Agnel Charities'
Fr. C.Rodrigues Institute of Technology, Vashi
Department of Computer Engineering
Mini-Project Report

Subject	**CSL405 - Open Source Technology Lab**	
Branch & Semester	COMP – IV (FH-2020)	
Group Members	**Roll Number**	**Name of the Student**
	101810	**CHARIVUKALAYIL JITIN JOHN**
	101815	**DSOUZA ARNOLD ASHOK**
	101816	**FERNANDES ARYAN AGNELO**
	101817	**FERNANDO EDWIN HIPSON**
	101819	**GEORGE SIBIN BABU**
	101820	**GOEL ESHAN**
Course Outcome		
Title of the Project	**To create a typing application**	
Subject Incharge Signature with Date	**Prof. Padmashree .N**	

Programming Language

Python (+ Additional languages if any)

Abstract

To create a GUI based typing application using python which would generate random one line sentences .The user must type the sentence in the textbox provided and after the user has finished typing the application would produce the user's typing results which includes: typing speed, accuracy and time taken to complete the sentence.

Description / Libraries Used

Description:

The application upon every new start or on clicking the reset button would generate random one line sentences. When the start button is clicked the cursor automatically focuses into the entry-box provided and now the user can start typing. If the user has finished typing then he/she can either press the enter button or click on the stop button and the typing result will be displayed.

The application also contains a login window in which you can enter your name and it will be displayed in the main typing window or else the word 'Guest' will be displayed.

Technique for generation of random sentences:

Multiple one line sentence text files have been created and stored inside a directory. All these files are given a number as their file name. Using randInt() function a random integer number is generated in the defined range and that particular file is opened and its contents are displayed for typing.

Technique for calculating speed and time taken:

Using the timer() (imported timeit as timer) function we calculate at what instance the start and stop buttons were clicked and find the difference between the two to obtain the total time taken between start and stop button click instances. This is the actual time taken by the user to complete typing.

Now whatever string the user has typed in the entry-box is taken and its length is calculated and is divided by 5*60 (assuming in general that a word contains 5 letters and 60 is for seconds to minute conversion) to obtain his typing speed in words per minute (WPM).

Technique for calculating accuracy:

For calculating accuracy we compare both the string i.e. the string to be typed and the string typed by the user. Here the ratio method of fuzz class is used which is present in fuzzywuzzy library, this returns the similarity ratio between the two strings which is found to be most accurate.

*Note: Every parameter calculated is converted into integer using ceil() function of math class

Precautionary measures:

1. The entry-box remains in the disabled state until the user clicks on the start button (This happens upon every new start or on clicking the reset button) and goes into the disabled state again when the user clicks on the stop button (This is done just to avoid typing before starting and after stopping).
2. Start button is disabled upon clicking the start button and the same applies for the stop button (If the start button remains enabled the user might click on it again to obtain a new instance of start time and the typing result will be inappropriate and the same applies for the stop button).
3. If the stop button is clicked before starting the user will get a warning message.

LIBRARIES USED:

1. tkinter (used for GUI)
2. PIL i.e. pillow (used for image setting)
3. random (Used for generation of random number)
4. math (used to obtain ceil values)
5. timeit (used for calculating time instances, this library is imported as timer in our code)
6. fuzzywuzzy (used for string comparisons)

Appendix (Code of the Project)

