



Bharati Vidyapeeth (Deemed To Be University)  
College Of Engineering, Pune  
Department of Computer Science & Engineering



Presentation for  
**(Computer Science Engineering)**

**SKILL LAB - II**  
**PBL**

03	ANUSHA ANAND	2114110004
05	GAURAV BAJAJ	2114110006
60	HARSHITA JAIN	2114110061
63	RAGHAV KWATRA	2214110584

GUIDED BY - PROF. SANA METKARI



# TYPING SPEED CALCULATOR

SKILL LAB-II PBL

# CONTENT

- INTRODUCTION
- OBJECTIVE
- LITERATURE STUDIES
- ALGORITHM
- METHODOLOGY USED
- IMPLEMENTATION
- RESULTS
- CONCLUSION





The background of the slide features a futuristic digital interface. On the left, there are several floating windows displaying various data visualizations, including bar charts, line graphs, and pie charts. A robotic hand, rendered in a metallic blue and white, is positioned in the center-right, with its index finger pointing towards the text. The overall aesthetic is high-tech and digital, with a blue color palette and glowing light effects.

# INTRODUCTION:

- In the digital age, where effective communication is often keystroke-dependent, one's typing proficiency plays a pivotal role in personal and professional spheres. Recognizing the significance of this skill, the development of a Typing Speed Calculator program becomes a compelling pursuit.
- This report delves into the comprehensive design, implementation, and testing of such a program, aiming to provide users with an insightful tool to measure their typing speed and identify areas for improvement.
- The primary objective of this endeavor is to create a Typing Speed Calculator program that goes beyond mere speed measurement. The program aims to provide users with a nuanced evaluation of their typing abilities, encompassing not only the words-per-minute metric but also an analysis of errors made during the typing process.
- By focusing on accuracy as well as speed, the program becomes a comprehensive tool for users to hone their typing skills systematically.

# OBJECTIVE

The Typing Speed Calculator project is a software application designed to measure and enhance the typing proficiency of individuals. This project aims to address the growing importance of typing skills in the digital age, where efficient and accurate typing is an essential skill for both personal and professional use. The Typing Speed Calculator offers a user-friendly interface that enables users to test their typing speed and accuracy while providing valuable feedback for improvement.

The core features of the Typing Speed Calculator include the ability to input custom texts or select from a diverse range of predefined texts for typing exercises. Users can set time limits for typing tests, and the system records the number of words typed, errors made, and the net words per minute (WPM) achieved. Additionally, the project provides graphical representation and statistical data to track typing progress over time, enabling users to set goals and measure their performance.



# LITERATURE STUDIES:

## How does Typing Speed Calculator work?

- **User Input:** Users start by entering the platform and can choose from various options. They can either input their custom text for typing practice or select from predefined texts available within the system.
- **Exercise Settings:** Users can configure exercise settings, including setting time limits for typing tests. This allows users to adapt the exercises to their specific needs and goals.

- **Typing Evaluation:** As users' type, the system records their input, calculating metrics such as the number of words typed, the number of errors made, and the net words per minute (WPM) achieved. This real-time evaluation provides users with immediate feedback on their performance.
- **Progress Tracking:** The Typing Speed Calculator project offers visual representations and statistical data to help users track their typing progress over time. This includes charts, graphs, and historical data to show improvements or areas that may need further attention.

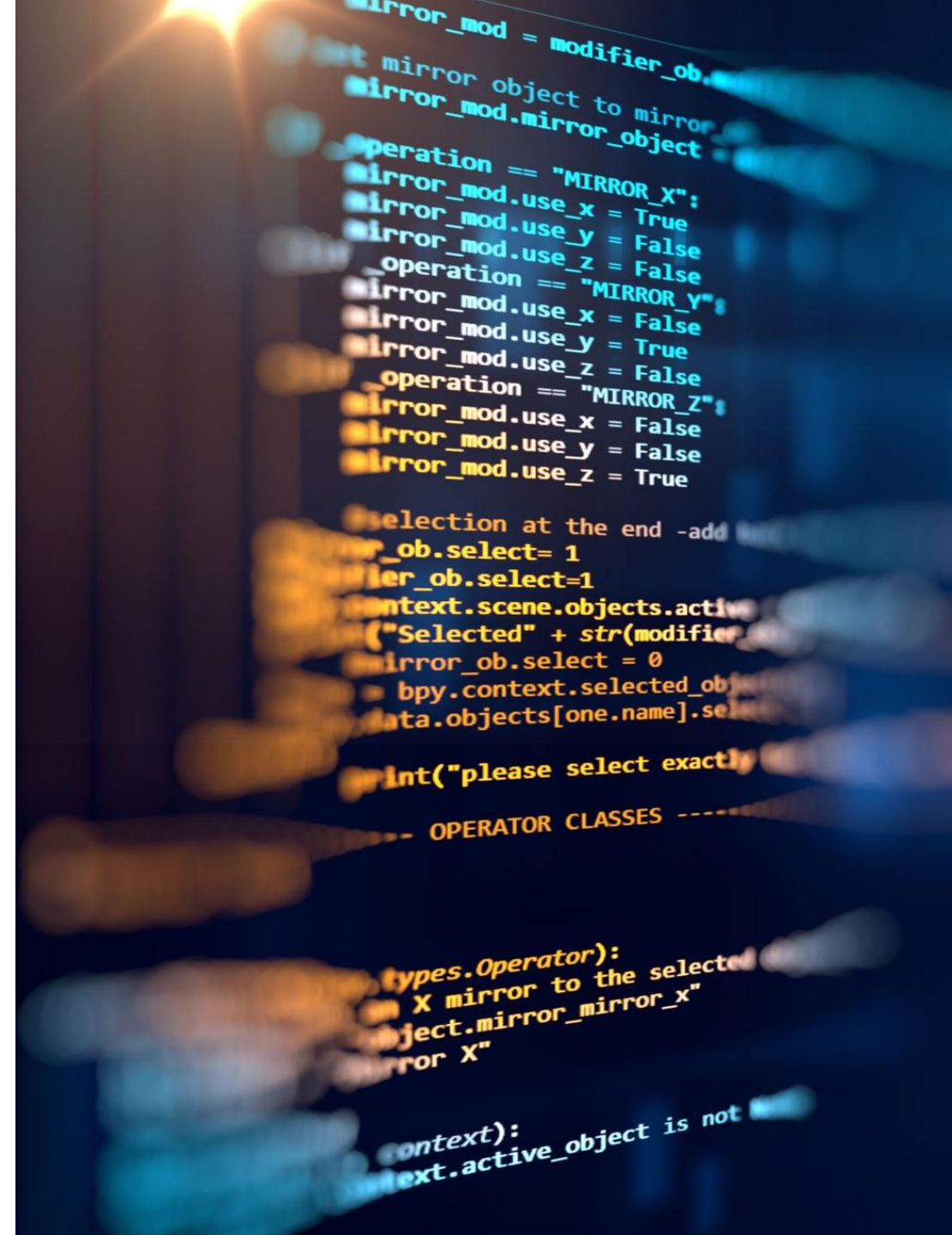


# TYPING SPEED TEST ALGORITHM

- **Timing Algorithm:** The core of a typing speed test is measuring the time it takes for the user to type a given text. Typically, a start time is recorded when the user begins typing, and an end time is recorded when they finish. The difference between these times provides the typing speed.
- **Text Comparison Algorithm:** To calculate accuracy, the typed text is compared to the provided text. A common method for this is Levenshtein distance or edit distance algorithm. It calculates the minimum number of single-character edits (insertions, deletions, or substitutions) needed to change one string into another. The ratio of correct characters to total characters typed is used to determine accuracy.
- **Random Text Generation:** To prevent users from memorizing the test text, the program may have algorithms for generating random sentences or paragraphs for the test.

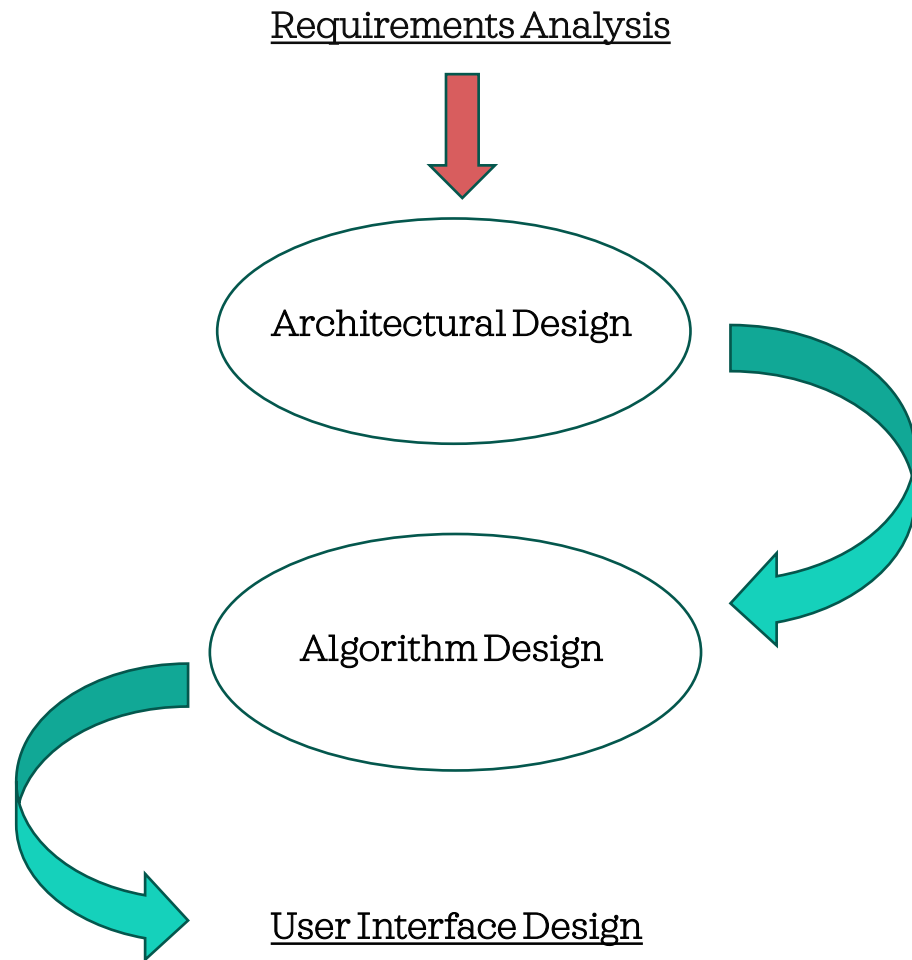
# TYPING SPEED TEST ALGORITHM

- **Words per Minute (WPM) Calculation:** WPM is a standard metric for typing speed. It is calculated by considering the number of words typed and the time taken. The formula often used is:  $\text{WPM} = (\text{Total characters typed} / 5) / (\text{Time in minutes})$ . The division by 5 is because, on average, English words are about 5 characters long.
- **User Interface:** The program's user interface includes components like a text input area, a timer, and a display to show the text to be typed. The user interface should also provide feedback on typing speed and accuracy. .
- **Difficulty Adjustment:** Some typing speed tests may dynamically adjust the difficulty based on the user's performance, making the text harder or easier to type as the user progresses.





## METHODOLOGY USED:



# METHODOLOGY USED

- **DESIGN PHASE:**

- 1.1 Requirements Analysis:**

- Identified user requirements through surveys, feedback, and analysis of similar tools.

- Defined key functionalities: input handling, speed calculation, error detection.

- 1.2 Architectural Design:**

- Chose a modular and scalable architecture to accommodate future enhancements.

- Separated concerns for input processing, speed calculation, and error detection.

- 1.3 Algorithm Design:**

- Developed algorithms for word count, speed calculation, and error detection.

- Implemented a dynamic formula for typing speed based on user input time.

- 1.4 User Interface Design:**

- Designed an intuitive and user-friendly interface to enhance user experience.

- Incorporated features for displaying typing speed, error count, and visual feedback.

# METHODOLOGY USED

- **IMPLEMENTATION PHASE:**

- 2.1 Programming Language:**

- Selected a programming language (e.g., Python) conducive to rapid development and ease of maintenance.

- 2.2 Input Handling:**

- Implemented robust input handling to accept diverse types of text input.

- Addressed challenges such as whitespace variations, punctuation, and special characters.

- 2.3 Speed Calculation:**

- Implemented the typing speed calculation algorithm based on the time taken and word count.

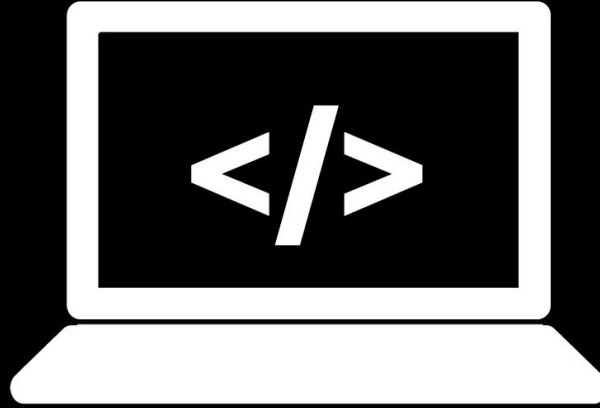
- Considered factors such as average word length to provide a nuanced speed metric.

- 2.4 Error Detection:**


- Developed algorithms to identify errors, including missing characters, capitalization errors, repeated words, and more.

- Utilized regular expressions and string manipulation techniques.



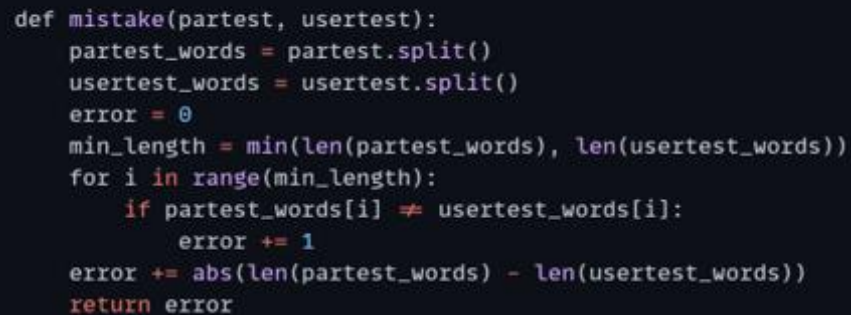


**IMPLEMENTATION**



```
import tkinter as tk
from tkinter import messagebox
import random as r
from time import time
```

- Import the Tkinter library and alternate name for it is 'tk' for convenience.
- Import the 'message box' module from Tkinter for displaying message box.
- Import the random module and alternate name for it is 'r'.
- Import the time function from the time module.

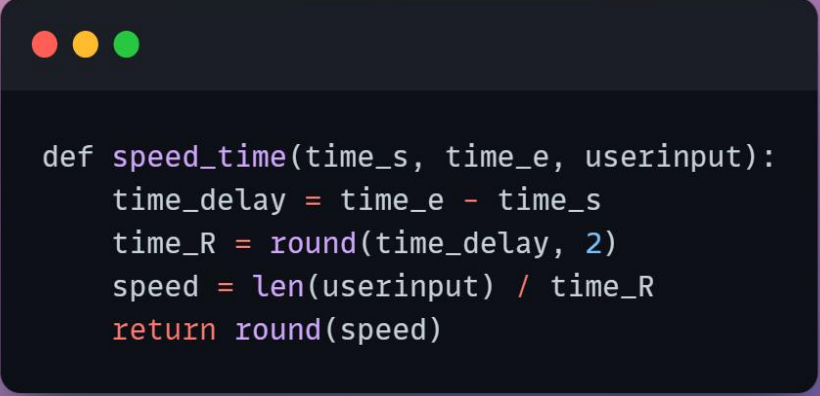


```
def mistake(partest, usertest):  
    partest_words = partest.split()  
    usertest_words = usertest.split()  
    error = 0  
    min_length = min(len(partest_words), len(usertest_words))  
    for i in range(min_length):  
        if partest_words[i] != usertest_words[i]:  
            error += 1  
    error += abs(len(partest_words) - len(usertest_words))  
    return error
```

snappify.com

- Mistake function takes two input parameters (“partest” and “usertest”), which represent the original sentence and the user's input.
- The sentences are split into lists of words using the split() method.
- It then iterates through the words up to the minimum length of the two lists, comparing each corresponding pair of words. If a difference is found, it increments the error count.
- After the loop, it adds the absolute difference in length between the two lists to the error count.
- The total error count is returned as the result.





```
def speed_time(time_s, time_e, userinput):  
    time_delay = time_e - time_s  
    time_R = round(time_delay, 2)  
    speed = len(userinput) / time_R  
    return round(speed)
```

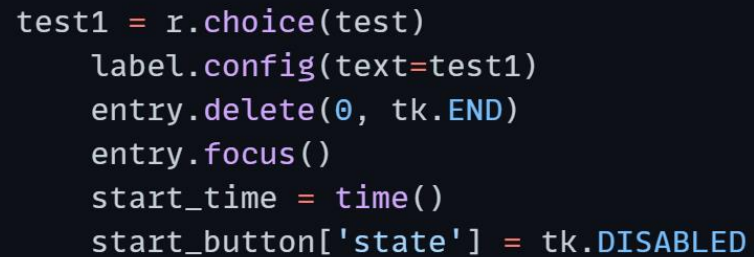
snappify.com

- The Speed-time function takes three input parameters: time\_s (start time), time\_e (end time), and userinput (the user's input).
- It calculates the time taken to complete the test by subtracting the start time from the end time.
- The time delay is rounded to two decimal places using the round function.
- The typing speed is calculated by dividing the length of the user's input by the rounded time delay.
- The final typing speed is rounded to the nearest whole number using the round function before being returned.

```
def start_test():  
    test = [  
        "The science and engineering of making intelligent machines, especially intelligent computer programs.",  
        "A computer program with AI can answer the generic questions it is meant to solve.",  
        "Artificial Intelligence is a developmental science.",  
        "Robot is an electromechanical technology .",  
        "Democracy is a political system .",  
        "Guitar is a musical instrument .",  
        "Solar energy is a renewable resource .",  
        "Psychology is a study of the human mind .",  
        "The Great Wall of China is a historical landmark.",  
        "Ice cream is a popular dessert .",  
        "Internet is a global network of computers .",  
        "Hiking is a recreational activity .",  
        "Pizza is a beloved Italian dish .",  
        "Language is a means of communication ."  
    ]
```

snappify.com

- The `start_test` function initializes a list of sentences that will be used as test prompts for typing.
- When this function is called, it randomly selects a sentence from the list, updates the label in the Tkinter interface with the selected sentence, and records the start time for measuring typing speed.



```
test1 = r.choice(test)
label.config(text=test1)
entry.delete(0, tk.END)
entry.focus()
start_time = time()
start_button['state'] = tk.DISABLED
```

snappify.com

- This block essentially prepares the interface for a new typing test by selecting a random sentence, updating the display, clearing the input area, setting the focus for user input, and disabling the "Start Test" button until the current test is completed. The start time is also recorded to measure the typing speed accurately.



```
def end_test():  
    end_time = time()  
    user_input = entry.get()  
    speed = speed_time(start_time, end_time, user_input)  
    errors = mistake(test1, user_input)  
    result = f"Speed: {speed} w/sec\nErrors: {errors}"  
    messagebox.showinfo("Test Result", result)  
    start_button['state'] = tk.NORMAL  
  
submit_button.config(command=end_test)
```

snappify.com

- The `end_test` function is called when the user presses the "Submit" button, and it calculates and displays the typing speed and any errors.
- This function provides feedback to the user by showing the typing speed and any errors in a message box after they submit their input. It also re-enables the "Start Test" button for initiating a new test.

```
root = tk.Tk()
root.geometry('940x735+200+10')

root.title("Typing Speed Test")

mainframe = tk.Frame(root, bd=4)
mainframe.grid()

titleframe = tk.Frame(mainframe, bg='pink')
titleframe.grid()

titleLabel = tk.Label(titleframe, text='TYPING TYCOON', font=('Times New Roman', 28, 'bold')
                      bd=10)
titleLabel.grid(pady=5)
```

- This part of the code initializes the main Tkinter window (root) and sets up a basic structure for the graphical user interface (GUI).
- This code sets up the basic structure of the Tkinter window, including frames and labels for organizing and displaying content.

```
label = tk.Label(root, text="", font=("Consolas", 22))
label.grid(pady=20)

entry = tk.Entry(root, font=("Consolas", 24), width=50, bd=2, fg='pink')
entry.grid(pady=20)

button_s = tk.Frame(root, bd=4)
button_s.grid()

start_button = tk.Button(button_s, text="Start Test", command=start_test, state=tk.NORMAL, font=("Consolas", 14),
                          width=10, height=1)
start_button.grid(row=0, column=0)

submit_button = tk.Button(button_s, text="Submit", font=("Consolas", 14), width=10, height=1)
submit_button.grid(row=0, column=1, padx=20)

exit_button = tk.Button(button_s, text="Exit", command=root.destroy, font=("Consolas", 14), width=10, height=1)
exit_button.grid(row=0, column=2)

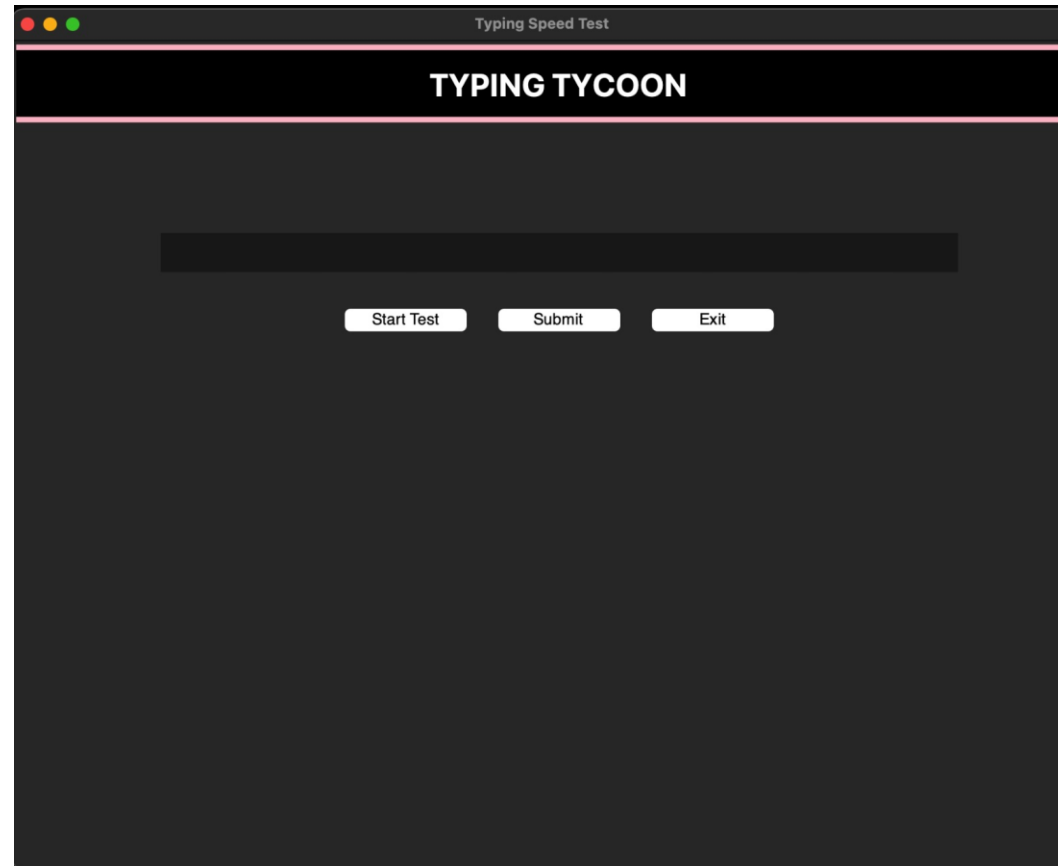
root.mainloop()
```

- This part of the code adds additional Tkinter widgets (Entry widget and Buttons) to the GUI, allowing users to interact with the typing speed test application.
- This code adds an entry widget for user input and three buttons ("Start Test," "Submit," and "Exit") to the Tkinter interface, enhancing the interactivity of the typing speed test application.



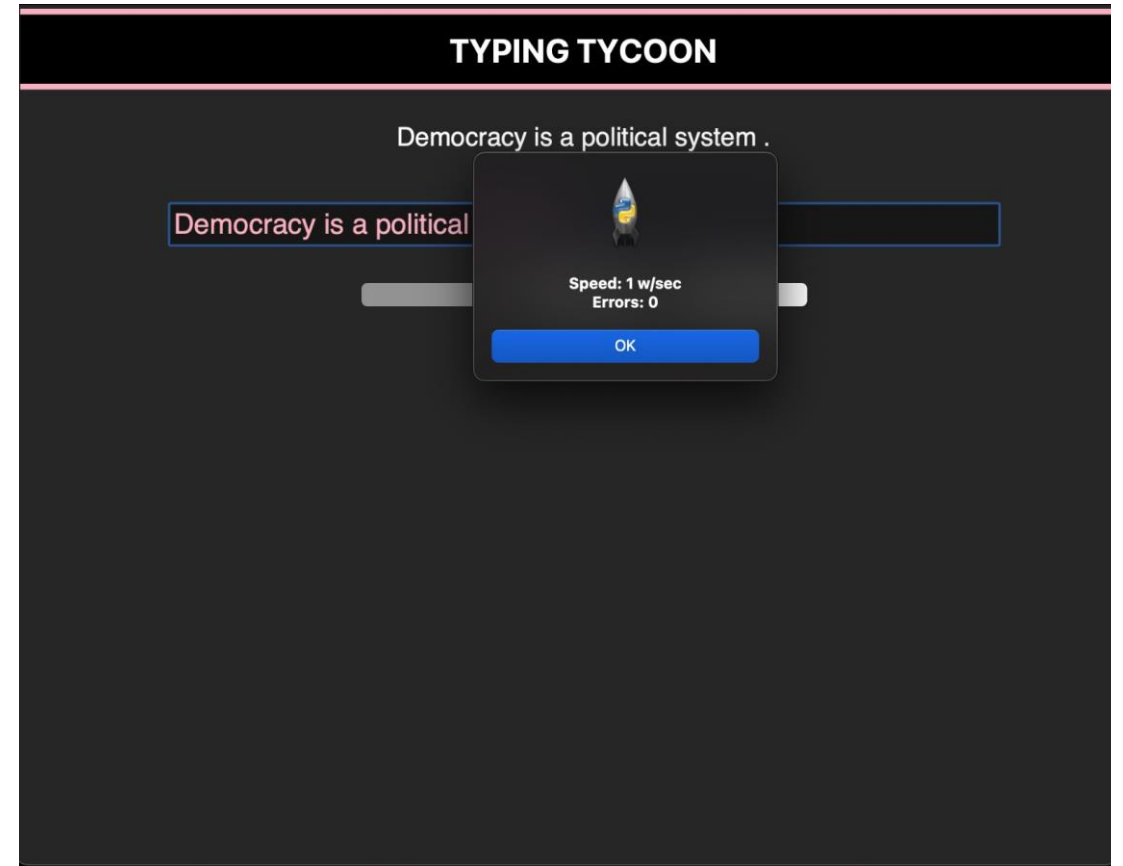
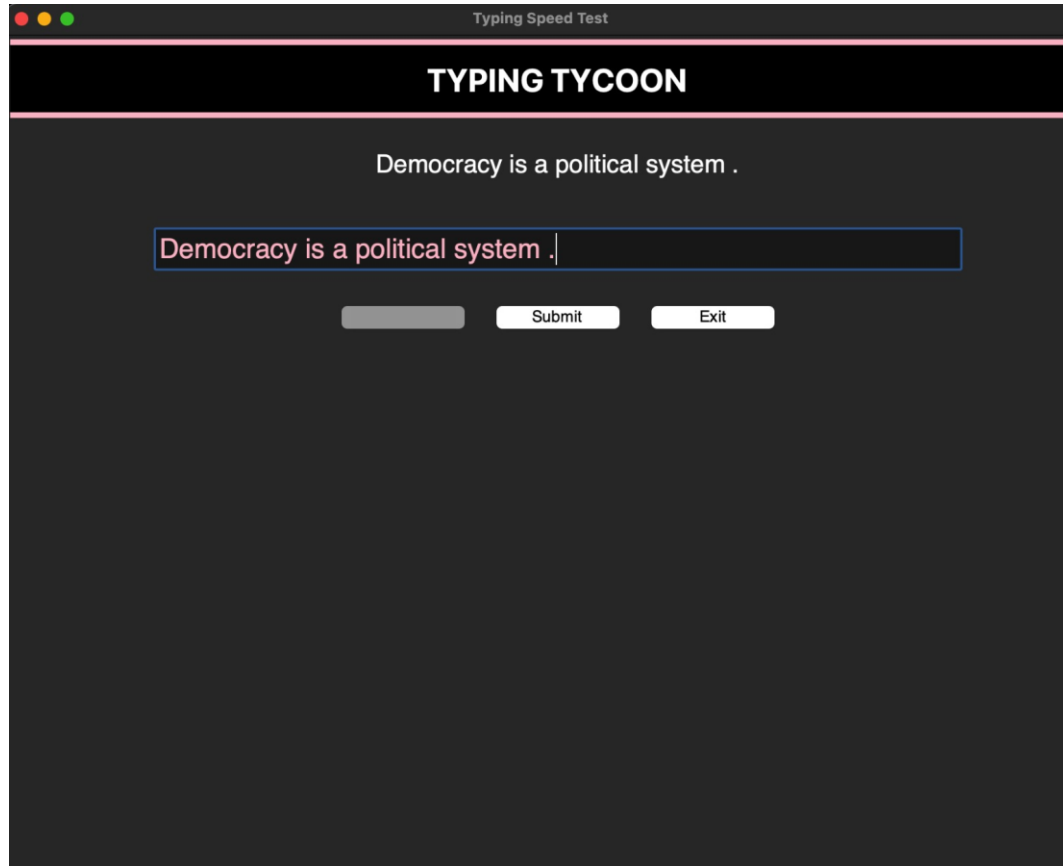
# RESULT:

---



# RESULT

---



# RESULT:

---





# CONCLUSION

The Typing Speed Calculator program demonstrates proficiency in accurately measuring typing speed while providing meaningful feedback on errors. The thorough testing process ensures the reliability and versatility of the program, positioning it as a valuable asset for users aiming to enhance their typing capabilities.

While the Typing Speed Calculator places a focus on speed, it is important to note its broader significance. The program contributes to digital literacy, effective communication, and personal development. The nuanced error analysis encourages users to not only type quickly but also accurately, emphasizing the importance of precision in written communication.

In an era where digital communication is integral to daily life, the Typing Speed Calculator stands as a relevant and valuable tool. Its impact spans across educational, professional, and personal spheres, aligning with the ever-increasing demand for efficient and accurate typing skills. The journey from conceptualization to implementation has been marked by meticulous attention to detail, a commitment to user-centric design, and a vision for the program's role in fostering digital proficiency.

In essence, the Typing Speed Calculator program represents more than just a technical solution; it embodies the intersection of technology and a fundamental skill that transcends industries and empowers individuals in their digital endeavors.





# Thank You