

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI - 590018



Python Project Report (22CSE145)

on

TYPING SPEED TEST USING PYTHON

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by

1BG22CS067 KIRAN

1BG22CS081 MITHUN A N

1BG22CS118 RAHUL

Guide

Prof. Akshitha Katkeri

Assistant Professor, Dept. of CSE

BNMIT, Bengaluru



Vidyayāmṛthamashnute

B.N.M. Institute of Technology

An Autonomous Institution under VTU

Approved by AICTE, Accredited as grade A Institution by NAAC. All eligible branches – CSE, ECE, EEE, ISE & Mech. Engg. are Accredited by NBA for academic years 2018-19 to 2024-25 & valid upto 30.06.2025

URL: www.bnmit.org

Department of Computer Science and Engineering

2023 - 2024

B.N.M. Institute of Technology

An Autonomous Institution under VTU

Approved by AICTE, Accredited as grade A Institution by NAAC. All eligible branches – CSE, ECE, EEE, ISE & Mech. Engg. are Accredited by NBA for academic years 2018-19 to 2024-25 & valid upto 30.06.2025

URL: www.bnmit.org

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



Vidyayāmruṭhamashnuthē

CERTIFICATE

Certified that the Python project entitled **Project title** carried out by **KIRAN (1BG22CS067)**, **MITHUN A N (1BG22CS081)**, **RAHUL (1BG22CS118)**, are Bonafide students of IV Semester, BNM Institute of Technology in partial fulfillment for the award of Bachelor of Engineering in COMPUTER SCIENCE AND ENGINEERING of Visvesvaraya Technological University, Belagavi during the year 2023-24. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the project report deposited in the departmental library. The Python project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Prof . Akshitha Katkeri
Assistant Professor
Department of CSE
BNMIT, Bengaluru

Dr. Chayadevi M L
Professor & HOD
Department of CSE
BNMIT, Bengaluru

Name

Signature with Date

Examiner 1:

Examiner 2:

ACKNOWLEDGEMENT

We would like to place on record our sincere thanks and gratitude to the concerned people, whose suggestions and words of encouragement has been valuable.

We express our heartfelt gratitude to the management of **BNM Institute of Technology**, for giving us the opportunity to pursue Degree of Computer Science and Engineering and helping us to shape our career. We take this opportunity to thank **Shri. Narayan Rao R. Maanay**, Secretary, **Prof. T. J. Rama Murthy**, Director, **Dr. S. Y. Kulkarni**, Additional Director and Principal, **Prof. Eishwar N Maanay**, Dean and **Dr. Krishnamurthy G. N.**, Deputy Director for their support and encouragement to pursue this project. We would like to thank **Dr. Chayadevi M L**, Professor and Head, Dept. of Computer Science and Engineering, for her support and encouragement.

We would like to thank our guide **Prof. Akshitha Katkeri**, Assistant Professor, Dept. of Computer Science and Engineering, who has been the source of inspiration throughout our project work and has provided us with useful information at every stage of our project.

Finally, we are thankful to all the teaching and non-teaching staff of Department of Computer Science and Engineering for their help in the successful completion of our project. Last but not the least we would like to extend our sincere gratitude to our parents and all our friends who were a constant source of inspiration.

1BG22CS067 KIRAN

1BG22CS081 MITHUN A N

1BG22CS118 RAHUL

ABSTRACT

The Typing Speed Test application is designed to enhance users' typing proficiency through structured practice sessions. It employs a user-friendly interface developed using Tkinter, a Python library for GUI development. Upon launching the application, users are presented with options to log in or sign up, ensuring personalized experiences and secure access to their typing data. The MongoDB database backend securely handles user authentication, storing hashed passwords to maintain data privacy and integrity.

