equivalent to JavaScript. Like JavaScript (and unlike Java), Python supports a programming style that uses simple functions and variables without engaging in class definitions. However, for JavaScript, that's all there is. Python, on the other hand, supports writing much larger programs and better code reuse through a true object-oriented programming style, where classes and inheritance play an important role.

## Perl

Python and Perl come from a similar background (Unix scripting, which both have long outgrown), and sport many similar features, but have a different philosophy. Perl emphasizes support for common application-oriented tasks, e.g. by having built-in regular expressions, file scanning and report generating features. Python emphasizes support for common programming methodologies

such as data structure design and object-oriented programming, and encourages programmers to write readable (and thus maintainable) code by providing an elegant but not overly cryptic notation. As a consequence, Python comes close to Perl but rarely beats it in its original application domain; however Python has applicability well beyond Perl's niche.

## **TCL**

Like Python, TCL is usable as an application extension language, as well as a stand-alone programming language. However, TCL, which traditionally stores all data as strings, is weak on data structures, and executes typical code much slower than Python. TCL also lacks features needed for writing large programs, such as modular namespaces. Thus, while a "typical" large application using TCL usually contains TCL extensions written in C or C++ that are specific to that application, an equivalent Python application can often be written in "pure Python". Of course, pure Python development is much quicker than having to write and debug a C or C++ component. It has been said that TCL's one redeeming quality is the Tk toolkit. Python has adopted an interface to Tk as its standard GUI component library.

TCL 8.0 addresses the speed issue by providing a bytecode compiler with limited data type support, and adds namespaces. However, it is still a much more cumbersome programming language.

## **Smalltalk**

Perhaps the biggest difference between Python and Smalltalk is Python's more "mainstream" syntax, which gives it a leg up on programmer training. Like Smalltalk, Python has dynamic typing and binding, and everything in Python is an object. However, Python distinguishes built-in object types from user-defined classes, and currently doesn't allow inheritance from built-in types. Smalltalk's standard library of collection data types is more refined, while Python's library has more facilities for dealing with Internet and WWW realities such as email, HTML and FTP.

Python has a different philosophy regarding the development environment and distribution of code. Where Smalltalk traditionally has a monolithic "system image" which comprises both the environment and the user's program, Python stores both standard modules and user modules in individual files which can easily be rearranged or distributed outside the system. One consequence is that there is more than one option for attaching a Graphical User Interface (GUI) to a Python program, since the GUI is not built into the system.



## C++

Almost everything said for Java also applies for C++, just more so: where Python code is typically 3-5 times shorter than equivalent Java code, it is often 5-10 times shorter than equivalent C++ code! Anecdotal evidence suggests that one Python programmer can finish in two months what two C++ programmers can't complete in a year. Python shines as a glue language, used to combine components written in C++.

## Common Lisp and Scheme

These languages are close to Python in their dynamic semantics, but so different in their approach to syntax that a comparison becomes almost a religious argument: is Lisp's lack of syntax an advantage or a disadvantage? It should be noted that Python has introspective capabilities similar to those of Lisp, and Python programs can construct and execute program fragments on the fly. Usually, real-world properties are decisive: Common Lisp is big (in every sense), and the Scheme world is fragmented between many incompatible versions, where Python has a single, free, compact implementation.

## Python Features

Python's features include –

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.

- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

# TESTING

## 14.TESTING

Testing is defined in several ways; some of them are as follows:

- The process of executing a system with the intent of finding an error is known as testing.
- Testing is defined as, the process in which defects are identified, isolated, subjected for rectification and ensure that product is defect free in order to produce the quality product and to satisfy customer.
- Quality is defined as justification of the requirements.
- Defect is nothing but deviation from the requirements.
- Defect is nothing but bug.
- Testing can demonstrate the presence of bugs, but not their absence.
- Debugging and Testing is not the same thing.
- Testing is a systematic attribute to break a program.
- Debugging is the art or method of uncovering why the program did not executed properly.

**System testing consists of the following steps:**

1. Program(s) testing.
2. String testing.
3. System testing.
4. System documentation.
5. User acceptance testing.

### **Field Testing:**

This is a special type of testing that may be very important in some projects. Here the system is tested in the actual operational surroundings. The interfaces with other systems and the real world are checked. This type of testing is very rarely used. So far our project is concerned; we haven't tested our project using the field testing.

**Acceptance Testing:**

After the developer has completed all rounds of testing and he is satisfied with the system, then the user takes over and re-tests the system from his point of view to judge whether it is acceptable according to some previously identified criteria. This is almost always a tricky situation in the project because of the inherent conflict between the developer and the user. In this project, it is the job of the bookstores to check the system that whether the made system fulfills the goals or not.

**Why System Testing?**

Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. Inadequate testing results in two types of problems

1. The time lag between the cause and the appearance of the problem.
2. The effect of system errors on the files and records within the system.

Another reason for system testing is its utility as a user-oriented vehicle before implementation.

**Activity Network for System Testing**

The test plan entails the following activities:

1. Prepare test plan.
2. Specify conditions for user acceptance testing.
3. Prepare test data for program testing.
4. Prepare test data for transaction path testing.
5. Plan user training.
6. Compile/assemble programs.
7. Prepare job performance aids.
8. Prepare operational documents.

**Prepare Test**

A workable test plan must be prepared in accordance with established design specifications. It includes the following items:

- Outputs expected from the system.
- Criteria for evaluating outputs.
- A volume of test data.
- Procedure for using test data.
- Personnel and training requirements.

**Specify Conditions for User Acceptance Testing**

Planning for user acceptance testing calls for the analyst and the user to agree on conditions for the test.

**Prepare Test Data for Program Testing**

As each program is coded, test data are prepared and documented to ensure that all aspects of the program are properly tested.

**Prepare Test Data for Transaction Path Testing**

This activity develops the data required for testing every condition and transactions to be introduced into the system. The path of each transaction from origin to destination is carefully tested reliable results.

**Plan User Training**

User training is designed to prepare the user for testing and converting the system. User involvement and training take place parallel with programming for three reasons:

- The system group has time available to spend on training while the programs are being written.
- Initiating a user-training program gives the systems group a clearer image of the user's interest in the new system.
- A trained user participates more effectively in system testing.

The training plan is followed by preparation of the user training manual and other text materials.

**Compile / Assemble Programs**

All programs have to be compiled / assembled for testing.

**Prepare Job Performance Aids**

In this activity the materials to be used by personnel to run the system are specified and scheduled. This includes a display of materials.

**Prepare Operational Documents**

During the test plan stage, all operational documents are finalized including copies of the operational formats required by the candidate system.

**TESTING METHODOLOGIES**

Testing methodologies includes the types of testing, which are as follows:

**1. *Black box testing:***

Black box testing is the process in which tester can perform on an application without having any internal structural knowledge of application.

Usually Test Engineers are involved in the black box testing.

**2. *White box testing:***

White box testing is the process in which tester can perform testing on an application with having internal structural knowledge.

Usually Developers are involved in white box testing.

**3. *Gray box testing:***

Gray box testing is the process in which the combination of black box and white box tonic's are used.

