**PROJECT REPORT**

**ON**

**SPEECH RECOGNITION**

Submitted by

**Name: JAGADISH JENA**

**University Roll No: 37UITM206025**

*for the partial fulfillment of the requirements for the degree of*

**Bachelor of Science in Information Technology & Management**

**(B.Sc. ITM)**

****

**LAKSHYA INSTITUTE OF TECHNOLOGY**

**M4/46, near water tank, Acharya Vihar, Bhubaneswar, Odisha 751013**

**YEAR: - 2023**

****

**CERTIFICATE**

***I hereby declare that the work which is being presented in the project report entitled “SPEECH RECOGNITION****”,* ***in partial fulfillment of the requirements for the award of******Degree of Bachelor of Computer Science (B.Sc.-Computer Science) submitted to the Department of Computer Science, Utkal University, Bhubaneswar is an authentic record of my own work. The content presented in this project report has not been submitted for the award of any other degree of this or any other University.***

***NAME: JAGADISH JENA***

***(Regd. No : )***

***It is certified that the above statement made by the student is correct to the best of my knowledge and belief.***

***(Mr.) Chandra K. Rout***

***Senior Lecturer***

*Countersigned by:*

**Faculty In-Charge Head of The Department**

**External Examiner**

ACKNOWLEDGEMENT

**I would like to thank Mr**. ***Chandra K. Rout*** , **Senior Lecturer, Department of Computer Science for the opportunity to work with him, for his invaluable guidance, encouragement, suggestions and untiring support. He has been an advisor in true sense both academically and morally though out this project work.**

**I am highly grateful to our Chairman Mr. *Susant K. Rout***, **for providing all the necessary infrastructure to complete my Project Work.**

**I extend my sincere thanks to Mr. *Chandra K. Rout* for their continuous inspiration and intellectual support during this project work.**

**Special thanks to my family members especially to my parents and friends whose uninterrupted love, inspiration and blessings helped me to complete my Project Work.**

**I thank and owe my deepest regards to all who have helped me directly or indirectly.**

**Full Signature of the Student**

**Date: -**

**(Roll No: 37UITM206025)**

**ABSTRACT**

The Project entitled SPEECH RECOGNITION is an application-based Tool that can perform varieties of task on the command ordered by the authenticated user. The project has been planned to be having the view of distributed architecture, with centralized storage of the database. The application for the storage of the data has been planned using the constructs of MySQL database and all the user interfaces have been designed using python supports “tkinter” library. The database connectivity is planned using the Apache server provided by XAMPP software. The standards of security and data protective mechanism have been given a big choice for proper usage. The application takes care of different action within the system capacity, which are ordered as per the command of user that are followed by the software assistant.

As we know Python is an emerging language so it becomes easy to write a script for Voice Assistant in Python. The instructions for the assistant can be handled as per the requirement of user. Speech recognition is the process of converting speech into text. This is commonly used in voice assistants like Alexa, Siri, etc. In Python there is an API called Speech-Recognition which allows us to convert speech into text. It was an interesting task to make my own assistant. It became easier to send emails without typing any word, Searching on Google without opening the browser, and performing many other daily tasks like playing music, opening your favorite IDE with the help of a single voice command. In the current scenario, advancement in technologies are such that they can perform any task with same effectiveness or can say more effectively than us. By making this project, I realized that the concept of AI in every field is decreasing human effort and saving time.

Functionalities of this project include:

1. It can send emails.

2. It can read PDF.

3. It can send text on WhatsApp.

4. It can open command prompt, your favorite IDE, notepad etc.

5. It can play music.

6. It can do Wikipedia searches for you.

7. It can open websites like Google, YouTube, etc., in a web browser.

8. It can give weather forecast.

9. It can give desktop reminders of your choice.

10. It can have some basic conversation.

Now the basic question arises in mind that how it is an AI? The virtual assistant that I have created is like if it is not an A.I, but it is the output of a bundle of the statement. But fundamentally, the mail purpose of A.I machines is that it can perform human tasks with the same efficiency or even more efficiently than humans. It is a fact that my virtual assistant is not a very good example of A.I., but it is an A.I.

Hardware Requirement: -

* Memory: 4 GB or higher
* Processor: Intel Core i3 or higher
* Hard drive: 16 GB or Higher Capacity
* Speaker & Mic: Realtek(R) or Better
* Monitor: 1024\*768 Pixels & Least (high color) 32 bit color system

Software Requirement: -

* Programming Language: Python 3
* Operating System: Microsoft Window XP
* Pycharm Community Edition
* Xammp or MySQL Workbench

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| Table 1.1. | PERT CHART | 25 |
| Table 1.2: | DFD CONSTRUCTION TABLE | 37 |
|  |  |  |

**:**

**:**

**LIST OF FIGURES**

Figure 1.1 PERT CHART 25

Figure 1.2 GANTT CHART 26

Figure 1.3 ER DIAGRAM 30

Figure 1.4 DFD LEVEL 0 DIAGRAM 38

Figure 1.5 DFD LEVEL 1 DIAGRAM 39

Figure 1.7 DFD MANAGE PRODUCTS 40

Figure 1.8 DFD MANAGE ORDERS 40

Figure 1.9 DFD LOGIN 41

Figure 1.10 DFD USER ACTIVITY 41

Figure 1.11 DFD FOR NEW USER REGISTRATION 42

Figure 1.12 USE CASE DIAGRAM 43

Figure 1.13 USE CASE COUSTOMER MANAGEMENT 44

Figure 1.14 USE CASE CONTENT MANAGEMENT 44

Figure 1.15 CONTENT MANAGEMENT 45

Figure 1.16 CLASS DIAGRAM 45

Figure 1.17 ACTIVITY DIAGRAM 46

Figure 1.18 REGISTRATION ACTIVITY DIAGRAM 47

Figure 1.19 ADMIN ACTIVITY DIAGRAM 48

Figure 1.20 USER ACTIVITY DIAGRAM 49

Figure 1.21 OUTPUT SCREENS 50-55

Figure 1.22 SDLC MODEL WATERFALL 80

**TABLE OF CONTENTS**

1. ICE CREAM PARLOUR 1-7

1.1 PREFACE 8

1.2 PURPOSE OF THE PROJECT 9

1.3 OBJECTIVE 10

1.4 INTRODUCTION 11

1.5 SCOPE OF THE PROJECT 14

1.6 THEORITICAL BACKGROUND 16

1.7 SYSTEM ANALYSIS AND DESIGN 21

1.8 PERT CHART 23

1.9 GANTT CHART 27

1.10 ER DIAGRAM 28

1.11 DATA FLOW DIAGRAM 34

1.12 USE CASE DIAGRAM 43

1.13 CLASS DIAGRAM 45

1.14 ACTIVITY DIAGRAM 46

1.15 OUTPUT SCREENS 50

1.16 CODING AND LIBRARIES 53

1.17 PROCESS INVOLBED 67

1.18 WATERFALL MODEL 70

1.19 SOFTWARE LIFE CYCLE 71

1.20 COST AND BENEFIT ANALYSIS 82

1.21 METHODS USED FOR TESTING 85

1.22 USER MANNUAL AND SECURITY POLICY 89

1.24 CONCLUSION 90

1.25 FUTURE DEVELOPMENT 91

1.26 REFERENCE 92

**PREFACE**

The presentation of this project report is an outcome of our hard work, study, and painful efforts taken by us for creating a project "SPEECH RECOGNITION" which forms the part of our B.Sc. ITM by doing work in computerized environment. This project fulfills the necessity of the knowledge.

**PURPOSE OF THE PROEJECT**

The **SPEECH RECOGNITION** help to perform our task by ordering a virtual assistant. It is easier to do all the software operations with some commands to a tool, who physically does not exist. Also it help to use the technology.

It was an interesting task to make my own assistant. It became easier to send emails without typing any word, Searching on Google without opening the browser, and performing many other daily tasks like playing music, opening your favorite IDE with the help of a single voice command. VERONICA is different from other traditional voice assistants in terms that it is specific to desktop and user does not need to make account to use this, it does not require any internet connection while getting the instructions to perform any specific task. The IDE used in this project is PyCharm. All the python files were created in PyCharm and all the necessary packages were easily installable in this IDE. For this project following modules and libraries were used i.e. pyttsx3, SpeechRecognition, Datetime, Wikipedia, Smtplib, pywhatkit, pyjokes, pyPDF2, pyautogui, pyQt etc. I have created a live GUI for interacting with the VERONICA as it gives a design and interesting look while having the conversation. With the advancement VERONICA can perform any task with same effectiveness or can say more effectively than us. By making this project, I realized that the concept of AI in every field is decreasing human effort and saving time. Functionalities of this project include, It can send emails, It can read PDF, It can send text on WhatsApp, It can open command prompt, your favorite IDE, notepad etc., It can play music, It can do Wikipedia searches for you, It can open websites like Google, YouTube, etc., in a web browser, It can give weather forecast, It can give desktop reminders of your choice. It can have some basic conversation.

The main purpose of the system is to help humans to interact with the machines and stepping forward to improve the next generation technology.

**OBJECTIVE**

The **SPEECH RECOGNITION** is a user-friendly tool which has the purpose of providing the site visitor with the information 'What you can get on this site?' This site has been purely built for the purpose of knowledge & to provide convenience for ordering the products just sitting at home. The site provides the visitor with much information along with images.

The objective of this project is to develop a general purpose e-commerce store where any product can be bought from the comfort of home through the Internet. The site provides the visitor with proper links, which will make accessing more efficient or rather to making effort to be more efficient. As its dawn of this new things aren't sound & perfect but I have tried my level best & promise to keep the same in the near future.

I made it easy for the user to access all the information at a click. Along with this the site provides the user with better acknowledging service. It can be a system to assist the customer in purchasing a particular handset by the means of information retrieval. Storing information of members, products, storing latest information about the products to the members. This project allows administrator to store and retrieve data with an instant.

**INTRODUCTION**

Artificial Intelligence when used with machines, it shows us the capability of thinking like humans. In this, a computer system is designed in such a way that typically requires interaction from human. As we know Python is an emerging language so it becomes easy to write a script for Voice Assistant in Python. The instructions for the assistant can be handled as per the requirement of user. Speech recognition is the Alexa, Siri, etc. In Python there is an API called Speech Recognition which allows us to convert speech into text. It was an interesting task to make my own assistant. It became easier to send emails without typing any word, Searching on Google without opening the browser, and performing many other daily tasks like playing music, opening your favorite IDE with the help of a single voice command. In the current scenario, advancement in technologies are such that they can perform any task with same effectiveness or can say more effectively than us. By making this project, I realized that the concept of AI in every field is decreasing human effort and saving time. As the voice assistant is using Artificial Intelligence hence the result that it is providing are highly accurate and efficient. The assistant can help to reduce human effort and consumes time while performing any task, they removed the concept of typing completely and behave as another individual to whom we are talking and asking to perform task. The assistant is no less than a human assistant but we can say that this is more effective and efficient to perform any task. The libraries and packages used to make this assistant focuses on the time complexities and reduces time. The functionalities include , It can send emails, It can read PDF, It can send text on WhatsApp, It can open command prompt, your favorite IDE, notepad etc., It can play music, It can do Wikipedia searches for you, It can open websites like Google, YouTube, etc., in a web browser, It can give weather forecast, It can give desktop reminders of your choice. It can have some basic conversation. Tools and technologies used are PyCharm IDE for making this project, and I created all py files in PyCharm. Along with this I used following modules and libraries in my project. pyttsx3, SpeechRecognition, Datetime, Wikipedia, Smtplib, pywhatkit, pyjokes, pyPDF2, pyautogui, pyQt etc. I have created a live GUI for interacting with the VERONICA as it gives a design and interesting look while having the conversation.

**Problem in existing system**

* The existing system is manual system. Needs to be converted into automated system.
* Risk of mismanagement of data.
* Less Security.
* No proper coordination between different Applications and Users.
* Fewer Users - Friendly.
* Accuracy not guaranteed.
* Not **in** reach of distant users.

**Solution of these problems**

The development of the new system contains the following activities, which try to automate the entire process keeping in view of the database integration approach.

* 1. User friendliness is provided in the application with various controls.

1. The system makes the overall project management much easier and flexible.
2. There is no risk of data mismanagement at any level while the project development is under process.
3. It provides high level of security with different level of authentication.
4. Users from any part of the world can make use of the system.
5. New system will be much better in performance as compared to existing one.

**SCOPE OF THE PROJECT**

* VERONICA, a desktop assistant is a voice assistant that can perform many daily tasks of desktop like playing music, opening your favorite IDE with the help of a single voice command. VERONICA is different from other traditional voice assistants in terms that it is specific to desktop and user does not need to make account to use this, it does not require any internet connection while getting the instructions to perform any specific task.

* 4.1. REAL LIFE APPLICATION
* 4.1.1. Saves time:VERONICA is a desktop voice assistant which works on the voice command offered to it, it can do voice searching, voice-activated device control and can let us complete a set of tasks.
* 4.1.2. Conversational interaction It makes it easier to complete any task as it automatically do it by using the essential module or libraries of Python, in a conversational interaction way. Hence any user when instruct any task to it, they feel like giving task to a human assistant because of the conversational interaction for giving input and getting the desired output in the form of task done.
* 4.1.3. Reactive nature: The desktop assistant is reactive which means it know human language very well and understand the context that is provided by the user and gives response in the same way, i.e. human understandable language, English. So user finds its reaction in an informed and smart way.
* 4.1.4. Multitasking: The main application of it can be its multitasking ability. It can ask for continuous instruction one after other until the user “QUIT” it.
* 4.1.5. No Trigger phase: It asks for the instruction and listen the response that is given by user without needing any trigger phase and then only executes the task.
* 4.2. DATA IMPLEMENTATION AND PROGRAM EXECUTION
* As the first step, install all the necessary packages and libraries. The command used to install the libraries is “pip install” and then import it. The necessary packages included are as follows:
* 4.2.1. LIBRARIES AND PACKAGES
* 4.2.2.1. pyttsx3: It is a python library which converts text to speech.
* 4.2.2.2. SpeechRecognition: It is a python module which converts speech to texarch option.

**Theoretical Background**

**System study**

The most important thing is to study a system thoroughly. Here we are studying both the existing and the proposed system so that the advantages and disadvantages both understand.

The first task was to identifying how the system is to be computerized. Some analysis and projection was done regarding changes to be made to the existing system.