Code for typing master window:

```
import tkinter
import typeLogin
from typeLogin import MainLogin
import random
import math
from timeit import default_timer as timer
from tkinter import*
from tkinter import messagebox
import fuzzywuzzy
from fuzzywuzzy import fuzz

def randomGenerator():
   var=(random.randint(1,30))
   print(var)
 file="Files\\" + str(var) + ".txt"
   f=open(file,'r')
 content=str(f.readlines())
   content=content.strip('[\']')
   print(content)
   return content

if(MainLogin.name==""):
   name="Welcome Guest"
else:
   name="Welcome " +str(MainLogin.name)

class Time:
   startTime=float(0)
   endTime=float(0)
   similarity=float(0)
   speed=float(0)
```

```
    def startNow(self):
        display_entry.configure(state='normal')
        display_entry.focus()
        Time.startTime=timer()
        start.configure(state='disable')

    def stopNow(self,*args):
        display_entry.configure(state='disable')
        if Time.startTime!=0:
            start.configure(state='disable')

Time.endTime=math.ceil(timer()-Time.startTime)
            print(("The total time is "
+str(Time.endTime)))
            print(display_label['text'])

Time.similarity=math.ceil(fuzz.ratio(display_label.cg
et("text"),result.get()))

Time.speed=math.ceil(len(result.get())/(5*Time.end
Time/60))
            print("The accuracy is: "
+str(Time.similarity)+ "%")
            endResult['text']="Accuracy: "
+str(Time.similarity)+ "%\nSpeed: "
+str(Time.speed)+ " wpm\nTime taken: "
+str(Time.endTime)+ " seconds"
            stop.configure(state='disable')
        else:
            messagebox.showwarning('Warning','Please
start the application before stopping')

t=Time()

def reset():
    display_label['text']=randomGenerator()
    display_entry.configure(state='normal')
    display_entry.delete(0,END)
    display_entry.configure(state='disable')
    start.configure(state='normal')
    stop.configure(state='normal')
    Time.startTime=0
    Time.endTime=0
    endResult['text']=""

main=tkinter.Tk()
main.title('Typing Master')
main.geometry('880x620')
main.resizable(0,0)
main.configure(bg='purple')

#welcome text
name_label=Label(main,text=name,font=("Times
New Roman bold",20),bg='purple')
name_label.place(x=50,y=40)

read=Label(main,text="Type the highlighted
text",font=("Times New Roman",15),bg='purple')
read.place(x=50,y=90)

#displays the text to be typed
display=StringVar()
display_label=Label(main,text=randomGenerator(),f
ont=("Times New
Roman",15),borderwidth=5,relief='ridge',bg='yellow
',highlightcolor='black')
display_label.place(x=50,y=120)
read=Label(main,text="Type here:",font=("Times
New Roman",15),bg='purple')
read.place(x=50,y=180)

#entry box to accept the text
result=StringVar()
display_entry=Entry(main,font=("Times New
Roman",15),textvariable=result,state='disabled')
display_entry.bind('<Return>',t.stopNow)
display_entry.place(x=50,y=210,width=740)

start=Button(main,bg='green',text='Start',font=("Ti
mes New
Roman",15),activebackground='blue',bd='3'
,command=t.startNow)
start.place(x=70,y=270)

stop=Button(main,bg='red',text='Stop',font=("Times
New Roman",15),activebackground='blue',bd='3',
command=t.stopNow)
stop.place(x=140,y=270)

reset=Button(main,bg='orange',text='Reset',font=("
Times New Roman",15),activebackground='blue',
bd='3', command=reset)
reset.place(x=210,y=270)

exit=Button(main,bg='black',fg='white',text='Exit',fo
nt=("Times New Roman",15),
activebackground='blue',
bd='3',command=main.destroy)
exit.place(x=290,y=270)

text=Label(main,text="Result: ",font=("Times New
Roman bold",20),bg='purple')
text.place(x=50,y=330,height=80)
```

```
endResult=Label(main,text="",font=("Times New
Roman",17,),bg='purple')
```

```
endResult.place(x=50,y=380,height=80)

main.mainloop()
```