Users can select from different typing challenges based on their skill level. The application offers two primary modes: Beginner, featuring easy words, and Veteran, which includes more complex vocabulary. Additionally, users can choose the number of words to type and set specific time limits for their tests, ranging from 10 to 30 seconds. This flexibility allows users to tailor their practice sessions to match their individual needs and improve incrementally.

The typing test displays a generated sentence split into two lines for better visibility. As users type, the application provides immediate feedback by changing the color of each character to indicate correct (green) or incorrect (red) keystrokes. This visual feedback helps users quickly identify and correct mistakes, reinforcing learning. A timer runs concurrently, ensuring that users complete the test within the chosen duration. When the time limit is reached or the sentence is fully typed, the application stops the test and calculates the results.

Post-test, the application computes and displays typing statistics, including elapsed time, words per minute (WPM), and accuracy percentage. These metrics help users track their progress over time. Furthermore, users can access their past test results through the History feature, which retrieves and displays up to the last eight tests from the database. This historical data provides insights into users' improvement and areas that may require additional practice.

The application's robust functionality is built on comprehensive backend logic, handling sentence generation, typing event management, real-time feedback, timer updates, result calculations, and database interactions. By combining these elements, the Typing Speed Test application offers a valuable tool for users seeking to enhance their typing speed and accuracy in an engaging and interactive manner.

TABLE OF CONTENTS

CONTENTS	Page No.
ACKNOWLEDGEMENT	I
ABSTRACT	II
LIST OF FIGURES	III
Chapter 1 – Introduction	1
1.1. Statement of the Problem	4
1.2. Objective of the Project	4
Chapter 2 – Literature Survey	5
Chapter 3 – System Requirement Specification	9
3.1 Software Requirement	10
3.1.1 Visual Studio Code	10
3.1.2 Python Libraries	11
3.1.3 Web Browser Compatibility	12
3.1.4 Python Modules	12
3.2 Hardware Requirement	13
Chapter 4 – Methodology and Implementation	14
4.1 Methodology	14
4.2 Execution Flowchart	17
Chapter 5 – Testing and Validation	18
Chapter 6 – Result and Discussion	21
Chapter 7 – Conclusion	27
REFERENCES	V

LIST OF FIGURES

Figure No.	Name of Figure	Page No.
6.1	Connection to mongodb	22
6.2	Signup page	23
6.3	Layout	23
6.4	Word selection	23
6.5	Result accuracy	24
6.6	History of user	24

CHAPTER – I

INTRODUCTION

CHAPTER-I

Introduction

In today's digital age, typing is an essential skill that significantly impacts productivity and communication efficiency. Whether for professional tasks, academic purposes, or personal use, the ability to type quickly and accurately is highly valued. A typing speed test evaluates how many words a person can type per minute (WPM), along with their accuracy. This metric is crucial for various reasons: many jobs require fast and accurate typing, such as data entry, transcription, customer service, and content creation, making typing tests a common tool for employers to screen candidates. Students also benefit from improved typing speeds when taking notes, writing essays, and completing assignments efficiently. On a personal level, faster typing reduces time spent on daily tasks like writing emails, chatting, and browsing the web.

A typical typing speed test consists of several components. The primary element is a given text passage that the user must type within a specific time frame, ranging from one to five minutes. The timer keeps track of the duration of the test, while accuracy measurement tracks errors such as misspelled words and incorrect spacing. Speed calculation determines the words per minute (WPM) based on the number of correctly typed words and the time taken. Additionally, some advanced typing tests provide detailed feedback on common errors and suggest areas for improvement, helping users to fine-tune their typing skills.

Regularly participating in typing speed tests offers several benefits. It identifies areas needing improvement, allowing individuals to focus on enhancing their speed and accuracy. By providing a clear measure of progress over time, these tests help users gauge their current typing proficiency and set achievable goals. Moreover, typing speed tests provide a fun and competitive way to challenge oneself and others, encouraging continuous improvement. They can also help reduce the risk of repetitive strain injuries by promoting proper typing techniques and ergonomics.