The new developed system "Specch Recognition" is simple without any complexities.

**Existing System**

This process is done manually. The existing system is time consuming and it is very costly, because it involves a lot of paperwork. To manually handle such a system was very difficult task.

But now-a-days because of computerization this job is becoming easier. The following are the reasons why the existing should be computerized

* To increase efficiency with reduced cost.
* To reduce the burden of paper work.
* To save time of management for recording details of each and every transaction.
* To check that particular product requested is available.

**Limitation of Existing System**

* **Time Consumption**

As the records are to be manually it consumes a lot of time.

* **Paper work**

Lot of paper work is involved as the records are maintained in the files and registers.

* **Storage requirement**

As files and registers are used the storage requirement is increased.

* **Less Reliable**

Use of paper for storing valuable data is not at all reliable.

* **Accuracy**

As the system is in manual, there are not many chances of human error. These cause eiwr in calculating mechanism or maintaining product and supplier data in register.

* **Difficulty** in **keeping new records**

It is difficult for keeping all the new entries of the products, their updates, reviews, feedback etc.

**Proposed system**

To reduce I.he inconvenience that were found i.n the current system, it has been automated so as to provide a user friendly GUI that will help data entry.

This also includes Member registration and Comment.

Creating a database for containing the info1mation present with them on the paper in the existing system.

The developed system will be a Time saving application for the user.

User or visitor can get a broad view about the handset which they requires.

**Advantages of Proposed System**

Time saving since all the detail information is stored in this system, it enables in significance reduction in the number of man-hours taken up for the storage and the retrieval of the information, as it removes the large registers from the scene.

* Accuracy and Reliability
* User-friendly , robust
* Security of data and Integration of all functions in to one system
* Remove redundancy and in-consistency
* Through proper validation, data inconsistency is reduced
* The proposed system would reduce the man-power, because a single trained person (Administrator) can handle whole database

**Feasibility Analysis**

A feasibility study is undertaken to determine the possibility of either improving the existing system or developing a complete new system. It helps to obtain an overview of the problem and to get rough assessment of whether a feasibility solution exits.

Need For Feasibility Study:

* Answer the question whether a new system is to be installed or not.
* Determine the potential of existing system.
* Improve the existing system.
* Know what should be embedded in the new system.
* Define the problem and objectives involved in the project.
* Avoid costly repairs at larger stage when the system in implemented.

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical and Operation feasibility for adding new modules and debugging old nmning system.

When the developer building any system, he/she make sure that the system he/she is going to build is feasible. Feasibility study mainly comprises of three kinds of feasibility.

* Technical Feasibility
* Economical Feasibi.lity
* Operational Feasibility

**Technical Feasibility:**

Technical Feasibility is concerned with the availability and capability of hardware, software and people.

**Hardware:**

A computer should be available that is powerfol enough to handle the proposed system. DBMS systems usually consume more computer power because of its data structure, security, rule and index pointers to the records. These take more machines processing time. ow almost any system is technically feasible.

**Software:**

According to convenience of the system, the system developer should decide which computer language or software he/she is going to use for the proposed project. The developer should keep in mind that the software or languages he/she is going to use should be easily available, efficient and portable and the database languages which he uses should be preferably Relational Database Language (RDBMS).

**People:**

The developers of the system should be ready to learn a new set of skill if necessary. It will be beneficial if he is thorough with the Operating System and Software Development Life Cycle.

**Economical Feasibility:**

Higher level of automation most often requires more funds. Hence based on the hardware and software specification a desirable alternative costs and benefits to see if the investment made in creating / developing a new system is costlier or more beneficial.

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit de1iverd from the new systems. Financial benefits must equal or exceed the costs.

Usefulness of the project is to be based on the cost benefit analysis from this perspective a project that costs greater than the benefits is only in feasible in the sense that it is not profitable to undertaken such project.

First, the cost and benefit ratio should be estimated and then the project should be undertaken. The present software developed to be economical beneficial.

**Operational Feasibility:**

The operational feasibility is obtained by consulting the system user whether it satisfies the user's requirements. A system with an easy interface will always help the user to use the system.

**System Analysis and Design**

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.



System Development Life Cycles (SDLC) describes the data design and applications design. SDLS is an interactive rather than a sequential process. Thus SDLS might help to refine Feasibility study to the user requirements.

Planning

Analysis

Detailed Systems Design

Implementation

Maintenance

Fig. System Development Life Cycles (SDLC)

**Planning:**

SDLS planning yields a general overview of the company and its objectives. An initial assessment of the information of the flow and intents requirement must be made during this discovery portions of SDLS.

**Analysis:**

Problems defined during the planning phase are examined in great details during analysis phase. Analysis phase of the SDLS is an effect, a though AUDIT of the users requirements.

**Detailed system Design:**

In this phase, the designer complete the design of the systems processes. This include all the necessary technical specification for the screens and reports.

**Implementations:**

In this phase hardware, DBMS software and the applications programs are installed and the database designs is being implementations .

**Maintenance:**

As soon as the system is operable, end users being to request in it. Those changes generates systems maintenance activities, which can be grouped into three type:

Adoptive maintenance due to changes in business environment Correctives maintenance in response to systems.

Perfective maintenance lo enhance the system.

**System Design**

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities - design, code and test that is required to build and verify software.

The importance can be stated with a single word "Quality". Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer's view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system - one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities - architectural design, data structure design, interface design and procedure design.

**PERT CHART**

A PERT chait is a project management tool used to schedule, organize and co-ordinate tasks within a project.

**PERT** stands for Program Evaluation Review Technique.

* A PERT chart presents a graphic illustration of a project as a network diagram consisting of numbered nodes(either circles or rectangles) representing events or milestones in the project linked by labeled vectors (directional lines) representing tasks in the project.
* The direction of the arrows on the lines includes the sequence of the tasks. In the diagram, the tasks which must be completed in sequence are called Dependent or Serial tasks.
* Tasks that must be completed in sequence but that don't require resource or completion time are considered to have event dependency.

These are represented by dotted lines with arrows and are called dummy activities.

* Number on the opposite sides of the vectors indicates the time allocated for the task.

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Start** | **Finish** | **Duration** |
| Analysis and requirements | 13/03/2023 | **l** 9/03/2023 | 6 days |
| Designing Tables | 20/03/2023 | 24/03/2023 | 4 days |
| Designing Forms | 25/03/2023 | 03/04/2023 | 8 days |
| System Coding | 04/04/2023 | 20/04/2023 | 16 days |
| Initial Testing | 21/04/2018 | 27/04/201 | 6days |
| Initial Implementation | 28/04/2018 | 07/05/2018 | 8 days |
| Final Testing | 08/05/2018 | 13/05/2018 | 5 days |
| Final  Implementation | 14/05/2018 | 18/05/2018 | 4days |

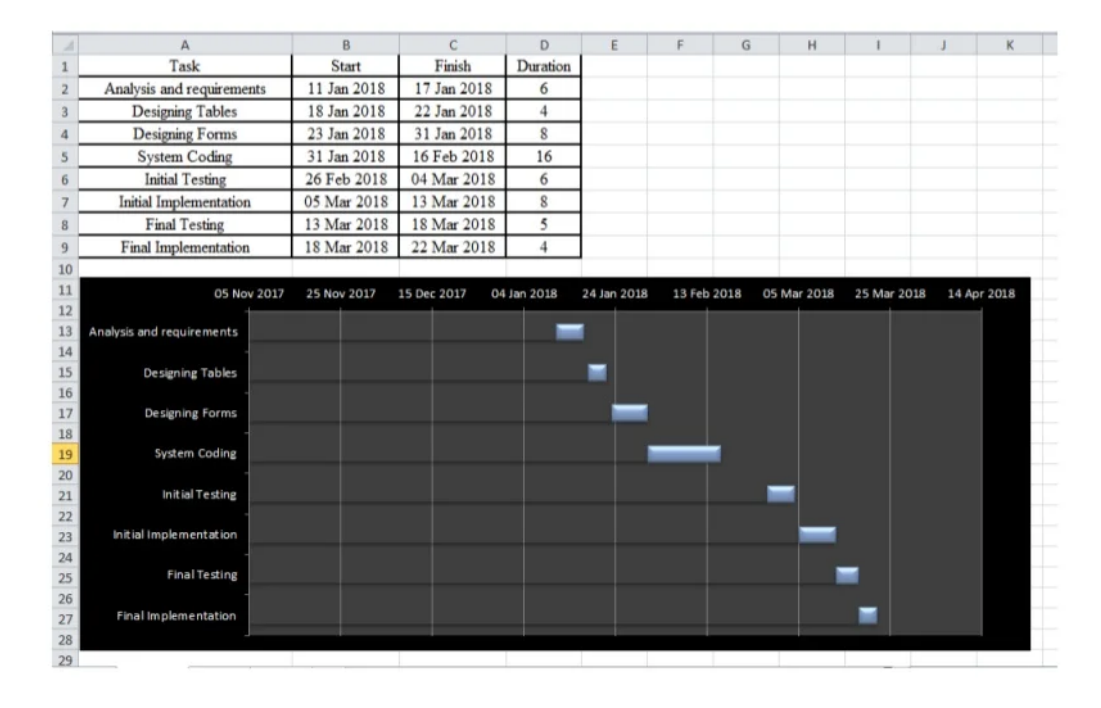
**Gantt Chart**

Gantt chart are useful in planning, how long project should take and helping to

sequence the events by laying them but in order in which the tasks needs to be completed.

Typically tasks are shown in vertical axis and the project time span represented on the horizontal axis. Each task has a corresponding bar that shows the time span required for that basic task. The bar can be filled into show the time span required

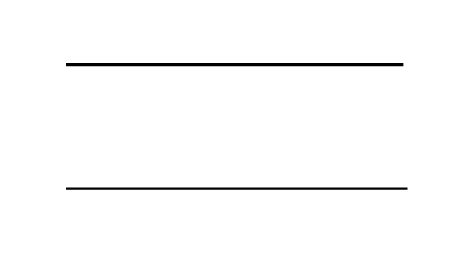
for that task.



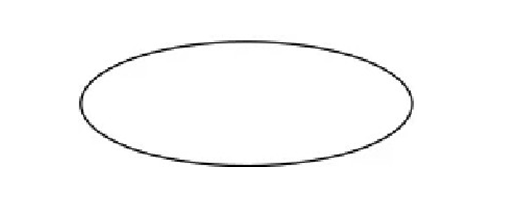
**ER Diagram**

**Symbols used in this E-R Diagram:**

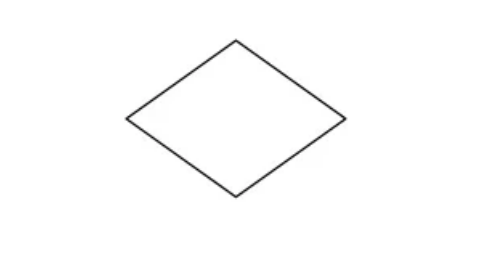
**Entity:** Entity is a "thing" in the real world with an independent existence. An entity may be an object with a physical existence such as person, car or employee. Entity symbol is as follows



**Attribute:** Attribute is a particular property that describes the entity. Attribute symbol is

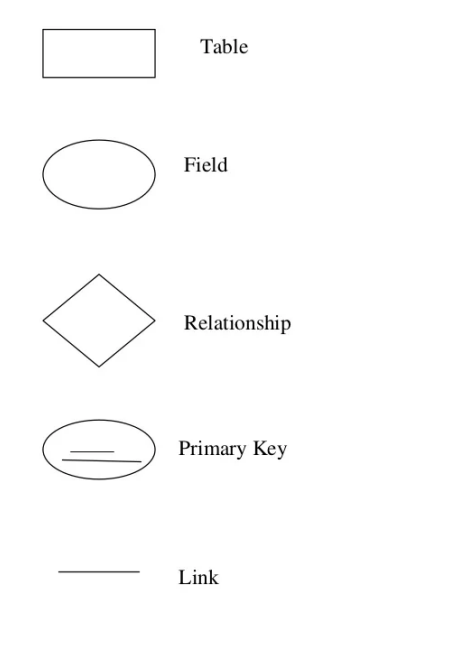


**Relationship:** Relationship will be several implicit relationships among various entity types whenever an attribute of one entity refers to another entity type some relationship exits. Relationship symbol is:

******

**Key attributes:** An entity type usually has an attribute whose values are distinct

for each individual entity in the collection. Such an attribute is called key attribute. Key attribute symbol is as follows

******

***Entity-Relationship Diagram***

***Entity-Relationship Dictionary***

ERD typically represents the relationships between entities, and a speech recognition project may not necessarily have a traditional ERD. Nevertheless, I can offer a high-level representation of the components involved in such a project:

1. User:
   * Represents the user interacting with the speech recognition system.
   * May initiate speech input and receive output from the system.
2. Microphone:
   * Represents the hardware component used for capturing audio input from the user.
3. Speech Recognition System:
   * Represents the core functionality of the project.
   * Utilizes various components for processing speech input and generating text output.
4. Speech-to-Text Engine:
   * Converts spoken words into text format.
   * May utilize pre-trained models or APIs for speech recognition.
5. Natural Language Processing (NLP) Module:
   * Performs analysis and processing of the recognized text.
   * May involve techniques such as text classification, sentiment analysis, or language understanding.
6. Language Model:
   * Represents a pre-trained model used for understanding and generating human-like text.
   * May include models like GPT-3 or other custom-trained models.
7. Text-to-Speech Engine:
   * Converts text output into synthesized speech.
   * Utilizes pre-trained models or APIs for generating speech from text.
8. Speaker/Headphones:
   * Represents the hardware component used for delivering audio output to the user.

It's important to note that the representation above is a simplified overview of the main components involved in a Python speech recognition project. The actual implementation and architecture may vary based on the specific requirements and technologies used.

**DATA FLOW DIAGRAM (DFD)**

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data

flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams.

Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD’S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The lop- level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system.

The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical from, this lead to the modular design.