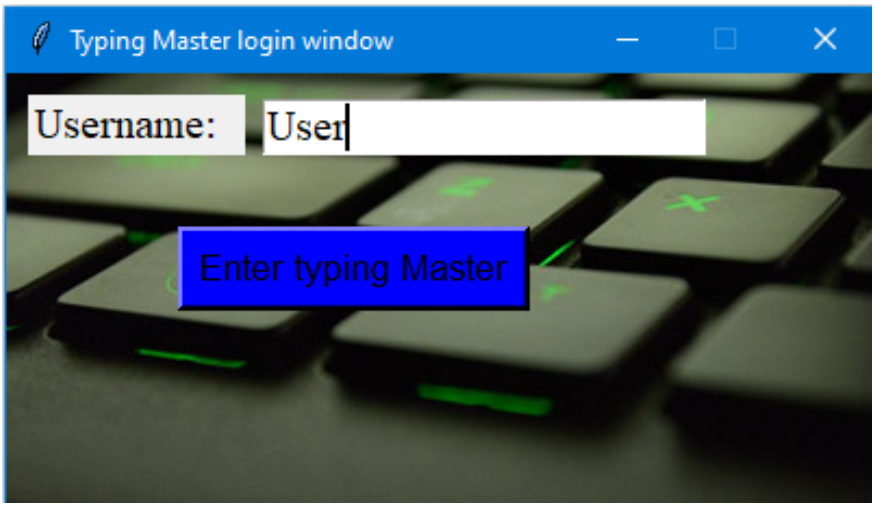
Code for login window:

```
import tkinter
from tkinter import*
from tkinter import messagebox
from PIL import ImageTk,Image

class MainLogin:
   name=""
   def Validate(self):
      MainLogin.name=user.get()
      if len(MainLogin.name)==0 or
MainLogin.name[0]==' ':
         messagebox.showerror('Verifying','Please
enter a valid username to continue')
      else:
         messagebox.showinfo('Verifying','Login
success!')
         login.destroy()
ml=MainLogin()

login=tkinter.Tk()
login.geometry('400x200')
```

```
login.title('Typing Master login window')
login.resizable(0,0)
image2=Image.open("C:\\Users\\SelvanEdwin2\\Do
wnloads\\type1.jpg")
image1=ImageTk.PhotoImage(image2)
photo=Label(login,image=image1)
photo.pack()

user=StringVar()
user_label=Label(login,text='Username:
',font=("Times New Roman",15),)
user_label.place(x=10,y=10)
user_entry=Entry(login,font=("Times New
Roman",15),textvariable=user)
user_entry.place(x=220,y=25,anchor='center')

Enter=Button(login,text='Enter typing
Master',font=("Arial",13),activebackground='green',
bg='blue',bd='2px',padx=3,pady=3,command=ml.Val
idate)
Enter.place(x=160,y=90,anchor='center')

login.mainloop()
```