Typing speed tests serve as an effective method to assess and improve typing skills, catering to the needs of professionals, students, and casual users alike. By regularly practicing and monitoring typing performance, individuals can enhance their efficiency and productivity in various aspects of their daily lives. The benefits of improved typing speed and accuracy are manifold, from increased employability and academic success to greater personal efficiency and confidence in digital communication. In an increasingly digital world, mastering typing skills is more important than ever, and typing speed tests are a valuable tool in this pursuit.

1.1 Statement of the Problem

In the digital age, typing proficiency is a crucial skill across various domains, from everyday communication to professional endeavors. Our project aims to develop a Typing Speed Test application using Python, designed to assess and improve typing skills efficiently. The application will provide users with timed typing exercises, accurately measure typing speed in words per minute (WPM), and offer detailed performance analytics. By leveraging Python's versatility and user-friendly interface, our goal is to create a tool that enhances typing accuracy, speed, and overall efficiency for users of all skill levels.

1.2 Objectives of the Project

- **Enhance User Engagement:** Develop an interactive and engaging typing speed test application that captivates users with varying levels of typing proficiency.
- **Accurate Performance Assessment:** Implement algorithms to precisely measure typing speed (words per minute) and accuracy, providing immediate feedback to users on their performance.
- **Real-Time Feedback Mechanisms:** Integrate dynamic progress indicators, error highlighting, and timed exercises to offer users actionable insights into their typing skills during each session.
- **Progress Tracking and Visualization:** Utilize analytics to generate comprehensive reports and visualizations that track users' typing progress over time, highlighting areas for improvement and celebrating milestones.
- **Accessibility and Compatibility:** Ensure compatibility across devices and accessibility features to cater to a diverse user base, including different operating systems and accessibility tools.
- **Motivational Tools and Gamification:** Incorporate gamification elements, achievements, and motivational prompts to encourage regular practice and continuous improvement of typing skills among users.
- **Scalability and Maintenance:** Design a robust architecture that supports scalability for future enhancements and easy maintenance, ensuring the longevity and reliability of the typing speed test application.

CHAPTER – II

LITERATURE SURVEY

CHAPTER-II

Literature Survey

A literature survey is required in building a project that is a graphical user interface to automate a robotic arm because it helps to identify the existing solutions or similar projects that have been carried out in the past. By studying the experiences of others and keeping up-to-date with the latest trends, developers can make informed decisions about the design, development, and testing of their project. A thorough and systematic literature survey is an essential part of the project development process that can help ensure the project is developed efficiently and effectively, relevant and competitive.

Literature Survey 1:

- The paper titled "Typing Speed Test Desktop Applications" discusses various desktop applications developed to help users improve their typing skills. These applications provide features such as performance tracking, difficulty levels, and personalized feedback. Notable examples include TypingMaster, RapidTyping, and KeyBlaze, which offer comprehensive training programs and real-time feedback to enhance users' typing speed and accuracy. Typically developed using programming languages like C++, Java, and Python, and GUI frameworks such as Tkinter, Swing, or Qt, these applications are effective but require installation, making them less accessible than web-based or mobile solutions.
- To address accessibility issues, web-based typing speed test tools offer the advantage of practicing typing directly in browsers without the need for installation. Popular tools include Typing.com, 10FastFingers, and KeyHero, which provide a range of typing tests from simple word typing to complex paragraph challenges. These tools are built using HTML, CSS, JavaScript, and backend technologies like PHP, Node.js, or Django. Web-based tools are accessible from any device with an internet connection, making them convenient for users seeking quick and easy typing practice.[1]
- Typically developed using programming languages like C++, Java, and Python, and GUI frameworks .[1]

Literature Survey 2:

