Programming Fundamentals

Lab Report

Team Members:

- **Ali Mubeen (201201037)**
- **Abdul Moiz Malik (210201082)**
- **Ahmad Hassan (201201072)**
- **Ahtisham Ali (210201086)**
- **Khizar Abbas (21201066)**

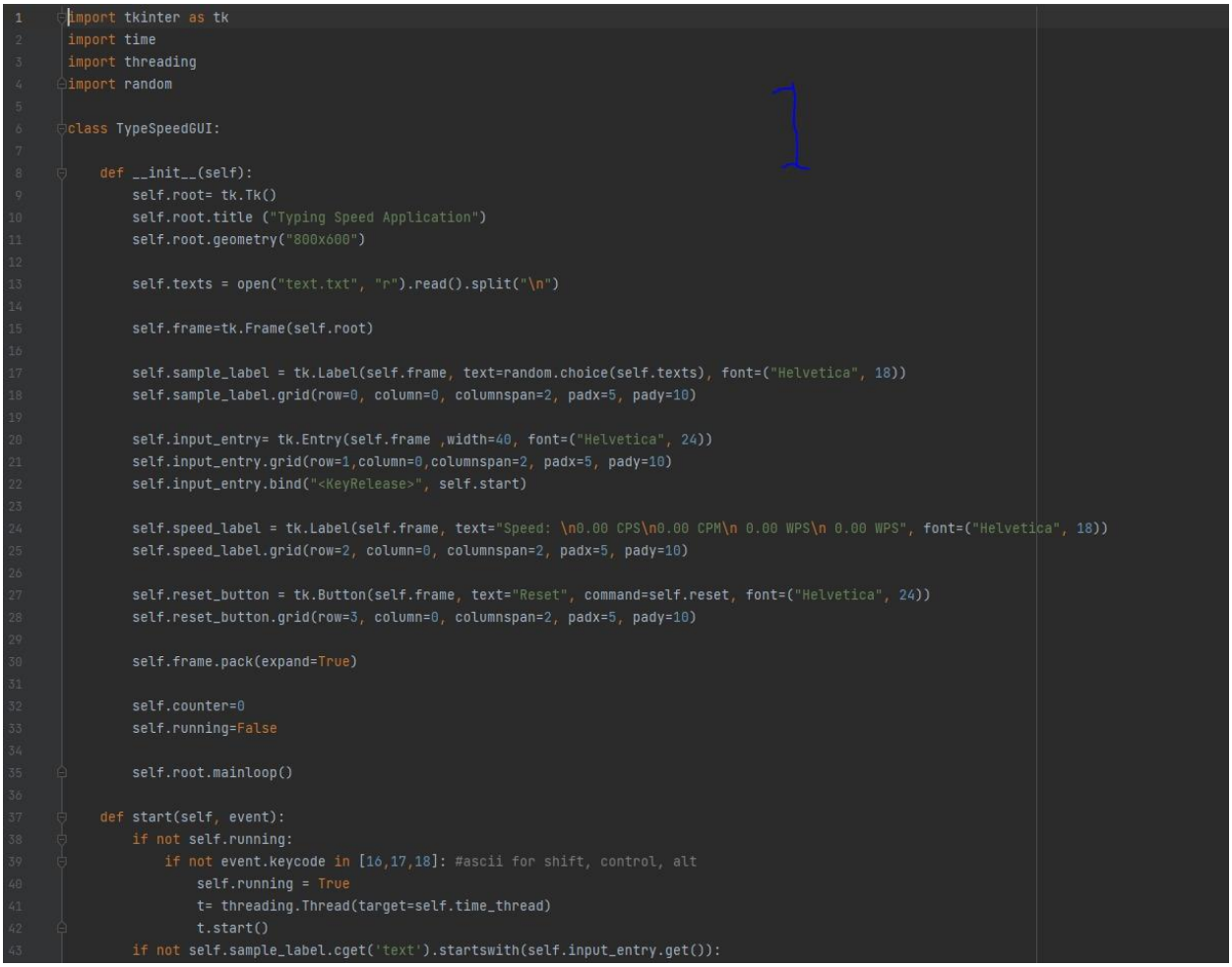
Project Description:

Typing speed is measured by the number of words you can type correctly in a set amount of time. A **word** is equivalent to five keystrokes. During a test, both speed and accuracy are measured. You will receive a number that indicates your average words per minute (WPM) and a percentage that indicates your accuracy.

