

# **TYPING SPEED TEST**

## **Using Python**

A Mini Project Report Submitted  
in partial fulfillment of the requirements for the award of the degree of

### **Bachelor of Technology**

**in**

### **Computer Science and Engineering**

**by**

**R. Sai Kiran**

**18N31A05J4**

**S. Mohith**

**19N35A0523**

**Sai Abhinav**

**18N31A05J7**

Under the esteemed guidance of

**J. Mahendar**

**Asst. Professor**



**Department of Computer Science and Engineering**

**Malla Reddy College of Engineering & Technology**

(Autonomous Institution- UGC, Govt. of India)

(Affiliated to JNTUH, Hyderabad, Approved by AICTE, NBA & NAAC with 'A' Grade)

Maisammaguda, Kompally, Dhulapally, Secunderabad – 500100

website: [www.mrcet.ac.in](http://www.mrcet.ac.in)

**2020-2021**



# **Malla Reddy College of Engineering & Technology**

(Autonomous Institution- UGC, Govt. of India)

(Affiliated to JNTUH, Hyderabad, Approved by AICTE, NBA & NAAC with 'A' Grade)

Maisammaguda, Kompally, Dhulapally, Secunderabad – 500100

website: [www.mrcet.ac.in](http://www.mrcet.ac.in)

## **CERTIFICATE**

This is to certify that this is the bonafide record of the project entitled “Typing Speed Game using python”, submitted by R. Sai Kiran(18N31A05J4), S. Mohith (19N35A0523) and Sai Abhinav (18N31A05J7) of B. Tech in the partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering, Department of CSE during the year 2020-2021. The results embodied in this project have not been submitted to any other university or institute for the award of any degree or diploma.

**Internal Guide**

**J. Mahendar**  
**Asst. Professor**

**Head of the Department**

**Dr. S. Shanthi**  
**Professor**

**External Examiner**

## **DECLARATION**

We hereby declare that the project titled “Typing speed Test using python” submitted to Malla Reddy College of Engineering and Technology (UGC Autonomous), affiliated to Jawaharlal Nehru Technological University Hyderabad (JNTUH) for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a result of original research carried-out in this thesis. It is further declared that the project report or any part thereof has not been previously submitted to any University or Institute for the award of degree or diploma.

**R. Sai Kiran - 18N31A05J4**

**S. Mohith - 19N35A0523**

**Sai Abhinav - 18N31A05J7**

## **ACKNOWLEDGEMENT**

We feel ourselves honored and privileged to place our warm salutation to our college Malla Reddy College of Engineering and Technology (UGC-Autonomous) and our principal Dr. VSK Reddy who gave us the opportunity to have experience in engineering and profound technical knowledge.

We express our heartiest thanks to our Head of the Department Dr. S. Shanthi for encouraging us in every aspect of our project and helping us realize our full potential.

We would like to thank our internal guide and Project Coordinator J Mahendar Sir for his regular guidance and constant encouragement. We are extremely grateful to his valuable suggestions and unflinching co-operation throughout project work.

We would like to thank our class in charge Mr. Abdul Saleem sir who in spite of being busy with his duties took time to guide and keep us on the correct path.

We would also like to thank all the supporting staff of the Department of CSE and all other departments who have been helpful directly or indirectly in making our project a success.

We are extremely grateful to our parents for their blessings and prayers for the completion of our project that gave us strength to do our project.

with regards and gratitude

**R. Sai Kiran - 18N31A05J4**

**S. Mohith - 19N35A0523**

**Sai Abhinav - 18N31A05J7**

## **ABSTRACT**

In this project, we present the detailed development and implementation of simple Speed Typing Test application. Speed Typing Test is a Desktop application which can be used by any naïve user. This game consists of Graphical user interface. We implemented this Speed typing game by using python's tkinter module. Tkinter is the standard GUI library for python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. This project helps you to improve your Typing speed i.e the player/user can improve their typing speed playing this game. This game usually takes input from user and compares that with predefined words. Once the game starts, one-by-one random words will pop-up on your screen and player has to type each word in the given span of seconds i.e 30 seconds to score. If not the game ends and in-order-to play again, you just have to click on play again. This project gives an insight into the different aspects of python programming and practical knowledge of Tkinter modules and classes. This application is easy to use for both beginners and advanced users.

## TABLE OF CONTENTS

<b>S.NO</b>	<b>TITTLE</b>	<b>PG.NO</b>
<b>1</b>	<b>INTRODUCTION</b>	<b>01</b>
	1.1 PURPOSE, AIM AND OBJECTIVES	01
	1.2 BACKGROUND OF PROJECT	01
	1.3 SCOPE OF PROJECT	02
	1.4 MODULES DESCRIPTION	02
<b>2</b>	<b>SYSTEM ANALYSIS</b>	<b>03</b>
	2.1 HARDWARE AND SOFTWARE REQUIREMENTS	03
	2.2 SOFTWARE REQUIREMENTS SPECIFICATION	04
<b>3</b>	<b>SYSTEM DESIGN &amp; ARCHITECTURE</b>	<b>06</b>
	3.1 SOFTWARE DESIGN	06
	3.2 ARCHITECTURE	06
<b>4</b>	<b>INPUT/OUTPUT DESIGN</b>	<b>08</b>
	4.1 INPUT DESIGN	08
	4.2 OUTPUT DESIGN	08
<b>5</b>	<b>IMPLEMENTATION</b>	<b>09</b>
<b>6</b>	<b>TESTING</b>	<b>19</b>
	6.1 TESTING OBJECTIVES	19
	6.2 TESTING METHODOLOGIES	19
	6.3 USER TRAINING	22
	6.4 MAINTAINENCE	22
	6.5 TESTING STRATEGY	23
<b>7</b>	<b>OUTPUT SCREENS</b>	<b>24</b>
<b>8</b>	<b>CONCLUSION &amp; FUTURE SCOPE</b>	<b>29</b>
<b>9</b>	<b>BIBLIOGRAPHY</b>	<b>30</b>

# **1. INTRODUCTION**

This chapter gives a scope description and overview of everything included in this project report. Also, the purpose, aim, objectives, background and operation environment of the system are listed.