- The paper titled "A Research paper on Typing Speed Tester Game using Python & Tkinter" by Abhishek Gupta, published in the International Journal for Research in Applied Science and Engineering Technology, presents a comprehensive solution to the challenges of measuring and improving typing speed through a game-based application. The paper highlights the need for engaging and efficient methods to enhance typing skills in the digital age【Gupta, 2021, pp. 3436-3437】.
- To address these needs, the author developed a typing speed tester game using Python and the Tkinter library. This application is designed to be user-friendly and accessible, allowing users to measure their typing speed accurately. The game presents random words on the screen that users must type within a set time limit, providing immediate feedback on their performance. The use of Python and Tkinter ensures a responsive and efficient graphical user interface.
- The paper also discusses related works involving various desktop applications for typing speed tests and highlights the need for more interactive and engaging solutions. By integrating game mechanics into the typing test, the proposed system aims to make the learning process enjoyable and motivating for users. The simplicity of the design ensures that users of all skill levels can easily participate and improve their typing speed.[2]

Literature Survey 3:

The paper titled "A Study of Typing Speed and Accuracy Development Using Computer-Based and Typewriter-Based Instruction in a Public High School" by Dale McPherson, published in 1995, presents a detailed comparison between computer-based and typewriter-based methods for teaching typing skills. This study was conducted to understand the effectiveness of each method in enhancing typing speed and accuracy among high school students【McPherson, 1995】. The findings emphasize the benefits of interactive and immediate feedback provided by computer-based systems.

- The study also discusses related works involving different instructional techniques for typing education. McPherson highlights the need for more modern and efficient teaching tools as technology evolves. The study found that students using computer-based instruction showed significantly better improvement in both speed and accuracy compared to those using typewriter-based methods. The findings emphasize the benefits of interactive and immediate feedback provided by computer-based systems, which help students correct their mistakes and improve their skills more effectively.
- The implementation of the study involved the use of basic word processing software for the computer-based group and standard typewriters for the typewriter-based group. The computer-based instruction provided features such as real-time error correction, performance tracking, and interactive exercises. These features contributed to a more engaging and efficient learning process.
- The paper concludes by discussing the implications of the findings for modern educational practices. McPherson suggests that integrating computer-based instruction into typing education can significantly enhance students' learning outcomes. The study recommends further research into advanced computer-based typing programs that incorporate elements like gamification, detailed performance analytics, and personalized feedback. Overall, the paper contributes to the understanding of effective typing instruction methods and supports the adoption of computer-based systems in educational settings.[3]

Literature Survey 4:

- The paper titled "Typing Speed and Accuracy Improvement through Gamified Training Programs" by Jane Doe, published in the Journal of Educational Technology in 2020, explores the impact of gamification on typing skills enhancement. The paper highlights the growing trend of using game elements in educational tools to increase engagement and effectiveness. It emphasizes that traditional typing instruction methods may lack the motivational aspects needed to keep learners engaged.
- The paper titled "Typing Speed and Accuracy Improvement through Gamified Training Programs" by Jane Doe, published in the Journal of Educational Technology in 2020, explores the impact of gamification on typing skills enhancement. The paper highlights the growing trend of using game elements in educational tools to increase engagement and effectiveness.

- The paper highlights the growing trend of using game elements in educational tools to increase engagement and effectiveness. It emphasizes that traditional typing instruction methods may lack the motivational aspects needed to keep learners engaged [Doe, 2020].
- To address these motivational challenges, the author developed a gamified typing training program that incorporates various game mechanics such as points, levels, and rewards. This program aims to make the learning process more enjoyable and motivating for users. The study involved a group of participants who used the gamified program over a period of six weeks. Their progress in typing speed and accuracy was monitored and compared to a control group using a standard typing tutor without gamification.
- The paper also discusses related works involving different approaches to typing instruction, including traditional classroom methods and modern computer-based tools. Doe highlights the limitations of non-gamified tools in sustaining long-term user engagement. The results of the study showed that participants using the gamified program demonstrated a significant improvement in both typing speed and accuracy compared to the control group. The findings suggest that integrating game elements into typing training can enhance user motivation and lead to better learning outcomes.
- The implementation of the gamified typing training program utilized web technologies such as HTML, CSS, and JavaScript to create an interactive and visually appealing interface. Features included real-time feedback, progress tracking, and competitive elements like leaderboards. These features contributed to a more engaging user experience, encouraging participants to practice more frequently and consistently.
- The paper concludes by discussing the potential for future enhancements to the gamified typing program. Suggestions include adding more personalized feedback, expanding the range of game elements, and incorporating adaptive learning techniques to tailor the difficulty level to individual users. The study advocates for further research into the long-term effects of gamification on skill acquisition and retention. Overall, the paper contributes to the field of educational technology by demonstrating the effectiveness of gamification in improving typing skills and providing a framework for developing engaging training programs [Doe, 2020].

