

MSPA-4

Course Name: Software Lab.

Batch: G18

Members and Roll no.:

Shruti Shrivastav-119

Saloni Urade-114

Parag Bandhate-174

Atharva Selokar-135

Application Name: Application for typing speed analysis.

Problem Statement:

Application for typing speed analysis is a simple application that is used to measure our typing speed and also improve it. The project involves the creation of an application that measures the typing speed and mistakes a user has done while typing a paragraph.

Introduction:

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. In this mini project we will see how we can create application for typing speed analysis using Tkinter.

Package Used:

Tkinter Python interface to Tcl/Tk:

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI. Tkinter is included with standard GNU/Linux, Microsoft Windows and macOS installs of Python.

The name Tkinter comes from Tk interface. Tkinter was written by Fredrik Lundh. Tkinter is free software released under a Python license. As with most other modern Tk bindings, Tkinter is implemented as a Python wrapper around a complete Tcl interpreter embedded in the Python interpreter. Tkinter calls are translated into Tcl commands, which are fed to this embedded interpreter, thus making it possible to mix Python and Tcl in a single application.

There are several popular GUI library alternatives available, such as wxPython, PyQt, PySide, Pygame, Pyglet, and PyGTK.

Timeit — Measure execution time of small code snippets:

This module provides a simple way to time small bits of Python code. It has both a Command-Line Interface as well as a callable one. It avoids a number of common traps for measuring execution times. See also Tim Peters' introduction to the "Algorithms" chapter in the second edition of *Python Cookbook*, published by O'Reilly.

Random – Generate pseudo-random numbers

This module implements pseudo-random number generators for various distributions.

For integers, there is uniform selection from a range. For sequences, there is uniform selection of a random element, a function to generate a random permutation of a list in-place, and a function for random sampling without replacement.

Code:

```
from tkinter import *
from tkinter import messagebox
from timeit import default_timer as timer
import random

root = Tk()#To create a window, tkinter offers a method Tk().
root.geometry("500x500")#To adjust size of window.
root.configure(bg="Black")#To setup the background color.

window = Tk()
window.geometry("550x500")
window.withdraw()

x = 0
def finish():
    window.destroy()#to destroy window
    root.destroy()#to destroy root window

def game():# This contains the actual code for speed test analysis
    global x
    if x==0:
        root.withdraw()
        x=x+1
    window.deiconify()
    def check_result():# This checks the final result
        j=0
        error=0
        answer=entry.get("1.0", 'end-1c')#Text we have entered.
        end = timer()
        time_taken=end-start#Time taken to write the passage.
```

```

word2=list(words[word].split(" "))
ans=list(answer.split(" "))
if len(word2)>len(ans):# this will count number of error made while
counting.
    # error=len(word2)-len(ans)
    for i in ans:
        if i == word2[j]:
            pass
        else:
            error+=1
            j+=1
    elif len(word2)<len(ans):
        # error=len(ans)-len(word2)
        for i in word2:
            if i == ans[j]:
                pass
            else:
                error+=1
                j+=1
    else:
        for i in word2:
            if i == ans[j]:
                pass
            else:
                error+=1
                j+=1
if error>len(ans):
    result="Try again."
else:
    wpm=len(answer)/5
    wpm=int(wpm/(time_taken/60))#Average speed of typing in wpm(Word per
minute)

    acc=len(ans)-error
    acc=(acc*100)/len(word2)
    acc="{:.2f}".format(acc)
    result="Your score is: "+str(wpm)+" WPM and accuracy is: "+str(acc)+"
%."

    messagebox.showinfo("Score",result)#The result will be shown in message
box.

#words=["The world is beautiful.", "The kite is drastic."]
words=["During the first part of your life, you only become aware of
happiness once you have lost it. Then an age comes, a second one, in which you
already know, at the moment when you begin to experience true happiness, that you