## **1.1 PURPOSE, AIM AND OBJECTIVES:**

The purpose of this document is to give a detailed description of Typing Speed Test Project. It will illustrate the purpose and complete declaration for the development of this application. This document is primarily intended to anyone who wants to get an overview of how this project works, its outcomes, and possible usages in future.

Typing speed test using Python is a Desktop based application. It allows any users to easily interact with the system. The main aim of the project is to develop an efficient application that can improve the user's typing speed. This application can be used by people with elementary level in typing such as teachers, students, staff, and parents.

→ This application aims to achieve these objectives

- ✓ This application is effective, efficient and initiating for helping users.
- ✓ Help the learners to enhance their learning process by developing their typing skills.
- ✓ Develop typing speed.
- ✓ Recognize position of the keys to use all fingers.
- ✓ Identify the location of different keys and different letters.
- ✓ To practice typing with all fingers.

## **1.2 BACKGROUND OF PROJECT:**

The current research aims to provide an effective and efficient typing experience to end user. In the current system the occurrences of different random words is done by importing random module. Here only one model is implemented with one application. Hence our approach is to do this application using Tkinter module from python.

### **1.3 SCOPE OF PROJECT:**

Our project mainly focuses on enhancing or improving end user's typing speed.

This project gives an insight in to the different aspects of python programming and practical knowledge of Tkinter modules and classes. Our application is capable of including future advancements.

### **1.4 MODULES DESCRIPTION:**

**This project is composed of following modules.**

1. **Tkinter**

Tkinter is the standard GUI library for python. Python when combined with Tkinter provides a fast and easy way to create GUI applications.

2. **Pillow**

The Python Imaging Library adds image processing capabilities to your Python interpreter.

3. **Random**

The random module is a built-in module to generate the pseudo-random variables. It can be used perform some action randomly such as to get a random number, selecting a random element from a list, shuffle elements randomly, etc.

4. **Pygame**

pygame is a free and open-source cross-platform library for the development of multimedia applications like video games using Python. It uses the Simple Direct Media Layer library and several other popular libraries to abstract the most common functions, making writing these programs a more intuitive task.



## 2. SYSTEM ANALYSIS

In this chapter, we will discuss and analyze about the developing process of Typing speed Test including software requirement specification (SRS) and comparison between existing and proposed system. The functional and non-functional requirements are included in SRS part to provide complete description and overview of system requirement before the developing process is carried out. Besides that, existing vs. proposed provides a view of how the proposed system will be more efficient than the existing one.

### 2.1 HARDWARE AND SOFTWARE REQUIREMENTS

#### 2.1.1 HARDWARE REQUIREMENTS:

<b>Processor</b>	:	AMD® processor or Intel® Core™ i3 processor
<b>Ram</b>	:	1 GB or above.
<b>Hard disk</b>	:	40GB or above.

#### 2.1.2 SOFTWARE REQUIREMENTS:

<b>Technology/Language</b>	:	Python.
<b>Operating System</b>	:	Windows, Linux, Mac.
<b>IDE</b>	:	PyCharm / Vscode / Sublime
<b>Python Version</b>	:	3.6.X and later versions

## **2.2 SOFTWARE REQUIREMENT SPECIFICATION:**

### **2.2.1 SRS:**

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase.)

The SRS phase consists of two basic activities:

#### **1) Problem/Requirement Analysis:**

The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.

#### **2) Requirement Specification:**

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity.

The Requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase.

### **2.2.2 ROLE OF SRS:**

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium through which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

### **2.2.3 SCOPE:**

This document is the only one that describes the requirements of the system. It is meant for the use by the developers, and will also be the basis for validating the final delivered system. Any changes made to the requirements in the future will have to go through a formal change

approval process. The developer is responsible for asking for clarifications, where necessary, and will not make any alterations without the permission of the client.

#### **2.2.4 EXISTING SYSTEM:**

In the existing system the application is complex not very user interactive. The user might feel difficult to understand the application as this is designed for experts. There are a lot of annoying advertisements. The existing system one of the biggest downsides is it is available for windows only and unfortunately cannot be accessed through Mac or Linux systems. This existing system does not provide any features to allow user to make any changes. According to user reviews the cursor movement was slow since it is an old application. Existing system uses a lot of memory space and ram. Since it is a web application there are more chances of virus attacks.

##### **2.2.4.1 DRAWBACKS OF EXISTING SYSTEM:**

- Interface not very user friendly
- Complex
- Advertisements
- Designed only for windows
- Uses more CPU memory

#### **2.2.5 PROPOSED SYSTEM:**

The proposed system is having the following features. It has more graphically attractive user interface. In this the game usually takes input from user and compares that with predefined words. Once the game starts, one-by-one random word will pop-up on your screen and player has to type each word in the given amounts of seconds i.e., 60s to score. Other-wise you will lose the game and to play again, you just have to click on play again. This project gives an insight in to the different aspects of python programming and practical knowledge of Tkinter modules and classes. This software is easy to use for both beginners and advanced users. Here only one model is implemented with one application.

There are less chances of virus attacks. As this application is made by ourself there will be no spasms and advertisements. It provides a space for more future enhancements.

### 3. SYSTEM DESIGN & ARCHITECTURE

System design is transition from a user-oriented document to programmers or data base personnel. The design is a solution, how to approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Designing goes through logical and physical stages of development, logical design reviews the present physical system, prepare input and output specification, details of implementation plan and prepare a logical design walkthrough.