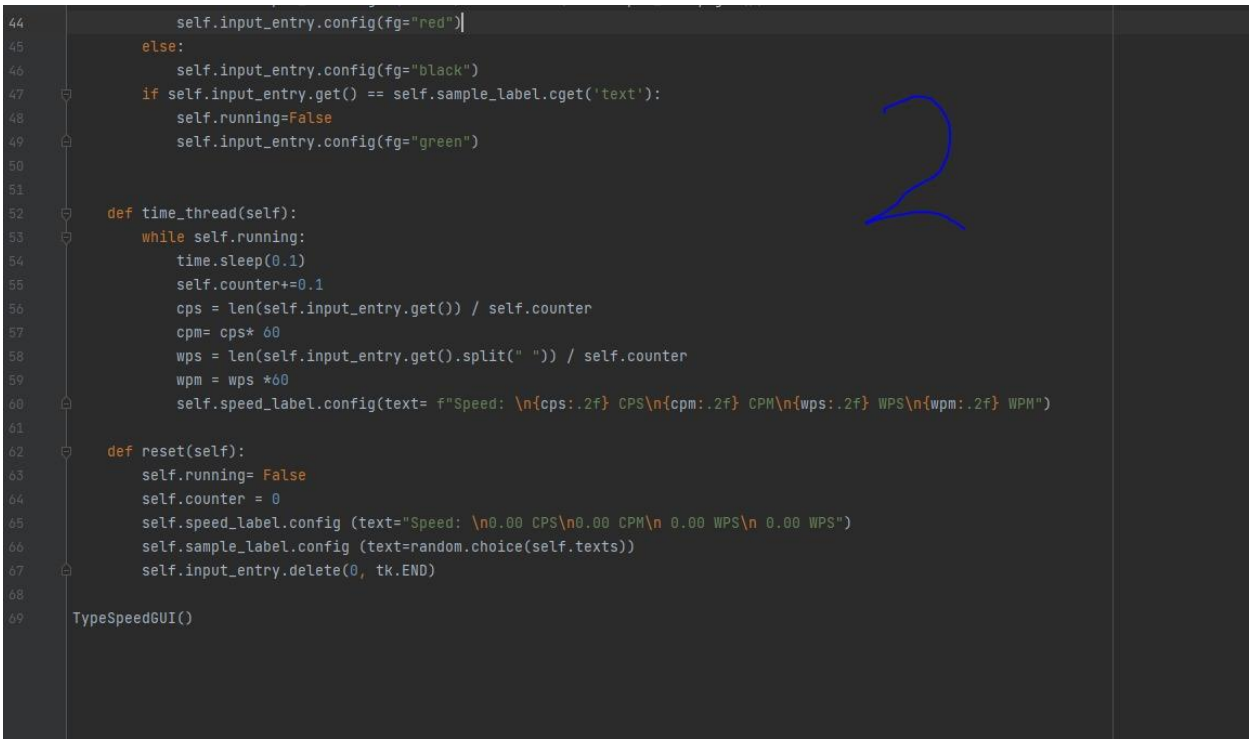
Functionality of the program:

This python code will conduct a typing test and measure the user's typing speed based on characters per minute (cpm), characters per second (cps), words per second (wps), words per minute (wpm) and display the results accordingly.