A DFD is also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

**DFD SYMBOLS:**

In the DFD, there are four symbols:

1. A square defines a source(originator) or destination of system data
2. An arrow identifies data flow. It is the pipeline through which the information flows
3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data

Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical from, this lead to the modular design. A DFD is also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

**DFD SYMBOLS:**

In the DFD, there are four symbols:

1. A square defines a source(originator) or destination of system data
2. An arrow identifies data flow. It is the pipeline through which the information flows
3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data
   1. Process that transforms data flow
   2. Source or Destination of data
   3. data flow
   4. Data store

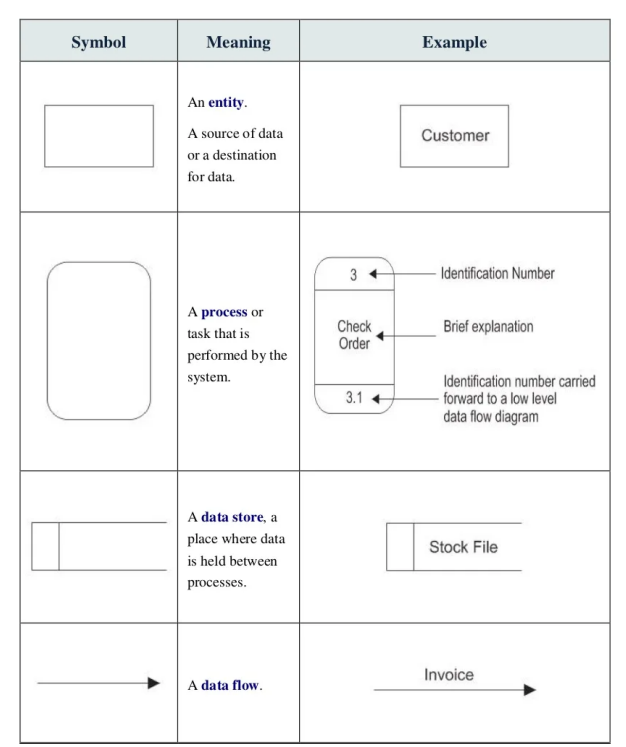


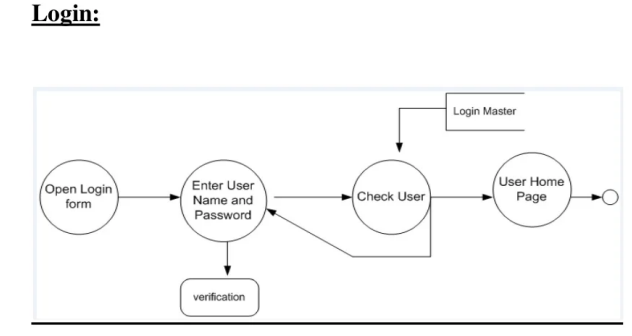
**CONSTRUCTING A DFD:**

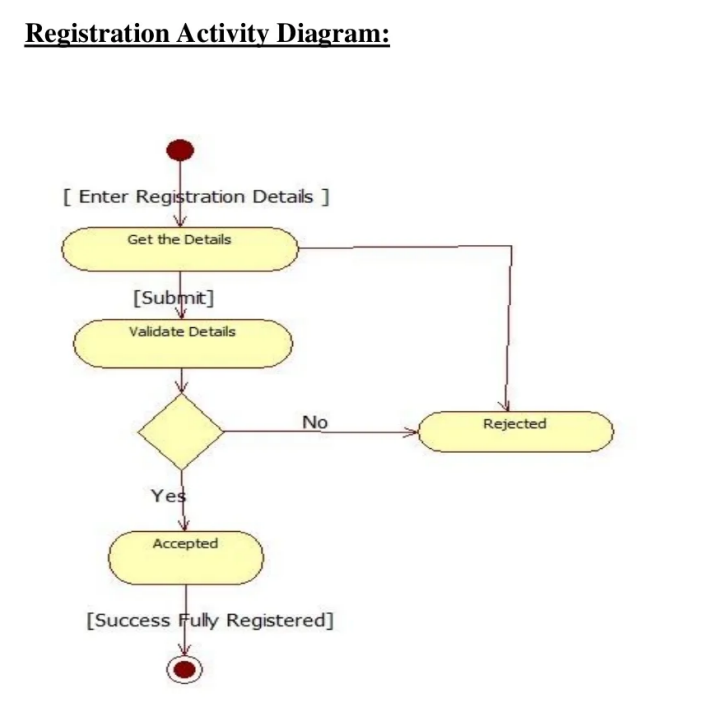
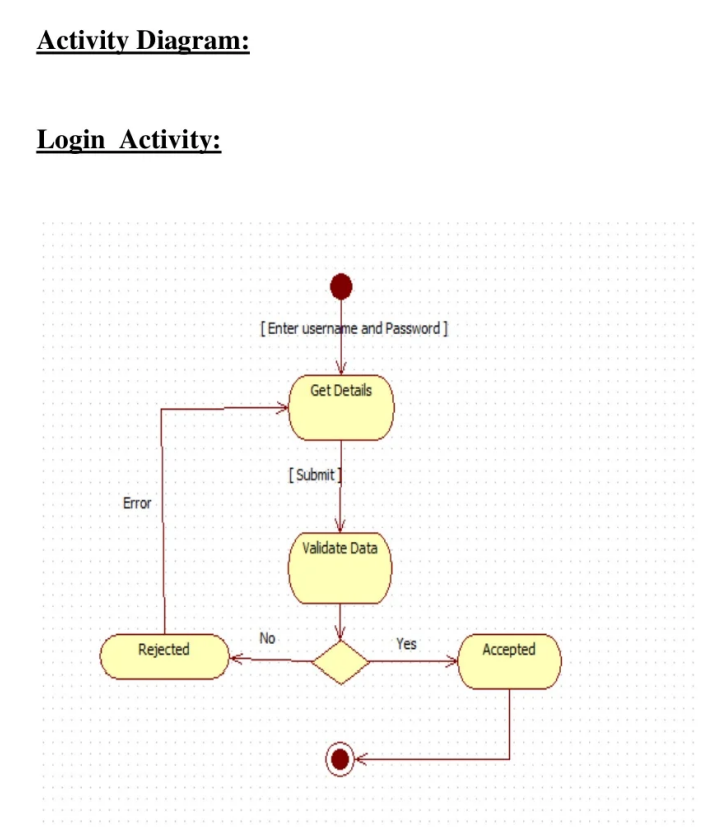
Several rules of thumb are used in drawing DFD’S:

1. Process should be named and numbered for an easy reference. Each name should be representative of the process.
2. The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way t indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.
3. When a process is exploded into lower level details, they are numbered.
4. The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each work capitalized.
5. A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out.

Questionnaires should contain all the data elements that flow in and out. Missing interfaces redundancies and like is then accounted for often through interviews.