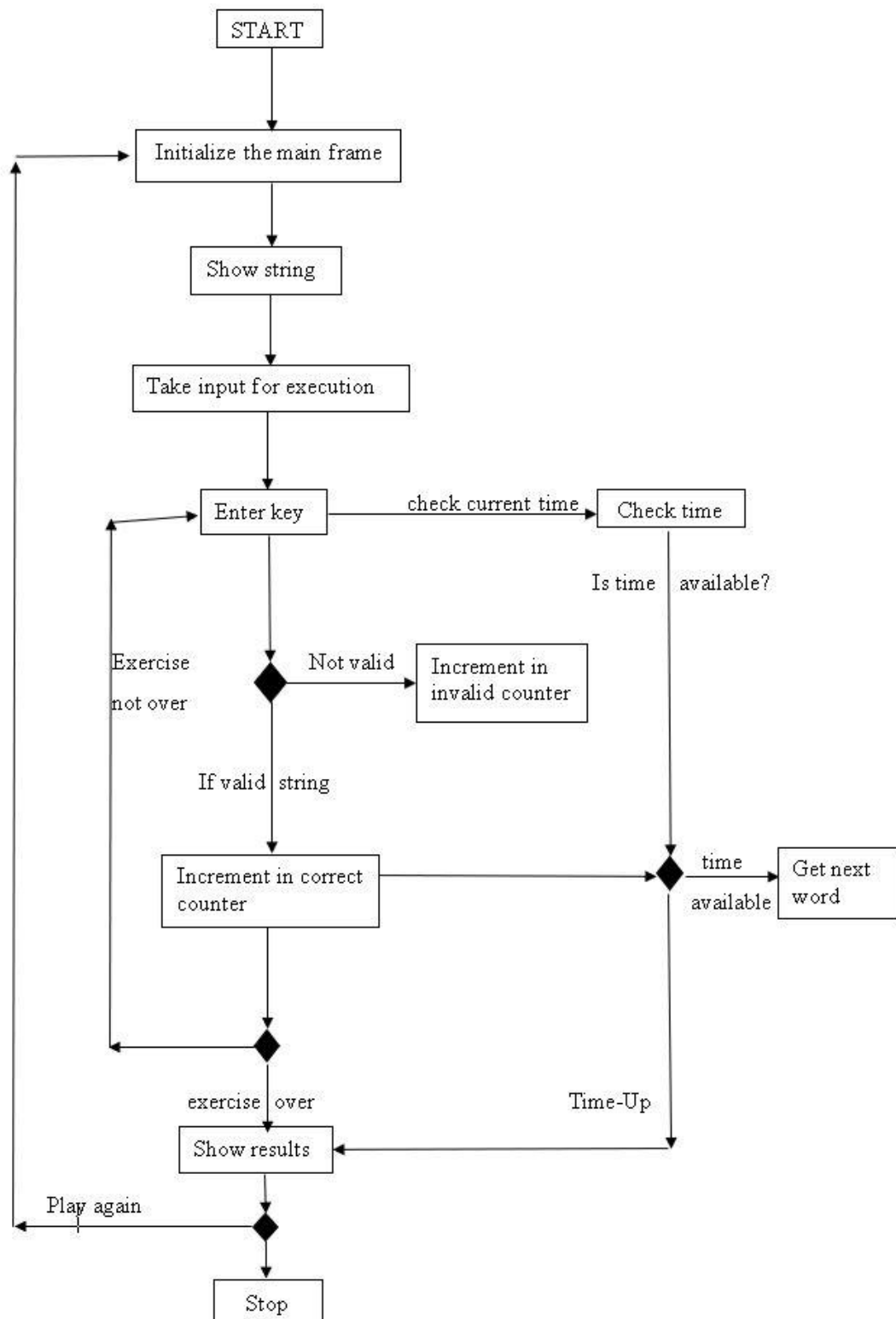
#### 3.1 SOFTWARE DESIGN:

In designing the software following principles are followed:

- 1) **Modularity and partitioning:** Software is designed such that, each system should consists of hierarchy of modules and serve to partition into separate function.
- 2) **Coupling:** Modules should have little dependence on other modules of a system.
- 3) **Cohesion:** Modules should carry out in a single processing function.
- 4) **Shared use:** Avoid duplication by allowing a single module be called by other that need the function it provides.

#### 3.2 ARCHITECTURE:

Architecture diagram is a diagram of a system, in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. The block diagram is typically used for a higher level, less detailed description aimed more at understanding the overall concepts and less at understanding the details of implementation.



**SYSTEM FLOW DIAGRAM**

## **4. INPUT/OUTPUT DESIGN**

### **4.1 INPUT DESIGN:**

Input Design plays a vital role in the life cycle of software development, it requires very careful attention of developers. The input design is to feed data to the application as accurate as possible. So, inputs are supposed to be designed effectively so that the errors occurring while feeding are minimized. According to Software Engineering Concepts, the input forms or screens are designed to provide to have a validation control over the input limit, range and other related validations. Input design is the process of converting the user created input into a computer-based format. The goal of the input design is to make the data entry logical and free from errors.

Validations are required for each data entered. Whenever a user enters an erroneous data, error message is displayed and the user can move on to the subsequent pages after completing all the entries in the current page.

### **4.2 OUTPUT DESIGN:**

The Output from the computer is required to mainly create an efficient method of communication within the company primarily among the project leader and his team members, in other words, the administrator and the clients. The output of VPN is the system which allows the project leader to manage his clients in terms of creating new clients and assigning new projects to them, maintaining a record of the project validity and providing folder level access to each client on the user side depending on the projects allotted to him. After completion of a project, a new project may be assigned to the client. User authentication procedures are maintained at the initial stages itself.

