

Project Name: GUI Smart-Voice Calculator

Table of Content

Demo

Overview

Motivation

Technical Aspect

Installation

Run/How to Use/Steps

Directory Tree/Structure of Project

To Do/Future Scope

Technologies Used/System Requirement/Tech Stack

Refer to requirements.txt

Credits

Demo

Refer to Video_Files Folder.

Overview

This is a smart-voice digital-ware for performing calculations.

This repository contains the code for smart calculator using one of the python libraries "Tkinter" Library. Also used pyttsx3, PyAudio and speechrecognition libraries to transform it from simple to smart.

The purpose of creating this repository is to help the people who are just new to tkinter along with exceeding capability to more than simple interface.

These python libraries helped to outdo from basic to advanced operations. Also raised knowledge in discovering these libraries with practical use of it.

The video will help you in understanding the flow of output.

Motivation

The idea of making smart calculator came to my mind as this is the device without voice but with hands that most of the local shopkeepers and even people uses in their daily life. That led me to create this particular project as to help people who do not have their hands or old age people can use calculator with their voice. Otherwise I also thought that normal/any people who have luggage in their hands and because of that reason if they cannot press button then by using voice also, they can do calculations. This just gives wings to simple app. It is a step to enter into revolutionary future technological world. It helped me to expand the scope of mental ability with technical skills. This taught me amalgamation of intangible concepts, logical thinking with python knowledge.

Technical Aspect

Python when combined with Tkinter library provides a fast and easy way and is used to create GUI applications.

Python when integrated with pyttsx3, PyAudio and SpeechRecognition libraries then provides userfriendly application.

It provides handy to use flow for anyone who is using it as you saw in demo.

A user can choose to use with voice or without voice.

pyttsx3 is cross-platform a text-to-speech conversion library and major advantage is it also works offline and it is compatible with python2 as well as python3. That is, it is platform independent.

PyAudio provides python bindings for PortAudio, the cross-platform audio I/O library. You can easily use python to play and record audio on a variety of platforms with PyAudio.

In SpeechRecognition library, Speech must be converted from physical sound to an electrical signal with a microphone and then to digital data with an analog-to-digital converter.

Installation

Using intel core i5 9th generation with NVIDIA GFORCE GTX1650.

Windows 10 Environment Used.

Already Installed Anaconda Navigator for Python 3.x

The Code is written in Python 3.8.

If you don't have Python installed then please install Anaconda Navigator from its official site.

If you are using a lower version of Python you can upgrade using the pip package, ensuring you have the latest version of pip, *python -m pip install --upgrade pip and press Enter.*

Run/How to Use/Steps

Keep your internet connection on while running or accessing files and throughout too.

Follow this when you want to perform from scratch.

Open Anaconda Prompt, Perform the following steps:

Creating Virtual Environment named "smartvoice". You can give any name of your choice.

```
conda create -n smartvoice python=3.6
```

```
y
```

```
conda activate smartvoice
```

```
pip install pyttsx3
```

```
pip install PyAudio
```

```
pip install speechrecognition
```

```
[PyAudio will only work with 3.6 environment]
```

```
cd <PATH>
```

You can also create requirement.txt file as, pip freeze > requirements.txt

run files.

Creating Virtual Environment is necessary so that you do not have to install packages everytime you run the code. Once all required packages are installed in virtual environment then you only need to access/open the virtual environment and run the final file.

Follow this when you want to just perform on local machine.

Download ZIP File.

Right-Click on ZIP file in download section and select Extract file option, which will unzip file. Move unzip folder to desired folder/location be it D drive or desktop etc.

Open Anaconda Prompt, write `cd <PATH>` and press Enter.

eg: `cd C:\Users\Monica\Desktop\Projects\Python Projects 1\Tkinter\Calculator_with_voice`

Now, open virtual environment that you have created ie

`conda activate smartvoice`

In Anconda Prompt, `pip install -r requirements.txt` to install all packages.

In Anaconda Prompt, write `python <filename>.py` and press Enter. That is,

In Anaconda Prompt, write `python GUI.py` and press Enter.

Then, you can see smart calculator GUI on desktop as well as console and you can perform relevant operations.

By Clicking on X button /Speaking Close, it will close the interface.

You can also minimize and maximize it.

You can also run all codes from Command Prompt instead of Anaconda Prompt after setting Environmental Variable Path Settings.

Note: I have created smartvoice virtual environment and used for more than one project and therefore you might see more than one unused library in requirements.txt especially for this project so do not worry because I am using them in another project under similar virtual environment. Whenever you get No Module <name of package> Error then see its PyPI Documentation and Install it using `pip install <package-name>` written there. In some cases, you need to install its .whl file which I will inform you if its necessary.

Note: `cd <PATH>`

[Go to Folder where file is. Select the path from top and right-click and select copy option and paste it next to `cd` one space <path> and press enter, then you can access all files of that folder] [cd means change directory]

Directory Tree/Structure of Project

Folder Name: Tkinter>Calculator_with_voice

calc.py

speak.py

speech.py

smartcalculator.py

GUI.py

To Do/Future Scope

It is possible to custom built it depending on various requirements. Then it will transcend all its execution.

Can also add air finger gesture function to it.

Can also add gaze-controlled ability to it.

Technologies Used/System Requirement/Tech Stack



Credits

RahulSingh