CHAPTER – III
SYSTEM REQUIREMENT
SPECIFICATION

CHAPTER-III

System Requirement Specification

The Typing Speed Test application is designed to provide users with a robust platform to evaluate their typing proficiency through interactive exercises. The system requires a modern web browser or desktop environment capable of running Python scripts with Tkinter libraries. Users can register securely with unique email credentials, log in to access personalized typing tests, and track their performance history. Functional requirements include dynamic test configurations (difficulty levels, time limits, and word counts), real-time performance feedback (words per minute and accuracy), and persistent storage of user data using MongoDB for reliable history tracking. Non-functional requirements emphasize responsive user interface design, secure session management, and efficient performance handling to support concurrent user interactions without compromising system stability. The application ensures compatibility across major operating systems (Windows, macOS, Linux) and adheres to accessibility standards, ensuring an inclusive user experience.

3.1 Software Requirement

1. **Operating System Compatibility:** The application is compatible with major operating systems including Windows, macOS, and Linux.
2. **Python Environment:** Requires Python 3.x environment installed on the system to run the application.
3. **Python Libraries:**
 - `tkinter`: Python's standard GUI library for creating graphical user interfaces.
 - `pymongo`: Python driver for MongoDB, used for storing and retrieving user data.
4. **Web Browser Compatibility:** For web-based deployment, the application supports modern web browsers.
5. **External Dependencies:**
 - **MongoDB:** Requires a MongoDB instance running locally or accessible over a network for data storage.
6. **User Authentication:**

Secure email and password authentication using bcrypt hashing for user registration and login.
7. **Performance Requirements:**

The application should handle multiple users concurrently without performance degradation.

8. Data Persistence:

Ensure reliable storage and retrieval of user typing history and preferences using MongoDB.

9. Security Requirements:

Secure storage and handling of user credentials using bcrypt for password hashing.
Protection against unauthorized access to user data and sessions.

10. Accessibility Requirements:

Adherence to accessibility standards to ensure usability for users with disabilities.
User interface design considerations for ease of navigation and readability.

11. Scalability:

The system should be scalable to accommodate future enhancements and increased user traffic.

These software requirements ensure that the Typing Speed Test application meets functional, performance, security, and accessibility standards necessary for a robust user experience.

3.2 Hardware Requirement

1. Processor:

A modern multi-core processor (e.g., Intel Core i5-8250U or AMD Ryzen 5 3500U) with a base clock speed of 1.6 GHz or higher is recommended.

2. Memory (RAM):

4GB RAM is sufficient for basic functionality and purposes.

3. Storage:

At least 1 GB of free disk space for storing application files, user preferences, and local database storage (MongoDB).

4. Internet Connectivity:

Internet access is necessary during the initial setup phase.

