***SHRI GURU TEGH BAHADUR INSTITUTE OF MANAGEMENT AND INFORMATION TECHNOLOGY***

***MAJOR PROJECT***

***VIRTUAL VOICE ASSISTANT***



***Under Supervision of - Submitted by: -***

***Dr HARSHITA TULI RAGHAV BABBAR***

***GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY***

***(2018-2021)***

DECLARATION

I hereby declare that this project report entitled “VIRTUAL VOICE ASSISTANT”, **submitted to the Sri Guru Tegh Bahadur Institute of Management & Information Technology, Guru Gobind Singh Indraprastha University** in the partial fulfilment for the award of the Degree of **Bachelor of Computer Application** is an authentic record of work done by me under the guidance of Dr Harshita Tuli. The project has not previously formed the basis for the award of any other degree, Diploma, Associate ship, Fellowship or other title.

**RAGHAV BABBAR**

**BCA 6th SEMESTER**

**06390202018**

CERTIFICATE

This is to certify that **Mr. RAGHAV BABBAR**, University Enrolment Number- **06390202018,** has worked under my supervision to prepare this project report of “VIRTUAL VOICE ASSISTANT”. The work embodied in this report is original and is of the standard expected of B.C.A student and has not been submitted in part or full to this or any university for the award of any degree or diploma, He has completed all requirement of guidelines for research project and the work is fit for evaluation.

Date: Signature of HOD (IT): Dr Raj Kumar

Place: Delhi

Signature of Guide: Dr HARSHITA TULI

ACKNOWLEDGEMENT

The project work in this report is an outcome of continual work and intellectual support from various sources. It is almost impossible to express adequately the debts owed to many persons who have been instrumental in imparting this work a successful status. It is however a matter of great pleasure to express our gratitude and appreciation to all those people who had helped in completion of this project.

I would like to take the opportunity to thank Dr Harshita Tuli for giving me an opportunity to work on this project, which not only has increased our awareness but also has taught us the importance of work, it is because of their guidance, constant encouragement and inspiration that I have been able to accomplish the task of completion this project.

I express our sincere thanks to our project mentor Dr Harshita Tuli a for their invaluable guidance and frequent suggestions during the course of the project. Their suggestions helped me to maintain a good quality of work. I express our deep gratitude to them.

I would also like to thank my project external examiner and all the people who provided me with the facilities being required and conductive conditions for my BCA (3rd YEAR) PROJECT ON **VIRTUAL VOICE ASSISTANT.**

ABSTRACT

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

There is an increased overall awareness and a higher level of comfort demonstrated specifically by millennial consumers. In this ever-evolving digital world where speed, efficiency, and convenience are constantly being optimized, it’s clear that we are moving towards less screen interaction.

This engages the ability to communicate socially through natural language processing, holding (and analysing) information within the context of the user. It is suggested that new technologies may soon make the idea of virtual personal assistants a reality. Experiments conducted on this system, combined with user testing, have provided evidence that a basic program with natural language processing algorithms in the form of a VPA, with basic natural language processing and the ability to function without the need for other type of human input (or programming) may already be viable.

TABLE OF CONTENT

|  |  |  |  |
| --- | --- | --- | --- |
| **S No** | | **Topic** | **Page No** |
| 1 | | Certificate | - |
| 2 | | Acknowledgements | - |
| 3 | | List of Tables | - |
| 4 | | List of Figures | - |
| 5 | | List of Abbreviations | - |
| 6 | | **Chapter-1: Introduction** | 1-2 |
|  | | 1.1 About Project  1.2 Objective  1.3 Scope  1.4 Features  1.5 Need of the System  1.6 Limitation of Existing System |  |
| 7 | | **Chapter-2: Requirements and Analysis** | 3-6 |
|  | | 2.1 Introduction  2.2 Software requirement Specification  2.3 Functional Requirement of the Project  2.4 Hardware Requirements  2.5 Software Requirements  2.6 Technologies used  2.7 Use Case Diagram |  |
| 8 | **Chapter-3: Software Design** | 7-8 |
|  | 3.1 Introduction  3.2 Module Description   * 1. Data Flow Diagram (DFD)      1. 0-Level      2. 1-Level |  |
| 9 | **Chapter-4: Testing** | 9-11 |
|  | 4.1 Introduction  4.2 Test Plan  4.3 Conclusions |  |
| 10 | **Chapter-5: Roles and Responsibilities** | 12 |
| 11 | **Chapter-6: Conclusion** | 13 |
|  | 6.1 Future Scope |  |
| 12 | Appendices  -Screenshot  -Codes  -References | 14-24 |

LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Title** | **Page No** |
| 1 | Use Case | 6 |
| 2 | Level “0” DFD | 8 |
| 3 | Level “1” DFD |
| 4 | Wakeup Validation | 11 |
| 5 | Front Page | 14 |
| 6 | Send Mail/Message Page |
| 7 | Screen Recorder Page | 15 |

LIST OF SYMBOLS

|  |  |  |
| --- | --- | --- |
| Entity Symbol | Name | Description |
|  | Relationship | Relationships are associations between or among entities |
| https://images.visual-paradigm.com/docs/vp_user_guide/11/94/2575/2745/uml_use_case_icon_19877.png | [Use Case](https://www.visual-paradigm.com/support/documents/vpuserguide/94/2575/84257_usecasediagr.html#uml-use-case) | Horizontally shaped ovals that represent the different uses that a user might have. |
| https://images.visual-paradigm.com/docs/vp_user_guide/11/94/2575/2745/uml_association_icon_19878.png | Association | A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases. |
| https://images.visual-paradigm.com/docs/vp_user_guide/11/94/2575/2745/uml_actor_icon_19879.png | [Actor](https://www.visual-paradigm.com/support/documents/vpuserguide/94/2575/84257_usecasediagr.html#uml-actor) | Stick figures that represent the people actually employing the use cases. |
| https://images.visual-paradigm.com/docs/vp_user_guide/11/94/2575/2745/uml_system_icon_19880.png | System | A specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario. |
|  | Data Flow | Movement of data between external entities, processes and data stores is represented with an arrow symbol, which indicates the direction of flow. |
|  | Process | An activity that changes or transforms data flows. Since they transform incoming data to outgoing data, all processes must have inputs and outputs on a DFD |
|  | Data Store | A data store does not generate any operations but simply holds data for later access |
| **Entity - ERD Symbol** | Entity | Also known as actors, sources or sinks, and terminators, external entities produce and consume data that flows between the entity and the system being diagrammed. |
| Attribute - ERD Symbol | Attribute | Attributes are characteristics of an entity, a many-to-many relationship, or a one-to-one relationship |

CHAPTER 1

INTRODUCTION

1.1 ABOUT PROJECT

In today’s era almost all tasks are digitalized. We have Smartphone in hands and it is nothing less than having world at your finger tips. These days we aren’t even using fingers. We just speak of the task and it is done. There exist systems where we can say Text Dad, “I’ll be late today.” And the text is sent. That is the task of a Virtual Assistant.

It also supports specialized task such as booking a flight, or finding cheapest book online from various e-commerce sites and then providing an interface to book an order are helping automate search, discovery and online order operations.

Virtual Assistants are software programs that help you ease your day to day tasks, such as showing weather report, creating reminders, making shopping lists etc. They can take commands via text (online chat bots) or by voice. Voice based intelligent assistants need an invoking word or wake word to activate the listener, followed by the command.

Formy project the wake word is DUDE. We have so many virtual assistants, such as Apple’s Siri, Amazon’s Alexa and Microsoft’s Cortana. For this project, wake word was chosen DUDE

* 1. OBJECTIVE

Main objective of building personal assistant software (a virtual assistant) is using semantic data sources available on the web, user generated content and providing knowledge. The main purpose of an intelligent virtual assistant is to answer questions that users may have. This may be done in a business environment, for example, other business website, with a chat interface. On the mobile platform, the intelligent virtual assistant is available as a call-button operated service where a voice asks the user “How can I help you?” and then responds to verbal input.

* Functionalities provided by VIRTUAL VOICE ASSISTANT are as follows:
* Open any website in the browser.
* Send an email to your contacts.
* Launch any system application.
* Tells you present time.
* Play you a song.
* Change background and voice
* Tells you latest news.
* Download Songs from YouTube
* Record Screen
* Write notes
* Tell jokes
* Tell Weather
* Search any information on Google or Wikipedia
  1. SCOPE

Voice assistants will continue to offer more individualized experiences as they get better at differentiating between voices. However, it’s not just developers that need to address the complexity of developing for voice as brands also need to understand the capabilities of each device and integration and if it makes sense for their specific brand. They will also need to focus on maintaining a user experience that is consistent within the coming years as complexity becomes more of a concern. This is because the visual interface with voice assistants is missing. Users simply cannot see or touch a voice interface.