## 5.IMPLEMENTATION

```
#importing all required libraries

#tkinter is a GUI library

import tkinter as tk

import random

from tkinter import messagebox

from tkinter.font import Font

import pygame

from pygame import mixer

from PIL import ImageTk, Image


#collection of words

words = ['Mobile', 'Paraphernalia', 'Indict', 'Laptop', 'Liquefy', 'Hyderabad', 'computer', 'Mango',
'Apple', 'samsung', 'Google', 'Microsoft', 'Lenovo', 'python', 'Java', 'message', 'Onomatopoeia',
'Android', 'gmail', 'email', 'Playwright', 'Albeit', 'Appease', 'Bemused', 'Contrived', 'Fuchsia',
'Minuscule', 'Colloquial', 'Conundrum', 'Dystopia', 'Egregious', 'Fortuitous', 'Incongruous',
'Nauseous', 'Obsolete', 'Vernacular', 'Ingenious', 'Sacrilegious', 'suburban', 'assuming',
'obstinance', 'foramens', 'Nauseous', 'Dilate', 'Orangutan', 'Mischievous', 'Gubernatorial',
'Acquiesce', 'Conscientious']

#function for title sliding effect

def labelSlider():

    global count, sliderWord

    text = "Typing Speed Increaser Game"

    if (count >= len(text)):
```

```

        count = 0

        sliderWord = ""

        sliderWord += text[count]

        count += 1

        fontLabel.configure(text=sliderWord)

        fontLabel.after(150, labelSlider)

#function for advice button

def reminder():

    global load4, image4, render4, img4

    open_window = tk.Toplevel(root)

    open_window.title('Intro')

    open_window.geometry('512x326+400+100')

    open_window.resizable(0, 0)

    load4 = Image.open('keyboard.jpg')

    image4 = load4.resize((512, 326), Image.ANTIALIAS)

    render4 = ImageTk.PhotoImage(image4)

    img4 = tk.Label(open_window, image=render4)

    img4.place(x=0, y=0)

#initialising pygame

pygame.init()

#play music function

def playMusic():

    mixer.music.load('bg.mp3')

```



```

        mixer.music.play(-1)

#stop music function
def stopMusic():
    mixer.music.stop()

#start game function
def startGame(event):
    global score, miss

    if (timeleft == 30):
        time()

        gamePlayLabel.configure(text="")

        if (wordEntry.get() == wordLabel['text']):
            score += 1

            scoreLabelcount.configure(text=score)

        else:
            miss += 1

        random.shuffle(words)

        wordLabel.configure(text=words[0])

        wordEntry.delete(0, tk.END)

#timer
def time():
    global timeleft, score, miss

    if (timeleft >= 11):
        pass

    else:
        timeLabelcount.configure(fg='red', font=('airal', 30, 'italic bold'))

```

```

if (timeleft > 0):

    timeleft -= 1

    timeLabelcount.configure(text=timeleft)

    timeLabelcount.after(1000, time)

else:

    last_window = tk.Toplevel(root)

    last_window.title('Game over!!')

    last_window.geometry('800x600+400+100')

    last_window.resizable(0, 0)


    load3 = Image.open('last.jpg')

    image3 = load3.resize((796, 598), Image.ANTIALIAS)

    render3 = ImageTk.PhotoImage(image3)

    img3 = tk.Label(last_window, image=render3)

    img3.place(x=0, y=0)


    output_label = tk.Label(last_window, text='Hit = { } \n Miss = { } \n Total   Score =
{ }'.format(score, miss, score - miss), font=('Castellar', 25, 'italic bold'), bg='#EEEEEE', fg='red')

    output_label.place(x=250, y=250)


    rr = messagebox.askretrycancel('Notification', 'For Play Again Hit Retry',
parent=last_window)

    if (rr == True):

        score = 0

        timeleft = 30

```

```

miss = 0

timeLabelcount.configure(text=timeleft)

wordLabel.configure(text=words[0])

scoreLabelcount.configure(text=score)

gamePlayLabel.configure(text='Hit Enter \n to \n start the game')

last_window.destroy()

else:

    last_window.destroy()

    root.destroy()

##### Root Method #####

root = tk.Tk()

root.geometry('800x600+400+100')

root.resizable(False, False)

load = Image.open('bg.jpg')

image = load.resize((850, 600), Image.ANTIALIAS)

render = ImageTk.PhotoImage(image)

img = tk.Label(root, image=render)

img.place(x=0, y=0)

root.wm_title('Typing Speed Increase Game')

root.iconbitmap('typingicon.ico')

```

```

score = 0

timeleft = 30

count = 0

sliderWord = "

miss = -1


#defining font

bigFont = Font(family="Times", size=26, weight="bold", slant="roman", underline=1)

##### Label Method for The UI #####

fontLabel = tk.Label(root, text="", font=bigFont, bg='#F6F6F6', fg="#D3212D", width=37)

fontLabel.place(x=15, y=25)

labelSlider()


random.shuffle(words)

wordLabel = tk.Label(root, text=words[0], font=('Helvetica', 35, 'bold'))

wordLabel.place(x=310, y=250)


scoreLabel = tk.Label(root, text='Your Score : ', font=('Helvetica', 24, 'bold'), fg='blue')

scoreLabel.place(x=30, y=130)


scoreLabelcount = tk.Label(root, text=score, font=('Helvetica', 24, 'bold'), fg='green')

scoreLabelcount.place(x=80, y=190)


timeLabel = tk.Label(root, text='Time Left : ', font=('Helvetica', 24, 'bold'), fg='blue')

timeLabel.place(x=550, y=130)