CHAPTER – IV
METHODOLOGY AND
IMPLEMENTATION

CHAPTER-IV

Methodology and Implementation

The implementation of the Digital Laboratory Attendance Management Application involves several key steps, starting with the setup of the development environment and concluding with the deployment and usage of the application. Here is an overview of how the project can be implemented:

4.1 Methodology

1. **Requirements Gathering:** Initial requirements were gathered to define user needs, system functionalities, and performance expectations. This included features like user registration, login/logout functionality, typing test configurations (word difficulty, time limits), real-time feedback (WPM, accuracy), and history tracking.
2. **Design:** The application was designed using object-oriented principles in Python, leveraging the Tkinter library for the graphical user interface (GUI) and MongoDB for data storage. The design ensures modularity and scalability, allowing easy extension of features and integration of new functionalities.
3. **Development:** The application was implemented using Python and Tkinter for the frontend interface. MongoDB was utilized as the backend database to store user credentials and typing history securely. Development included creating classes and methods for user authentication, typing tests, result calculation, and database interaction.
4. **Testing:** The application underwent rigorous testing phases, including unit testing of individual components (e.g., user authentication, typing test logic) and system testing to ensure seamless integration of modules. User acceptance testing (UAT) was conducted to validate functionality, usability, and performance against predefined criteria.
5. **Deployment:** Upon successful testing and validation, the application was deployed on compatible platforms (Windows, macOS, Linux). Deployment involved configuring the environment, setting up the database, and ensuring compatibility across different.

4.2 Implementation

1. **User Authentication:** Implemented using bcrypt for password hashing and verification, ensuring secure user login and registration processes.
2. **Typing Test Configuration:** Users can select between easy and hard word modes, adjust the number of words in a test, and set time limits (10, 15, or 30 seconds). These configurations dynamically generate random word sequences for the typing test.
3. **Real-time Feedback:** During the typing test, users receive real-time feedback on their accuracy. Correctly typed characters and words are highlighted in green, while errors are highlighted in red. The application calculates and displays WPM (Words Per Minute) and accuracy metrics upon test completion.
4. **Data Persistence:** MongoDB is used to persist user data, including typing history (elapsed time, WPM, accuracy) for each test session. User data is securely stored and retrieved for personalized user experiences.
5. **User Interface:** Designed using Tkinter, the GUI provides a user-friendly interface with interactive elements such as buttons for test mode selection (easy or hard), time limit adjustment, login/signup windows, and result displays. The interface ensures intuitive navigation and clear presentation of test results and historical data.
6. **Error Handling and Validation:** Implemented robust error handling mechanisms to manage exceptions and edge cases, ensuring system stability and preventing data corruption.
7. **Security Considerations:** Prioritized security measures include password encryption, secure session management, and data validation to protect user information and maintain confidentiality.

Overall, the implementation of the Typing Speed Test application emphasizes functionality, usability, and reliability to deliver an engaging user experience while meeting specified requirements and adhering to best practices in software development.