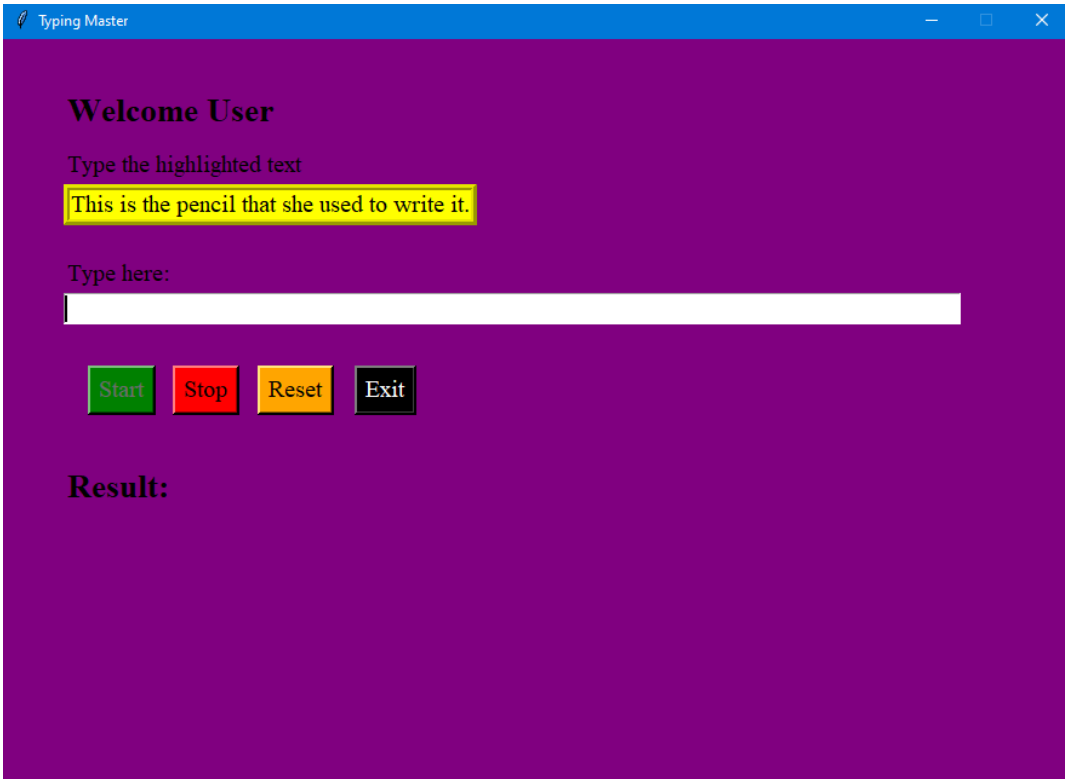
Implementation Screenshots

1. login window



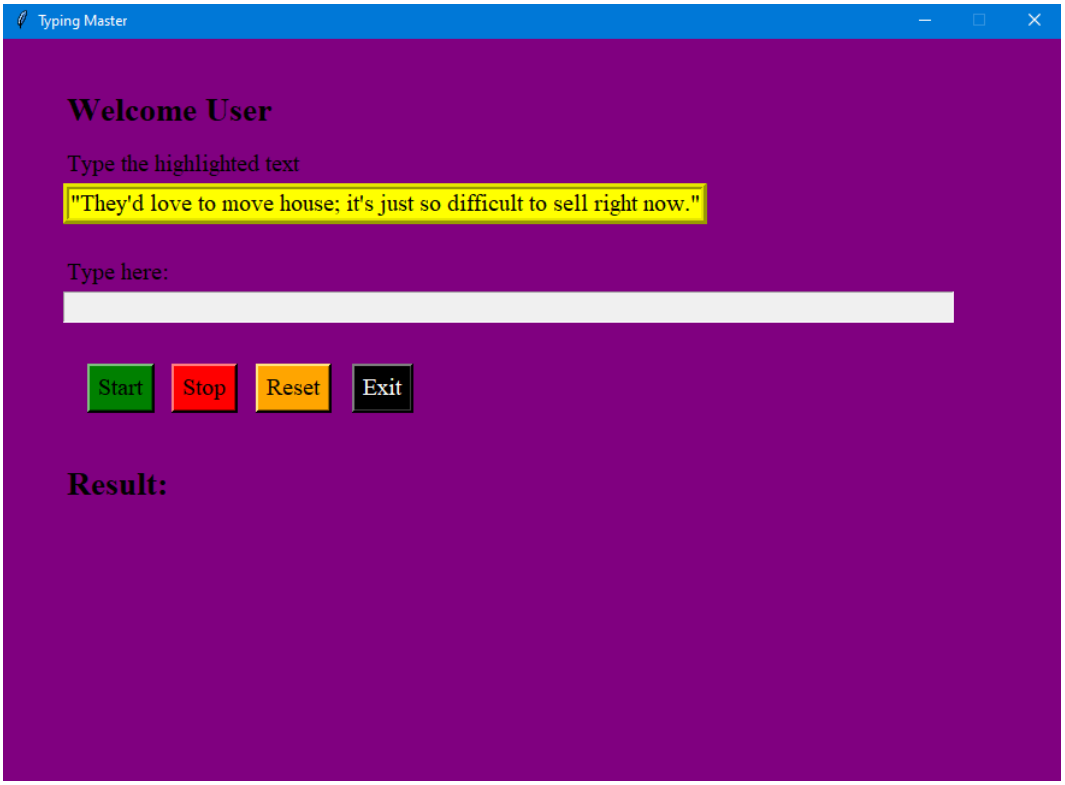
2. Typing screen window with entry-box disabled at the beginning

3. Entry-box enabled on clicking start button with cursor focussed into it and start button is disabled



4. Entry-box is disabled upon clicking stop or pressing enter button and also stop button is disabled and typing results are displayed

5. All back to normal with new random sentence to type upon clicking reset button



6. User receives a warning when trying to stop before starting