```

are, at the end of the day, going to lose it.", "Opera refers to a dramatic art form, originating in Europe, in which the emotional content is conveyed to the audience as much through music, both vocal and instrumental, as it is through the lyrics.", "Dolphins are regarded as the friendliest creatures in the sea and stories of them helping drowning sailors have been common since Roman times. The more we learn about dolphins, the more we realize that their society is more complex than people previously imagined.", "Naval architects never claim that a ship is unsinkable, but the sinking of the passenger-and-car ferry Estonia in the Baltic surely should have never have happened. It was well designed and carefully maintained.", "Erosion of America's farmland by wind and water has been a problem since settlers first put the prairies and grasslands under the plow in the nineteenth century."]

```
word=random.randint(0, len(words)-1)

x2= Label(window, text=words[word], bg='black', fg="white", height=7,
width=47, font="times 15", wraplength=500)#Contains the text we have to type.
x2.place(x=15, y=10)

x3=Button(window, text="Submit", font="times 20", bg="white",
command=check_result)#Button to check result.
x3.place(x=220,y=350)

entry = Text(window)#The text entry box
entry.place(x=25, y=180, height=150, width=500)

b2 = Button(window, text="Done", font="times 13", bg="#ffc003", width=12,
command=finish)#Button to destroy the window.
b2.place(x=155, y=420)

b3 = Button(window, text="Another One", font="times 13", bg="#ffc003",
width=12, command=game)#Button to start game command.
b3.place(x=265, y=420)

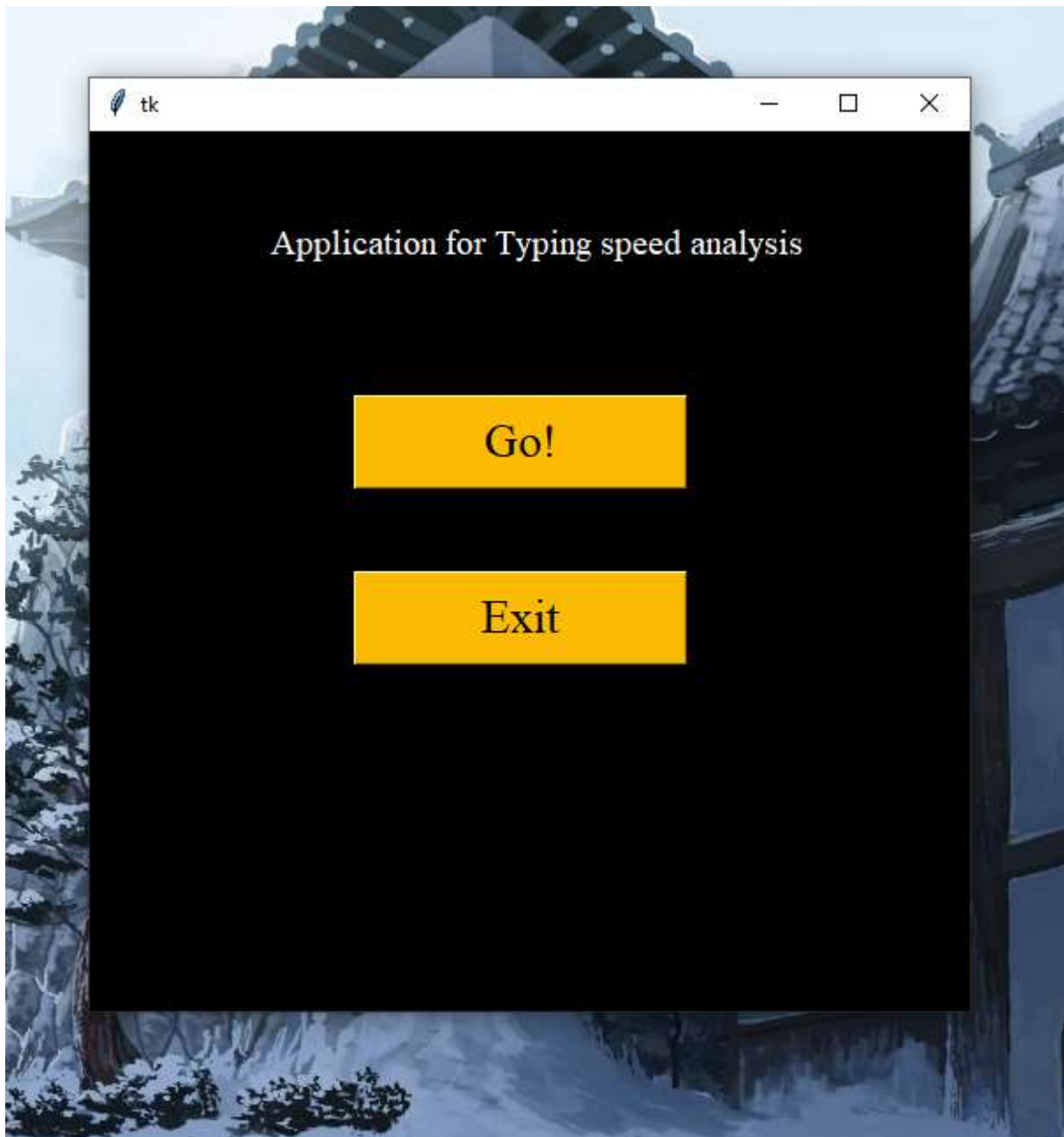
start = timer()

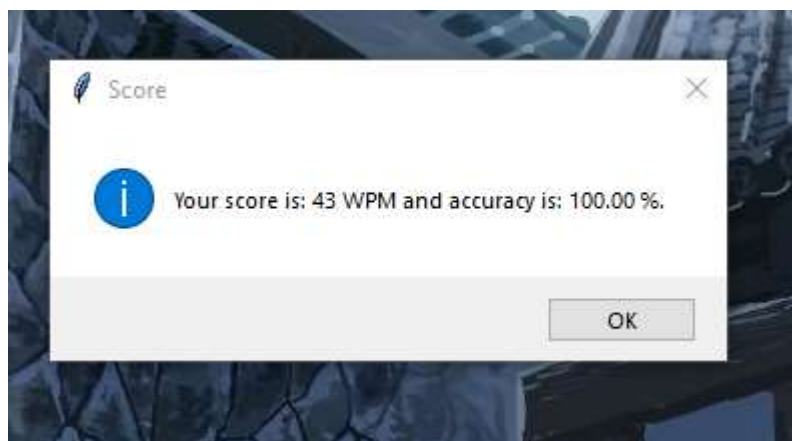
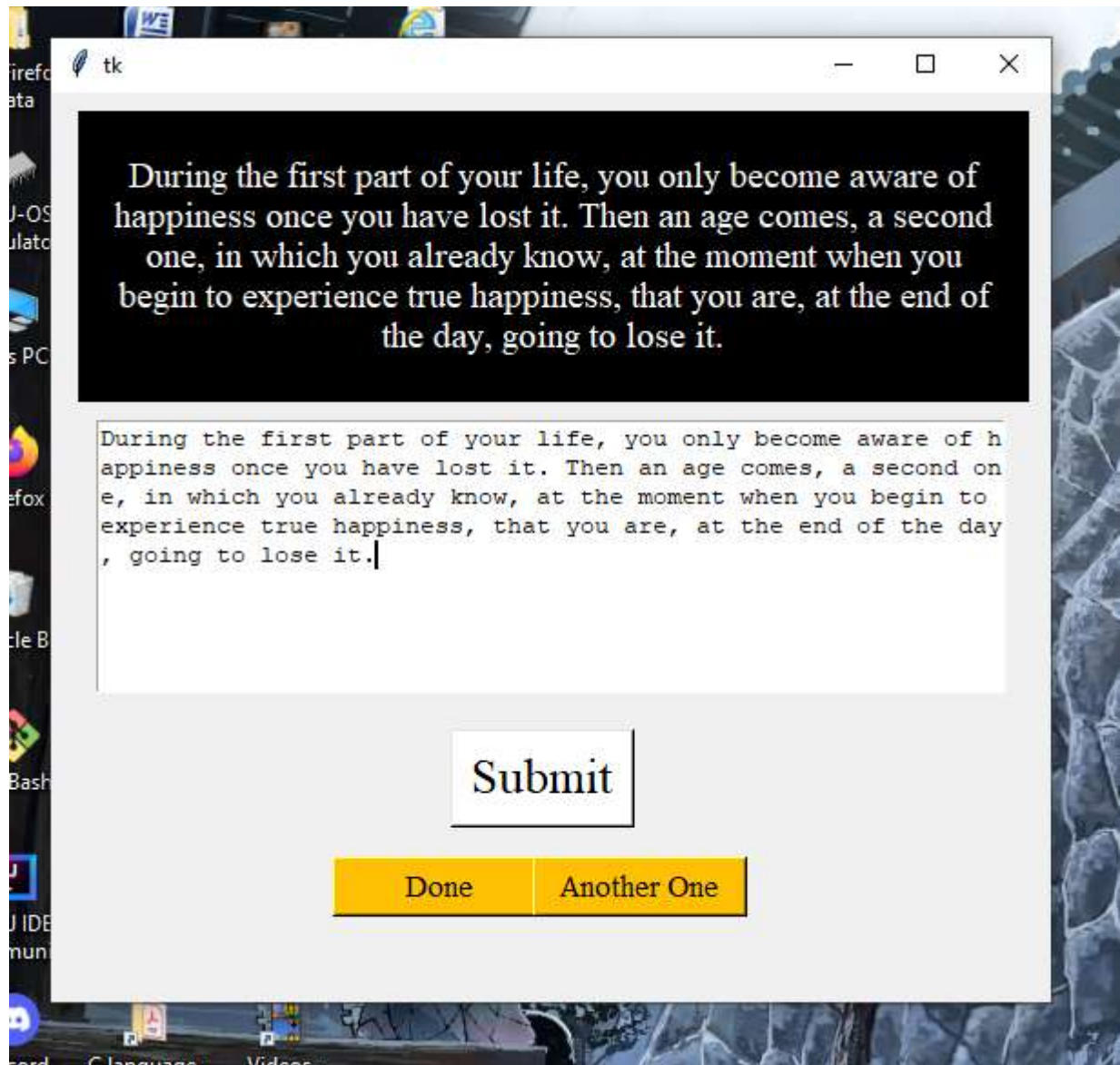
window.mainloop()
x1 = Label(root, text="Application for Typing speed analysis", bg="black",
fg="white", font="times 15")
x1.place(x=100,y=50)

b1 = Button(root, text="Go!", width=12, bg="#fcba03", font="times 20",
command=game)
b1.place(x=150, y=150)
```