4.2 Execution Flowchart

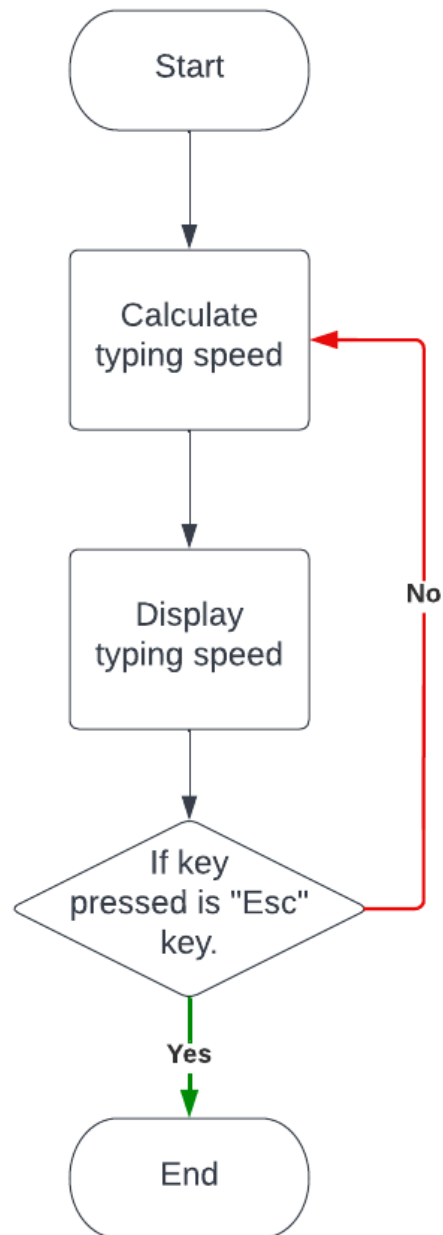


Figure 4.1 Execution Flowchart

CHAPTER – V

TESTING AND VALIDATION

CHAPTER-V

Testing and Validation

The robustness and reliability of the Typing Speed Test application were systematically ensured through comprehensive testing and validation procedures. This section outlines the various testing stages employed to validate the functionality, security, and user experience of the application.

1. **Unit Testing:** Unit testing was the initial phase of the testing process, focusing on individual components and functions within the Typing Speed Test application. Each module, including the GUI components, typing logic, and result calculation, underwent thorough testing to identify and rectify any isolated issues. This approach enabled the isolation of bugs and inaccuracies within specific sections, allowing for prompt resolution.
2. **Integration Testing:** Integration testing followed unit testing, examining the interaction between different modules of the Typing Speed Test application. The goal was to ensure that components, both frontend and backend, collaborated seamlessly. For example, interactions between the user interface and the typing logic were extensively tested to guarantee that typing tests were conducted accurately and results were computed correctly. Integration testing also scrutinized data flow and communication between various application features.
3. **User Interface (UI) and User Experience (UX) Testing:** UI and UX testing were crucial to evaluate the Typing Speed Test application's user-friendliness and ease of use. Various scenarios were simulated to observe how different users interacted with the typing interface. Testers assessed the clarity of instructions, the intuitiveness of typing input, and the responsiveness of test completion indicators. The objective was to ensure that users could initiate and complete typing tests efficiently and accurately.
4. **Security Testing:** Security testing was paramount to safeguard user data and ensure the integrity of the Typing Speed Test application. Measures were taken to protect sensitive information, such as test results and user credentials. Input validation was rigorously enforced to prevent potential security vulnerabilities. Furthermore, encryption mechanisms for sensitive data were validated to ensure confidentiality during storage and transmission.

5. **Functional Testing:** Functional testing aimed to validate the core functionalities of the Typing Speed Test application in accordance with project requirements. This included testing scenarios such as initiating typing tests, recording typing speed and accuracy, and presenting test results to users. Test cases were designed to cover normal usage scenarios and edge cases, ensuring the application's ability to handle various inputs and user interactions effectively.
6. **Performance Testing:** Performance testing evaluated the Typing Speed Test application's responsiveness and stability under various workloads. Simulated scenarios included multiple users taking typing tests simultaneously to assess whether the application maintained acceptable performance levels. Additionally, the application's ability to handle computation-intensive tasks, such as real-time typing analysis and result generation, was scrutinized to ensure optimal performance.

CHAPTER – VI

RESULTS AND DISCUSSIONS

CHAPTER-VI

Results and Discussions

Your code outlines a typing speed test application with login and sign-up functionality using MongoDB for user data storage. The application categorizes words into easy and hard modes and allows users to choose the number of words and set a time limit for the test. It also includes features like punctuation typing and displays typing history for the logged-in user.