Of course, one have to look at the human interaction that humans provide in chat service. This isn't an issue with Desktop Assistants because of the wide range of services it provides and it is available 24\*7. It would be wrong to say that Desktop Assistant is evolving. Desktop Assistant are more intelligent. Even there are reports that 80- 85% of businesses will be giving out more enhanced Desktop Assistants by 2020.

* 1. FEATURES
* Open any website in the browser.
* Send an email to your contacts.
* Launch any system application.
* Tells you present time.
* Play you a song.
* Change background and voice
* Tells you latest news.
* Download Songs from YouTube
* Record Screen
* Write notes
* Tell jokes
* Tell Weather
* Search any information on Google or Wikipedia

1.5 LIMITATION OF EXISTING SYSTEM

Voice recognition software enables hands free use of the applications, lets users to query or command the agent through voice interface. This helps users to have access to the agent while performing other tasks and thus enhances value of the system itself. DUDE also have ubiquitous connectivity through Wi-Fi or LAN connection, enabling distributed applications that can leverage other APIs exposed on the web without a need to store them locally

CHAPTER 2

REQUIREMENTS AND ANALYSIS

2.1 INTRODUCTION

Requirement is determining the needs or conditions to meet for a new or altered product. Requirements analysis is critical to the success of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

2.2 SOFTWARE REQUIREMENT SPECIFICATION

The virtual voice assistant should satisfy the following requirements which are discussed below**:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Requirement** | **Essential or**  **Desirable** | **Description of the Requirement** | **Remarks** |
| 1 | Send an email to your contacts | Essential | Dude should be able to send email to his/her contact | Dude is able to send email to his/her contact |
| 2 | Play song | Essential | Dude should be able to play any song | Dude can play any song |
| 3 | Download song from YouTube | Essential | Dude should be able to download any song from YouTube | Dude can download any song from YouTube |
| 4 | Tell News | Essential | Dude should be able to tell you latest new | Dude can tell latest news |
| 5 | Record Screen | Essential | Dude should be able to record your screen | Dude can record your screen |
| 6 | Tell weather | Essential | Dude should be able to tell weather | Dude can tell weather |
| 7 | Launch Application | Essential | Dude should be able to launch system application | Dude can launch any application in system |
| 8 | Open Websites | Essential | Dude should be able to open any website | Dude can open any website |
| 9 | Track Phone | Essential | Dude should be able to find user phone | Dude can track user phone |

2.3 FUNCTIONAL REQUIREMENTS OF THE PROJECT

Required dude is for providing info to users about activities. The system should satisfy the following requirements:

* Open any website in the browser.
* Send an email to your contacts.
* Launch any system application.
* Tells you present time.
* Play you a song.
* Change background and voice
* Tells you latest news.
* Download Songs from YouTube
* Record Screen
* Write notes
* Tell jokes
* Tell Weather
* Search any information on Google or Wikipedia

2.4 HARDWARE REQUIREMENT

* **Processor:** Pentium-4 class processor or Above
* **RAM:** 2GB of RAM or Above
* **Operating System:** Windows XP or Above

2.5 SOFTWARE REQUIREMENT

* **Front End:** Tkinter
* **Language:** Python v3.8.5

2.6 USE CASE DIAGRAM

The use case model for any system consists of “use cases”. Use cases represent different ways in which the system can be used by the user. A simple way to find all the use case of a system is to ask the questions “What the user can do using the system?” The use cases partition the system behaviour into transactions such that each transaction performs some useful action from the users’ point of view.

The purpose of the use case to define a piece of coherent behaviour without revealing the internal structure of the system. A use case typically represents a sequence of interaction between the user and the system. These interactions consist of one main line sequence is represent the normal interaction between the user and the system. The use case model is an important analysis and design artefact (task). Use cases can be represented by drawing a use case diagram and writing an accompany text elaborating the drawing.

