Voice Based OS

Rishik TS - 191IT254
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
rishik.tulaa@gmail.com

Gosu Praneeth - 191IT117
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
praneethgosu322@gmail.com

Shashi Prakash A - 191IT204
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
shashiprakash1729@gmail.com

Naveen R - 191IT133
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
veninavi36@gmail.com

***Abstract*—Voice Based OS project is about designing a voice control which works like an OS and better than many voice assistants. It can do many tasks which an OS can do and some tasks which the existing voice assistants cannot do. We can even use it to fill a form by our voice. We can communicate with this assistant to search on google or YouTube or anything. We can use it to create, open folders and files, run files, delete folders and file, shutdown. Many things are possible.**

I. Introduction

Voice based applications and virtual assistants are taking over the market. Voice based things will be replaced by everything in the near future. Everything is now at our finger tips these days. In future, all the electronic gadgets like Laptops, TVs, Mobiles, Cars will be used by our voice.
We have many voice assistants like Siri, Ok Google, Cortana on our phones already. But these are just voice assistants.
Things like creating, opening, deleting folders, searching on google, filling a form can't be done by the voice assistants that we are familiar with.
Our idea is basically to step forward and to try to build a voice control which works like an OS, which can do tasks that make our working environment easier and efficient.

There are some cases where people think that why to operate our PCs only using the keyboard and mouse, why can't it understand our language and perform actions according to the commands we give. We thought to add the foundation stone for this idea. We tried our best to implement this. And also in future, technology will be developed in a way that our personal computers and mobile phones interact with us in a way that they will become our friends for general communication also.

So, we are doing this project where we can create, edit, delete folders, files and renaming them using the voice commands. And we can also open the applications in our computer.

II. Literature Survey

Voice is replacing everything in our surroundings. In near future, everything that we operate with our hands will be operated by using voice. Many voice assistants like Cortana, Siri, Ok Google are being evolved in the market. As voice provides authentication compared to text, electronic gadgets are being developed to work with voice.

A research was done on Speech Recognition:
[1] Parwinder pal singh has written a paper on **Speech Recognition as Emerging Revolutionary Technology**. In this paper he has described that Speech recognition is the translation of spoken words into text. It is also known as ”automatic speech recognition”, ”ASR”, ”computer speech recognition”, ”speech to text”, or just ”STT”. Speech Recognition is technology that can translate spoken words into text.

III. Problem Statement

“Building a Voice control which works like an OS and better than existing voice assistants”.

IV. Methodology and Work Done

A. Voice Control

Basically, this is done in 2 python files i.e, ”functions.py” and ”sdf.py”.
• ”**functions.py**” - consists of the shell commands that are needed for the operations that are done by OS. This will be imported into ”sdf.py” and appropriate functions are called whenever needed.The above needs can be fulfilled by using python library called ”os” and using the methods in them for the required file-handling operations and also to execute various other commands related to running commands in gnome-terminal.
• ”**sdf.py**” - this is the heart of our project. This is where the recognition of the user's requests, reply to user happens.

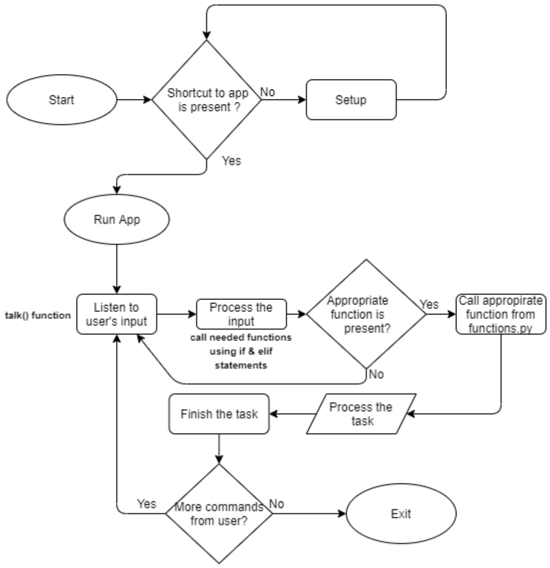
Functions which it can perform:

- Create, delete, rename, Open folders.
- Create, delete, rename, Open, run, edit files like Python, C, C++.
- Change the directory, Go to previous directory.
- Opens already installed applications in PC.
- Search on web - Google, YouTube etc.
- Shutdown in (user input) minutes.

The code for all the functions specified above are basically written in functions.py.

The code in sdf.py (talk() function) will recognize what the user is saying and will pass the parameters in appropriate syntax that the functions written in functions.py ask for. This is basically done by searching for key words in the input text using if-else statements and the functions are called.

In this way, our voice control can perform the functions said above which makes it equal to an OS (in some aspects) and makes it higher the voice assistants that we are familiar with.



Start
Shortcut to app is present ?
No
Setup
Yes
Run App
talk() function
Listen to user's input
Process the input
call needed functions using if & elif statements
Appropriate function is present?
Yes
No
Call appropriate function from functions.py
Process the task
Finish the task
More commands from user?
Yes
No
Exit

Fig. 1: Flow chart for Working of Voice control

B. Form Filling by Voice

We can use this feature by saying ”I want to fill a form” or user can just include the word ”form” in his input voice. It recognizes and opens a UI where the user has to enter the URL of the form that he wants to use it for.

Basically, when the user clicks ”Submit”, callback() function is called which will search for input fields inside the form tag in the web page.

Now, it uses the talk() function which we defined already to understand the user’s voice and it fills in the input field there. It is made to search only for non-password type (input) fields to protect the privacy of the user.

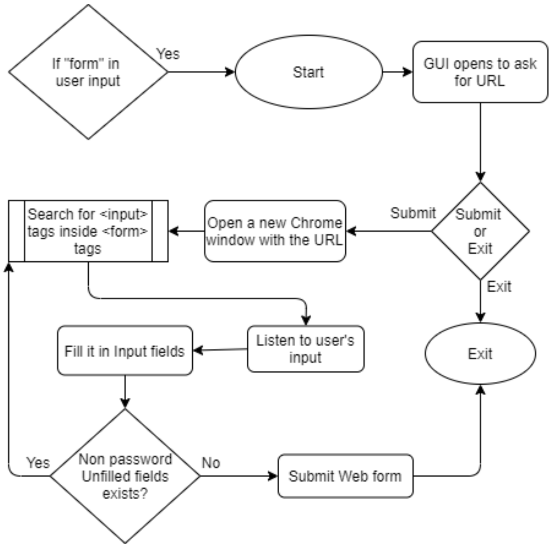


Fig. 2: Flow chart for Form filling

Tech stack:

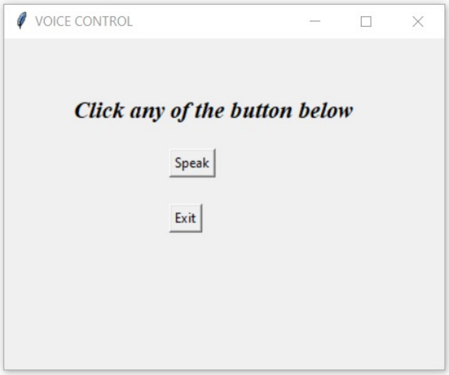
- os - Implement shell commands
- speech recognition - To recognize user’s input
- gTTs - Voice of program
- selenium - Form filling
- tkinter - UI

By the combination of all the above said libraries, the Idea is executed.

V. Execution and Results

First, when the user executes the program, it fetches the name of user from our PC using os.getlogin() and greets the user.

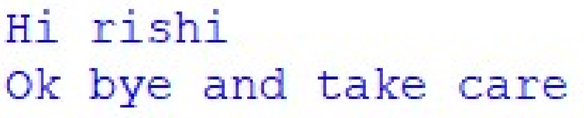
```
Hi rishi
```



Click any of the button below

Fig. 3: Greeting the user and asking to speak

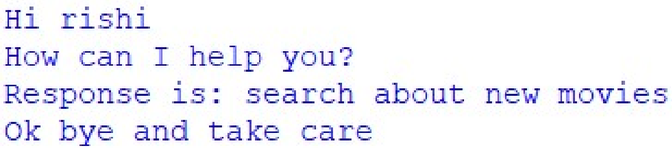
It asks if the user wants to speak or not. If the user clicks on ”Speak” then it asks ”How can I help you” else it exits by saying ”Ok bye and Take care”.



```
Hi rishi
Ok bye and take care
```

Fig. 4: Response when user clicks ”Exit”

Suppose, user wants to search about new movies



```
Hi rishi
How can I help you?
Response is: search about new movies
Ok bye and take care
```

Fig. 5: User says ”Search about new Movies”

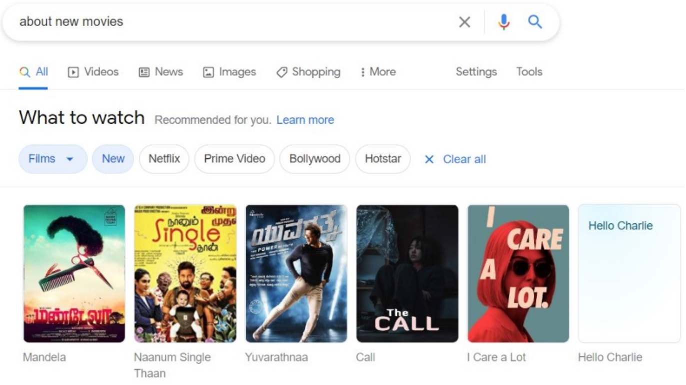
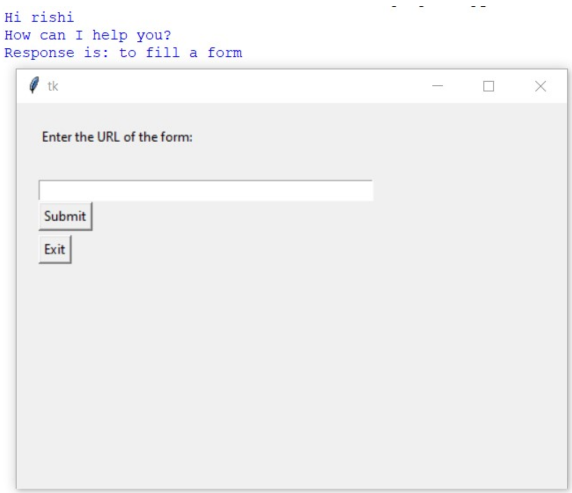


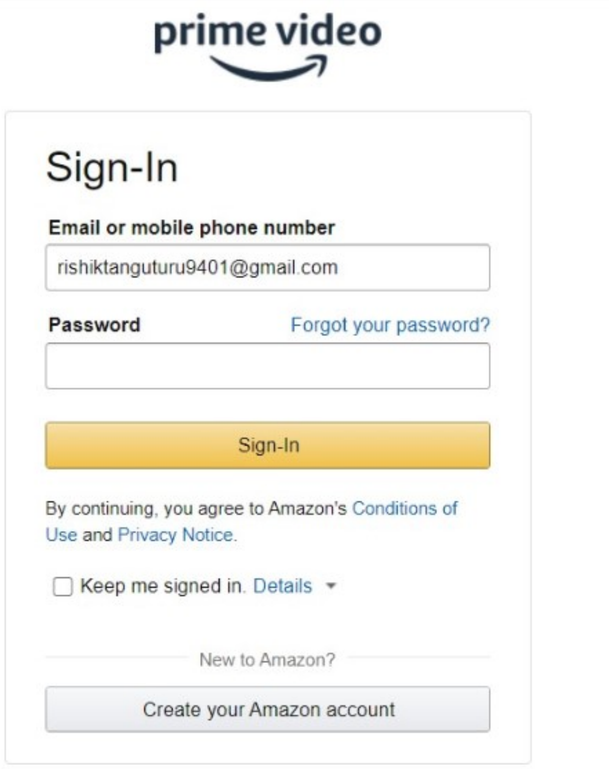
Fig. 6: About new movies on Google

So it searches about new movies.
Now, Lets see how it can be used to fill a form.



```
Hi rishi
How can I help you?
Response is: to fill a form
```

Fig. 7: User asking to ”Fill a form”



Sign-In

Email or mobile phone number

rishiktanguturu9401@gmail.com

Password Forgot your password?

Sign-In

By continuing, you agree to Amazon's Conditions of Use and Privacy Notice.

□ Keep me signed in. Details

New to Amazon?

Create your Amazon account

Fig. 8: User filling the form using Voice

Error Handling

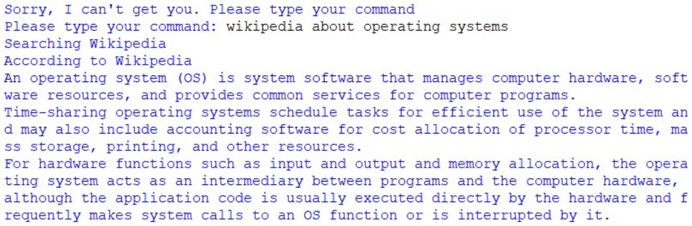
• Incase, input field is Email then user’s input ”at” is replaced by ”@”.

• Only lowercase letters are considered to remove the ambiguity since a voice doesn’t know capital or small.

Note:

Since, user is giving input by voice, there might be some situations where the voice control is unable to understand user’s input.

We handle this situation by putting a flag which basically checks if the voice control fails to work on the input given by him 2 times consecutively, then it asks user to type his/her request.



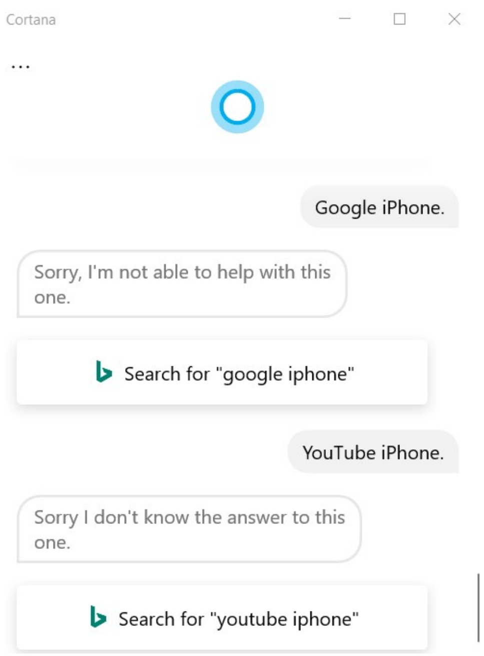
```
Sorry, I can't get you. Please type your command
Please type your command: wikipedia about operating systems
Searching Wikipedia
According to Wikipedia
An operating system (OS) is system software that manages computer hardware, soft
ware resources, and provides common services for computer programs.
Time-sharing operating systems schedule tasks for efficient use of the system an
d may also include accounting software for cost allocation of processor time, ma
ss storage, printing, and other resources.
For hardware functions such as input and output and memory allocation, the opera
ting system acts as an intermediary between programs and the computer hardware,
although the application code is usually executed directly by the hardware and f
requently makes system calls to an OS function or is interrupted by it.
```

Fig. 9: Error handling

In this way, problem in recognizing voice by the system can be dealt.

VI. Conclusion

So, basically in this way we can implement a voice control which can do tasks that are done by OS. And the tasks which can’t be done by the existing Voice assistants.



Google iPhone.

Sorry, I'm not able to help with this one.

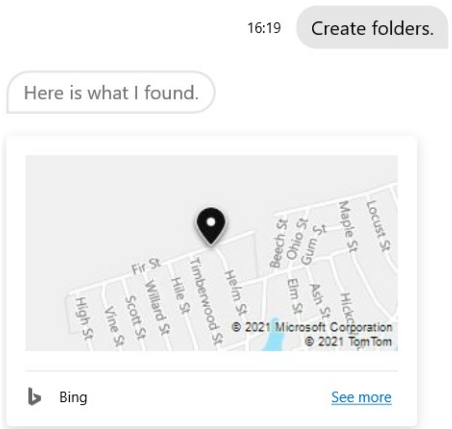
Search for "google iphone"

YouTube iPhone.

Sorry I don't know the answer to this one.

Search for "youtube iphone"

Fig. 10: Google and YouTube - Cortana



16:19 Create folders.

Here is what I found.

Fig. 11: Create folder - Cortana

We get such errors and wrong outputs if we use existing voice assistants.
But this is not the case with the voice control built.
This makes this project equal to an OS (in some aspects) and above than existing Voice assistants.

VII. References

[1] Selenium 3.141 documentation . *https://www.selenium.dev/selenium/docs/api/py/api.html*.

[2] Miscellaneous operating system interfaces. *https://docs.python.org/3/library/os.html* .

[3] How to type in a textbox with Selenium. *https://pythonspot.com/selenium-textbox/*.

[4] Tutorials Teacher,*Create UI in Python-Tkinter* . https://www.tutorialsteacher.com/python/create-gui-using-tkinter-python

[5] Speech Recognition *https://pypi.org/project/SpeechRecognition/*.