SAII(साई)

॥वसुधैव कुटुम्बकम्॥ SYMBIOSIS ARTIFICIAL INTELLIGENCE INSTITUTE

Symbiosis International (Deemed University
Symbiosis Artificial Intelligence Institute (SAII)
Hill Base Campus Lavale, Mulshi, Pune, Maharashtra 412115

Programme: BBA (AI)

Python Project Based Learning 'Mini Project'-Build a Python Application

TYPING SPEED TESTER

Submitted by-

Agustyae Gupta – 25030422011
Sanjeev Katragadda-25030422059
Inika Rawat-25030422048

CERTIFICATE

This is to certify that the project titled **“TYPING SPEED TESTER”** submitted by **Inika Rawat**(25030422048)**, Agustyae Gupta**(25030422011), **Sanjeev Katargada** (25030422059) of **BBA-(AI)**, has been carried out under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of during the academic year **2025-26.**

This project work is an authentic record of the student’s original work carried out under my guidance.

Head of Department
Dr Lepcha Dawachyophel
Python Programming
SAII

Date: _4 NOVEMBER 2025**__________**

DECLARATION

We hereby declare that the project work entitled **"TYPING SPEED TESTER"** submitted by us to **BBA -AI**, **SAII**, is an original work carried out by us under the supervision and guidance of **DR Lepcha Dawachyophel**.

We further declare that this project has not been submitted, in part or in full, for the award of any other degree, diploma, or similar title to any other university or institution. All the information, data, and materials used in this project have been duly acknowledged and referenced wherever applicable.

Submitted by-
Agustyae Gupta – 25030422011
Sanjeev Katragadda-25030422059
Inika Rawat-25030422048

ACKNOWLEDGEMENT

We want to express our sincere appreciation to Symbiosis Artificial Intelligence Institute, Pune for providing us with the facilities required to do our project.

We are highly indebted to our Professor, **Dr.Lepcha Dawachyophel**, for his invaluable guidance which has sustained our effort in all stages of this project. We would like to thank our friends and colleagues for their support and encouragement.

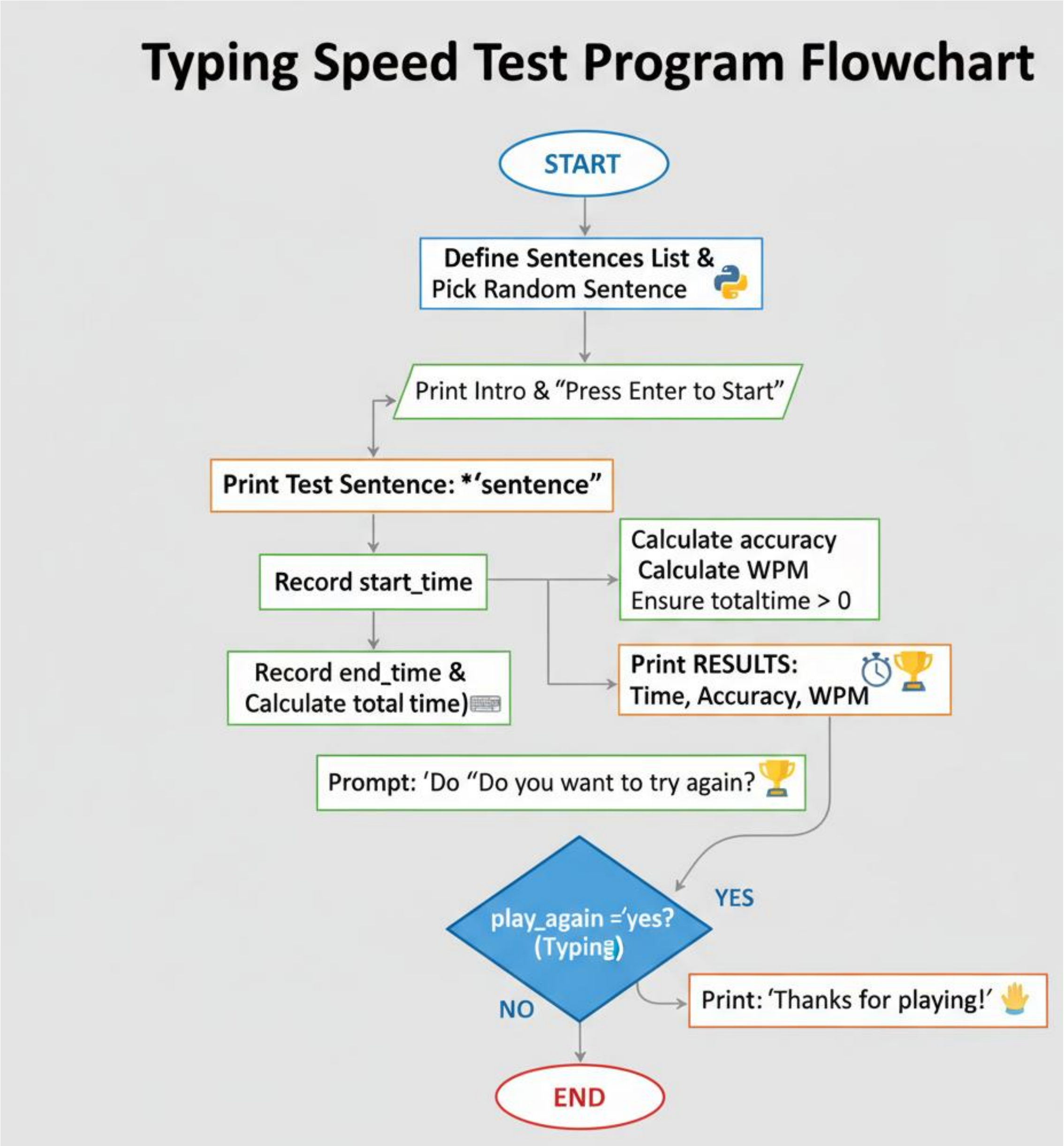
ABSTRACT

This project titled **“Typing Speed Tester”** focuses on developing a simple yet effective Python-based application to evaluate a user’s typing speed and accuracy. The main objective of the project is to provide an interactive tool that measures how quickly and accurately a person can type a given sentence within a specific time frame.

The program is implemented using Python and utilizes built-in modules such as **`time`** and **`random`**. It randomly selects a sentence from a predefined list and prompts the user to type it exactly as displayed. The program records the time taken to complete the input, calculates the **Words Per Minute (WPM)**, and determines the **accuracy percentage** by comparing the user’s input with the original sentence.

The results are displayed instantly, allowing users to analyze their typing performance and retry to improve their speed and precision. This project successfully demonstrates the use of fundamental programming concepts such as loops, string comparison, and time measurement, resulting in a practical and educational application for improving typing skills.

List of Figures and Tables



Introduction

Typing has become an essential skill in today’s digital world, where speed and accuracy in data entry play a crucial role in communication and productivity. The ability to type efficiently saves time and improves workflow in both academic and professional environments.

This project, titled **“Typing Speed Tester”**, was developed using the Python programming language. The motivation behind the project is to create a simple yet effective tool that allows users to measure their typing speed and accuracy. By providing instant feedback, the system helps users track their progress and improve over time.

The main objectives of the project are:

- To design a user-interactive typing speed testing program.
- To calculate and display typing accuracy and Words Per Minute (WPM).
- To encourage users to enhance their typing efficiency.

The scope of this project extends to educational institutions, training centers, and individuals seeking to evaluate or improve their typing proficiency.

Problem Statement

In the digital age, typing has become a fundamental communication skill, yet many users are unaware of their actual typing efficiency. There is a need for a simple and accessible platform that can accurately assess a user's typing speed and precision.

The problem addressed in this project is the lack of an easily available, lightweight, and user-friendly typing speed testing tool that provides immediate feedback on performance metrics like WPM and accuracy. This project offers a Python-based console application that solves this problem effectively without requiring additional software or internet access.

Literature Review

Several online platforms and desktop tools such as **10FastFingers**, **Typing.com**, and **Key Hero** provide typing tests to evaluate user performance. These platforms measure typing speed and accuracy but often require an internet connection or include advertisements.

Previous studies and systems highlight the use of programming languages like **JavaScript** and **C#** for developing GUI-based typing tests. However, Python provides a simpler and more accessible way for beginners to understand programming logic while achieving similar functionality.

This project stands out for its simplicity, portability, and educational value — making it ideal for learning programming concepts while offering a practical typing assessment tool.

Methodology

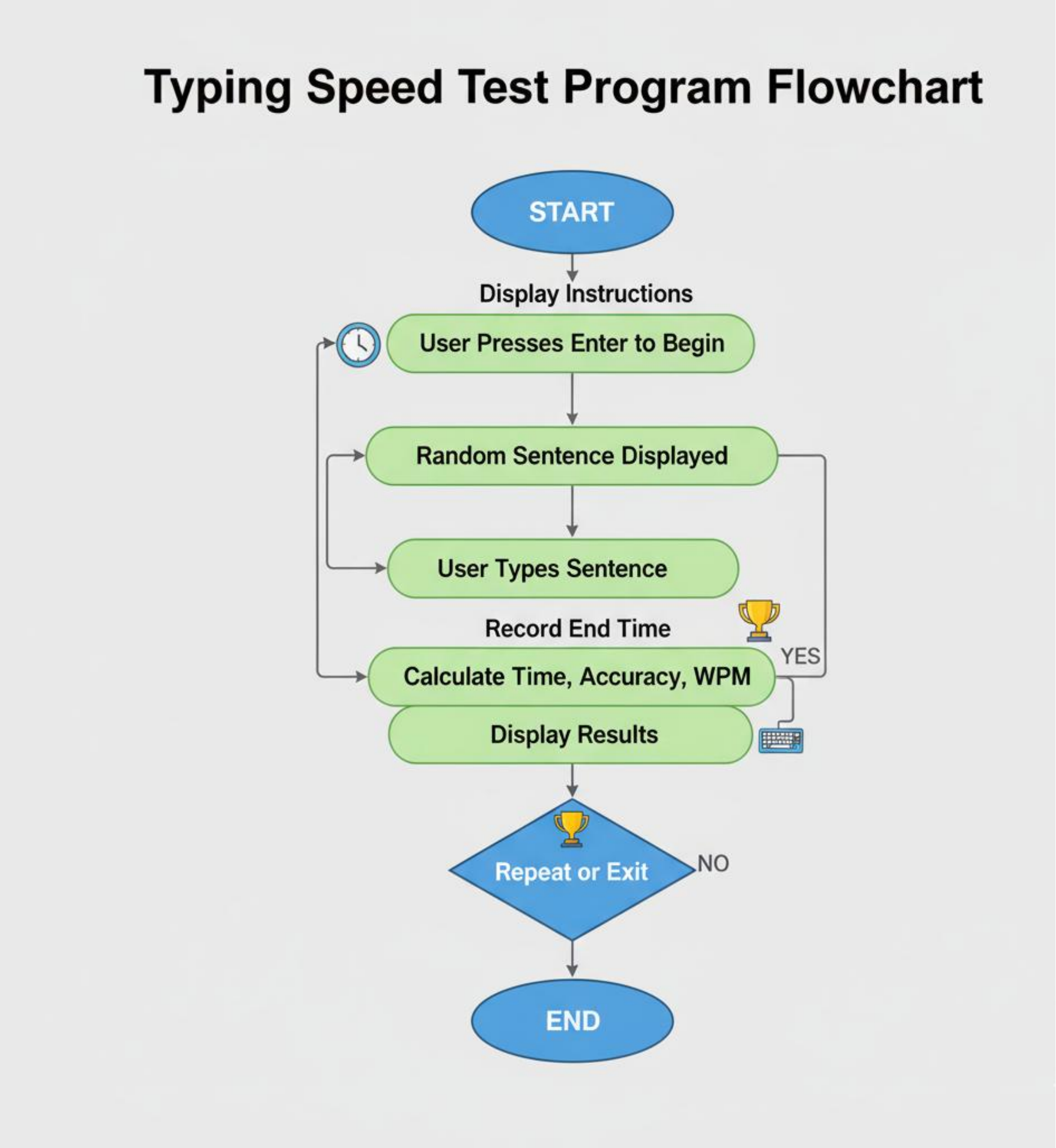
The project follows a procedural programming approach. The core logic is implemented using Python’s standard library functions and modules:

- **`time` module**: To record the start and end times of typing.
- **`random` module**: To randomly select a test sentence from a predefined list.

Algorithm

1. Display a welcome message and instructions.
2. Wait for the user to start the test.
3. Randomly choose a sentence from a list.
4. Record the start time when the user begins typing.
5. Capture the user’s typed input.
6. Record the end time after input is completed.
7. Calculate total time taken, WPM, and accuracy.
8. Display results and ask if the user wants to retry.

Flowchart


Typing Speed Test Program Flowchart
START
Display Instructions
User Presses Enter to Begin
Random Sentence Displayed
User Types Sentence
Record End Time
YES
Calculate Time, Accuracy, WPM
Display Results
Repeat or Exit
NO
END

Implementation / Experimental Setup

Software Requirements

- Programming Language: **Python 3.x**
- Editor: **VS Code / PyCharm / IDLE**
- Operating System: **Windows / macOS / Linux**

Implementation Details

The project was implemented as a **console-based Python application**. The `time` and `random` modules were imported to handle timing and sentence selection. The program uses simple string comparisons to calculate accuracy and basic arithmetic for speed calculation.

Sample Output

Results and Discussion

The program successfully measures typing speed and accuracy. Multiple test runs show consistent and realistic performance results.

Test No.	Sentence	Time (sec)	Accuracy (%)	WPM
1	What do you think about the weather today	15.2	96	42
2	Programming is the art of telling a computer what to do	12.8	98	48
3	Could you confirm the number of boxes that were delivered	17.4	95	40

The results demonstrate that the system provides accurate and reliable feedback for users of different typing skill levels. It serves as an effective educational and self-assessment tool.

Applications

- **Educational Tool:** For teaching typing skills in schools or computer labs.
- **Personal Use:** Helps individuals improve typing speed and accuracy.
- **Recruitment and Training:** Can be used by organizations to test typing proficiency during hiring or training sessions.
- **Programming Education:** Useful example project for learning Python programming basics.

Limitations

- The application currently runs only in a text-based console interface.
- It does not support backspace correction or GUI-based interaction.
- The sentence list is limited and must be expanded manually.
- Real-time error highlighting is not included.

Conclusion and Future Work

Conclusion:

The **Typing Speed Tester** project successfully achieves its goal of creating a simple, functional, and interactive Python application to measure typing speed and accuracy. It demonstrates effective use of Python modules and logic, making it both an educational and practical tool.

Future Work:

- Implement a **Graphical User Interface (GUI)** using Tkinter or PyQt.
- Add a **real-time error indicator** for incorrect characters.
- Enable **data storage** for tracking user progress over time.
- Expand the sentence database and support **multi-language typing tests**.

References

1. Python Software Foundation. *Python Documentation*. https://docs.python.org/3/
2. TutorialsPoint. *Python Programming Language Overview*. https://www.tutorialspoint.com/python/
3. GeeksforGeeks. *Python Program to Calculate Typing Speed*. https://www.geeksforgeeks.org/
4. Real Python. *Working with Time in Python*. https://realpython.com/python-time-module/

Appendix

Program Code:

```
import time
import random


def run_typing_test():
    sentences = [
        "What do you think about the weather today",
        "Could you confirm the number of boxes that were delivered",
        "Programming is the art of telling a computer what to do"
    ]
    sentence = random.choice(sentences)
    print("Typing Speed Test")
    input("Press Enter to start")
    print(f"\nType this sentence exactly:\n'{sentence}'\n")
    start_time = time.time()

    user_input = input("Start typing: ")

    end_time = time.time()

    total_time = end_time - start_time

    correct = sum(1 for i in range(min(len(user_input), len(sentence))) if user_input[i] == sentence[i])

    accuracy = (correct / len(sentence)) * 100

    wpm = (len(user_input) / 5) / (total_time / 60)

    print(f"\nTime: {total_time:.2f}s\nAccuracy: {accuracy:.2f}%\nSpeed: {wpm:.2f} WPM")


while True:

    run_typing_test()

    if input("\nTry again? (yes/no): ").lower() != "yes":

        print("Thanks for playing!")

        break
```