## **STLC (Software Testing Life Cycle)**

### **Test Planning:**

1. Test Plan is defined as a strategic document which describes the procedure how to perform various testing on the total application in the most efficient way.
2. This document involves the scope of testing.
3. Objective of testing.
4. Areas that need to be tested.
5. Areas that should not be tested.
6. Scheduling Resource Planning.
7. Areas to be automated, various testing tools used.

### **Test Development**

1. Test Case Development (Checklist).
2. Test Procedure Preparation (Description of the test cases).
3. Implementation of test cases (Observing the result).

### **Result Analysis:**

1. Expected value; is nothing but expected behaviour of application.
2. Actual Value; is nothing but actual behaviour of application.

**Bug tracing:** collect all the failed cases, prepare documents.

**Reporting:** prepare documents (status of the application).

### **Types of Testing**

#### **○ Smoke Testing:**

It is the process of initial testing in which tester looks for availability of all the functionality of the application in order to perform detailed testing on them. (Main check is for available forms).

#### **○ Sanity Testing:**

It is a type of testing that is conducted on an application initially to check for the proper behaviour of an application that is to check all the functionality are available before the detailed testing is conducted by on them.



- **Regression Testing:**

It is one of the best and important testing. Regression testing is the process in which the functionality, which is already tested before, is once again tested whenever some new change is added in order to check whether the existing functionality remains same.

- **Re-Testing:**

It is the process in which testing is performed on some functionality which is already tested before to make sure that the defects are reproducible and to rule out the environments issues if at all any defects are there.

- **Static Testing:**

It is testing, which is performed on an application when it is not been executed. Ex: GUI, Document Testing.

- **Dynamic Testing:**

It is testing which is performed on an application when it is being executed. Ex: Functional testing.

- **Alpha Testing:**

It is a type of user acceptance testing, which is conducted on an application when it is just before released to the customer.

- **Beta Testing:**

It is a type of UAT that is conducted on an application when it is released to the customer, when deployed in to the real time environment and being accessed by the real time users.

- **Monkey Testing:**

It is the process in which abnormal operations, beyond capacity operations are done on the application to check the stability of it in spite of user's abnormal behaviour.

- **Compatibility Testing:**

It is the testing process in which usually the products are tested on the environments with different combinations of databases (application servers, browsers, etc). In order to check how far the product is compatible with all these environments platform combination.

- **Installation Testing:**

It is the process of testing in which the tester try to install or try to deploy the module into the corresponding environment by following the guidelines produced in the deployment document and check whether the installation is successful or not.

- **Adhoc Testing:**

It is the process of testing in which unlike the formal testing where in test case document is used, without that test case document testing can be done of an application, to cover that testing of the future which are not covered in that test case document. Also it is intended to perform GUI testing which may involve the osmotic issues.

### **13.1 TCD (Test Case Document):**

#### **Test Case Document Contains**

- **Test Scope or Test Objective**
- **Test Scenario**
- **Test Procedure**
- **Test case**

This is the sample test case document for the advertise details of Blazon Agency Project:

#### **Test Scope:**

- Test coverage is provided for the screen “Login check” form of a Administration module of Blazon Agency application.
- Areas of the application to be tested.

#### **Test Scenario:**

- When the office personals use this screen for the payment, tariff and company registration and web advertisements on basis of requirements and quit the form.

**Test Procedure:**

- The procedure for testing this screen is planned in such a way that the data entry, status calculation functionality, saving and quitting operations are testing in terms of GUI testing. Positive testing, Negative testing using the corresponding GUI test cases, Positive test cases, Negative test cases respectively.