In the use case diagram, each use case is represented by an ellipse with the name of use case written inside the ellipse. All the ellipses of the system are enclosed with in a rectangle which represents the system boundary. The name of the system being modules appears inside the rectangle. The different users of the system are represented by using stick person icon. The stick person icon is normally referred to as an Actor. The line connecting the actor and the use cases is called the communication relationship. When a stick person icon represents an external system, it is annotated by the stereo type<<external system>>.

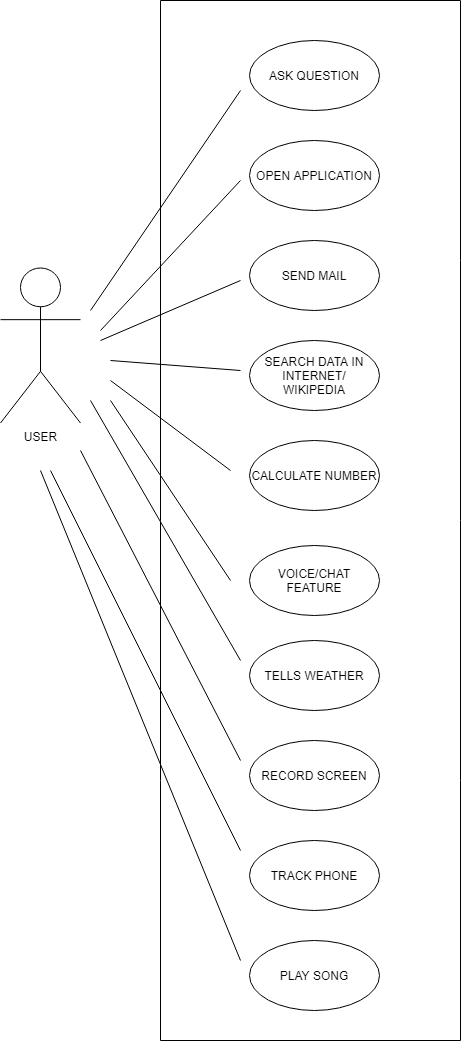


Fig.1 USE CASE DIAGRAM

CHAPTER 3

SOFTWARE DESIGN

3.1 INTRODUCTION

Software design is an engineering process by which we are representing of something that is to be built. It is a blueprint for constructing the software... After making Software Requirement Specification of the project, now we are in position to design the software.

The system design shows how Software will be structured to satisfy the requirements identified during the analysis phase. The design process is a translation of requirements into a description of the software structure, software components, interfaces and data necessary for the implementation phase. The design phase provides a complete blueprint for the implementation activity.

3.2 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.

It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

* The following observations about DFDs are essential:

1. All names should be unique. This makes it easier to refer to elements in the DFD.
2. Remember that DFD is not a flow chart. Arrows is a flow chart that represents the order of events; arrows in DFD represents flowing data. A DFD does not involve any order of events.
3. Suppress logical decisions. If we ever have the urge to draw a diamond-shaped box in a DFD, suppress that urge! A diamond-shaped box is used in flow charts to represents decision points with multiple exists paths of which the only one is taken. This implies an ordering of events, which makes no sense in a DFD.
4. Do not become bogged down with details. Defer error conditions and error handling until the end of the analysis.

3.2.1 0-LEVEL DFD

A level 0 DFD is called fundamental system model or context model represents entire software element as a single bubble with input and output data indicating by incoming and outgoing arrows. It is also known as fundamental system model, or context diagram represents the entire software requirement as a single bubble with input and output data denoted by incoming and outgoing arrows. Then the system is decomposed and described as a DFD with multiple bubbles.



Fig.2 0-Level DFD

3.2.2 1-LEVEL DFD

In 1-level DFD, a context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main objectives of the system and breakdown the high-level process of 0-level DFD into subprocesses.

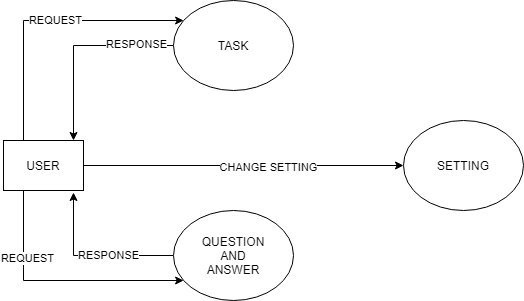


Fig.3 1-Level DFD

CHAPTER 4

TESTING

4.1 INTRODUCTION

Executing a program with the intent of finding errors is called testing. Testing is vital to the success of any system. Testing is done at different stages within the development phase. System testing makes a logical assumption that if all parts of the system are correct, the goals will be achieved successfully. Inadequate testing or no testing at all leads to errors that may come up after a long time when correction would be extremely implementation. The testing of the system was done on both artificial and live data. In order to test data test cases are developed.

4.2 TEST PLANS

The test strategy implementation of the project that defines how the testing has been carried out. We followed the Reactive approach to testing. This means that we continued testing each component or module of our project during its creation. However, we are only able to test it completely once the coding and designing part of the project is complete. The testing methodologies in the development process which make sure that the software can successfully operate in multiple environments and across different platforms. We are trying to develop a project which will work on the web and as a mobile application. This will help the end-users to access the platform with ease and flexibility. Also, the offline mode is available for users to access the network anywhere and anytime.

4.2.1 WHITE BOX TESTING

It is a testing approach in which internal structure is known to the tester. It is best suited for a lower level of testing like Unit Testing, Integration testing. White box testing is usually done by testers and developers. It is a structural test of the software. As developers, we realised that testing through the white-box testing procedure is more feasible than black-box testing because we developed the code and therefore have all the knowledge about the internal structure. The features of white-box testing are in favour of the developers. However, when we learn about other kinds of testing such as non-functional testing, we realise that it is also possible to state some information and facts about them based on our project.

4.2.2 Unit Testing

Unit testing is the first level of testing which ensures that the individual components of a piece of software at the code level are functional and work as they were designed. The unit testing was done during the development of the specific components of each module by an individual. This was conducted manually by repeatedly debugging the errors for the proper functionality of the module components. We typically wrote and aimed to execute the tests prior to the software being deployed in front of the evaluators or testers. This testing helped us debug the code at an early stage so that we don’t find errors when we combine all the components of the module together. E.g., Creation and testing of the login page and forgot password page.

4.2.3 Functional Testing

This testing is based on the requirements specified by various stakeholders and incorporates testing to ensure that all the components are working properly.

4.2.4 Integrating Testing

Integration testing is the next step after unit testing. These are then tested as a group through integration testing to ensure whole segments of an application behave as expected. Once all the components were tested, we began combining those components into one module. Manual tests were also conducted for integration. This test enabled us to run all the components one after the other smoothly. For e.g., creation and testing of the homepage and the subpages individually and then combining it to test it for proper functionality.

4.2.5 SYSTEM TESTING

It is executing programs to check logical changes made in it with intention of finding errors. a system is tested for online response, volume of transaction, recovery from failure etc. System testing is done to ensure that the system satisfies all the user requirements.

4.3 TEST CASES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No. | Module Name | Test Case No | Test Case Description | Expected Result |
| 1 | Response Time | TC1 | To make sure that the system respond back time is efficient | Response Time is good |
| 2 | Accuracy | TC2 | To assure that answers retrieved by system are accurate as per gathered data | Gives accurate answer to the question |
| 3 | Wake Up Validation | TC3 | To wake up Dude when call “Activate Dude” | Wake Up successfully when call |
| 4 |  |  |  |  |

4.4 TEST REPORT

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Test Case No.** | **Test Status** | **Test Report** |
| 1 | TC1 | Successful | - |
| 2 | TC2 | Successful | - |
| 3 | TC3 | Successful | Fig 4 |



Fig.4 Wake-Up Validation

CHAPTER 5

ROLES AND RESPONSIBILITIES

5.1 ROLES

While doing my project on VIRTUAL VOICE ASSISTANT, my responsibilities towards my project are:

* I was involved in collection of information related to my project.
* I have used very easy and user-friendly designing mechanism.
* I am responsible for approving the designing tool.
* Planning and Execution of these page were done by me.
* I am responsible for managing instances of the above-mentioned modules.

5.1 RESPONSIBILITIES

* Work on definition of development requirements and priorities.
* Interfaces with other systems.
* Set up and maintenance of security rights and access permission.
* Contributing to technical strategy, policy and procedure.
* Development and operation of technical testing programmes.
* Production of technical documentation to agreed quality standards.
* Reporting on progress/issues to management and users.