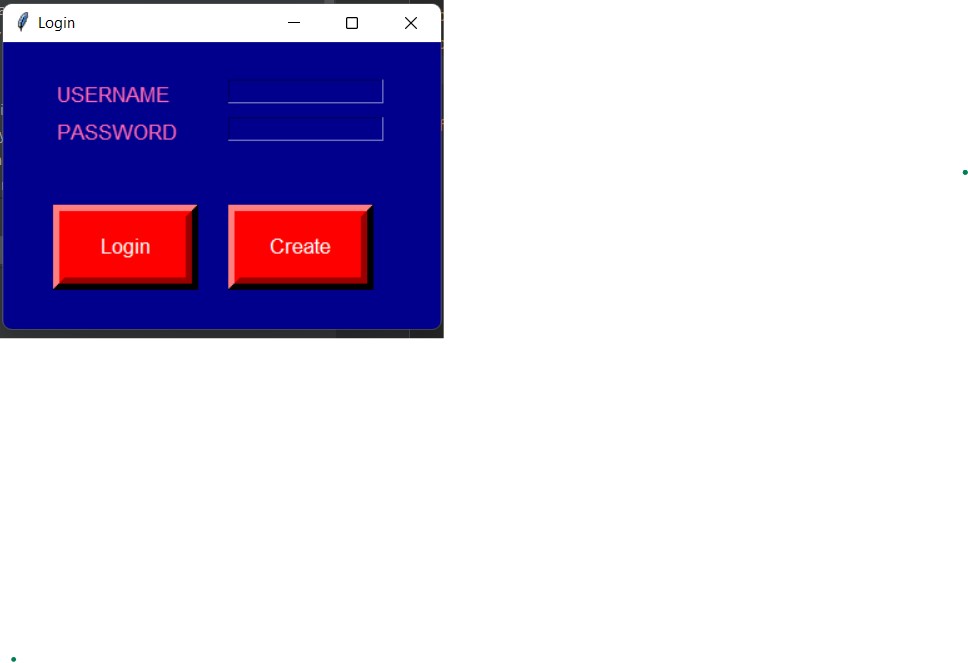




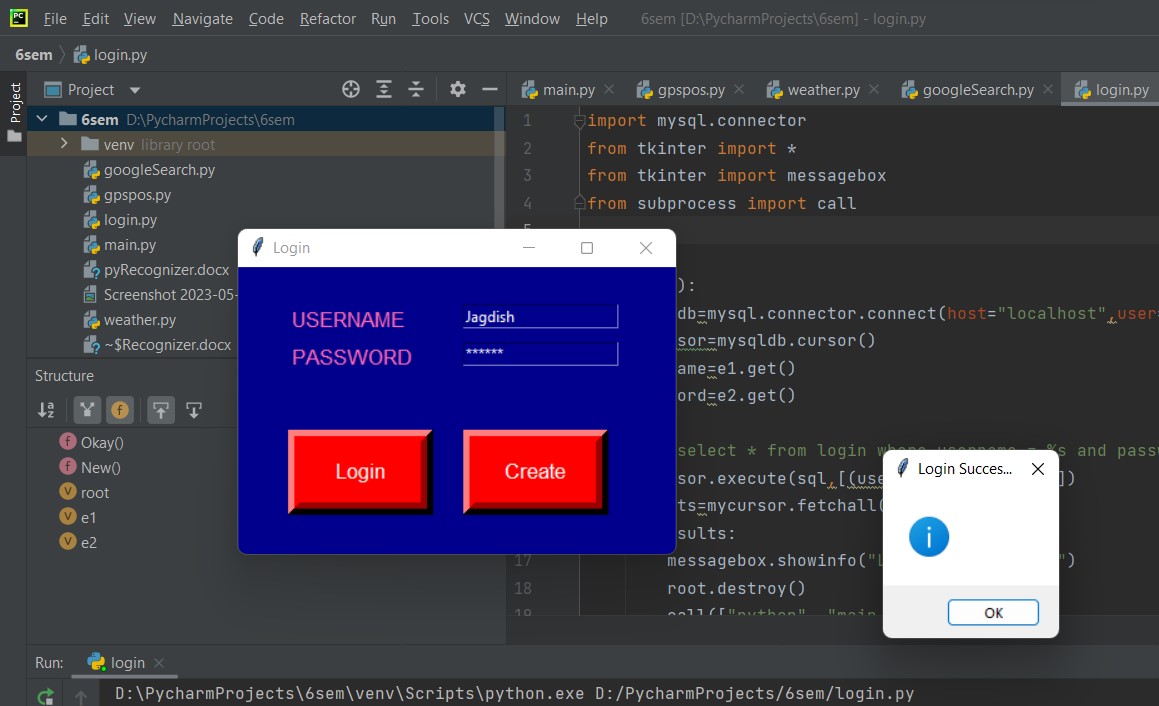


**OUT PUT SCREEN**

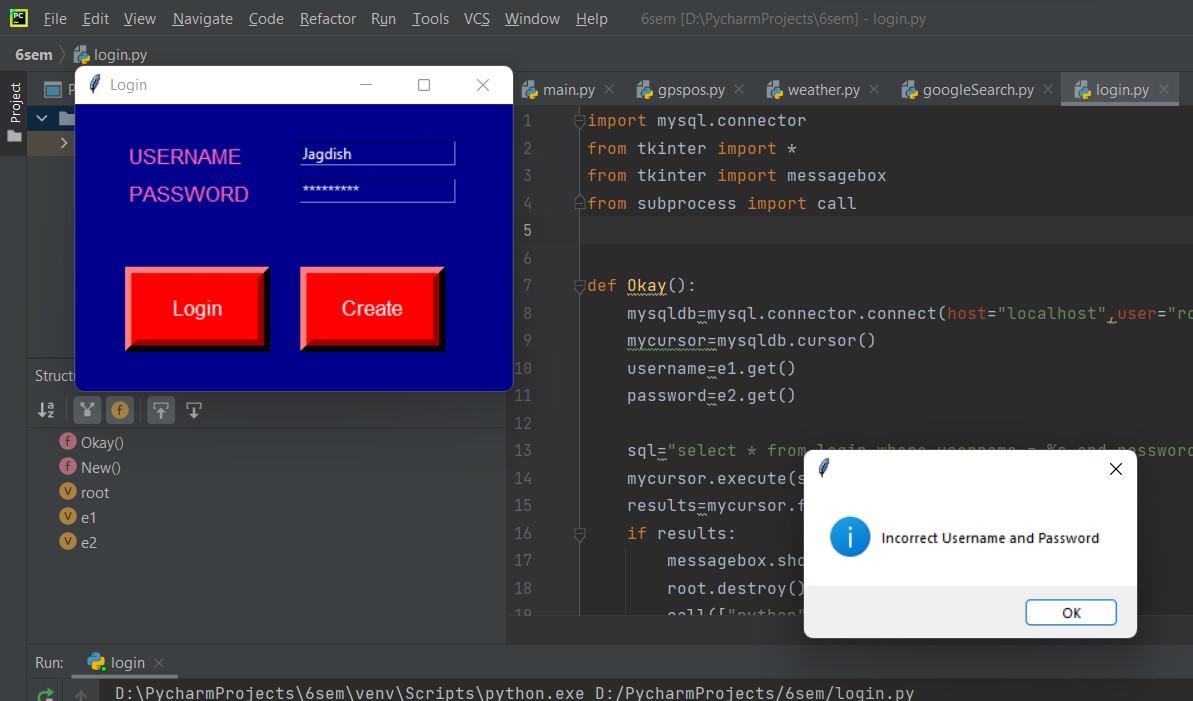
**Login Page**



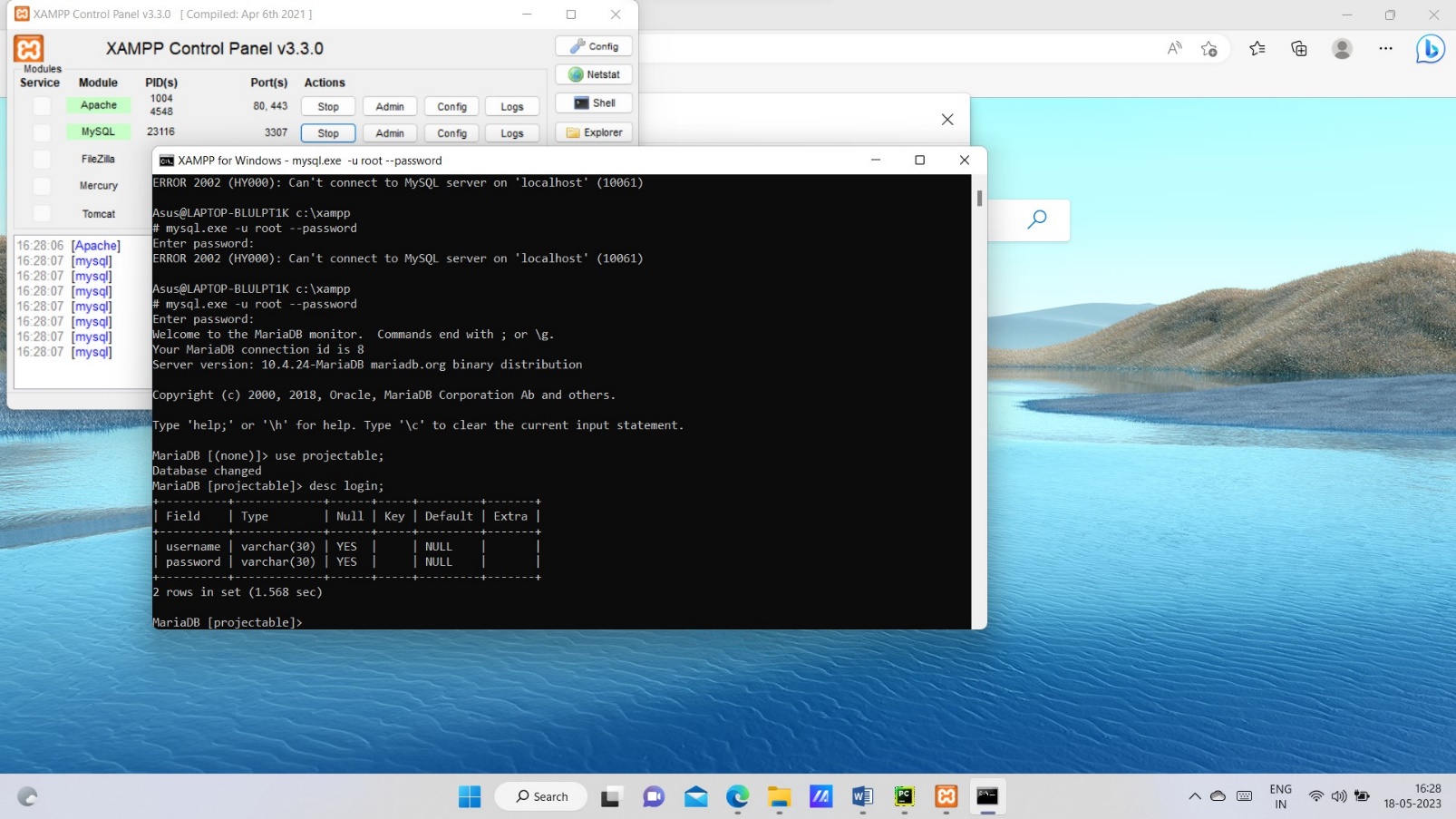
**Successful Authentication**

****

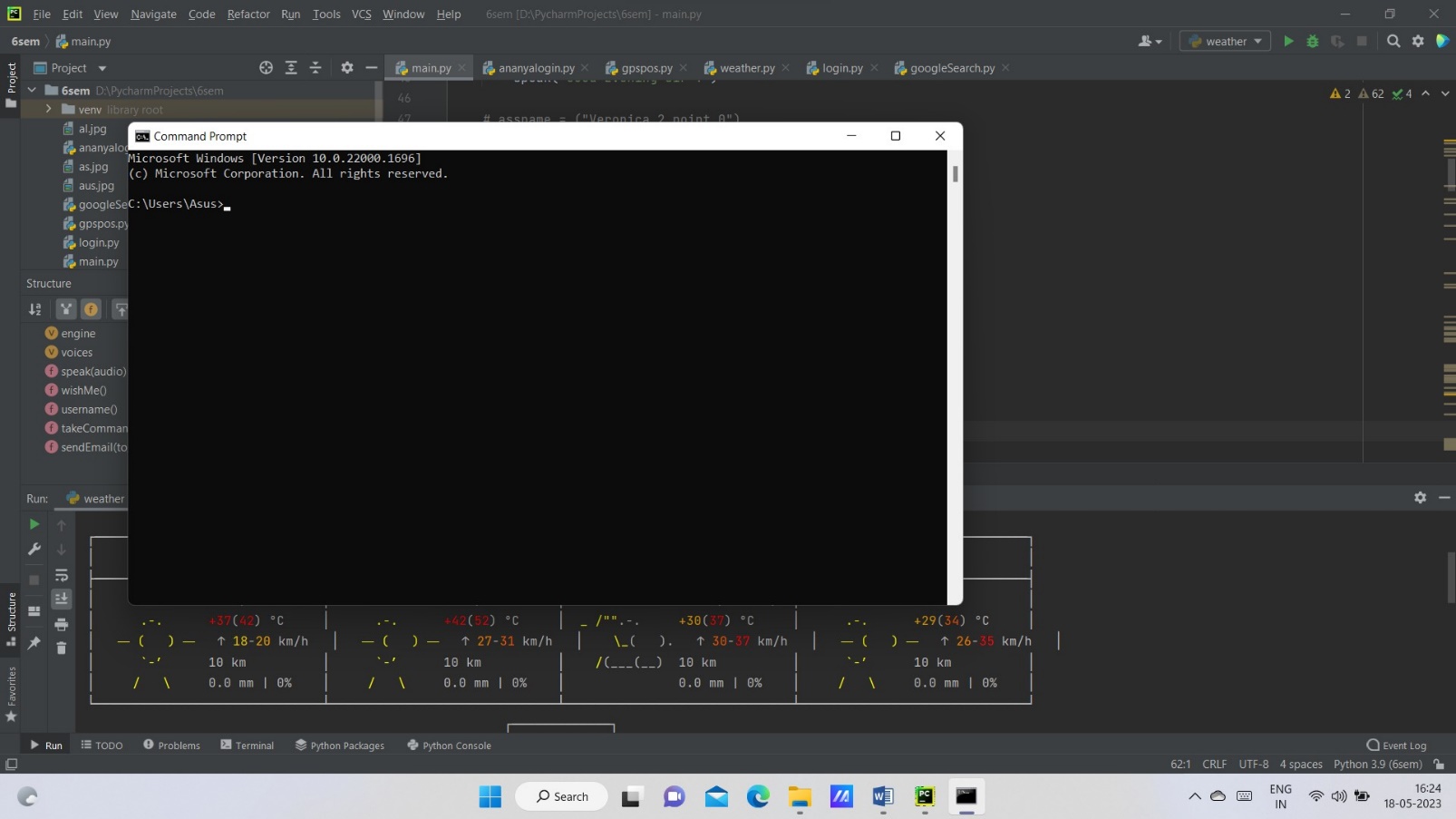
**Unsuccessful Authentication**

****

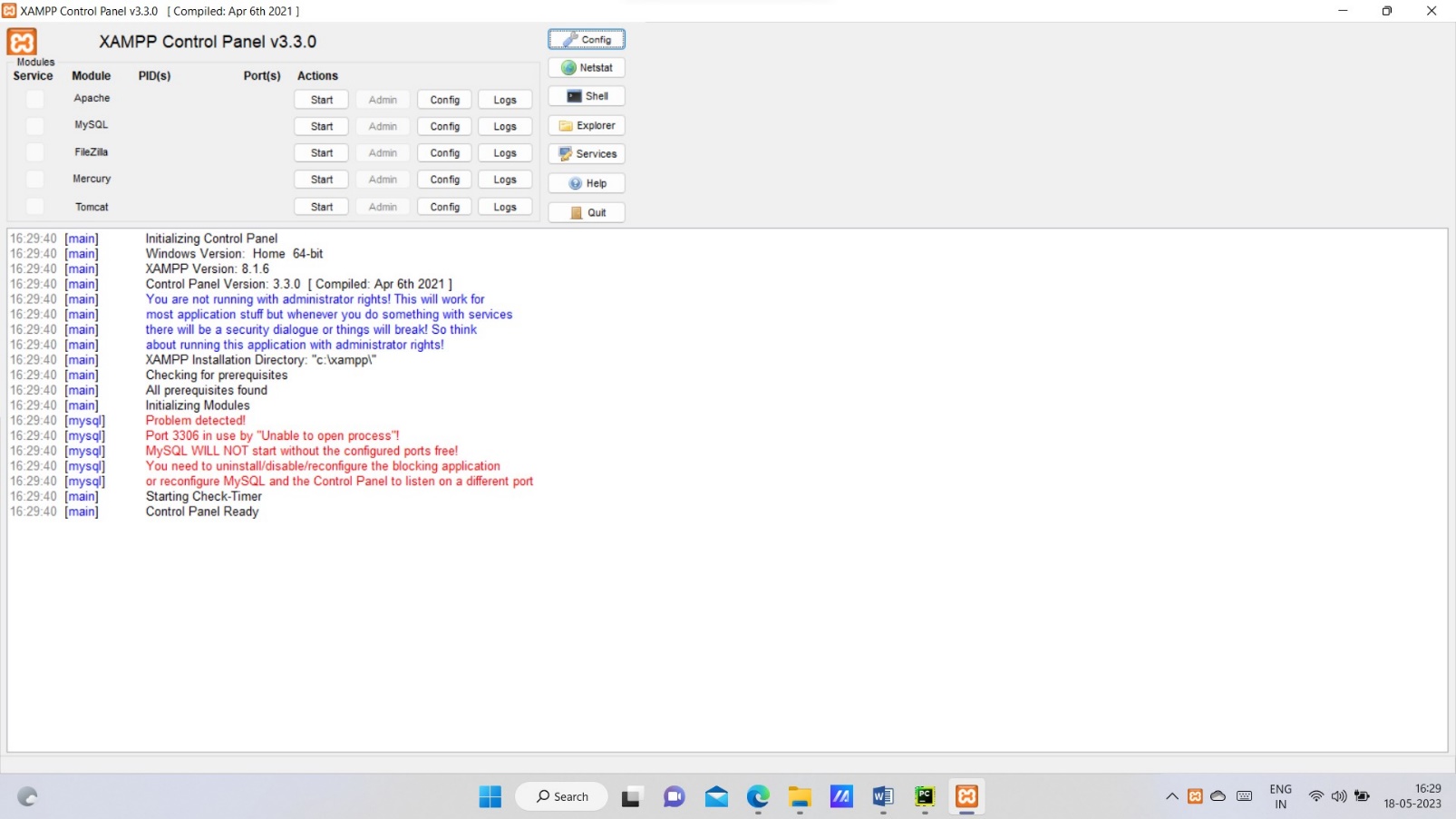
**DataBase**

****

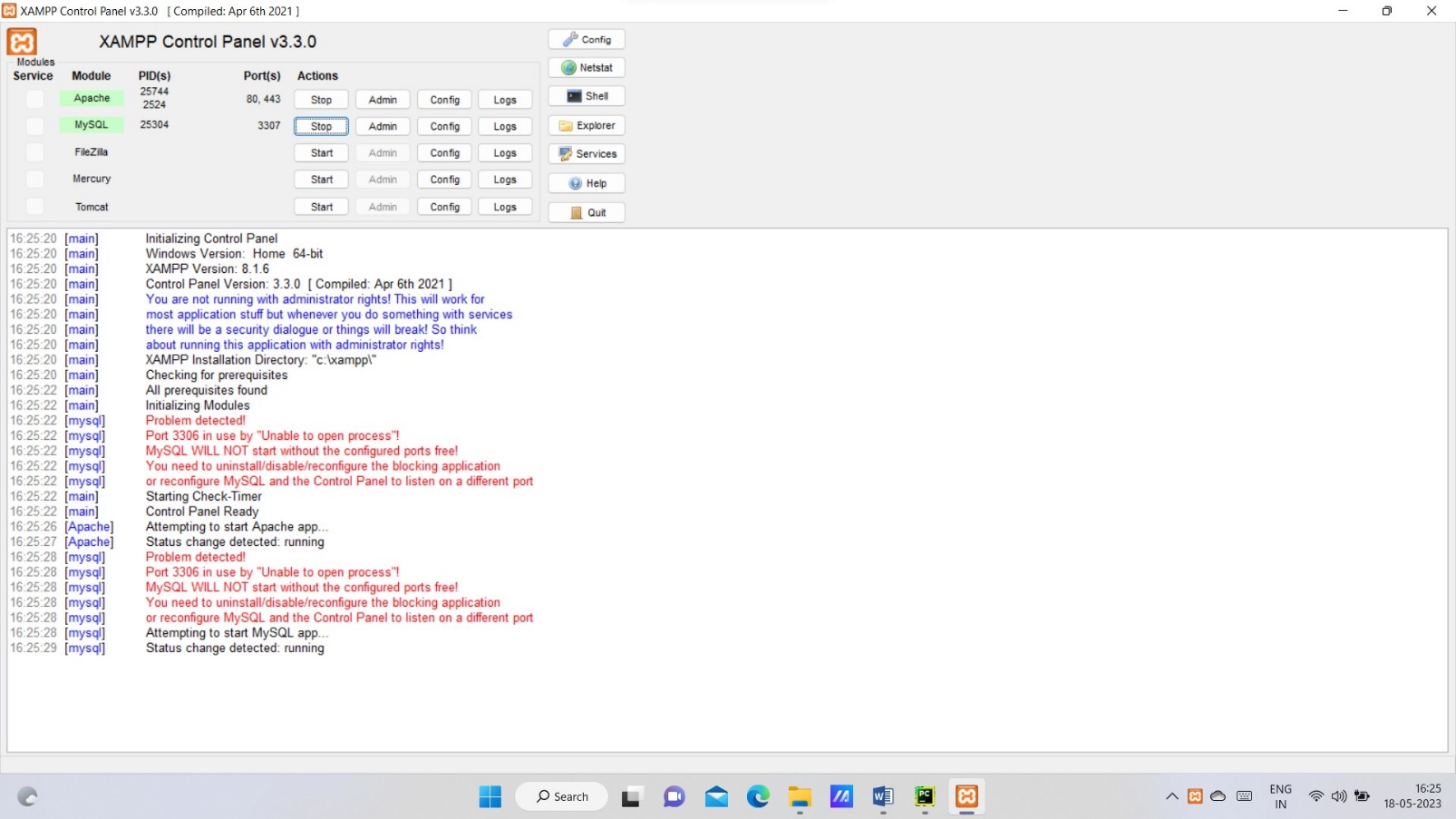
**Command Prompt**

****

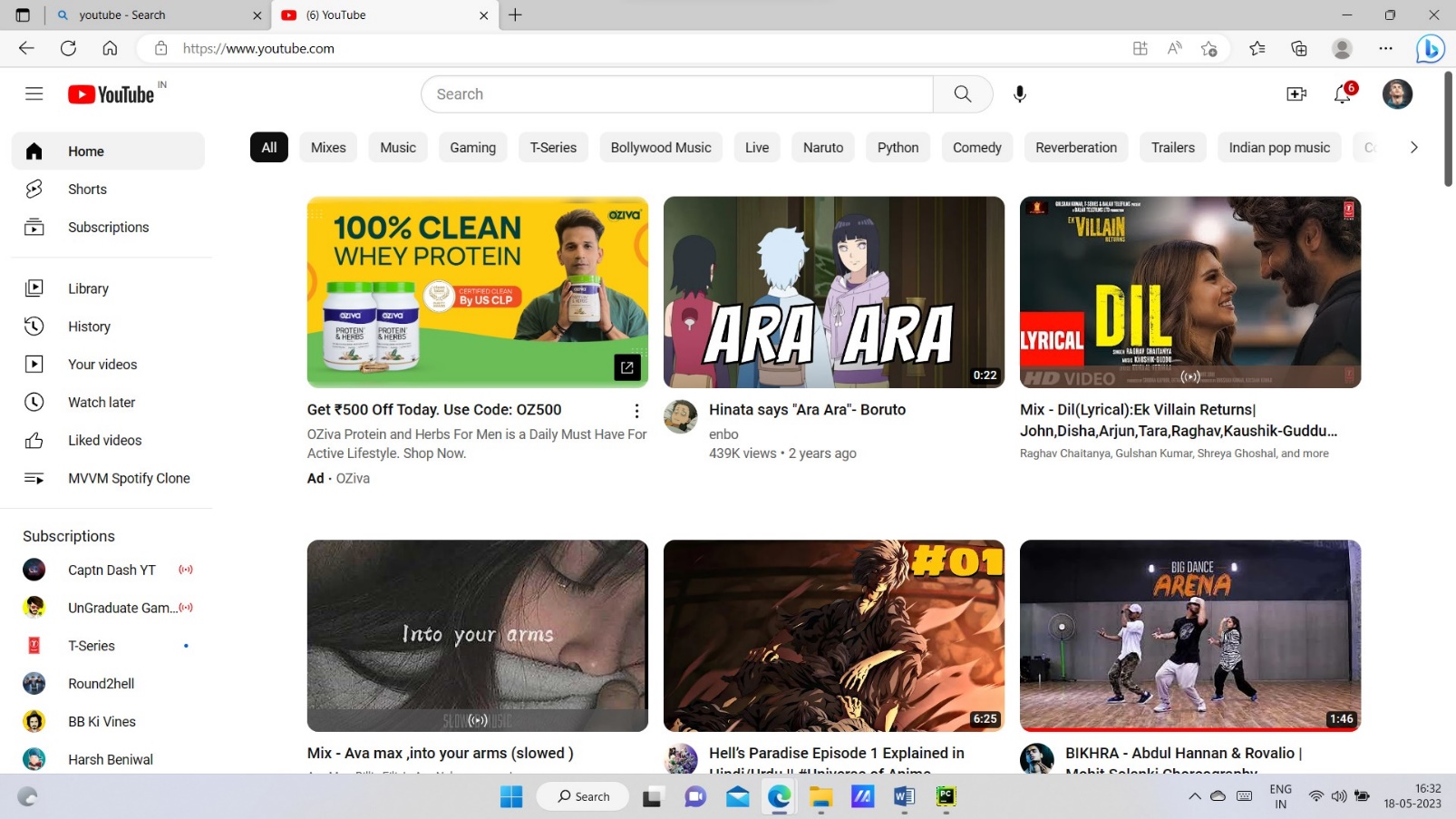
**XAMPP**

****

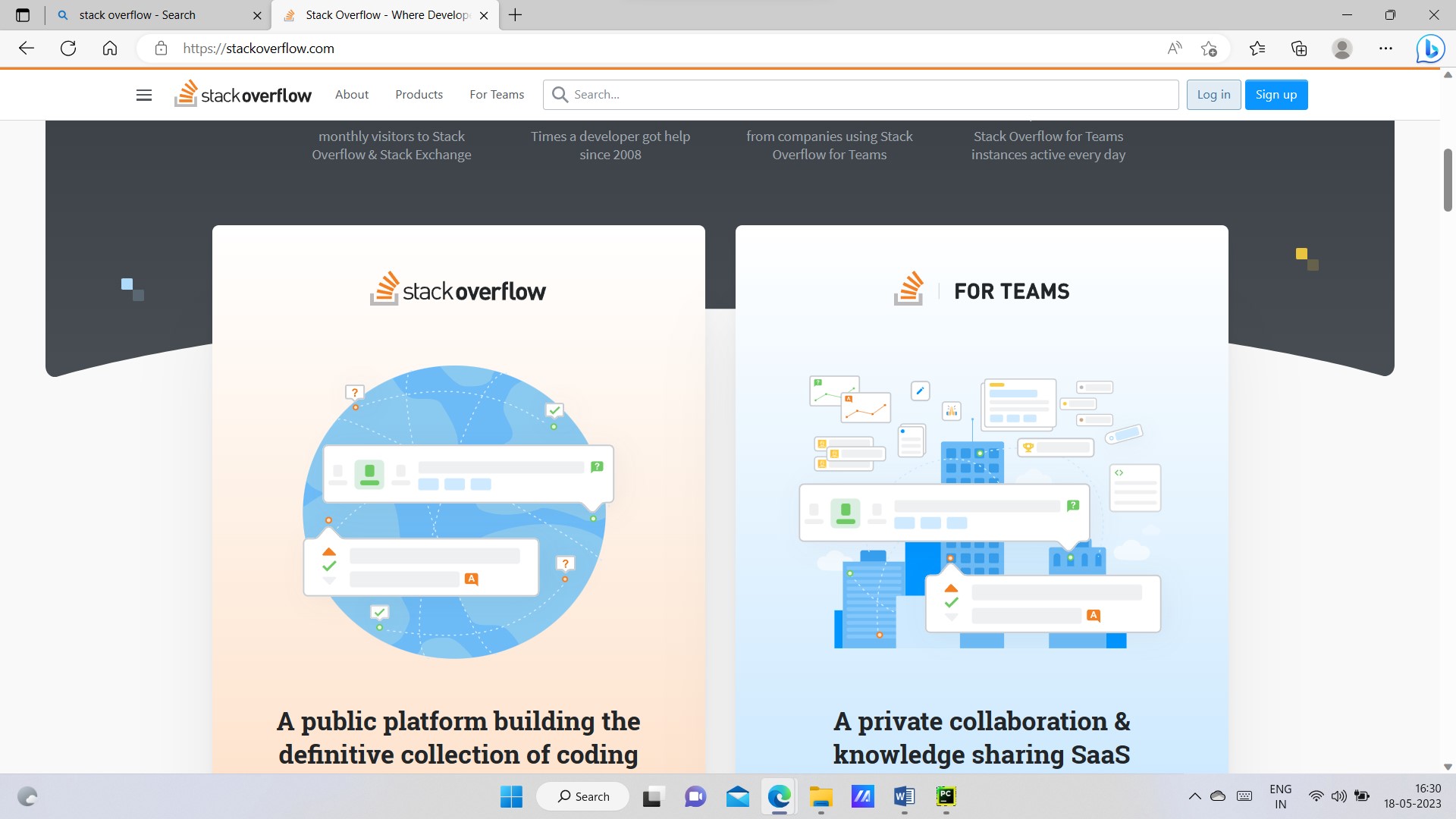
**SERVER**

****

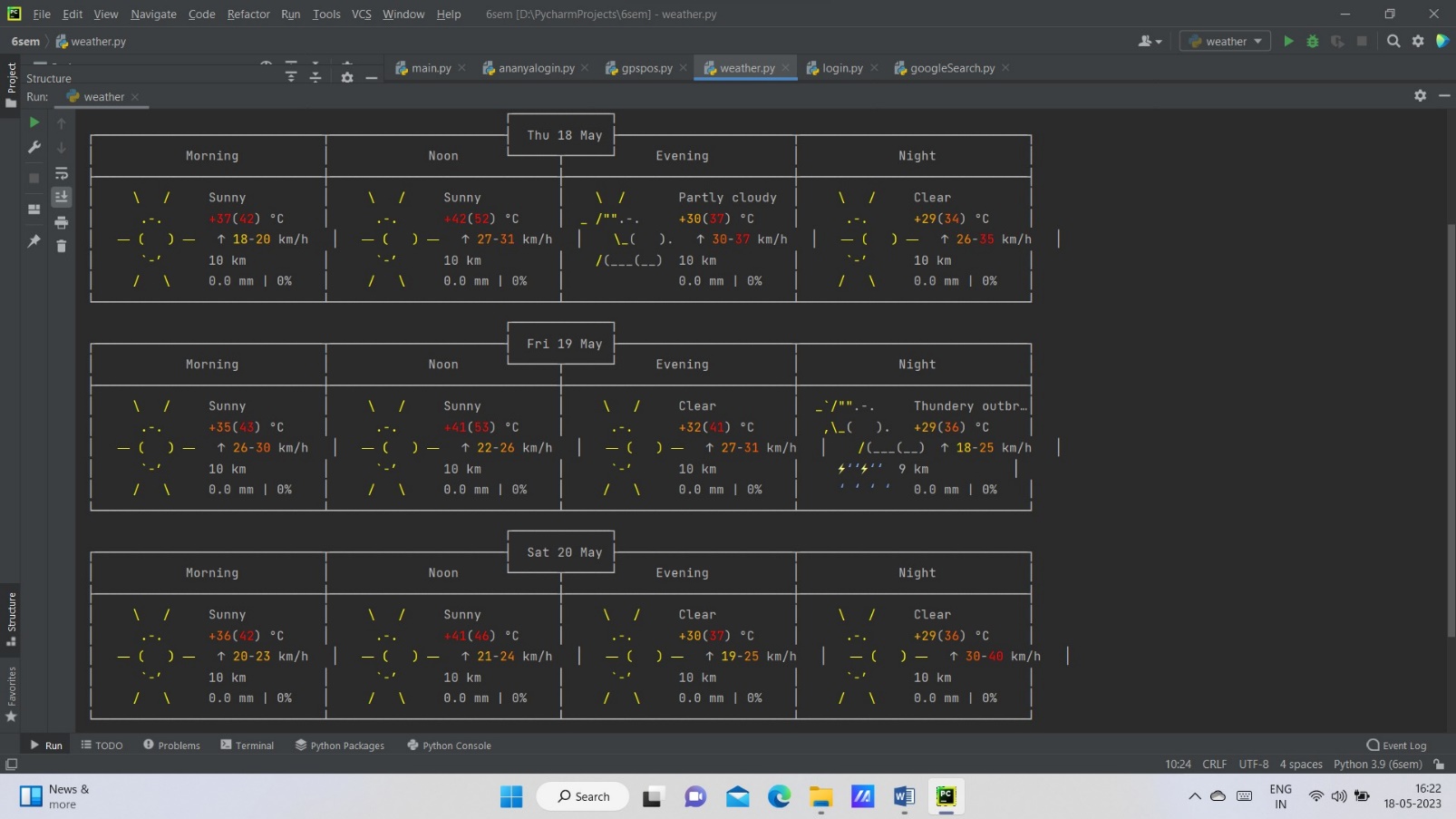
**YouTube**

****

**StackOverflow**

****

**Weather**

****

**CODING**

**Login.py**

import mysql.connector  
from tkinter import \*  
from tkinter import messagebox  
from subprocess import call  
  
def Okay():  
 mysqldb=mysql.connector.connect(host="localhost",user="root",password="",port=3307,database="projectable")  
 mycursor=mysqldb.cursor()  
 username=e1.get()  
 password=e2.get()  
  
 sql="select \* from login where username = %s and password = %s"  
 mycursor.execute(sql,[(username),(password)])  
 results=mycursor.fetchall()  
 if results:  
 call(["python","main.py"])  
 messagebox.showinfo("Login Successfully")  
 root.destroy()  