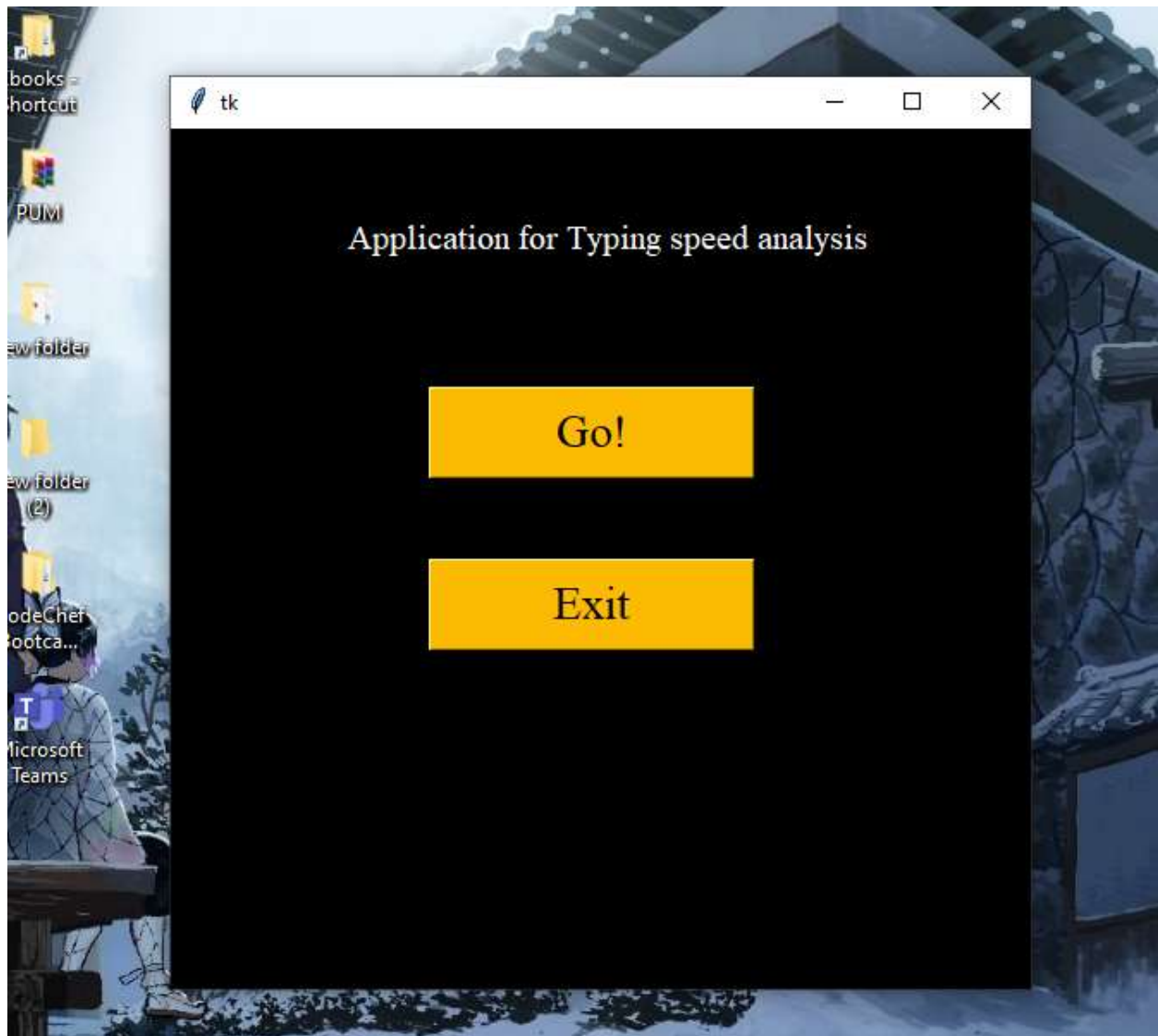
```
b1 = Button(root, text="Exit", width=12, bg="#fcb03", font="times 20",  
command=finish)  
b1.place(x=150, y=250)  
  
root.mainloop()
```