CHAPTER 6

CONCLUSION

The main aim of the project was to develop a Desktop Assistant that will be used to identify answers related to user submitted questions. To provide with sufficient information that is required by the user. A background research took place, which included an overview of the conversation procedure and any relevant desktop Assistant available. A desktop Assistant already in user were excellent service that is provided. The developed system is made on python programming language to be more specific Python 3.8. Different libraries where used such as Speech Recognition, Text to Speech convertor, Short Mail Transferring Protocols (SMTP), pyjokes, Wikipedia and many more. It provides information regarding the weather, News, it can play music, it can search for topics on Wikipedia, can setup an alarm, Display the current date and time. User can collect information through this application. It reduces both man power and time. Due to support of NLP user can ask queries in very formal way. No need ask queries in very strict and specific way. The user should aware of general rules of English Language. The goal is to provide people a quick and easy way to have their questions answered.

6.1 FUTURE SCOPE

Of course, one have to look at the human interaction that humans provide in chat service. This isn't an issue with Desktop Assistants because of the wide range of services it provides and it is available 24\*7. It would be wrong to say that Desktop Assistant is evolving. Desktop Assistant are more intelligent. Even there are reports that 80- 85% of businesses will be giving out more enhanced Desktop Assistants by 2020. Natural Language Programming (NLP) helps in giving a raised human involvement, hence making the Desktop Assistants more communicative. Undoubtedly, Desktop Assistants are a great help for e-commerce stores. The goal is to provide people a quick and easy way to have their questions answered. It can also be incorporated with the college's website so that we can give user a better experience. Then users do not want to install this application they can use this application via college website on any system.