```

```
timeLabelcount = tk.Label(root, text=timeleft, font=('Helvetica', 24, 'bold'), fg='green')
```

```
timeLabelcount.place(x=600, y=190)
```

```
gamePlayLabel = tk.Label(root, text='Hit Enter \n to \nStart the game', font=('Helvetica', 20,  
'italic bold'), bg='#F6F6F6', fg='red')
```

```
gamePlayLabel.place(x=360, y=420)
```

```
#making an advice button
```

```
btn = tk.Button(root, text="Advice", command=reminder)
```

```
btn.place(x=700, y=560)
```

```
#making a button to play music
```

```
play_music = tk.PhotoImage(file='play.png')
```

```
play_label = tk.Label(image=play_music)
```

```
play_btn = tk.Button(root, image=play_music, command=playMusic, borderwidth=0)
```

```
play_btn.place(x=5, y=540)
```

```
#making a button to stop music
```

```
stop_music = tk.PhotoImage(file='mute.png')
```

```
stop_label = tk.Label(image=stop_music)
```

```
stop_btn = tk.Button(root, image=stop_music, command=stopMusic, borderwidth=0)
```

```
stop_btn.place(x=60, y=540)
```

##### Entry Method #####

“””Here in this line of code we define a word entry Entry on the GUI window and set the font, bd and justify of Entry. The entry box used to take an entry from the user or also we can set some value on the Entry Box.”””

```
wordEntry = tk.Entry(root, font=('airal', 28, 'italic bold'), bd=8, justify='center')
```

```
wordEntry.place(x=210, y=340)
```

```
wordEntry.focus_set() #this means the entry label is given main focus
```

#####

“””

Define bind function on the event <Return> to a Button, when the command to be run after hitting Enter. When we click on the enter button then the program gives some execution so the main work of the bind function, we bind our code with events.

“””

```
root.bind('<Return>', startGame)
```

“””

Now come to the last line of code of the project is the main loop function that provides an infinite loop. mainloop() is an infinite loop used to run the application, using this function window is not closed as long as.

“””

```
root.mainloop()
```

## Making our project (.py) to executable(.exe) application

### Steps to convert .py to .exe in Python

1. Install [Python 3.6](#) or higher.
2. You need have pyinstaller (to install type pip install pyinstaller in cmd)
3. Install cx\_Freeze, (open your command prompt and type pip install cx\_Freeze.
4. Install idna, (open your command prompt and type pip install idna.
5. Write a .py program i.e your main project code. (TypingSpeed.py)
6. Create a new python file named typingsetup.py on the current directory of your script.
7. In the typingsetup.py file, copy the code below and save it

-----Code-----

```
import sys
from cx_Freeze import *
#add all external image files here

includefiles = ['typingicon.ico', 'A.ico', 'bg.jpg', 'bg.mp3', 'keyboard.jpg', 'last.jpg',
'mute.png', 'play.png']
base = None
if sys.platform == "win32":
    base = "Win32GUI"
shortcut_table = [
    ("DesktopShortcut", # Shortcut
    "DesktopFolder", # Directory_
    "Typing Speed", # Name
    "TARGETDIR", # Component_
    "[TARGETDIR]\TypingSpeed.exe", # Target
    None, # Arguments
    None, # Description
    None, # Hotkey
    None, # Icon
    None, # IconIndex
    None, # ShowCmd
    "TARGETDIR", # WkDir
    )
]
msi_data = {"Shortcut": shortcut_table}

# Change some default MSI options and specify the use of the above defined tables
bdist_msi_options = {'data': msi_data}
setup(
    version="1.0",
```

```

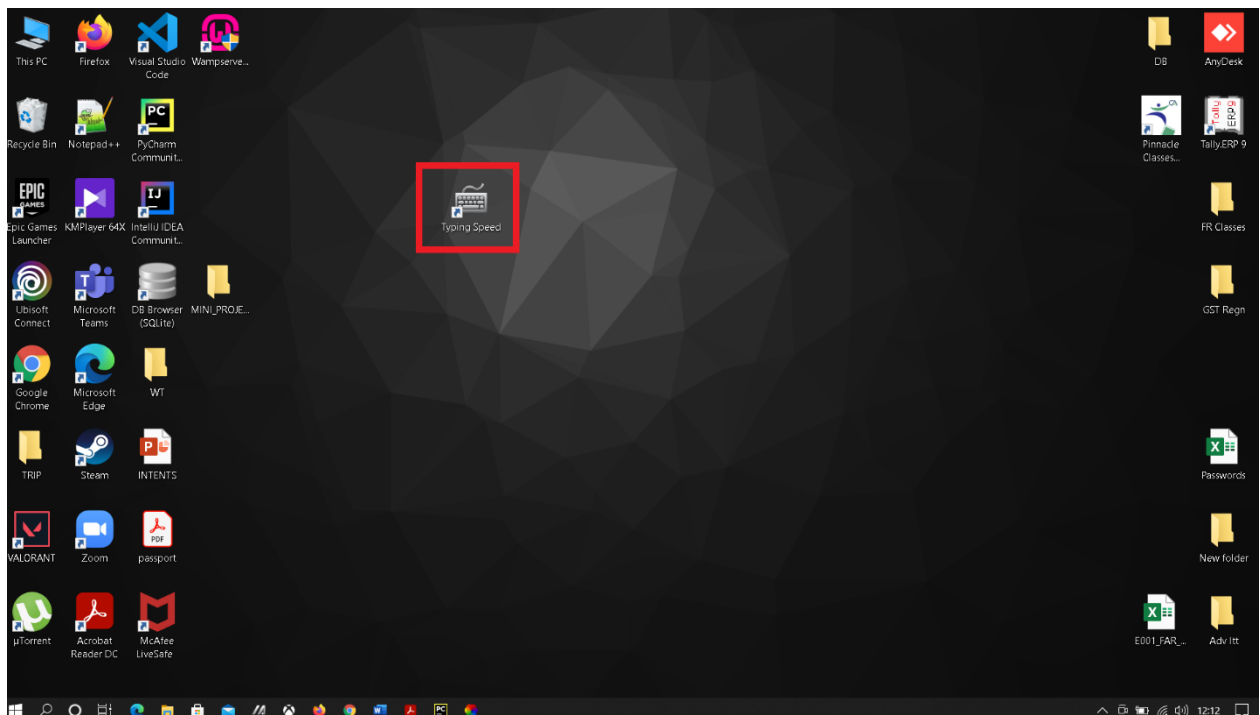
description="Typing Speed Increaser Game",
author="ForCodeCoder",
name="Typing Speed",
options={'build_exe': {'include_files': includefiles}, "bdist_msi":
bdist_msi_options, },
executables=[
    Executable(
        script="TypingSpeed.py",
        base=base,
        icon='A.ico',
    )
]
)

```

8. With shift pressed right click on the same directory, so you are able to open a command prompt window.
9. In the prompt, type `python typingsetup.py bdist_msi` and enter.
10. If your script is error free, then there will be no problem on creating application.
11. Check the newly created folder dist. Within that folder you can find your application. Run it. Make yourself happy.