 return True  
 else:  
 messagebox.showinfo("","Incorrect Username and Password")  
 return False  
  
def New():  
 mysqldb = mysql.connector.connect(host="localhost", user="root", password="", port=3307,database="projectable")  
 mycursor = mysqldb.cursor()  
 username = e1.get()  
 password = e2.get()  
  
 sql = "insert into login values(%s,%s)"  
 mycursor.execute(sql, [(username), (password)])  
 mysql.commit()  
 messagebox.showinfo("User added successfully")  
  
root =Tk()  
root.title("Login")  
root.config(bg="blue4")  
root.geometry("350x230")  
global e1  
global e2  
  
Label(root, text="USERNAME",font=('italic', 13), fg='hotpink',bg='blue4').place(x=40,y=30)  
Label(root, text="PASSWORD",font=('italic', 13), fg='hotpink',bg='blue4').place(x=40,y=60)  
  
e1=Entry(root)  
e1.place(x=180,y=30)  
e1.config(bg='blue4',fg='white')  
  
e2=Entry(root)  
e2.place(x=180,y=60)  
e2.config(show='\*',bg='blue4',fg='white')  
  
Button(root,text="Login",font=('bold', 13), bg='red', fg='white',relief=RAISED,borderwidth=10,  
 command=Okay,height=2,width=10).place(x=40,y=130)  
Button(root,text="Create",font=('bold', 13), bg='red', fg='white',relief=RAISED,borderwidth=10,  
 command=New,height=2,width=10).place(x=180,y=130)  
root.mainloop()