APPENDICES

**SCREENSHOT**



Fig.5 Front Page

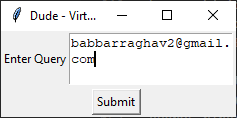


Fig. 6 Sending Email/Message Page

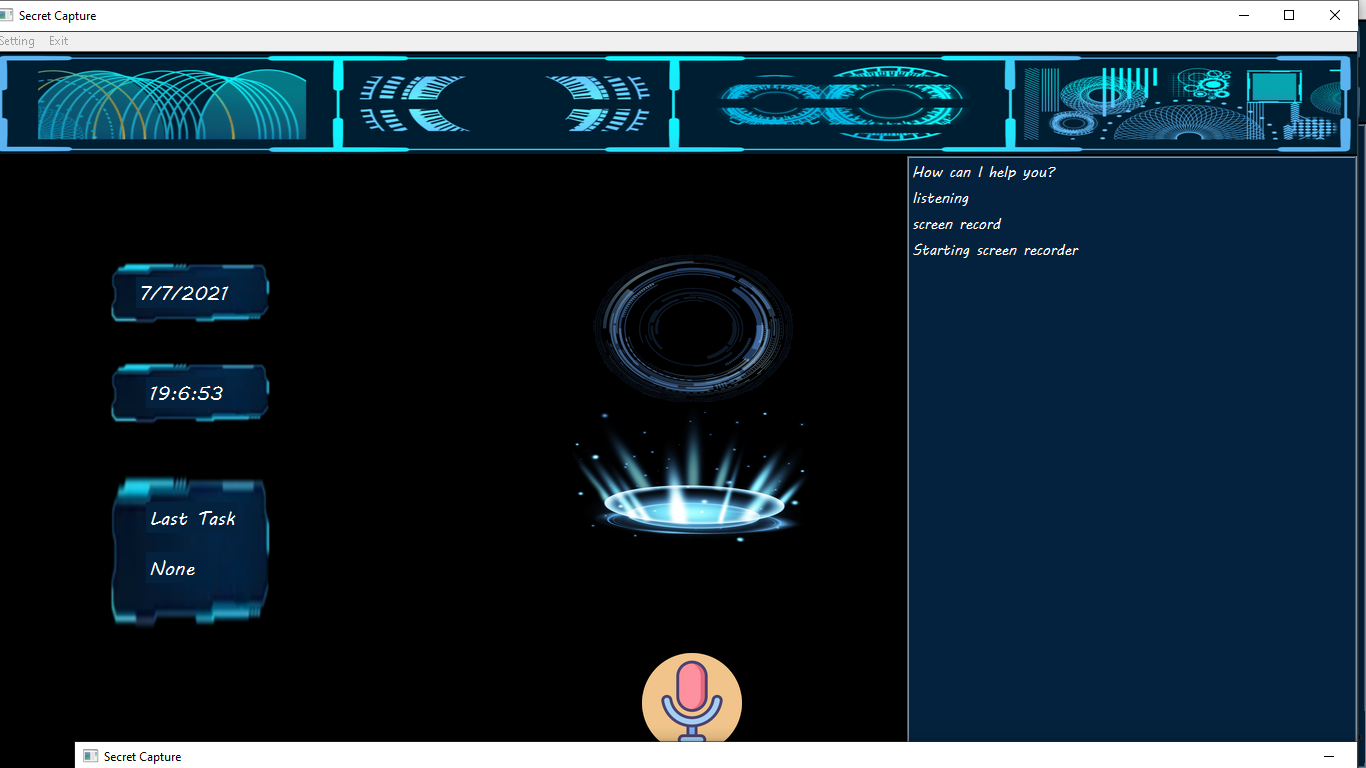


Fig 7 Screen Recorder Page

**CODES**

#function where task for dude

def task(query):

    query = query.lower()

    speak(query)

    #search on wikipeadia

    if 'wikipedia' in query:

        speak('Searching for wikipedia query...')

        try:

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

            speak("According to Wikipedia")

            msg(result)

            root.update\_idletasks()

            root.update()

            speak(result)

            last\_task.config(text="Wikipedia")

        except Exception as e:

            speak("Due to some problem, I am not able to access wikipedia regarding")

            speak(query)

            print(e)

    #tells joke

    elif 'joke' in query:

        joke = pyjokes.get\_joke()

        msg(joke)

        root.update\_idletasks()

        root.update()

        speak(joke)

        last\_task.config(text="Joke")

    #tells weather

    elif 'weather' in query or 'temperature' in query:

        wolfram(query)

        last\_task.config(text="Weather")

    #calculate the value

    elif 'calculate' in query:

        wolfram(query)

        last\_task.config(text="Calculate")

    #search for value in google

    elif 'search' in query and 'for' in query:

        try:

            last\_task.config(text="search for " + str(query))

            query = query.replace("search for", "")

            msg('Searching for ' + query)

            speak('Searching for ' + query)

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

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

        except Exception as e:

            print(e)

            speak("Due to some problem, I am not able to search")

    #play video in youtube and also can download it!

    elif 'video on' in query:

        try:

            last\_task.config(text="Video on " + str(query))

            query = query.replace('play video on ', '')

            query = query.replace(' ', '+')

            html = urllib.request.urlopen("https://www.youtube.com/results?search\_query=" + query)

            video = re.findall(r"watch\?v=(\S{11})", html.read().decode())

            speak("Enjoy the video on " + query)

            wb.open("https://www.youtube.com/watch?v=" + video[0])

            time.sleep(10)

            msg("Do you want to download this song")

            msg("If yes then press Y else press N")

            root.update\_idletasks()

            root.update()

            speak("Do you want to download this song")

            speak("If yes then press Y else press N")

            value = root2()

            top.destroy()

            if 'y' in value or 'Y' in value:

                yt = YouTube("https://www.youtube.com/watch?v=" + video[0])

                speak("Downloading " + query)

                msg("Downloading " + query)

                videos = yt.streams.filter(only\_audio=True).first()

                videos.download('C:\\Users\\raghav\\Music\\Playlists')

                speak("Downloaded")

                msg("Downloaded")

            else:

                speak("OK")

        except Exception as e:

            print(e)

            speak("Due to some problem, I am not able to open")

    elif 'play' in query and 'song' in query:

        path = "C:\\Users\\raghav\\Music\\Playlists"

        speak("Playing a song. Please wait")

        msg("Playing a song. Please wait")

        root.update\_idletasks()

        root.update()

        song = os.listdir(path)

        value = random.choice(song)

        os.startfile(os.path.join(path, value))

        last\_task.config(text="Played Song")

    elif 'track' in query and 'phone' in query:

        try:

            speak("Tracking your Phone")

            msg("Tracking your phone")

            root.update\_idletasks()

            root.update()

            number = '9953181244'

            client = Client(acc\_id, acc\_token)

            call = client.calls.create(

                    twiml='<Response><Say>I am DUDE, virtual artificial intelligence made by Raghav</Say></Response>',

                    to = '+91'+number,

                    from\_ = '+19138082981'