Note: Make sure that all the packages of the project are installed in your python/script folder.

→This is how it looks after installing .exe application.





## 6. TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation.

### 6.1 TESTING OBJECTIVES:

- To ensure that during operation the system will perform as per specification.
- To make sure that system meets the user requirements during operation
- To make sure that during the operation, incorrect input, processing and output will be detected
- To see that when correct inputs are fed to the system the outputs are correct
- To verify that the controls incorporated in the same system as intended
- Testing is a process of executing a program with the intent of finding an error
- A good test case is one that has a high probability of finding an as yet undiscovered error

The software developed has been tested successfully using the following testing strategies and any errors that are encountered are corrected and again the part of the program or the procedure or function is put to testing until all the errors are removed. A successful test is one that uncovers an as yet undiscovered error.

Note that the result of the system testing will prove that the system is working correctly. It will give confidence to system designer, users of the system, prevent frustration during implementation process etc.

### 6.2 TESTING METHODOLOGIES:

- ✓ White box testing.
- ✓ Black box testing.
- ✓ Unit testing.
- ✓ Integration testing.
- ✓ User acceptance testing.
- ✓ Output testing.
- ✓ Validation testing.
- ✓ System testing.

#### 1) White Box Testing:

White box testing is a testing case design method that uses the control structure of the procedure design to derive test cases. All independent paths in a module are exercised at least once, all logical decisions are exercised at once, execute all loops at boundaries and within their operational bounds exercise internal data structure to ensure their validity. Here the customer is given three chances to enter a valid choice out of the given menu. After which the control exits the current menu.

## **2) Black Box Testing:**

Black Box Testing attempts to find errors in following areas or categories, incorrect or missing functions, interface error, errors in data structures, performance error and initialization and termination error. Here all the input data must match the data type to become a valid entry.

## **3) Unit Testing:**

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

## **4) Integration Testing:**

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and build a program structure that has been dictated by design.

### **The following are the types of Integration Testing:**

#### **✓ Top Down Integration:**

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module.

#### **✓ Bottom Up Integration:**

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated.

#### **5) User acceptance Testing:**

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

#### **6) Output Testing:**

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

#### **7) Validation Testing:**

Validation testing is generally performed on the following fields:

##### **✓ Text Field:**

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

##### **✓ Numeric Field:**

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform.

##### **✓ Preparation of Test Data:**

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

✓ **Using Live Test Data:**

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

✓ **Using Artificial Test Data:**

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.

### **6.3 USER TRAINING:**

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose, the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

### **6.4 MAINTAINENCE:**

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user's requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development

in technology, it may be possible to add many more features based on the requirements in future. The coding and designing are simple and easy to understand which will make maintenance easier.

## **6.5 TESTING STRATEGY:**

A strategy for system testing integrates system test cases and design techniques into a well-planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation. A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding.

### **6.5.1 SYSTEM TESTING:**

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

### **6.5.2 UNIT TESTING:**

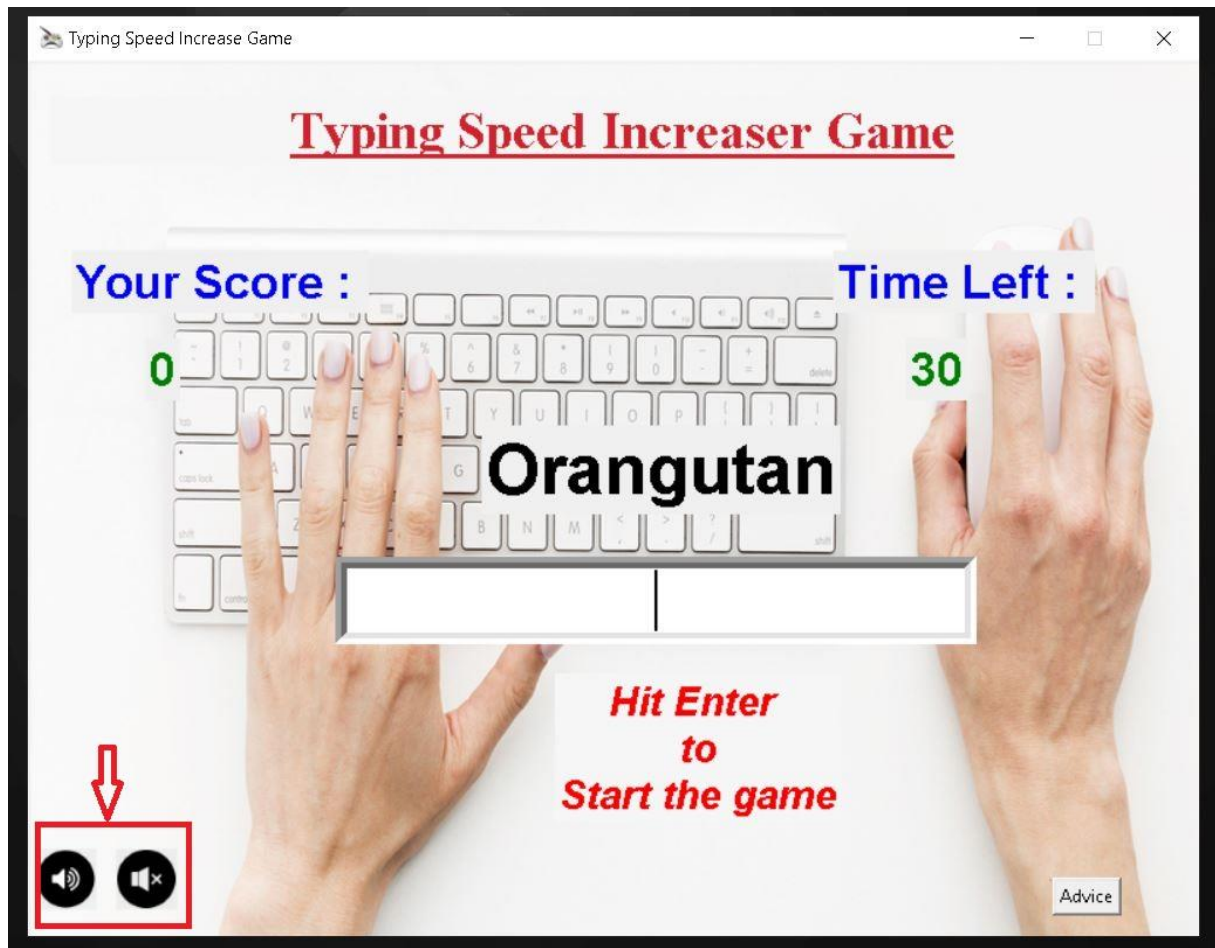
In unit testing different are modules are tested against the specifications produced during the design for the modules. Unit testing is essential for verification of the code produced during the coding phase, and hence the goals to test the internal logic of the modules. Using the detailed design description as a guide, important Conrail paths are tested to uncover errors within the boundary of the modules. This testing is carried out during the programming stage itself. In this type of testing step, each module was found to be working satisfactorily as regards to the expected output from the module. In Due Course, latest technology advancements will be taken into consideration. As part of technical build-up many components of the networking system will be generic in nature so that future projects can either use or interact with this.