**Main.py**

import ctypes  
import subprocess  
import time  
  
import pyttsx3  
import speech\_recognition as sr  
import datetime  
  
import winshell  
import wolframalpha  
from selenium import webdriver  
import webbrowser  
import shutil  
import smtplib  
import wikipedia  
from PyDictionary import PyDictionary  
import pyjokes  
  
import weather  
  
  
engine = pyttsx3.init('sapi5')  
voices = engine.getProperty('voices')  
engine.setProperty('voice', voices[1].id)  
  
def speak(audio):  
 engine.say(audio)  
 engine.runAndWait()  
  
  
def wishMe():  
 hour = int(datetime.datetime.now().hour)  
 if hour >= 0 and hour < 12:  
 speak("Good Morning Sir !")  
  
 elif hour >= 12 and hour < 18:  
 speak("Good Afternoon Sir !")  
  
 else:  
 speak("Good Evening Sir !")  
  
 assname = ("Veronica 2 point 0")  
 speak("I am your Assistant")  
 speak(assname)  
  
  
def username():  
 speak("What should i call you sir")  
 uname = takeCommand()  
 speak("Welcome Mister")  
 speak(uname)  
 columns = shutil.get\_terminal\_size().columns  
  
 print("#####################".center(columns))  
 print("Welcome Mr.", uname.center(columns))  
 print("#####################".center(columns))  
  
 speak("How can i Help you, Sir")  
  
  
def takeCommand():  
 r = sr.Recognizer()  
  
 with sr.Microphone() as source:  
  
 print("Listening...")  
 r.pause\_threshold = 1  
 audio = r.listen(source)  
  
 try:  
 print("Recognizing...")  
 query = r.recognize\_google(audio, language='en-in')  
 print(f"User said: {query}\n")  
  
 except Exception as e:  
 print(e)  
 print("Unable to Recognize your voice!!! Say that again please")  
 speak("Unable to Recognize your voice!!! Say that again please")  
 # takeCommand().lower()  
 return "None"  
  
 return query  
  
def sendEmail(to, content):  
 server = smtplib.SMTP('smtp.gmail.com', 587)  
 server.ehlo()  
 server.starttls()  
 # Enable low security in gmail  
 server.login('your email id', 'your email password')  
 server.sendmail('your email id', to, content)  
 server.close()  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 wishMe()  
 username()  
  
 while True:  
 query = takeCommand().lower()  
  
 # Logic for executing tasks based on query  
 if "introduce yourself" in query:  
  
 speak("I am veronica. A virtual artificial intelligence."  
 "i'm here to assist you verities of task as my best!!! "  
 "24 hours in a day, 7 days in a week, any time i'm ready for you!!!"  
 "all operations are optimized!!! order me, sir")  
  
 elif "veronica" in query:  
 speak("yes sir!!! veronica 1 point o in your service. Order me!!!")

elif 'take a rest' in query:

speak("thank you sir!!! have a good day.")  
 exit()  
  
 elif 'shutdown system' in query:  
 speak("Hold On a Sec ! Your system is on its way to shut down")  
 subprocess.call('shutdown / p /f')  
  
 elif 'lock window' in query:  
 speak("locking the device")  
 ctypes.windll.user32.LockWorkStation()  
  
 elif "restart" in query:  
 subprocess.call(["shutdown", "/r"])  
  
 elif "log off" in query or "sign out" in query:  
 speak("Make sure all the application are closed before sign-out")  
 time.sleep(5)  
 subprocess.call(["shutdown", "/l"])  
  
 elif 'empty recycle bin' in query:  
 winshell.recycle\_bin().empty(confirm=False, show\_progress=False, sound=True)  
 speak("Recycle Bin Recycled")  
  
 elif "write a note" in query:  
  
 speak("What should i write, sir")  
 note = takeCommand()  
 file = open('veronica.txt', 'w')  
 speak("Sir, Should i include date and time")  
 snfm = takeCommand()  
 if 'yes' in snfm or 'sure' in snfm:  
 strTime = datetime.datetime.now().strftime("% H:% M:% S")  
 file.write(strTime)  
 file.write(" :- ")  
 file.write(note)  
 else:  
 file.write(note)  
  
 elif "tell me about weather" in query:  
 speak("about wich city")  
 city=takeCommand()  
 weather.Gen\_report(city)  
  
 elif 'tell a joke' in query:  
 My\_joke = pyjokes.get\_joke(language="en", category="neutral")  
 speak(My\_joke)

elif 'forecast the weather' in query:  
 speak('about which city')  
 city=takeCommand()  
 weather.Gen\_report(city)  
  
  
 elif 'the time' in query:  
 strTime = datetime.datetime.now().strftime("%H:%M:%S")  
 speak(f"Sir, the time is {strTime}")  
  
 elif "show note" in query:  
  
 speak("Showing Notes")  
 file = open("veronica.txt", "r")  
 print(file.read())  
 speak(file.read(6))  
  
 elif 'open youtube' in query:  
 webbrowser.open("youtube.com")  
  
 elif 'open google' in query:  
 webbrowser.open("google.com")  
  
 elif 'open stackoverflow' in query:  
 webbrowser.open("stackoverflow.com")  
  
 elif 'wikipedia' in query:  
 speak('Searching Wikipedia...')  
 query = query.replace("wikipedia","")  
 results = wikipedia.summary(query, sentences=2)  
 speak("According to Wikipedia")  
 # print(results)  
 speak(results)  
  
 elif 'find the a meaning' in query:  
  
 speak("Which word do u want to find the meaning sir")  
 word = takeCommand()  
 dictionary = PyDictionary()  
 word\_mean = dictionary.meaning(word)  
 print(word\_mean)  
 speak(word\_mean)  
  
 elif 'search on google' in query:  
  
 speak("what will i search")

search\_string=takeCommand().lower()

search\_string = search\_string.replace(' ', '+')  
 browser = webdriver.Chrome('chromedriver')  
 for i in range(1):  
 matched\_elements = browser.get("https://www.google.com/search?q=" + search\_string + "&start=" + str(i))  
 speak(matched\_elements)  
  
 elif 'search' in query or 'play' in query:  
  
 query = query.replace("search", "")  
 query = query.replace("play", "")  
 webbrowser.open(query)  
  
 elif "where is" in query:  
  
 query = query.replace("where is", "")  
 location = query  
 speak("User asked to Locate")  
 speak(location)  
 webbrowser.open("https://www.google.nl / maps / place/" + location + "")  
  
 elif "what is" in query or "who is" in query:  
 client = wolframalpha.Client("API\_ID")  
 res = client.query(query)  
 try:  
 print(next(res.results).text)  
 speak(next(res.results).text)  
 except StopIteration:  
 print("No results")  
  
  
 elif 'send email to' in query:  
 try:  
 speak("What should I say?")  
 content = takeCommand()  
 speak("whom i send")  
 to=takeCommand() #"name@gmail.com"  
 sendEmail(to, content)  
 speak("Email has been sent!")  
 except Exception as e:  
 print(e)  
 speak("sorry mail"  
 " can't be send")  
  
 else:  
 speak("Sorry!!! Sir i can not help you")

**Weather.py**

import pyttsx3  
import requests  
  
engine = pyttsx3.init('sapi5')  
voices = engine.getProperty('voices')  
engine.setProperty('voice', voices[0].id)  
  
def speak(audio):  
 engine.say(audio)  
 engine.runAndWait()  
  
def Gen\_report(C):  
 url = 'https://wttr.in/{}'.format(C)  
 try:  
 data = requests.get(url)  
 T = data.text  
 except:  
 T = "Error Occurred"  
 print(T)  
 # speak(T)

**GoogleSearch.py**

import imdb  
import pyttsx3  
import speech\_recognition as sr  
import datetime

# Function for speaking  
def speak(text):  
 engine = pyttsx3.init()  
 voices = engine.getProperty('voices')  
 engine.setProperty('voice', voices[1].id)  
 rate = engine.getProperty('rate')  
  
 engine.setProperty('rate', rate-20)  
  
 engine.say(text)  
 engine.runAndWait()  
  
  
# calling the speak() function  
speak("Say the movie name")  
  
  
# Function to get input in the audio format  
def get\_audio():  
 r = sr.Recognizer()  
 with sr.Microphone() as source:  
 r.pause\_threshold = 1  
 r.adjust\_for\_ambient\_noise(source, duration=1)  
 audio = r.listen(source)  
 said = ""  
  
 try:  
  
 # will recognize the input  
 said = r.recognize\_google(audio)  
 print(said)  
  
 except:  
 speak("Didn't get that")  
 # will return the input in lowercase  
 return said.lower()  
  
  
# Function for searching movie  
def search\_movie():

# gathering information from IMDb  
 moviesdb = imdb.IMDb()  
  
 # search for title  
 text = get\_audio()  
  
 # passing input for searching movie  
 movies = moviesdb.search\_movie(text)  
  
 speak("Searching for " + text)  
 if len(movies) == 0:  
 speak("No result found")  
 else:  
  
 speak("I found these:")  
  
 for movie in movies:  
  
 title = movie['title']  
 year = movie['year']  
 # speaking title with releasing year  
 speak(f'{title}-{year}')  
  
 info = movie.getID()  
 movie = moviesdb.get\_movie(info)  
  
 title = movie['title']  
 year = movie['year']  
 rating = movie['rating']  
 plot = movie['plot outline']  
  
 # the below if-else is for past and future release  
 if year < int(datetime.datetime.now().strftime("%Y")):  
 speak(  
 f'{title}was released in {year} has IMDB rating of {rating}.\  
 The plot summary of movie is{plot}')  
 print(  
 f'{title}was released in {year} has IMDB rating of {rating}.\  
 The plot summary of movie is{plot}')  
 break

else:

speak(  
 f'{title}will release in {year} has IMDB rating of {rating}.\  
 The plot summary of movie is{plot}')  
 print(  
 f'{title}will release in {year} has IMDB rating of {rating}.\  
 The plot summary of movie is{plot}')  
 break  
  
  
search\_movie()

**LIBRARIES**

**Ctypes :**

[ctypes](https://docs.python.org/3/library/ctypes.html#module-ctypes) is a foreign function library for Python. It provides C compatible data types, and allows calling functions in DLLs or shared libraries. It can be used to wrap these libraries in pure Python.

**Os :**

This module provides a portable way of using operating system dependent functionality. If you just want to read or write a file see [open()](https://docs.python.org/3/library/functions.html#open), if you want to manipulate paths, see the [os.path](https://docs.python.org/3/library/os.path.html#module-os.path) module, and if you want to read all the lines in all the files on the command line see the [fileinput](https://docs.python.org/3/library/fileinput.html#module-fileinput) module. For creating temporary files and directories see the [tempfile](https://docs.python.org/3/library/tempfile.html#module-tempfile) module, and for high-level file and directory handling see the [shutil](https://docs.python.org/3/library/shutil.html#module-shutil) module.

**Subprocess :**

The [subprocess](https://docs.python.org/3/library/subprocess.html#module-subprocess) module allows you to spawn new processes, connect to their input/output/error pipes, and obtain their return codes. This module intends to replace several older modules and functions:

os.system

os.spawn\*

**time :**

This module provides various time-related functions. For related functionality, see also the [datetime](https://docs.python.org/3/library/datetime.html#module-datetime) and [calendar](https://docs.python.org/3/library/calendar.html#module-calendar) modules.

Although this module is always available, not all functions are available on all platforms. Most of the functions defined in this module call platform C library functions with the same name. It may sometimes be helpful to consult the platform documentation, because the semantics of these functions varies among platforms.

**Pyautogui :**

PyAutoGUI is a cross-platform GUI automation Python module for human beings. Used to programmatically control the mouse & keyboard.

**pyttsx3 :**

pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline, and is compatible with both Python 2 and 3.

**Requests :**

Requests allows you to send HTTP/1.1 requests extremely easily. There’s no need to manually add query strings to your URLs, or to form-encode your PUT & POST data — but nowadays, just use the json method!

Requests is one of the most downloaded Python packages today, pulling in around 30M downloads / week— according to GitHub, Requests is currently [depended upon](https://github.com/psf/requests/network/dependents?package_id=UGFja2FnZS01NzA4OTExNg%3D%3D) by 1,000,000+ repositories. You may certainly put your trust in this code.

**Datetime :**

The [datetime](https://docs.python.org/3/library/datetime.html#module-datetime) module supplies classes for manipulating dates and times.

While date and time arithmetic is supported, the focus of the implementation is on efficient attribute extraction for output formatting and manipulation.

**Winshell :**

The winshell module is a light wrapper around the Windows shell functionality.

It includes convenience functions for accessing special folders, for using the shell’s file copy, rename & delete functionality, and a certain amount of support for structured storage.

**Selenium :**

Selenium requires a driver to interface with the chosen browser. Firefox, for example, requires [geckodriver](https://github.com/mozilla/geckodriver/releases), which needs to be installed before the below examples can be run.

**Webbrowser :**

The [webbrowser](https://docs.python.org/3/library/webbrowser.html#module-webbrowser) module provides a high-level interface to allow displaying web-based documents to users. Under most circumstances, simply calling the [open()](https://docs.python.org/3/library/webbrowser.html#webbrowser.open) function from this module will do the right thing.

Under Unix, graphical browsers are preferred under X11, but text-mode browsers will be used if graphical browsers are not available or an X11 display isn’t available. If text-mode browsers are used, the calling process will block until the user exits the browser.

**Shutil :**

The [shutil](https://docs.python.org/3/library/shutil.html#module-shutil) module offers a number of high-level operations on files and collections of files. In particular, functions are provided which support file copying and removal. For operations on individual files, see also the [os](https://docs.python.org/3/library/os.html#module-os) module.

**Smtplib :**

The [smtplib](https://docs.python.org/3/library/smtplib.html#module-smtplib) module defines an SMTP client session object that can be used to send mail to any internet machine with an SMTP or ESMTP listener daemon. For details of SMTP and ESMTP operation, consult [**RFC 821**](https://datatracker.ietf.org/doc/html/rfc821.html) (Simple Mail Transfer Protocol) and [**RFC 1869**](https://datatracker.ietf.org/doc/html/rfc1869.html) (SMTP Service Extensions).

**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](https://www.mediawiki.org/wiki/API) so you can focus on using Wikipedia data, not getting it.

**PyDictionary :**

PyDictionary is a Dictionary Module for Python 2/3 to get meanings, translations, synonyms and Antonyms of words. It uses WordNet for getting meanings, Google for translations, and synonym.com for getting synonyms and antonyms.