            )

            speak("Calling your Phone")

            last\_task.config(text="Phone Track")

        except Exception as e:

            print(e)

            speak("Due to some problem, I am not able to call")

    elif 'message' in query:

        try:

            msg("Enter the number whom you want to send message")

            root.update\_idletasks()

            root.update()

            speak("Enter the number whom you want to send message")

            number = root2()

            top.destroy()

            msg("What message you want to send")

            root.update\_idletasks()

            root.update()

            speak("What message you want to send")

            message = root2()

            top.destroy()

            client = Client(acc\_id, acc\_token)

            message\_no = client.messages.create(

                    body = message,

                    from\_ = '+19138082981',

                    to = '+91'+number

            )

            print(message\_no.sid)

            speak("Messaging")

            msg("Messaging")

            last\_task.config(text="Sending Message")

        except Exception as e:

            print(e)

            speak("Due to some problem, I am not able to send message")

    elif 'send' in query and 'mail' in query:

        msg("Who do you want to send email")

        root.update\_idletasks()

        root.update()

        speak("Who do you want to send email")

        to = root2()

        top.destroy()

        msg("what message you want to send")

        root.update\_idletasks()

        root.update()

        speak("what message you want to send")

        letter = root2()

        top.destroy()

        email(letter, to)

        last\_task.config(text="Send Mail")

    elif 'screen' in query and 'record' in query:

        speak("Starting screen recorder")

        msg("Starting screen recorder")

        root.update\_idletasks()

        root.update()

        width = GetSystemMetrics(0)

        height = GetSystemMetrics(1)

        time\_stamp = datetime.datetime.now().strftime('%Y-%m-%d %H-%M-%S')

        file\_name = f'{time\_stamp}.mp4'

        fourcc = cv2.VideoWriter\_fourcc('m', 'p', '4', 'v')

        captured\_video = cv2.VideoWriter(file\_name, fourcc, 20.0, (width, height))

        while True:

            img = ImageGrab.grab(bbox=(0, 0, width, height))

            img\_np = np.array(img)

            img\_final = cv2.cvtColor(img\_np, cv2.COLOR\_BGR2RGB)

            cv2.imshow('Secret Capture', img\_final)

            captured\_video.write(img\_final)

            if cv2.waitKey(10) == ord('q'):

                break

        last\_task.config(text="Record Screen")

    elif 'notes' in query:

        speak("What should i write")

        msg("What should i write")

        root.update\_idletasks()

        root.update()

        note = root2()

        top.destroy()

        speak("What should i name the file")

        msg("What should i name the file")

        root.update\_idletasks()

        root.update()

        file\_names = root2()

        top.destroy()

        file = open(str(file\_names) + '.txt', 'w')

        speak("Should i include date and time")

        msg("Should i include date and time")

        root.update\_idletasks()

        root.update()

        snfm = root2()

        top.destroy()

        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)

        speak("File Created")

        msg("File Created")

        root.update\_idletasks()

        root.update()

        last\_task.config(text="Notes")

    elif 'news' in query:

        news\_url = "https://news.google.com/news/rss"

        speak("Ok, i am telling you top 5 news")

        msg("Ok, i am telling you top 5 news")

        root.update\_idletasks()

        root.update()

        client = urllib.request.urlopen(news\_url)

        xmlpage = client.read()

        client.close()

        page = soup(xmlpage, "html.parser")

        list = page.findAll("item")

        for i in range(5):

            lists = list[i]

            msg("News " + str(i+1) + ":" + lists.title.text)

            root.update\_idletasks()

            root.update()

            speak("News " + str(i+1) + ":" + lists.title.text)

        last\_task.config(text="News")

REFERENCES

* <https://docs.djangoproject.com/en/3.1/>
* https://books.goalkicker.com/PythonBook/
* <https://developer.paypal.com/docs/archive/checkout/integrate/>
* <https://getbootstrap.com/docs/5.0/getting-started/introduction/>
* https://developer.mozilla.org/en-US/docs/Web/CSS/Reference