Here are the pictures of the constructed code:



```python
import tkinter as tk
import time
import threading
import random

class TypeSpeedGUI:

    def __init__(self):
        self.root= tk.Tk()
        self.root.title ("Typing Speed Application")
        self.root.geometry("800x600")

        self.texts = open("text.txt", "r").read().split("\n")

        self.frame=tk.Frame(self.root)

        self.sample_label = tk.Label(self.frame, text=random.choice(self.texts), font=("Helvetica", 18))
        self.sample_label.grid(row=0, column=0, columnspan=2, padx=5, pady=10)

        self.input_entry= tk.Entry(self.frame ,width=40, font=("Helvetica", 24))
        self.input_entry.grid(row=1,column=0,columnspan=2, padx=5, pady=10)
        self.input_entry.bind("<KeyRelease>", self.start)

        self.speed_label = tk.Label(self.frame, text="Speed: \n0.00 CPS\n0.00 CPM\n 0.00 WPS\n 0.00 WPS", font=("Helvetica", 18))
        self.speed_label.grid(row=2, column=0, columnspan=2, padx=5, pady=10)

        self.reset_button = tk.Button(self.frame, text="Reset", command=self.reset, font=("Helvetica", 24))
        self.reset_button.grid(row=3, column=0, columnspan=2, padx=5, pady=10)

        self.frame.pack(expand=True)

        self.counter=0
        self.running=False

        self.root.mainloop()

    def start(self, event):
        if not self.running:
            if not event.keycode in [16,17,18]: #ascii for shift, control, alt
                self.running = True
                t= threading.Thread(target=self.time_thread)
                t.start()
        if not self.sample_label.cget('text').startswith(self.input_entry.get()):
```



```
            self.input_entry.config(fg="red")
        else:
            self.input_entry.config(fg="black")
        if self.input_entry.get() == self.sample_label.cget('text'):
            self.running=False
            self.input_entry.config(fg="green")


    def time_thread(self):
        while self.running:
            time.sleep(0.1)
            self.counter+=0.1
            cps = len(self.input_entry.get()) / self.counter
            cpm= cps* 60
            wps = len(self.input_entry.get().split(" ")) / self.counter
            wpm = wps *60
            self.speed_label.config(text= f"Speed: \n{cps:.2f} CPS\n{cpm:.2f} CPM\n{wps:.2f} WPS\n{wpm:.2f} WPM")

    def reset(self):
        self.running= False
        self.counter = 0
        self.speed_label.config (text="Speed: \n0.00 CPS\n0.00 CPM\n 0.00 WPS\n 0.00 WPS")
        self.sample_label.config (text=random.choice(self.texts))
        self.input_entry.delete(0, tk.END)

TypeSpeedGUI()
```

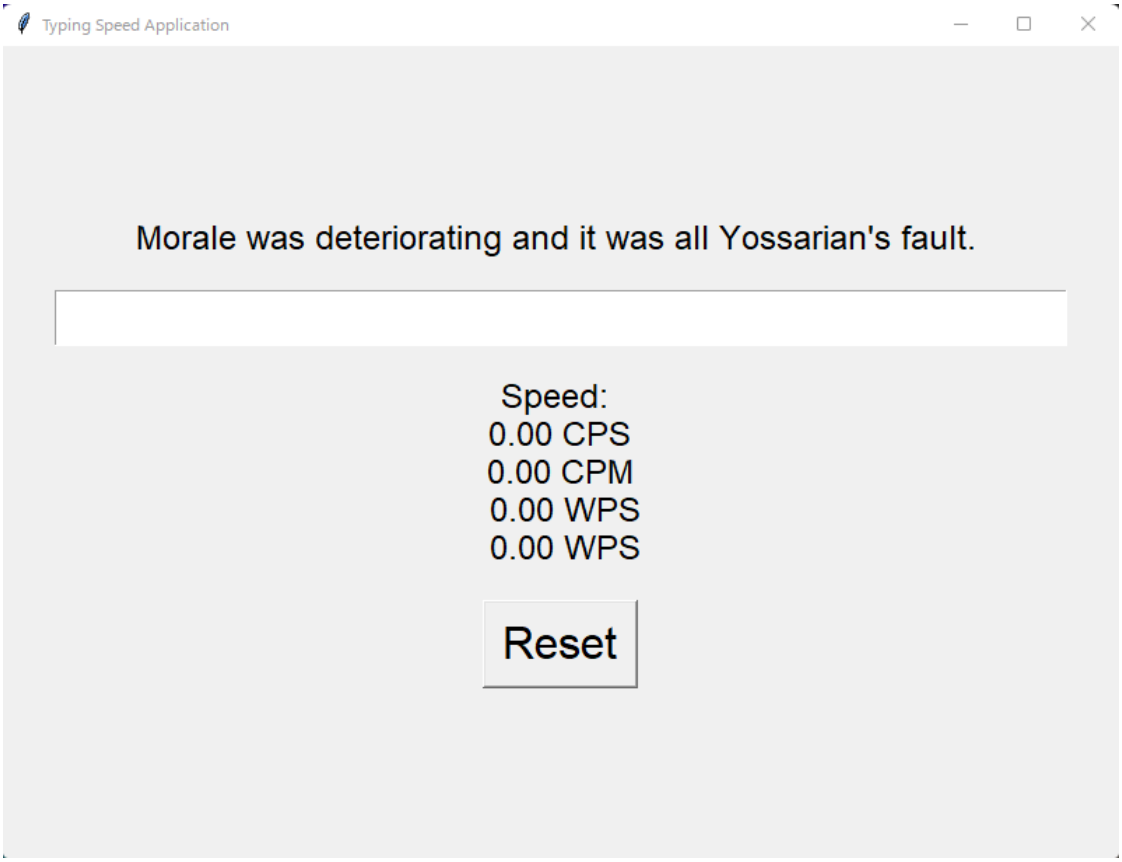
We have used several libraries to make this typing test.

