

***Operating Systems Project Report***

***[ Project Name: Voice Controlled Shell]***

***Instructor***

***Miss Safia Baloch***

***Sir Ali Shah Fatimi***

***Project Team***

***Syed Muhammad Faheem 20K-1054***

***Mohsin Ali Mirza 20K-0353***

***Ahmad Aleem 20K-0169***

***Submission Date***

***27-05-2022***

1. **Introduction**

In computing, a shell is a user interface for access to an operating system's services. In general, operating system shells use either a command-line interface (CLI) or graphical user interface (GUI), depending on a computer's role and particular operation. Our project is a new take on the traditional methods to use shells. Our voice-controlled shell uses voice commands to run bash scripts that perform various tasks like informing the user of the time, date, changing directories, shutting the machine down etc. We are using a Google API for Speech-to-text (STS) and Text-to-speech (TTS) services.

1. **Problem Statement**

To create a voice-controlled shell that responds to voice commands and executes shell scripts/commands in order to simplify the process of executing processes and to automate the use of the computer itself. The crux of our problem is to ease the life of the user and to ensure that people who are not familiar with scripting can utilize the major commands through ordinary statements using their voice.

1. **Methodology**

Our project is implemented using python. We have utilized the Google Speech-To-Text (GSTT) and Google Text-To-Speech (GTTS) libraries in order to decode our voice commands, change them into scripts and vice versa.

Moreover, we have used the py-audio, subprocess and system libraries that are built-in in python. The purpose of the py-audio library is to detect the microphone and capture the sound effectively. The purpose of the subprocess and system library is to run bash commands.

The procedure of use is as follows:

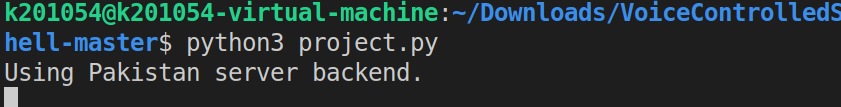
* PC has a microphone and a speaker connected to it which is always listening.
* Next the user will speak one of the valid commands from the list of commands that we have programmed.
* The program will verify if the command is valid and then find the action to be performed.
* To stop the process, you have to call ‘GOODBYE”.

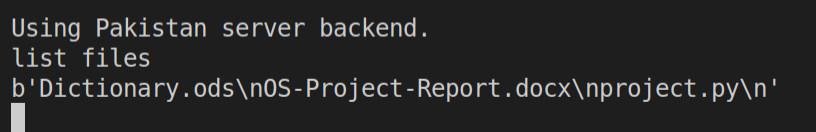
1. **Settings or Configuration Steps**

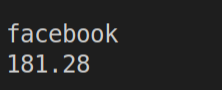
First of all, for this program to run, we need to have python3 environment set up on your computer and some set of external python modules for the API and functionalities to work. But before that, we need to have the most recent version of Ubuntu on our system. We ran this project on Ubuntu 22.04 64-bit version. Then, we set up the latest version of python which is Py 3.11.0b1. We, then installed the latest pip version (a tool to install python modules). Then, we moved on to installing the relevant modules with which we were able to support voice commands, scraping of information etc.

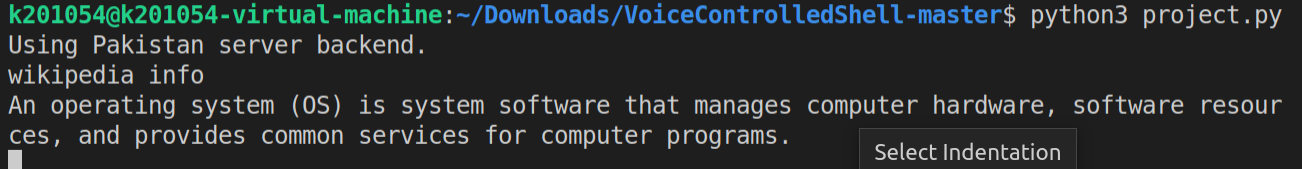
1. **Project Result**

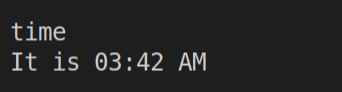
As a result of our efforts, we have created a voice-controlled shell that supports over sixty **(60)** commands. These voice commands correspond to a list of shell scripts and commands that then execute on the terminal.

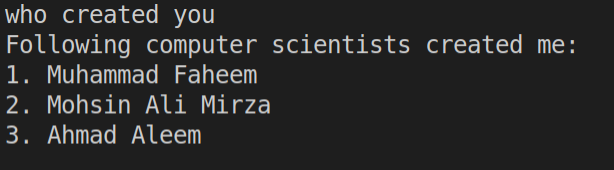












1. **References**

<https://atwing.net/home%20automation/shell-commands-assistant/>

<http://www.kscst.iisc.ernet.in/spp/40_series/39S_bestprojreports/39S_BE_1732.pdf>

<https://www.noobslab.com/2014/06/control-your-ubuntulinux-mint-system.html>