This module uses Python Requests, BeautifulSoup4 and goslate as dependencies

**Pyjokes :**

One line jokes for programmers (jokes as a service)

**Pywhatkit :**

[PyWhatKit](https://pypi.org/project/pywhatkit/) is a Python library with various helpful features. It's easy-to-use and does not require you to do any additional setup. Currently, it is one of the most popular library for WhatsApp and YouTube automation. New updates are released frequently with new features and bug fixes.

**PROCESS INVOLBED**

**Analysis Model**

Software Development Life Cycle or SDLC is a model of a detailed plan on how to create, develop, implement and eventually fold the software. It's a complete plan outlining how the software will be born, raised and eventually be retired from its function.

**Waterfall Model**

This is also known as Classic Life Cycle Model (or) Linear Sequential Model (or) System Development Life Cycle Model. This model has the following activities.

1. **System/Information Engineering and Modeling**

As software is always of a large system (or business), work begins by establishing the requirements for all system elements and then allocating some subset of these requirements to software. This system view is essential when the software must interface with other elements such as hardware, people and other resources. System is the basic and very critical requirement for the existence of software in any entity. So if the system is not in place, the system should be engineered and put in place. In some cases, to extract the maximum output, the system should be re-engineered and spruced up. Once the ideal system is engineered or tuned, the development team studies the software requirement for the system.

1. **Software Requirement Analysis**

The requirements gathe1ing process is intensified and focused specifically on software. To understand the nature of the program to build, the software engineer must understand the information domain for the software as well as required function, behavior, perfo1mance, and interface. Requirement for both the system and the software are documented and reviewed with the customer.

1. **System Analysis and Design**

In this phase, the software development process, the software's overall structure and its nuances are defined. A software development model is thus created.

Analysis and Design are very crucial in the whole development cycle. Much care is taken during this phase. The logical system of the product is developed in this phase.

1. **Code Generation**

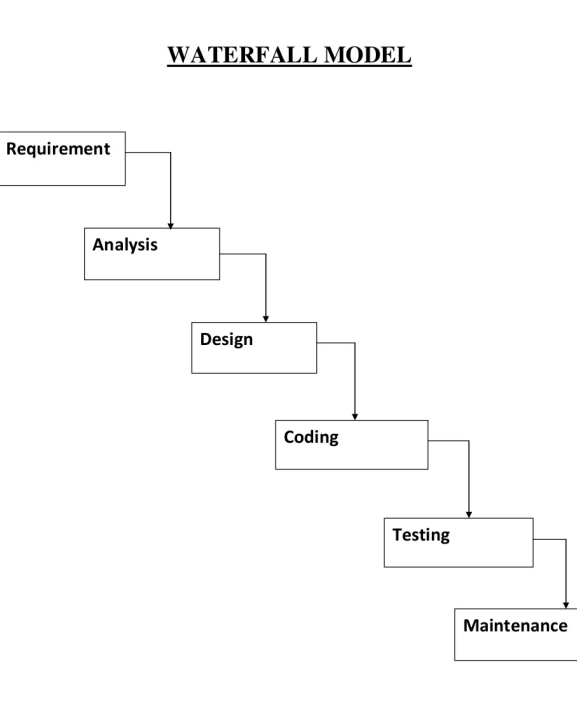
The design must be translated into a machine-readable form. The code generation step performs this task. If designed is performed in a detailed manner, code generation can be accomplished mechanistically.

1. **Testing**

Once code has been generated, program testing begins. The testing process focuses on the logical internals of the software, ensuring that all statements have been tested, and on the functional externals; that is, conducting tests to uncover errors and ensure that defined input will produce actual results that agree with required results.

1. **Maintenance**

The software will definitely undergo change once it is delivered to the customer. There can be many reasons for this change to occur. Change could happen because of some unexpected input values into the system. In addition, the changes in the system could directly affect the software operations. The software should be developed to accommodate changes that could happen during the post implementation period.



**SOFTWARE LIFE CYCLE**

This is the phase in the software life cycle where the actual software is implemented. The result of this phase consists of source code, together with documentation to make the code reliable. Implementation is the action that must follow any preliminary in order for something to actually happen. It encompasses all the processes involved in getting new software and hardware operating properly in its environment, including installation, configuration, running, testing and making necessary changes.

The word deployment is sometimes used to mean the same thing. Implementation refers to post sales process of guiding a client from purchase to use of the software or hardware that was purchased.

This includes Requirement Analysis, Scope Analysis, Customizations, System Integrations, User Policies, User Training and Delivery. These steps are often overseen by a Project Management Methodologies set forth in the Project Management Body of Knowledge. Software Implementations involve several professionals that are relatively new to the knowledge based economy such as business Analysts, Technical Analysts, Solution Architect and Project Managers.

There are 6 Stages in Software Implementation:

1. **verification and validation**

Purpose:

Verify that a system confirms that to its specification and validate that the systems is what the customer wanted.

Activities:

* + Inspections and reviews.
  + Testing: Unit Testing, Integration Testing, System Testing, Acceptance Testing, etc.

1. **Integration**

Purpose:

Assemble the system from subsystems.

Approaches

Incremental integration or big-bang?

Problems:

* + lntetface mismatches or misunderstanding.
  + Resources Conflicts: Memory, Control, Physical
  + Communication Problems.
  + Integration with COTS.

1. **Installation**

Purpose:

System is installed in the environment and made for ready for use.

Problems

* Unexpected issues such as too little space, power, heat, etc.
* Configuration problems and tuning.
* Training

**4) Software Evolution and Maintenance**

Purpose

Change the system while it is first in use to handle changing requirements.

Covers:

* Trouble fixing (Corrective maintenance).
* Adaptation to new platform/technology(Adaptive Maintenance).
* ew requirements (Perfective Maintenance).

Activities:

* Assess existing software.
* Propose change, Modify System.

**5) System De-commissioning**

Purpose:

* Take the system out of use at the end of its life-time.
* Should be carefully planned for large systems.

Issues:

* Safety or secmity considerations
* Reuse of software or data.

**6) Project Management**

Planning

Approaches, Resources, Schedule, Increments or Deliveries and Budget and Managing all these.

Cost Estimation

**HW,** SW, Organization

**Hardware:**

Processor: Intel Core i3 0r Ryzen 5 or Higher

RAM: 2GB

Hard disk space : 2-3 GB

**Software:**

Tool: Pycharm Community Edition

:XAMPP

Database: MySQL Workbench

Operating System: Windows 2007/2010

Web browser: Google Chrome

**About Python Tkinter as GUI**

.Tkinter-What is it?

.The [tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter) package (“Tk interface”) is the standard Python interface to the Tcl/Tk GUI toolkit. Both Tk and [tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter) are available on most Unix platforms, including macOS, as well as on Windows systems.

.Running python -m tkinter from the command line should open a window demonstrating a simple Tk interface, letting you know that [tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter) is properly installed on your system, and also showing what version of Tcl/Tk is installed, so you can read the Tcl/Tk documentation specific to that version.

.Tkinter supports a range of Tcl/Tk versions, built either with or without thread support. The official Python binary release bundles Tcl/Tk 8.6 threaded. See the source code for the \_tkinter module for more information about supported versions.

.Tkinter is not a thin wrapper, but adds a fair amount of its own logic to make the experience more pythonic. This documentation will concentrate on these additions and changes, and refer to the official Tcl/Tk documentation for details that are unchanged.

**.Tkinter Architecture**

Tcl/Tk is not a single library but rather consists of a few distinct modules, each with separate functionality and its own official documentation. Python’s binary releases also ship an add-on module together with it.

Tcl

Tcl is a dynamic interpreted programming language, just like Python. Though it can be used on its own as a general-purpose programming language, it is most commonly embedded into C applications as a scripting engine or an interface to the Tk toolkit. The Tcl library has a C interface to create and manage one or more instances of a Tcl interpreter, run Tcl commands and scripts in those

instances, and add custom commands implemented in either Tcl or C. Each interpreter has an event queue, and there are facilities to send events to it and process them. Unlike Python, Tcl’s execution model is designed around cooperative multitasking, and Tkinter bridges this difference (see [Threading model](https://docs.python.org/3/library/tkinter.html#threading-model) for details).

Tk

Tk is a [Tcl package](https://wiki.tcl-lang.org/37432) implemented in C that adds custom commands to create and manipulate GUI widgets. Each [Tk](https://docs.python.org/3/library/tkinter.html#tkinter.Tk) object embeds its own Tcl interpreter instance with Tk loaded into it. Tk’s widgets are very customizable, though at the cost of a dated appearance. Tk uses Tcl’s event queue to generate and process GUI events.

Ttk

Themed Tk (Ttk) is a newer family of Tk widgets that provide a much better appearance on different platforms than many of the classic Tk widgets. Ttk is distributed as part of Tk, starting with Tk version 8.5. Python bindings are provided in a separate module, [tkinter.ttk](https://docs.python.org/3/library/tkinter.ttk.html#module-tkinter.ttk).

Internally, Tk and Ttk use facilities of the underlying operating system, i.e., Xlib on Unix/X11, Cocoa on macOS, GDI on Windows.

When your Python application uses a class in Tkinter, e.g., to create a widget, the [tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter) module first assembles a Tcl/Tk command string. It passes that Tcl command string to an internal \_tkinter binary module, which then calls the Tcl interpreter to evaluate it. The Tcl interpreter will then call into the Tk and/or Ttk packages, which will in turn make calls to Xlib, Cocoa, or GDI.

## Tkinter Modules

Support for Tkinter is spread across several modules. Most applications will need the main [tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter) module, as well as the [tkinter.ttk](https://docs.python.org/3/library/tkinter.ttk.html#module-tkinter.ttk) module, which provides the modern themed widget set and API:

**from** **tkinter** **import** \*

**from** **tkinter** **import** ttk

*class*tkinter.**Tk**(*screenName=None*, *baseName=None*, *className='Tk'*, *useTk=True*, *sync=False*, *use=None*)

Construct a toplevel Tk widget, which is usually the main window of an application, and initialize a Tcl interpreter for this widget. Each instance has its own associated Tcl interpreter.

The [Tk](https://docs.python.org/3/library/tkinter.html#tkinter.Tk) class is typically instantiated using all default values. However, the following keyword arguments are currently recognized:

screenName

When given (as a string), sets the DISPLAY environment variable. (X11 only)

baseName

Name of the profile file. By default, baseName is derived from the program name (sys.argv[0]).

className

Name of the widget class. Used as a profile file and also as the name with which Tcl is invoked (argv0 in interp).

useTk

If True, initialize the Tk subsystem. The [tkinter.Tcl()](https://docs.python.org/3/library/tkinter.html#tkinter.Tcl) function sets this to False.

sync

If True, execute all X server commands synchronously, so that errors are reported immediately. Can be used for debugging. (X11 only)

use

Specifies the id of the window in which to embed the application, instead of it being created as an independent toplevel window. id must be specified in the same way as the value for the -use option for toplevel widgets (that is, it has a form like that returned by winfo\_id()).

Note that on some platforms this will only work correctly if id refers to a Tk frame or toplevel that has its -container option enabled.

[Tk](https://docs.python.org/3/library/tkinter.html#tkinter.Tk) reads and interprets profile files, named .*className*.tcl and .*baseName*.tcl, into the Tcl interpreter and calls [exec()](https://docs.python.org/3/library/functions.html#exec) on the contents of .*className*.py and .*baseName*.py. The path for the profile files is the HOME environment variable or, if that isn’t defined, then [os.curdir](https://docs.python.org/3/library/os.html#os.curdir).

**tk**

The Tk application object created by instantiating [Tk](https://docs.python.org/3/library/tkinter.html#tkinter.Tk). This provides access to the Tcl interpreter. Each widget that is attached the same instance of [Tk](https://docs.python.org/3/library/tkinter.html#tkinter.Tk) has the same value for its [tk](https://docs.python.org/3/library/tkinter.html#tkinter.Tk.tk) attribute.

**master**

The widget object that contains this widget. For [Tk](https://docs.python.org/3/library/tkinter.html#tkinter.Tk), the master is [None](https://docs.python.org/3/library/constants.html#None) because it is the main window. The terms master and parent are similar and sometimes used interchangeably as argument names; however, calling winfo\_parent() returns a string of the widget name whereas [master](https://docs.python.org/3/library/tkinter.html#tkinter.Tk.master) returns the object. parent/child reflects the tree-like relationship while master/slave reflects the container structure.

**children**

The immediate descendants of this widget as a [dict](https://docs.python.org/3/library/stdtypes.html#dict) with the child widget names as the keys and the child instance objects as the values.

tkinter.**Tcl**(*screenName=None*, *baseName=None*, *className='Tk'*, *useTk=False*)

The [Tcl()](https://docs.python.org/3/library/tkinter.html#tkinter.Tcl) function is a factory function which creates an object much like that created by the [Tk](https://docs.python.org/3/library/tkinter.html#tkinter.Tk) class, except that it does not initialize the Tk subsystem. This is most often useful when driving the Tcl interpreter in an environment where one doesn’t want to create extraneous toplevel windows, or where one cannot (such as Unix/Linux systems without an X server). An object created by the [Tcl()](https://docs.python.org/3/library/tkinter.html#tkinter.Tcl) object can have a Toplevel window created (and the Tk subsystem initialized) by calling its loadtk() method.

The modules that provide Tk support include:

[tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter)

Main Tkinter module.

[tkinter.colorchooser](https://docs.python.org/3/library/tkinter.colorchooser.html#module-tkinter.colorchooser)

Dialog to let the user choose a color.

[tkinter.commondialog](https://docs.python.org/3/library/dialog.html#module-tkinter.commondialog)

Base class for the dialogs defined in the other modules listed here.

[tkinter.filedialog](https://docs.python.org/3/library/dialog.html#module-tkinter.filedialog)

Common dialogs to allow the user to specify a file to open or save.

[tkinter.font](https://docs.python.org/3/library/tkinter.font.html#module-tkinter.font)

Utilities to help work with fonts.

[tkinter.messagebox](https://docs.python.org/3/library/tkinter.messagebox.html#module-tkinter.messagebox)

Access to standard Tk dialog boxes.

[tkinter.scrolledtext](https://docs.python.org/3/library/tkinter.scrolledtext.html#module-tkinter.scrolledtext)

Text widget with a vertical scroll bar built in.

[tkinter.simpledialog](https://docs.python.org/3/library/dialog.html#module-tkinter.simpledialog)

Basic dialogs and convenience functions.