## 7.OUTPUT SCREENS

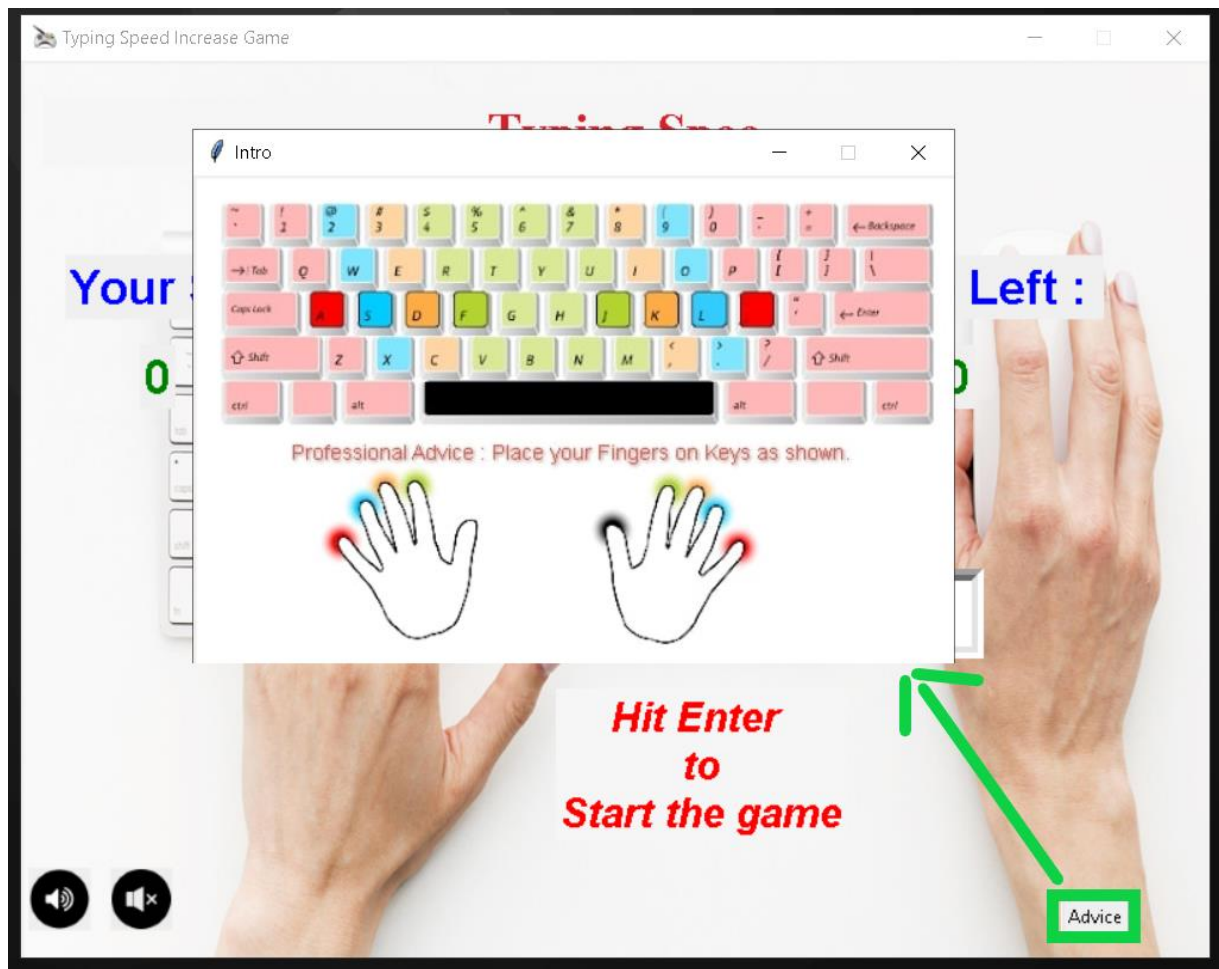
### ➤ Main window



- **Controllable sound i.e play and stop music**

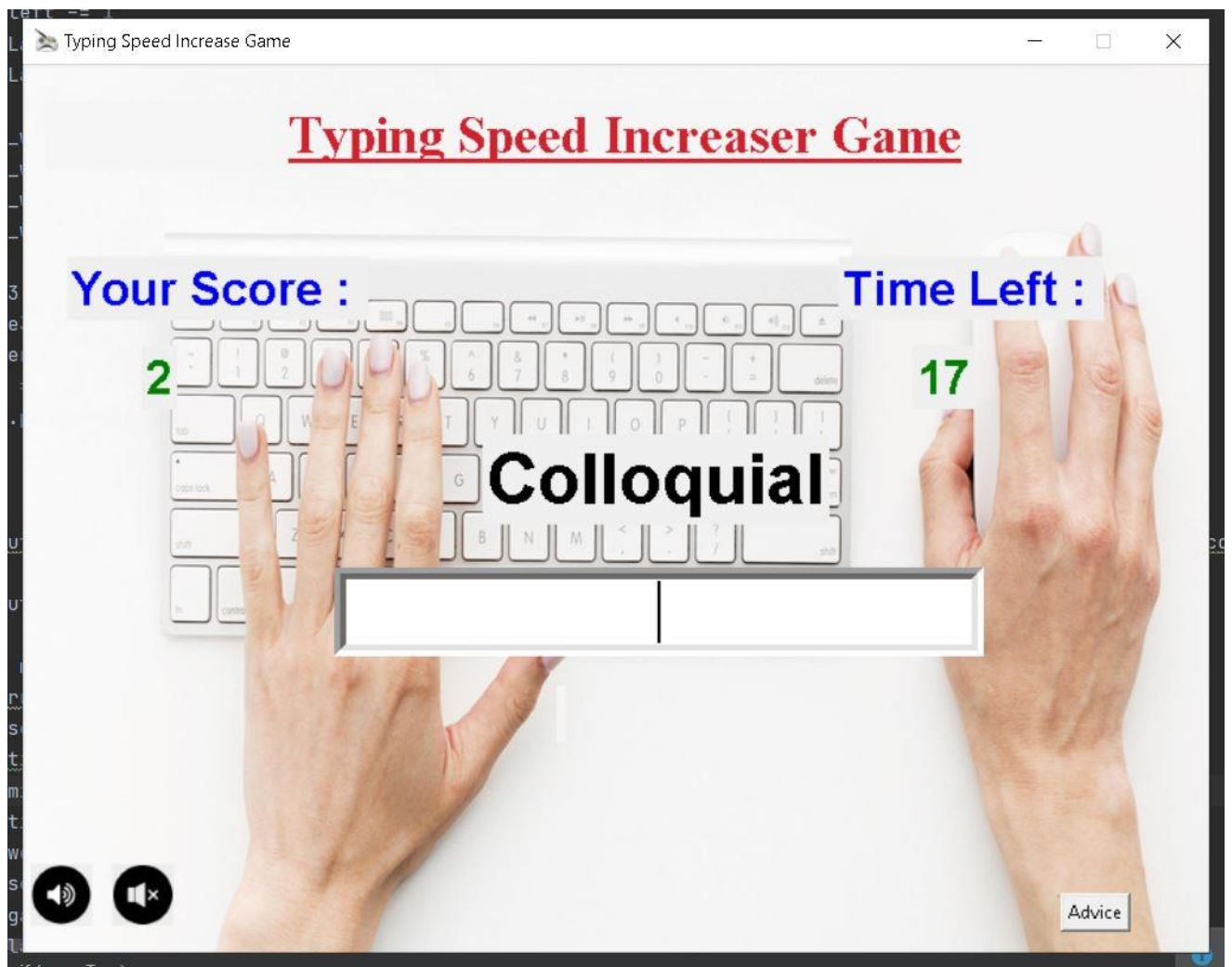


## ➤ Advice button





- After enter the game starts.



➤ **Game over window**

**Here hit retry to restart the game and close to close the game.**



## **8. CONCLUSION & FUTURE SCOPE**

### **8.1 CONCLUSION:**

This application is designed to help anyone to improve their typing speed. The speed typing game based on python Tkinter module is very simple and user friendly and is easy to use for both beginners and advanced users. This project is developed mainly for a single purpose i.e to increase the typing speed of an individual user. This project gives an insight in to the different aspects of python programming and practical knowledge of Tkinter modules and classes.

### **8.2 FUTURE SCOPE:**

- Make the interface look more attractive.
- Adding difficulty levels
- connect the application to a database and store the user's performance/progress
- Include many exercises and games.
- provide feedback in a good way to reinforce learners.
- provide help button in each section
- Guided practice i.e. introducing a AI tutor

## 9. BIBLIOGRAPHY

### 9.1 WEBSITES:

- ✓ [https://www.tutorialspoint.com/python/python\\_gui\\_programming.htm](https://www.tutorialspoint.com/python/python_gui_programming.htm)
- ✓ <https://docs.python.org/3/library/tkinter.html>
- ✓ <https://realpython.com/python-gui-tkinter/>
- ✓ <https://www.geeksforgeeks.org/python-gui-tkinter/>

### 9.2 REFERENCES:

- ✓ Python Crash Course: A Hands-On, Project-Based Introduction to Programming (2nd Edition)
- ✓ Python GUI Programming with Tkinter  
by Alan D. Moore
- ✓ <https://www.youtube.com/playlist?list=PLCC34OHNcOtoC6GglhF3ncJ5rLwQrLGnV>

### GitHub link :

[https://github.com/kiranRudraram/Mini\\_Project/](https://github.com/kiranRudraram/Mini_Project/)