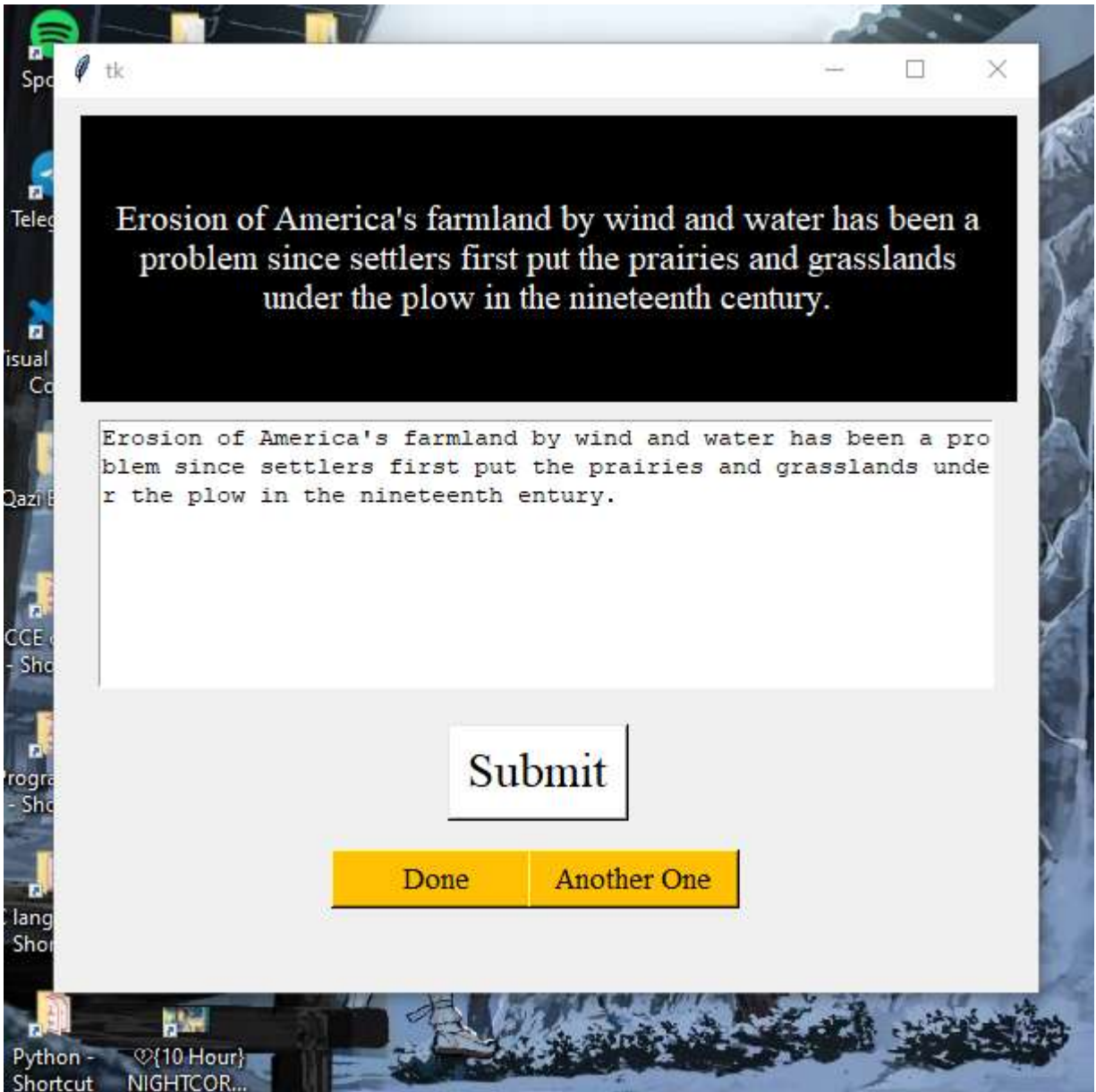
Output 1:

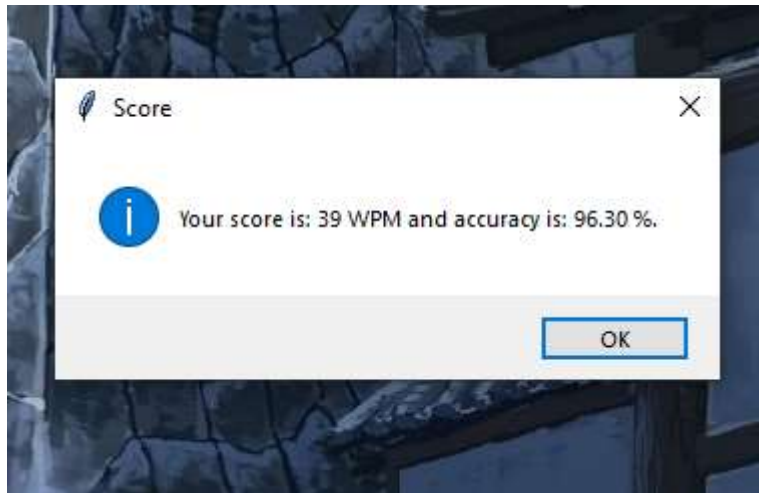




Output 2:







Reference:

1. <https://docs.python.org/3/library/random.html>
2. <https://docs.python.org/3/library/timeit.html>
3. <https://docs.python.org/3/library/tkinter.html>