[tkinter.ttk](https://docs.python.org/3/library/tkinter.ttk.html#module-tkinter.ttk)

Themed widget set introduced in Tk 8.5, providing modern alternatives for many of the classic widgets in the main [tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter) module.

Additional modules:

\_tkinter

A binary module that contains the low-level interface to Tcl/Tk. It is automatically imported by the main [tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter) module, and should never be used directly by application programmers. It is usually a shared library (or DLL), but might in some cases be statically linked with the Python interpreter.

[idlelib](https://docs.python.org/3/library/idle.html#module-idlelib)

Python’s Integrated Development and Learning Environment (IDLE). Based on [tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter).

tkinter.constants

Symbolic constants that can be used in place of strings when passing various parameters to Tkinter calls. Automatically imported by the main [tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter) module.

[tkinter.dnd](https://docs.python.org/3/library/tkinter.dnd.html#module-tkinter.dnd)

(experimental) Drag-and-drop support for [tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter). This will become deprecated when it is replaced with the Tk DND.

[tkinter.tix](https://docs.python.org/3/library/tkinter.tix.html#module-tkinter.tix)

(deprecated) An older third-party Tcl/Tk package that adds several new widgets. Better alternatives for most can be found in [tkinter.ttk](https://docs.python.org/3/library/tkinter.ttk.html#module-tkinter.ttk).

[turtle](https://docs.python.org/3/library/turtle.html#module-turtle)

Turtle graphics in a Tk window.

**MySOL 2008 as DataBase:**

MySQL server is scalable database system whose primary purpose is to serve as a back-end database for client program, such as your Web browser, an accounting program like online Ice-Cream Parlour-anything that makes use of the data. In the most common usage scenario, a client program connects to

MySQL server and requests some information, whereupon MySQL server processes the requests and returns results. The client must then interpret and display these results- eg. Displaying list of Users in alphabetical order. MySQL server can structured information. Eg. You can instantly search through millions ofrecords and view the results of the search in many different formats.

You can combine different data into one set; you can transform some formats into others; you can set security rules to be enforced by MySQL server; and so on.

**FEATURES OF APACHE SERVER 2008:**

* Information representation
* Unique definition of rows
* Systematic treatment of Null values
* Guaranteed access
* High level Update, Insert, and Delete Retrieving information from the database.
* Accepting query language statements.
* Enforcing security specifications.
* Enforcing data integrity specifications
* Enforcing transaction consistency
* Managing data sharing
* Optimizing queries

**SYSTEM MAINTAINANCE AND EVOLUTION**

Maintenance is not a pan of software development, but is an extremely important activity in the life of a software product. Maintenance includes all the activities after installation of the software that is performed to keep the system operational. Maintenance activities related to fixing of errors modifications that may occur due to changes in the requirements is called 'Adaptive Maintenance'. It may happen that the application may fail after development. A software product wears out due to age. In software, failures occur due to bugs or errors that get introduced during the design and development process. Software may fail after operating correctly for some time. It is necessary to check that the vendor's computer system is having the requirements specified that too in full version and not trial period version. If the requirements are met then check for the bugs and errors do possible modifications. These bugs should be removed. So as not to encounter such problems in the future it its necessary to visit the organization frequently to check whether the system is maintained properly and is error free. In most software organizations, the budget for software maintenance is much larger than for software development.

The Maintenance can be classified as Corrective, Adaptive, Perfective and Prevention.

**Corrective Maintenance:**

Corrective Maintenance changes the software to connect defects. Means repairing, processing or Performance failures or making alteration because of previously ill-defined problems.

**Adaptive Maintenance:**

Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

**Perfective or Enhancement Maintenance:**

Enhancing the performance or modifying the programs according to the user's additional or changing needs. Perfect maintenance extends the software beyond its original functional requirements.

**Prevention Maintenance:**

Computer Software deteriorates due to change, and because of this, preventive maintenance must be conducted to enable the software to serve the needs of its end users. In essence, preventive maintenance makes changes to computer programs so that they can be more easily connected, adapted and enhanced.

**COST AND BENEFIT ANALYSYS**

Why cost & benefit analysis ... ??

Why should you do a cost-benefit analysis for your project? IT projects frequently over promise and under deliver. Executive managers have become aware of this performance issue and the cost-benefit analysis is their guarantee that the project team has carefully evaluated the project before commencement, studying the whole life cycle costs and the expected benefits.

The most common way to caJTying out an economic assessment of a proposed information system, or other development, is by comparing the expected cost of development and operation of the system with the benefits of having it in place. Assessment is based upon the question of whether the estimated costs are exceeded by the estimated income and other benefits. Additionally, it is manually necessary to ask whether or not the project under consideration is the best of number of options.

Identifying and estimating all of the costs and benefits of carrying out the project: This includes development costs of the system, the operating costs and the benefits that are expected to accrue from the operation of the system. Where the proposed system is replacing an existing one, these estimates should reflect the cost and benefits due to the

new system. A sales order processing system, for example, could not claim to benefit an organization by the total value of sales - only by the increase due to the use of new system.

Expressing these costs and benefits in common units:- We must the net benefit, which is the difference between the total benefit and total cost. To do this, we must express each cost and benefit in memory terms,

Most costs are relatively easy to identify and quantify in approximate monetary te1ms. It is helpful to categorize costs according to where they originate in the life of the project.

Setup costs:- Include the costs of putting the system into place. These consist mainly of the any new hardware and ancillary equipment but will also include cost of file conversion, recruitment and staff training.

Operational costs: - consists of the costs of operating the system once it has been installed.

Direct benefits:- these accure directly from the operation of the proposed system. These could, for example, include the reduction in salary bills though the introduction of a new, computerized system.

**Cost Analysis**

The cost can be calculated with respect to amount of hours spent while doing the project. It is the cost which is calculated with the respect to the amount of work done by each person on the project and also the number of days taken for completing each form and each module. It also takes the time required for designing each fo1m and validating them. Also the time spent in testing each module with all its forms. The amount of hours spent can be calculated with respect to the number of forms and modules created dming this curses

It also coJTesponds to the amount of consumption of electricity, printout, telephone bills travelling expenses, internet, stationary etc.

**Cost Estimation**

For a given set of requirements it is desirable to know how much it will cost to develop the software to satisfy a given requirements, and how much time development will take. The cost of a project is a function of many parameters. Foremost among them is the size of the project. Other factors that affect the cost are programmer ability, experience of the developers in the area, complexity of the project, and reliability requirements. It is also due to the requirements of software, hardware and human resources.

Cost required for the project is to install Microsoft Visual Studio software. **SQL**

express comes inbuilt into it.

**Benefit analysis**

There are many benefits the company will have by using our software

I. System will helping in reducing the file work.

1. System will make the job easier than other system
2. It will save time
3. Job of searching records and its information will be much easier, hence time for retrieving particular record will be less.
4. Records can be saved which solves the problem of misplacing.

Because lack of accuracy & information in improper format the existing system is computerized to reduce the manual record keeping work to improve timelessness during implementation of an organization employee details.

Cost and benefits Analysis is a three step Process-

**1st step:**

Estimate the anticipate development and operational cost

* **Development cost:**

Cost occurred during the development of the system.

* **Operational Cost:**

Cost that will occur after the installing of the system.

**2nd Step:**

Estimate the anticipate financial benefits.

* **Financial Benefits:**

They are expected annual saving or increase in revenue derived from the installation of the new system.

**3rd Step:**

The cost and benefit analysis is calculated based on detailed estimate cost and benefits.

**METHODS USED FOR TESTING**

**Introduction**

Testing is the process of mnning a system with the intention of finding errors. Testing enhances the integrity of a system by detecting deviations in design and e1rnrs in the system. Testing aims at detecting error-prone areas. This helps in the prevention of errors in a system. Testing also adds value to the product by conforming to the user requirements.

The main purpose of testing is to detect errors and error-prone areas in a system. Testing must be thorough and well-planned. A partially tested system is as bad as an untested system. And the price of an untested and under-tested system is high.

Software Testing is a process of executing the program with objective finding an e1rnr. Software testing is successful only if all error from software are removed. Good testing techniques are available which find maximum uncovered error but no techniques is available which will find all error so we have to used set of setting technique to find errors.

**Objectives of Testing:**

The objectives of testing are:

* Testing is a process of executing a program with the intent of finding enurs.
* A Successful test case is one that uncovers an as- yet-undiscovered enur.
  1. **White Box Testing (WBT)**

White Box Testing is related with a su-ucrure or internal logic. Various test cases designed which takes care of the following:

**1.** Every statement in a program must be executed at least once.

ii. Every path in program must be executed at least once which is called as path analysis.

111. Every logical decision must be executed on their true or false sides.

iv. Executed all loops their boundaries and within operational boundaries

Static analyzers is one of the tool used for WBT which is used to check function calls, and initialized variables, variables defines **but** not used etc.

**Advantages of WBT**

l) As all statement or logic is verified it is possible to give guarantees of error free software.

2) WBT is helpful for improving the quality of system in terms if increasing

executions speed by selecting proper logic, minimizing memory space by removing unnecessary statement etc.

**Disadvantages of WBT**

I) Only experts persons able to do WBT because understanding of the code is necessary.

1. Each and every statement and path is checked this requires lot of time for testing.
2. Cost of testing is high because we have to pay expert persons.
3. **Black Box Testing**

Black Box Testing is related with input and output only and not related with internal structure of the program. In BET it is checked some input is produce by the program or not various set of input test cases are prepared and applied on the program and coJTesponding output is verified. BBT is used to find the error in data structure, performances errors etc.

**Methods for BBT**

* 1. **Partioning**

In this input domain of the program is divided into different categorized from which test cases can be de.rived same as input portioning. Output ponioning is also done for output domain.

* 1. **Error Guessing**

Possible error are listed and inputs conditions can based on this are constructed.

* 1. **Boundary value Analysis**

1. Test cases should be design in such way that input conditions are either just above or just below of the actual value.
2. **4. Random Testing**
3. Program is tested by randomly selecting some subset of the possible input values test data may be chosen randomly or by sampling process.

**Advantage of BBT**

1. Expert persons are not requiring for testing because internal logic or statement are not verified
2. Cost of BBT may be less as expert persons are not required.

**Disadvantages of BBT**

**l)** It is not possible to give guarantee of error free software because possibility of the error still remains logic is not checked.

2) Requires large test data even after we have applied large test data we can't says that this is sufficient.

**a-p Testing**

After software development is complicated is completed during implementing of the system a proper training is given to the users get early training before implementing at the developer side. Though proper training is given to the customer there is the no guarantee that they will handle the system property. Due to implementing also there are no chances of the errors in handling the systems. So the methods ofo- testing is applied.

***a* Testing**

A customer conducts this test at the developer side. *a* test are conducts in a controls enrollment. Customer operates the software and developer watches and usages problems.

**p testing**

This test is conducted at one user sides by the end user of the software.

During this, developer is not presents so user fields free in handling the system and records all the systems these problems are then reported to the developer. Developers solve the problems and makes necessary changes in software and then modified software is given to all customers.

**USER MANUAL AND SECURITY POLICY**

The tool has various users and hence the tool is divided based on the login provided by the user. The access is different for different users based on their login type. The two types of user regular-Admin and irregular-Admin.

This is a user manual to describe how website works:

* When regular-Admin enters into the tool :

When a user wants to login, he has to give his respective username and password. The username and password fetch to the database, if they matched then user can login to the main program and we called him the regular admin. Then user will able to order any command, if that exist within the query then the task will going to perform or in else case it will return negative statement to the user.

The user can perform both online and offline task py providing commands. Also he can update his password at the time of user login by giving update information.

* When irregular-Admin enters into tool :

Unlike a regular user, the permission to access all the commands and use the main program is not possible in case of an irregular admin. In case of using the tool he has to create new account by providing his username and password at the time of user login.

**CONCLUSION**

VERONICA is a very helpful voice assistant without any doubt as it saves time of the user by conversational interactions, its effectiveness and efficiency. But while working on this project, there were some limitations encountered and also realized some scope of enhancement in the future which are mentioned below:

9.1. LIMITATIONS

9.1.1. Security is somewhere an issue, there is no voice command encryption in this project.

9.1.2. Background voice can interfere

9.1.3. Misinterpretation because of accents and may cause inaccurate results.

9.1.4. VERONICA cannot be called externally anytime like other traditional assistants like Google Assistant can be called just by saying, “Ok Google!”

9.2. SCOPE FOR FUTURE WORK

9.2.1. Make VERONICA to learn more on its own and develop a new skill in it.

9.2.2. VERONICA android app can also be developed.

9.2.3. Make more VERONICA voice terminals.

9.2.4. Voice commands can be encrypted to maintain security

FUTURE DEVELOPEMENT

*As* for other future developments, the following can be done:

* The Administrator of the web site can be given more functionality, like looking at a specific customer's profile, the books that have to be reordered, etc.
* Multiple Ordering Carts can be allowed.
* A console for the data center may be made available to allow the personnel to monitor on the sites which were cleared for hosting during a particular period.

Moreover, it is just a beginning; further the system may be utilized in various other types of auditing operation viz. Network auditing or similar process/workflow based applications...

**REFERENCE**

**Textbook reference:**

* Software Engineering, Fifth Edition by Roger S. Pressman, Tata McGraw - Hill Publication
* SQL Complete Reference Second Edition by James R. Groff and Paul Weinberg Tata McGraw - Hill Publication

**Web Reference:**

[www.google.com](http://www.google.com/)

<http://msdn.microsoft.com>

[www.webschools.com](http://www.webschools.com)

**ACKNOWLEDGEMENT**

With due to regards I express my sincere thanks to **Mr. DILELSWAR SIR,** (Principal of LAKSHYA INSTITUTE OF TECHNOLOGY**)** and the Management for having facilitated me with the essential infrastructure resources without which this project would not have seen light of the day.

**I** am at the outset on the completions of "Online Ice-Cream Parlor" project, express our deep regard to those who have offered their invaluable guidance in hour of needs. I would like to thank **Prof. Mrs. CHANDRA KANTA ROUT** who gave me an opportunity to developed the Software Projects.

**My** sincere thanks to all teaching and non-teaching staff members of the **Information Technology Department LIT** for their encouragement and whole hearted support for this project.

Finally, I express my whole hearted gratitude towards the entire Computer Department and my Friends, Relatives and people who are directly or indirectly in the successful with completion of the projects.

**Jagadish Jena,**

**Roll – 37UITM206025**

**Lakshya Institute of Technology,**

**Bhubaneswar. Odisha.**