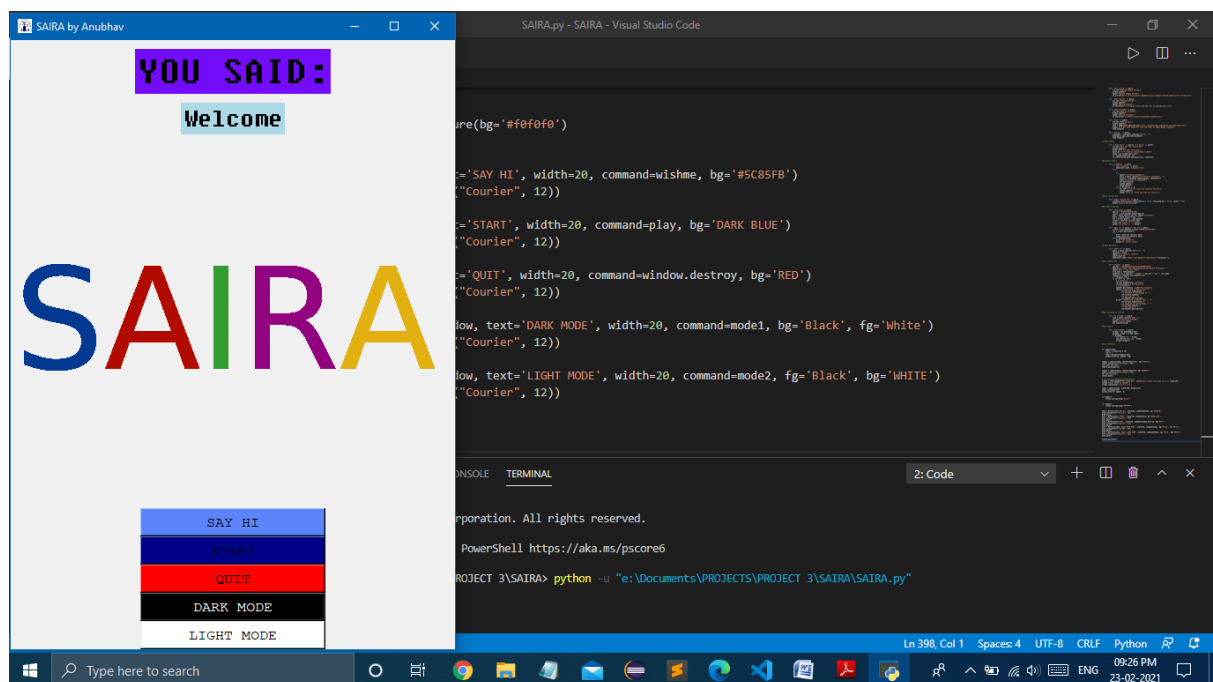
# OUTPUT SCREENS

## 15. SCREENS

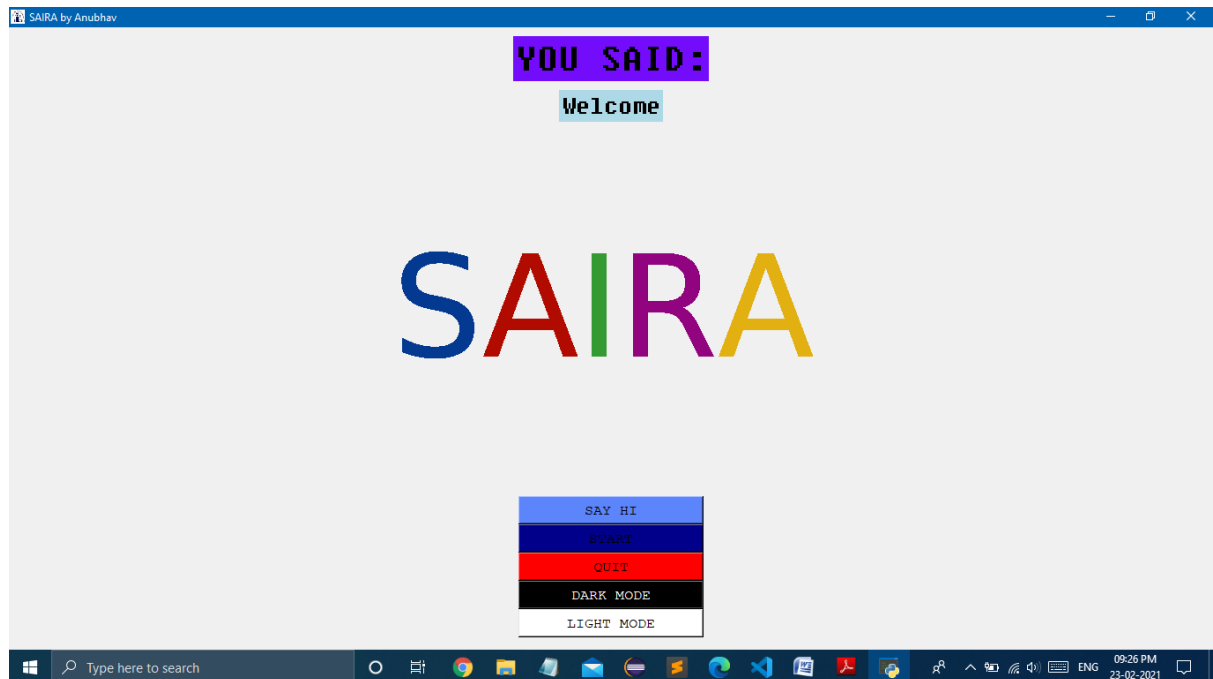
### 1. MAIN SCREEN IMAGE

# SAIRA

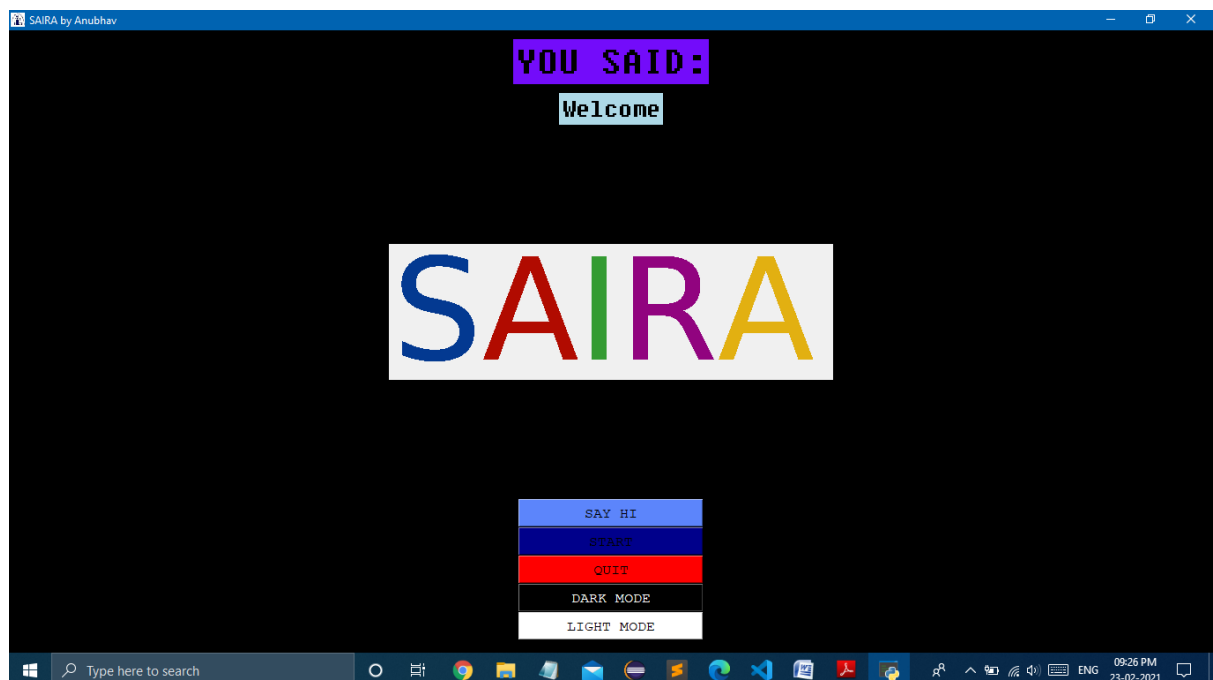
### 2. MINI SCREEN GUI WHILE RUN



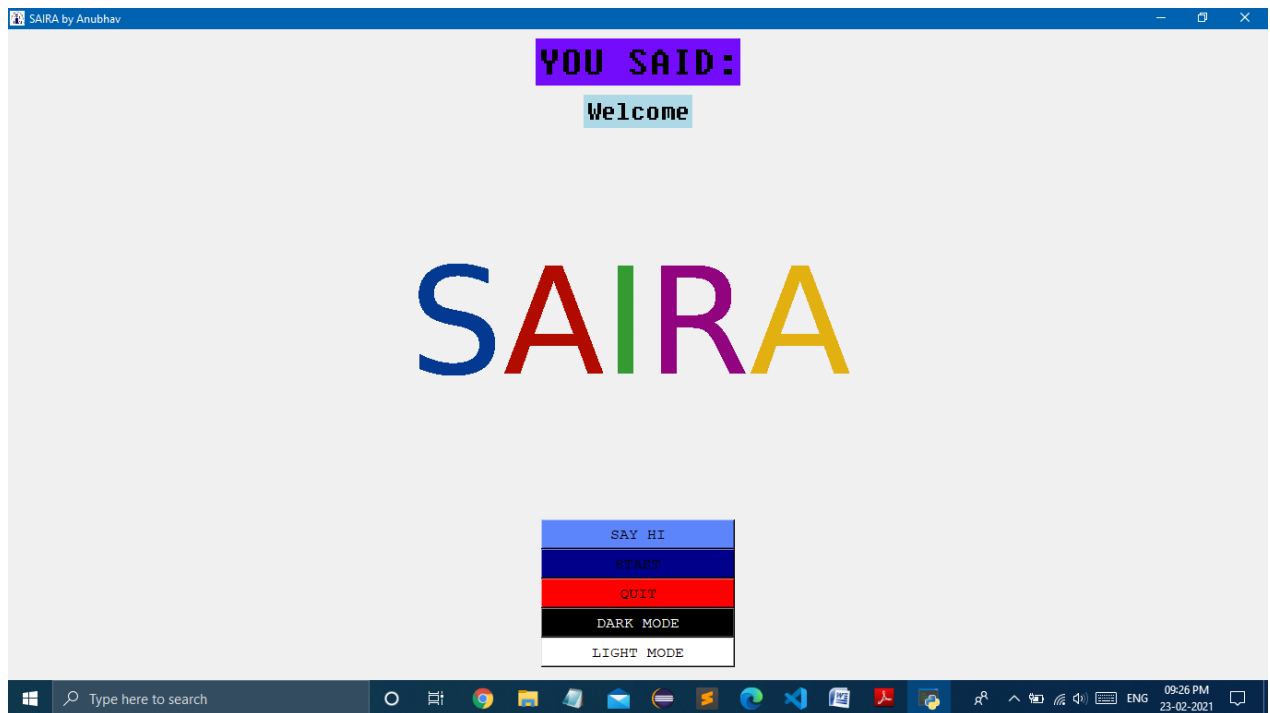
### 3. FULLSCREEN HOME WINDOW



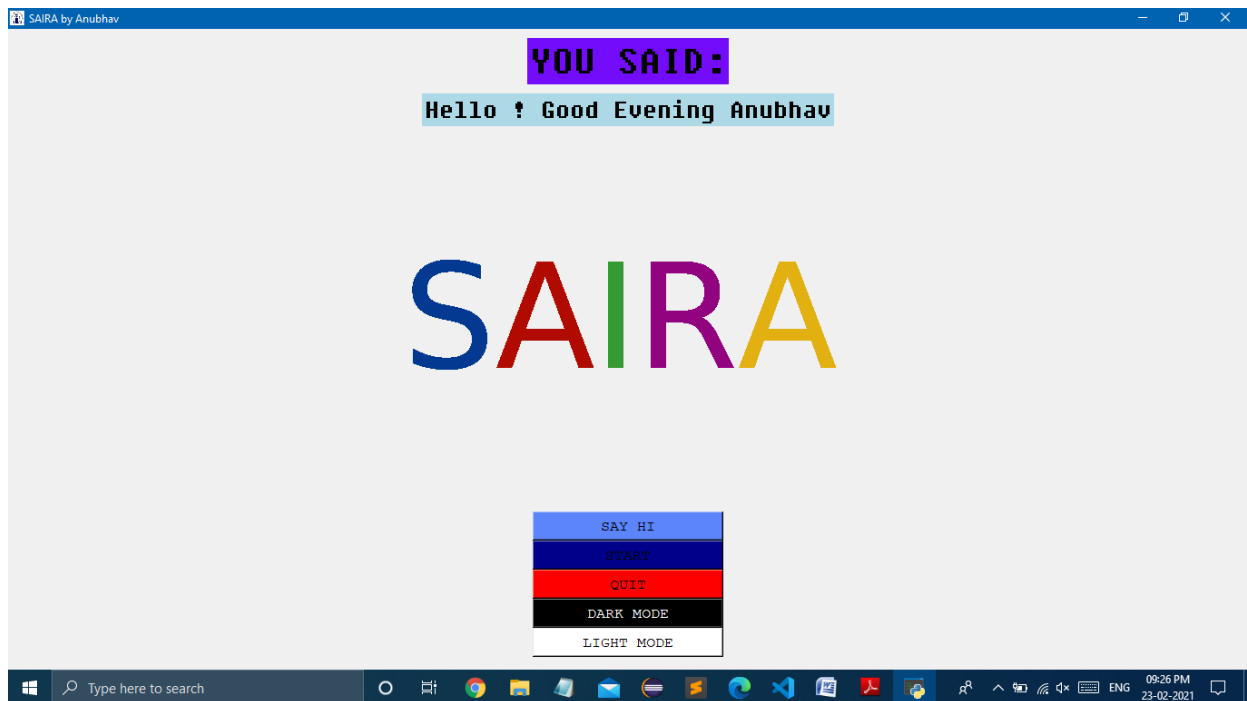
### 4. DARK MODE



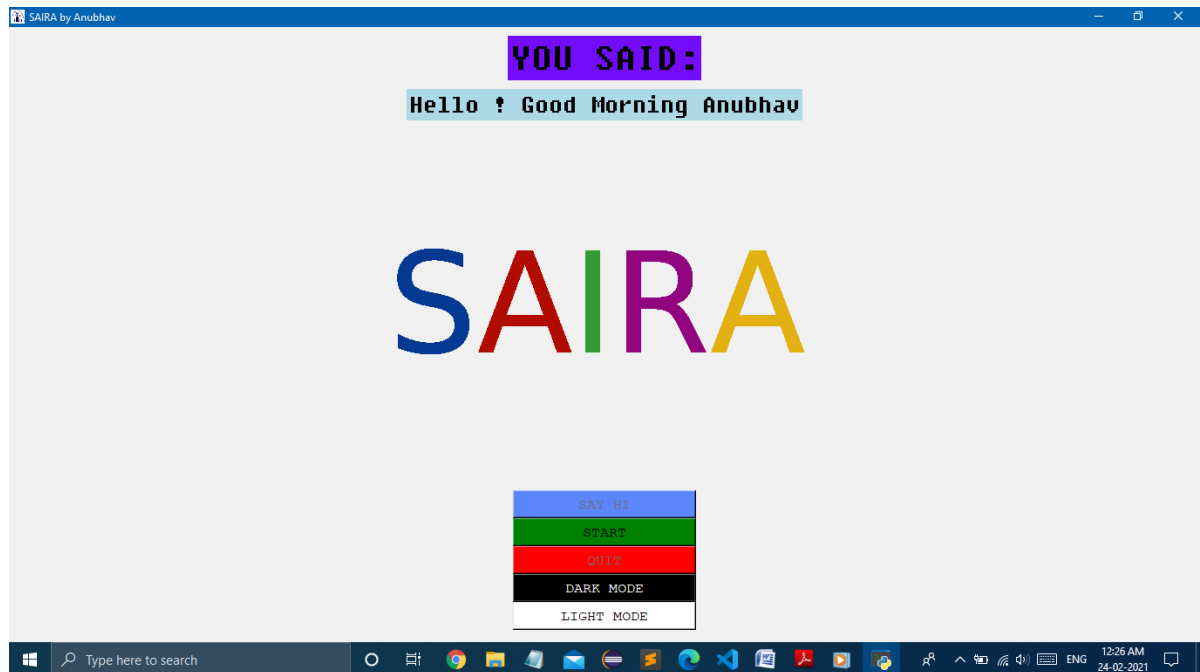
## 5. LIGHT MODE



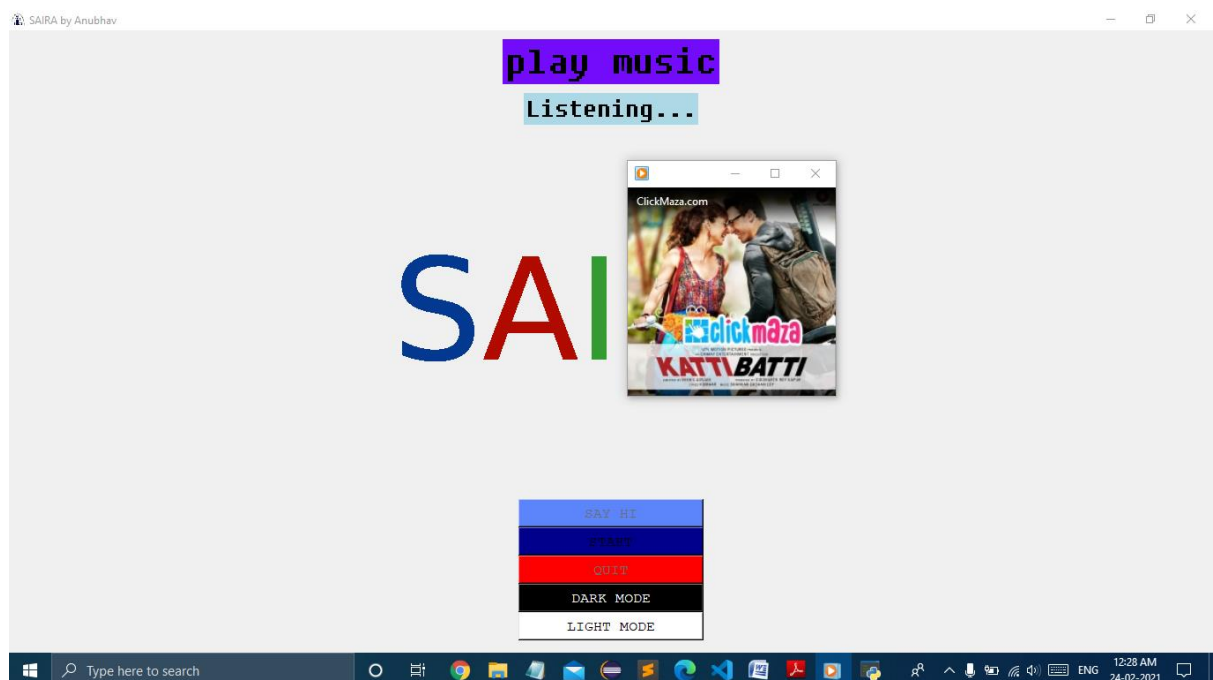
## 6. SAY HI



## 7. COLOR CHANGE WHILE PRESSING START BUTTON AND SAY HI AND QUIT WILL GET DISABLED:

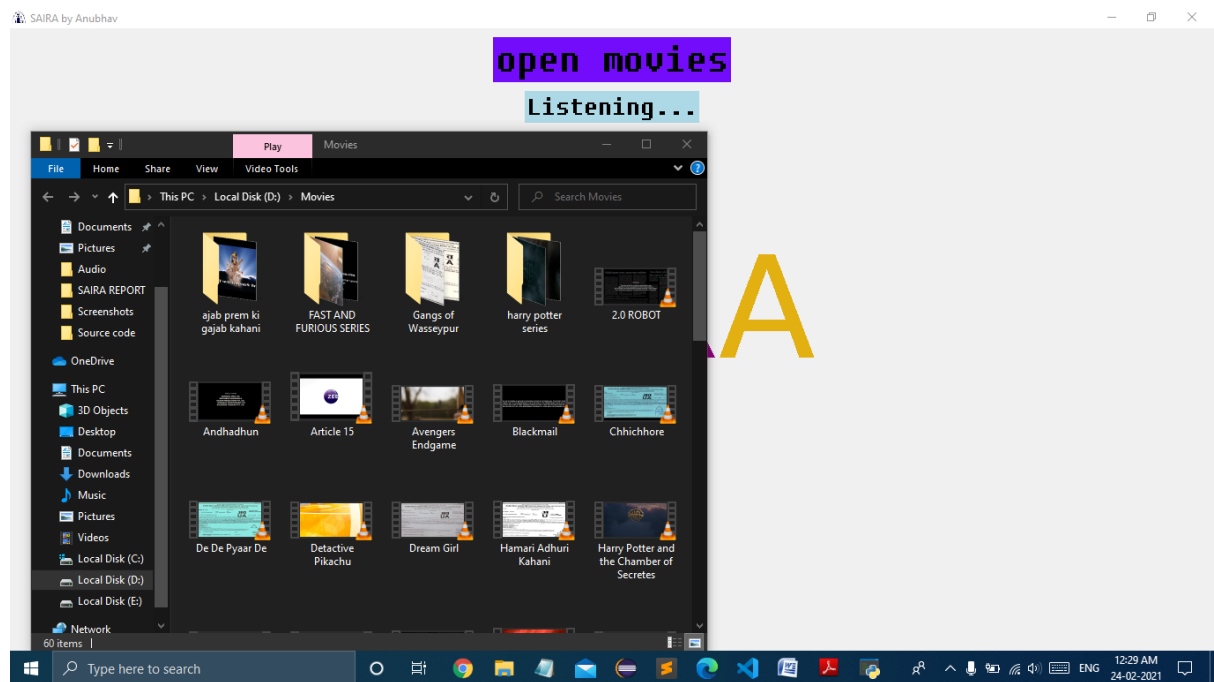


## 8. OPEN MUSIC

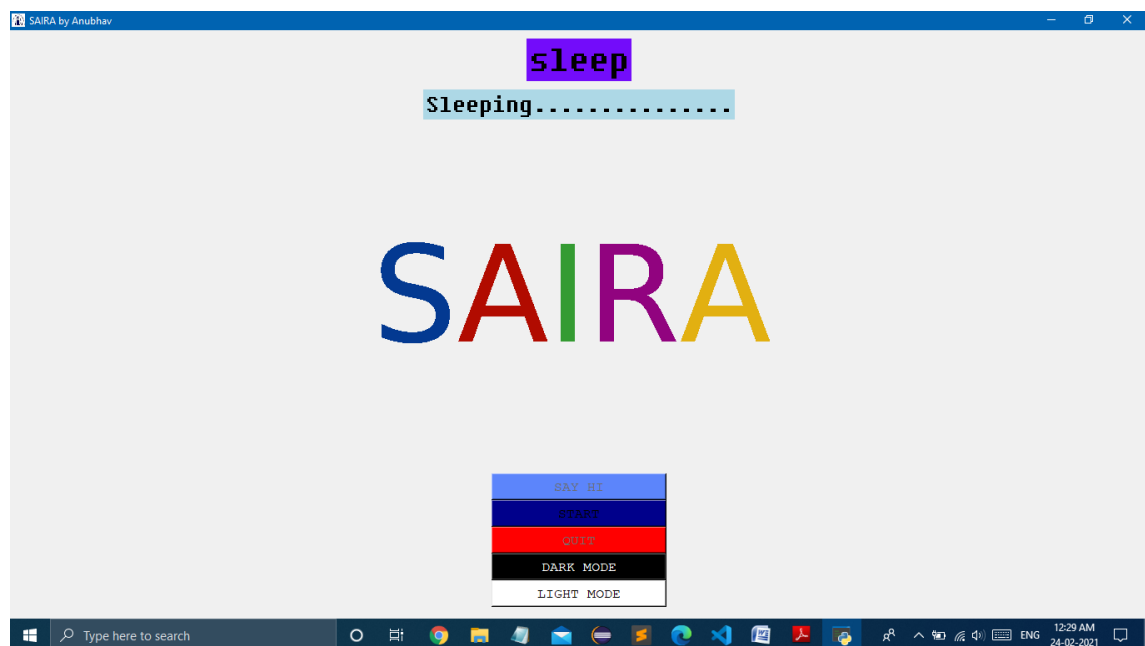




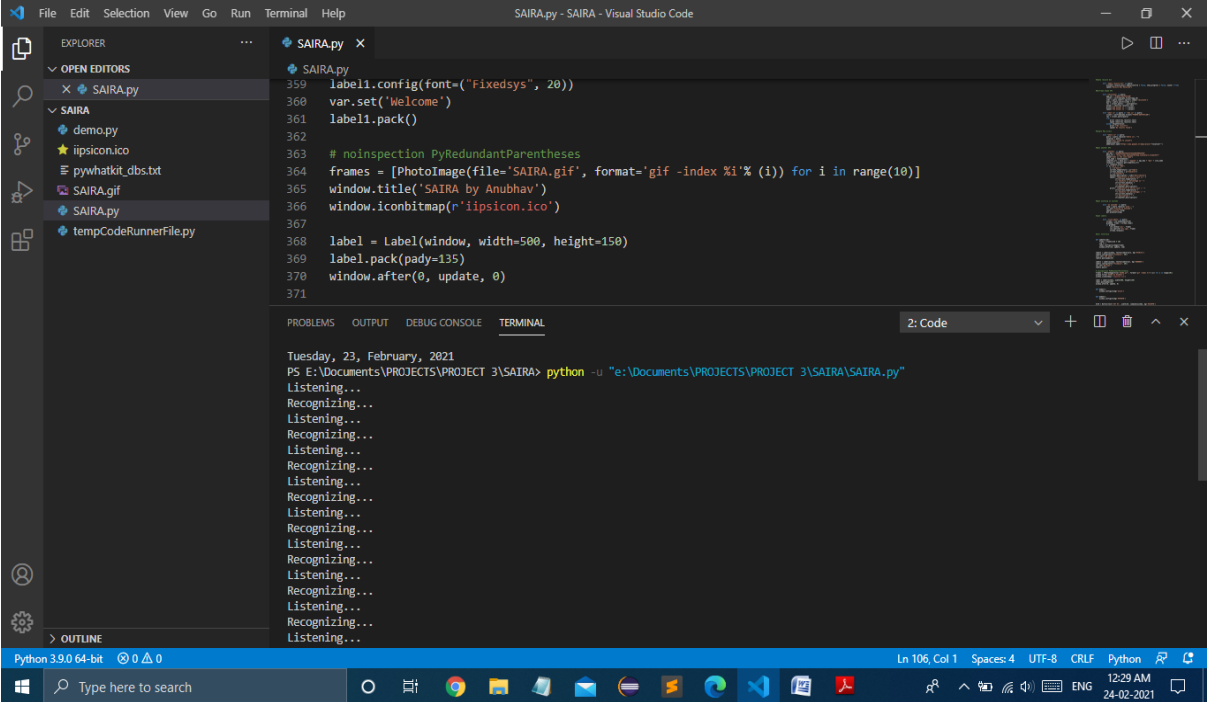
## 9. OPEN MOVIES



## 10. SLEEP



## 11. BACKEND PROCESSING



The screenshot displays the Visual Studio Code interface with the file `SAIRA.py` open. The Explorer sidebar on the left shows the project