Firstly, we have used the **tkinter** library for designing the GUI of the program, in order to give the user an interactive experience.

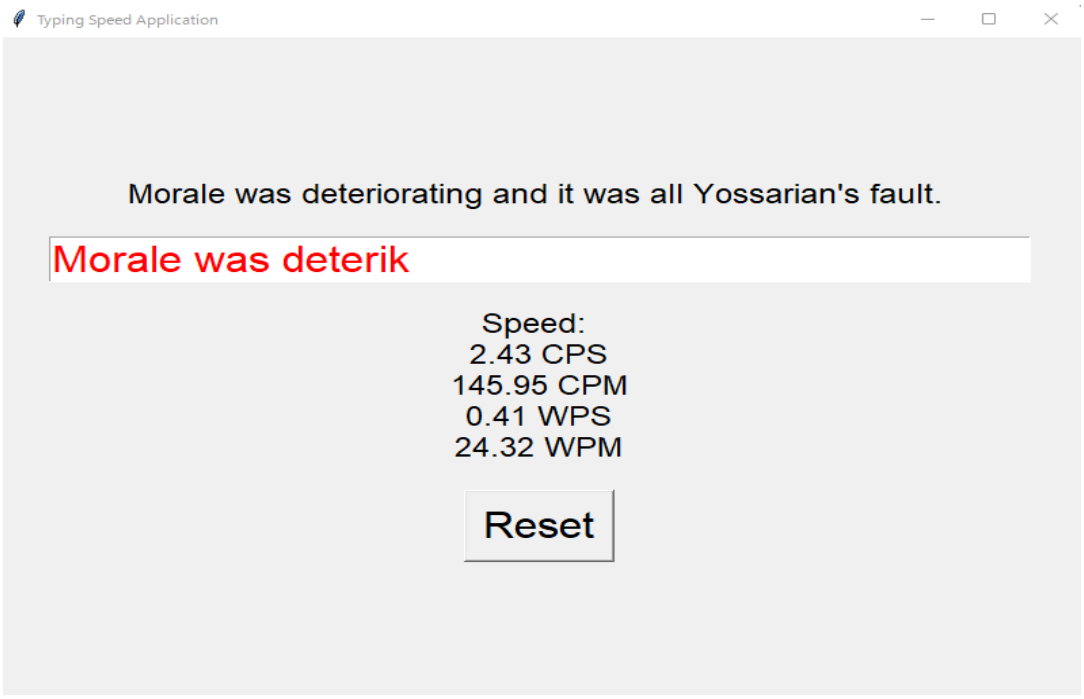
Secondly, we have used time and threading libraries for measuring the typing speed and other tasks.

Lastly, we have used random for choosing random pieces of texts for the typing test.

Here is a graphical representation of the result of the constructed code:



Here, the text turns red when a mistake is made:



And the text turns green when the test is completed: