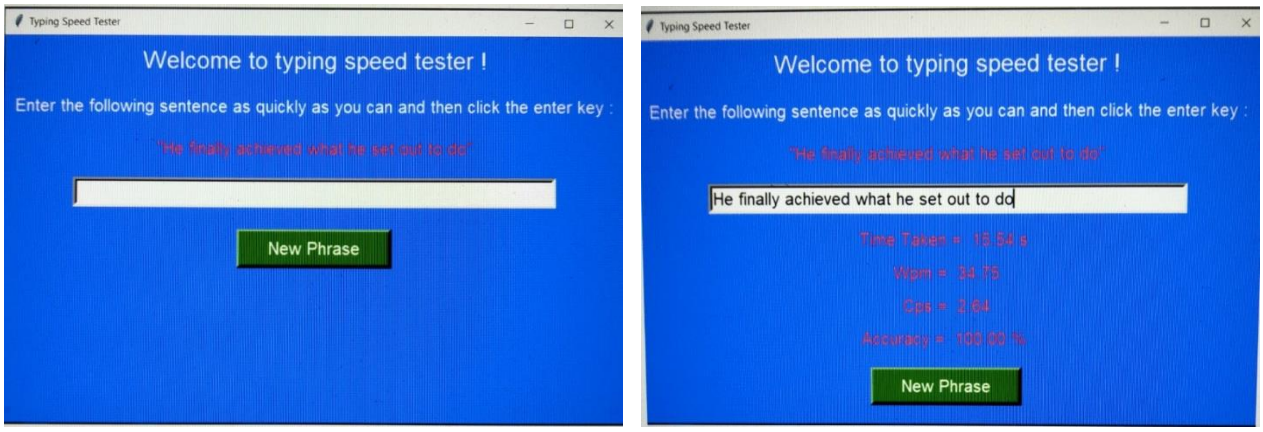
Project Name: GUI Typing Speed Checker

Table of Content

Demo



Once interface is visible then type sentence as shown and then press Enter, which will display result.
For getting new sentence, click on 'New Phrase' Button.

Overview

This is a Typing Speed Checker for performing calculations.
This repository contains the code for Typing Speed Checker using one of the python libraries "Tkinter" Library. Also used bs4, requests, lxml libraries as sentences are been taken up directly from online resources instead of offline pre-installed book.
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.
It helps to perform not only normal operations but also computational skills.
The screenshot will help you in understanding the flow of output.

Motivation

The reason behind making Typing Speed Checker is there are many stationary shops who keep Typist and while hiring them this app will provide exact idea of how fast they are and also be helpful in hiring by comparing scores between two or more candidates before appointing one of them. It also helps in making decision for owner of shop. It will also boost up confidence for candidates or students who are practicing Typing. Every-time they practice, they can check their scores and it will be beneficial for them to improve on further. Building such kind of apps not only helps to improve my python skills but also boosts my integration skill with routine observation in surroundings.

Technical Aspect

Python when combined with Tkinter library provides a fast and easy way and is used to create GUI applications.
Beautiful Soup or bs4 library is for pulling data out of HTML or other markup files.
It provides user-friendly flow for anyone who is using it as you saw in demo.
Beautiful Soup saves programmers hours and days and one of the most popular and used library. bs4 works with favorite parser to provide idiomatic ways of navigating, searching and modifying the parse tree. bs4 relies on a parser, the default is lxml.
lxml is most feature-rich package and allows easy handling of markup files.
It turns objectified (module) XML into python objects.
lxml can interface to the parsing capabilities of bs4.
Requests library is designed to be used by humans to interact with the language. It allows you to send HTTP/1.1 requests extremely easily. It builds reliable HTTP-speaking applications.

Installation

Using intel core i5 9th generation with NVIDIA GFORECE 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 bs4
pip install requests
pip install lxml
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 every-time 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\Typing_Speed_Checker
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_Typing_Speed_Checker.py and press Enter.
Then, you can see Typing Speed Checker GUI on desktop 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: Tkinter>Typing_Speed_Checker
GUI_Typing_Speed_Checker.py

To Do/Future Scope

Can make it big paragraphs test.
Also deploy it on Heroku so that it will be available to needy.

Technologies Used/System Requirement/Tech Stack

Credits

Anik31