structure, including `SAIRA.py`, `demo.py`, `iipsicon.ico`, `pywhatkit_dbstxt`, `SAIRA.gif`, `SAIRA.py`, and `tempCodeRunnerFile.py`. The main editor window shows the code for `SAIRA.py`, which includes a Tkinter window titled "SAIRA by Anubhav" and a loop that displays a GIF for 10 frames. The terminal at the bottom shows the command `python -u "e:\Documents\PROJECTS\PROJECT 3\SAIRA\SAIRA.py"` being executed, resulting in a series of "Listening..." and "Recognizing..." messages.

```
SAIRA.py
359 label1.config(font=("Fixedsys", 20))
360 var.set("Welcome")
361 label1.pack()
362
363 # noinspection PyRedundantParentheses
364 frames = [PhotoImage(file='SAIRA.gif', format='gif -index %i' % (i)) for i in range(10)]
365 window.title('SAIRA by Anubhav')
366 window.iconbitmap(r'iipsicon.ico')
367
368 label = Label(window, width=500, height=150)
369 label.pack(pady=135)
370 window.after(0, update, 0)
371
```

```
Tuesday, 23, February, 2021
PS E:\Documents\PROJECTS\PROJECT 3\SAIRA> python -u "e:\Documents\PROJECTS\PROJECT 3\SAIRA\SAIRA.py"
Listening...
Recognizing...
Listening...
Recognizing...
Listening...
Recognizing...
Listening...
Recognizing...
Listening...
Recognizing...
Listening...
Recognizing...
Listening...
Recognizing...
Listening...
Recognizing...
Listening...
Recognizing...
Listening...
Recognizing...
Listening...
```

# **SOURCE CODE**

## 16.SOURCE CODE

### MODULES TO IMPORT

```
1  from tkinter import *
2  import tkinter as tk
3  import cv2
4  import pyttsx3
5  import datetime
6  import speech_recognition as sr
7  import wikipedia
8  import webbrowser
9  import random
10 import os
11 import winshell
12 import requests
13 import wolframalpha
14 import pywhatkit as pwt
```

### SETTING UP THE VOICE

Sapi5 is the voice we are using in this project where voice 0 is of male and voice 1 is of female.

```
16 engine = pyttsx3.init('sapi5')
17 voices = engine.getProperty('voices')
18 engine.setProperty('voice', voices[1].id)
19
20 window = tk.Tk()
21
22 global var
23 global var1
24
25 var = StringVar()
26 var1 = StringVar()
27
28
29 def speak(audio):
30     engine.say(audio)
31     engine.runAndWait()
32
33
```

## WISHME BUTTON

```
34 def wishme():
35     hour = int(datetime.datetime.now().hour)
36     if 0 <= hour <= 12:
37         var.set("Hello ! Good Morning Anubhav")
38         window.update()
39         speak("Hello ! Good Morning Anubhav!")
40     elif 12 <= hour <= 18:
41         var.set("Hello ! Good Afternoon Anubhav!")
42         window.update()
43         speak("Hello ! Good Afternoon Anubhav!")
44     else:
45         var.set("Hello ! Good Evening Anubhav")
46         window.update()
47         speak("Hello ! Good Evening Anubhav!")
48     speak("Myself SAIRA! How may I help you sir")
49
50
```

## SETTING UP SPEECH RECCOGNITION

```
51 def takeCommand():
52     r = sr.Recognizer()
53     with sr.Microphone() as source:
54         var.set("Listening...")
55         window.update()
56         print("Listening...")
57         r.pause_threshold = 1
58         r.energy_threshold = 400
59         audio = r.listen(source)
60     try:
61         var.set("Recognizing...")
62         window.update()
63         print("Recognizing...")
64         query = r.recognize_google(audio, language='en-in')
65     except Exception as e:
66         speak("Pardon me, please say that again")
67         return "None"
68     var1.set(query)
69     window.update()
70     return query
71
72
```

## START BUTTON

### EXIT TASK

```

73  def play():
74      btn2['state'] = 'disabled'
75      btn0['state'] = 'disabled'
76      btn1.configure(bg='green')
77      wishme()
78      while True:
79          btn1.configure(bg='dark blue')
80          query = takeCommand().lower()
81          if 'exit' in query:
82              var.set("Bye sir")
83              speak("Bye sir")
84              btn1.configure(bg='#5C85FB')
85              btn2['state'] = 'normal'
86              btn0['state'] = 'normal'
87              window.update()
88              break
89

```

### GENERAL CONVERSATION

```

92      elif 'hello' in query:
93          var.set('Hello Sir')
94          window.update()
95          speak("Hello Sir")
96
97      elif 'thank you' in query:
98          var.set("Welcome Sir")
99          window.update()
100         speak("Welcome Sir")
101
102      elif ('old are you' in query) or ('version' in query):
103          var.set("Version 0.1.1 ")
104          window.update()
105          speak("I am a newbie sir ! Version 0.1.1")
106
107      elif 'your name' in query:
108          var.set("Myself SAIRA")
109          window.update()
110          speak("Myself SAIRA")
111
112      elif 'who made you' in query:
113          var.set("My Creator is Anubhav Singh")
114          window.update()
115          speak("My Creator is Anubhav Singh")
116
117      elif 'sleep' in query:
118          var.set('Sleeping.....')
119          window.update()
120          speak("OK Anubhav!! time to sleep have a good day")
121          quit()
122

```

## SYSTEM DATE AND TIME

```
123 #System date and time
124     elif 'time' in query:
125         strtime = datetime.datetime.now().strftime("%I %M %S %p")
126         var.set("Sir the time is %s" % strtime)
127         window.update()
128         speak("Sir the time is %s" % strtime)
129
130     elif 'date' in query:
131         strdate = datetime.datetime.today().strftime("%d %m %y")
132         var.set("Sir today's date is %s" % strdate)
133         window.update()
134         speak("Sir today's date is %s" % strdate)
135
```

## SYSTEM FOLDERS AND SOFTWARES

```
136 #open system folders and softwares
137
138     elif 'open movies' in query:
139         var.set("Opening Movies")
140         window.update()
141         speak("Opening Movies")
142         os.startfile("D:\\Movies")
143
144     elif 'open software' in query:
145         var.set("Opening Software")
146         window.update()
147         speak("Opening Software")
148         os.startfile("D:\\Software")
149
150     elif 'series' in query:
151         var.set("Opening OTD Series")
152         window.update()
153         speak("Opening OTD Series")
154         os.startfile("E:\\OTD")
155
156     elif 'code' in query:
157         var.set("Opening V S Code")
158         window.update()
159         speak("Opening V S Code")
160         os.startfile("C:\\Users\\Lenovo\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe")
161
```

## SOFTWARES AND WEBSITES

```

162         elif 'open youtube' in query:
163             var.set('opening Youtube')
164             window.update()
165             speak('opening Youtube')
166             webbrowser.open("youtube.com")
167
168         elif 'open google' in query:
169             var.set('opening google')
170             window.update()
171             speak('opening google')
172             webbrowser.open("google.com")
173
174         elif 'open stackoverflow' in query:
175             var.set('opening stackoverflow')
176             window.update()
177             speak('opening stackoverflow')
178             webbrowser.open('stackoverflow.com')
179
180         elif 'open github' in query:
181             var.set('opening github')
182             window.update()
183             speak('opening github')
184             webbrowser.open('https://github.com/anubhavv1998')
185

```

## SOFTWARES

```

186         elif 'open media player' in query:
187             var.set("opening VLC media Player")
188             window.update()
189             speak("opening V L C media player")
190             os.startfile("C:\\Program Files (x86)\\VideoLAN\\VLC\\vlc.exe")
191
192         elif 'open python' in query:
193             var.set("Opening Python")
194             window.update()
195             speak("Opening Python")
196             os.startfile("C:\\Users\\Lenovo\\AppData\\Local\\Programs\\Python\\Python39\\python.exe")
197
198         elif 'open chrome' in query:
199             var.set("Opening Google Chrome")
200             window.update()
201             speak("Opening Google Chrome")
202             os.startfile("C:\\Users\\Lenovo\\AppData\\Local\\Google\\Chrome\\Application\\chrome.exe")
203
204         elif 'open sublime' in query:
205             var.set('Opening Sublime')
206             window.update()
207             speak('opening Sublime')
208             os.startfile("C:\\Program Files\\Sublime Text 3\\sublime_text.exe")
209

```



## SOFTWARES, NEWS AND SEARCH TASK

```
210 elif 'open anaconda' in query:
211     var.set('Opening Anaconda')
212     window.update()
213     speak('opening anaconda')
214     os.startfile("C:\\Users\\Lenovo\\anaconda3\\python.exe")
215
216 elif 'news' in query:
217     var.set('Opening news')
218     window.update()
219     news = webbrowser.open_new_tab("https://timesofindia.indiatimes.com/home/headlines")
220     speak('Here are some headlines from the Times of India,Happy reading')
221     time.sleep(6)
222
223 elif 'search' in query:
224     statement = statement.replace("search", "")
225     webbrowser.open_new_tab(statement)
226     time.sleep(5)
227
```

## PLAY MUSIC BY RANDOM SELECT

```
228 #random module
229
230 elif ('play music' in query) or ('music' in query):
231     var.set('Here are your favorites')
232     window.update()
233     speak('Here are your favorites')
234     music_dir = 'C:\\Users\\Lenovo\\Music\\Audio'
235     songs = os.listdir(music_dir)
236     n = random.randint(0, 71)
237     os.startfile(os.path.join(music_dir, songs[n]))
238
```

## WIKIPEDIA

```
239 #Wikipedia module
240
241     elif 'wikipedia' in query:
242         if 'open wikipedia' in query:
243             webbrowser.open('wikipedia.com')
244         else:
245
246             try:
247                 speak("searching wikipedia")
248                 query = query.replace("according to wikipedia", "")
249                 results = wikipedia.summary(query, sentences=1)
250                 speak("According to wikipedia")
251                 var.set(results)
252                 window.update()
253                 speak(results)
254             except Exception as e:
255                 var.set('sorry sir could not find any results')
256                 window.update()
257                 speak('sorry sir could not find any results')
258
```

## EMPTY RECYCLE BIN

```
259 #empty recycle bin
260
261     elif 'empty recycle bin' in query:
262         winshell.recycle_bin().empty(confirm = False, show_progress = False, sound = True)
263         speak("Recycle Bin Recycled")
264
```

## WOLFRAM ALPHA TO SEARCH FOR WHAT, WHO AND CALCULATE

```
265 #Wolframe Alpha API
266
267     elif "calculate" in query:
268         app_id = "PTW6PG-X63YYGTL6W"
269         client = wolframalpha.Client(app_id)
270         indx = query.lower().split().index('calculate')
271         query = query.split()[indx + 1:]
272         res = client.query(' '.join(query))
273         answer = next(res.results).text
274         print("The answer is " + answer)
275         speak("The answer is " + answer)
276
277     elif "what is" in query or "who is" in query:
278         client = wolframalpha.Client("PTW6PG-X63YYGTL6W")
279         res = client.query(query)
280         try:
281             print (next(res.results).text)
282             speak (next(res.results).text)
283         except StopIteration:
284             print ("No results")
285             speak('No results found')
286
```

## GOOGLE MAP

```
287 #Google Map access
288
289     elif "where is" in query:
290         query = query.replace("where is", "")
291         location = query
292         speak("User asked to Locate")
293         speak(location)
294         webbrowser.open("https://www.google.nl/maps/place/"+location+"")
295
```

## OPEN WEATHER

```
296 #Open weather API
297
298 elif 'weather' in query:
299     api_key = "659d4ba65a35ab7af4a4a6926062d70e"
300     base_url = "http://api.openweathermap.org/data/2.5/weather?"
301     speak("what is the city name")
302     city_name = takeCommand()
303     complete_url = base_url + "appid=" + api_key + "&q=" + city_name
304     response = requests.get(complete_url)
305     x = response.json()
306     if x["cod"] != "404":
307         y = x["main"]
308         current_temperature = y["temp"]
309         current_humidiy = y["humidity"]
310         z = x["weather"]
311         weather_description = z[0]["description"]
312         speak(" Temperature in kelvin unit is " +
313               str(current_temperature) +
314               "\n humidity in percentage is " +
315               str(current_humidiy) +
316               "\n description  " +
317               str(weather_description))
318         print(" Temperature in kelvin unit = " +
319               str(current_temperature) +
320               "\n humidity (in percentage) = " +
321               str(current_humidiy) +
322               "\n description = " +
323               str(weather_description))
```

## OPEN ANYTHING ON YOUTUBE

```
325 #Open anything on youtube
326
327 elif 'on youtube' in query:
328     song = query.replace('play','')
329     var.set('Playing on Youtube')
330     speak('playing'+song)
331     pwt.playonyt(song)
332
```

## OPEN CAMERA AND CLICK PHOTO

```
333 #open camera
334
335     elif 'click photo' in query:
336         stream = cv2.VideoCapture(0)
337         grabbed, frame = stream.read()
338         if grabbed:
339             cv2.imshow('pic', frame)
340             cv2.imwrite('pic.jpg', frame)
341             stream.release()
```

## USER INTERFACE

### SETTING FRAMES AND BUTTONS

```
343 #User Interface
344
345
346 def update(ind):
347     frame = frames[ind % 10]
348     ind += 1
349     label.configure(image=frame)
350     window.after(10, update, ind)
351
352
353 label2 = Label(window, textvariable=var1, bg='#730cfa')
354 label2.config(font=("Fixedsys", 30))
355 var1.set('YOU SAID:')
356 label2.pack(pady=10)
357
358 label1 = Label(window, textvariable=var, bg='#ADD8E6')
359 label1.config(font=("Fixedsys", 20))
360 var.set('Welcome')
361 label1.pack()
362
```

## SETTING GIF IN GUI WITH TITLE AND ICON

```
363 # noinspection PyRedundantParentheses
364 frames = [PhotoImage(file='SAIRA.gif', format='gif -index %i' % (i)) for i in range(10)]
365 window.title('SAIRA by Anubhav')
366 window.iconbitmap(r'iipsicon.ico')
367
368 label = Label(window, width=500, height=150)
369 label.pack(pady=135)
370 window.after(0, update, 0)
371
372
```

## SETTING MODE (DARK AND LIGHT)

```
373 def mode1():
374     window.configure(bg='black')
375
376
377 def mode2():
378     window.configure(bg='#f0f0f0')
379
380
```

## SETTING UP ALL THE BUTTONS

```
381 btn0 = Button(text='SAY HI', width=20, command=wishme, bg='#5C85FB')
382 btn0.config(font=("Courier", 12))
383 btn0.pack()
384 btn1 = Button(text='START', width=20, command=play, bg='DARK BLUE')
385 btn1.config(font=("Courier", 12))
386 btn1.pack()
387 btn2 = Button(text='QUIT', width=20, command=window.destroy, bg='RED')
388 btn2.config(font=("Courier", 12))
389 btn2.pack()
390 btn3 = Button(window, text='DARK MODE', width=20, command=mode1, bg='Black', fg='White')
391 btn3.config(font=("Courier", 12))
392 btn3.pack()
393 btn4 = Button(window, text='LIGHT MODE', width=20, command=mode2, fg='Black', bg='WHITE')
394 btn4.config(font=("Courier", 12))
395 btn4.pack()
396
397 window.mainloop()
```

# GANTT CHART

## 17. Gantt Chart

|                |        |        |        |        |        |
|----------------|--------|--------|--------|--------|--------|
| Analysis       | 30 NOV |        |        |        |        |
| Design         |        | 10 DEC |        |        |        |
| Implementation |        |        | 30 DEC |        |        |
| Testing        |        |        |        | 20 JAN |        |
| Report Making  |        |        |        |        | 20 FEB |



# **FUTURE ENHANCEMENT**

## 18. FUTURE ENHANCEMENTS

### Future Enhancements:

It is not possible to develop a system that makes all the requirements of the user. User requirements keep changing as the system is being used. Some of the future enhancements that can be done to this system are:

- As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment.
- It is based on object-oriented design, any further changes can be easily adaptable.
- Based on the future security issues, security can be improved using emerging technologies.

With time and with increase of processing speed of system we can enhance capacity of SAI'RA. Since, this is a window's based AI so; I will try to in co-operate SAI'RA with Linux operating system. Since, we have virtual assistant in all other operating system but not for Linux operating system. It will be a difficult task to do but I will try my best to do such thing.

# CONCLUSION

## 19. CONCLUSION

The SAIRA Virtual assistant has been computed successfully and was also tested successfully by taking “test cases”. It is user friendly and has required options, which can be utilized by the user to perform the desired operations.

The software is developed using PYTHON in Windows environment. The goals that are achieved by software are:

- ✓ Optimum utilization of resources.
- ✓ Efficient management of records.
- ✓ Simplification of the operations.
- ✓ Less processing time and getting required information.
- ✓ User friendly.
- ✓ Portable and flexible for future enhancement.

# **BIBLIOGRAPHY**

## 20. Bibliography

- Automate the Boring Stuff with Python : By Al Sweigar
- Python Courses : Google Python classes
- The Python Standard Library : By Doug Hellmann
- Online Resources :  
Geeks for Geeks  
Python.org  
Python community  
Python libraries  
youtube.com  
google.com  
towardsdatascience.co  
m