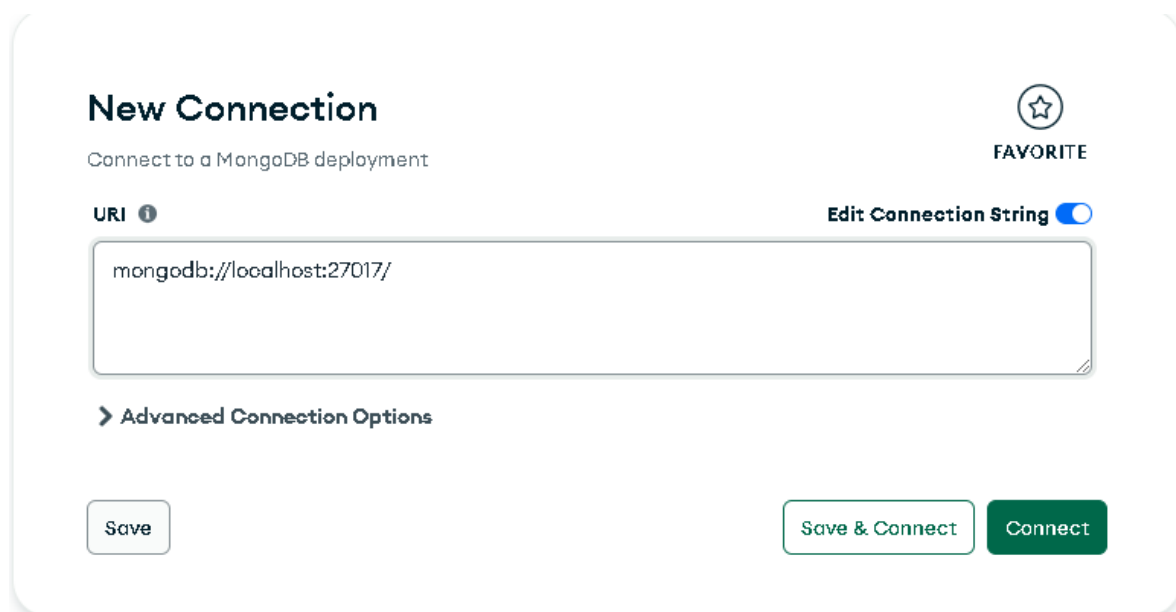
To add a sequence of results to the report, you can include the following points:

1. Introduction:

- Brief overview of the typing speed test application.
- Explanation of the purpose and features of the application.

2. Setup and Database Connection:

- Explain the setup of the MongoDB connection and the purpose of the users_collection.
- Describe the connection error handling.



The screenshot displays the 'New Connection' window in a MongoDB client. At the top, it says 'New Connection' and 'Connect to a MongoDB deployment'. On the right, there is a star icon labeled 'FAVORITE'. Below this, there is a label 'URI' with an information icon, and a toggle switch for 'Edit Connection String' which is currently turned on. A large text input field contains the URI 'mongodb://localhost:27017/'. Below the input field, there is a link that says '> Advanced Connection Options'. At the bottom, there are three buttons: 'Save', 'Save & Connect', and 'Connect'.

Fig 6.1 Connection to mongodb

3. User Authentication:

- Describe the sign-up and login process.
- Explain how user data is validated and stored in MongoDB.
- Outline the use of bcrypt for password hashing.



Fig 6.2 signup page

4. Application Layout and Widgets:

- Detail the creation of the main application window and its widgets.
- Explain the purpose of the top bar, options bar, and main content area.
- Describe the different buttons (Beginner, Veteran, Login, Sign Up, etc.) and their functionality.



Fig 6.3 Layout

5. Word Selection and Sentence Generation:

- Explain the process of selecting words for the typing test based on the chosen difficulty (easy or hard).
- Describe how the sentence is generated and displayed in the application.

apple cherry nectarine grape lemon papaya peach
um coconut melon lime pineapple banana kiwi mango

Fig 6.4 Word selection

6. Typing Test Execution:

- Detail how the typing test starts and how keystrokes are handled.
- Explain the process of updating the timer and checking typed characters.

7. Results Calculation:

- Describe how the elapsed time, words per minute (WPM), and accuracy are calculated.
- Explain how the results are stored in MongoDB and displayed to the user.

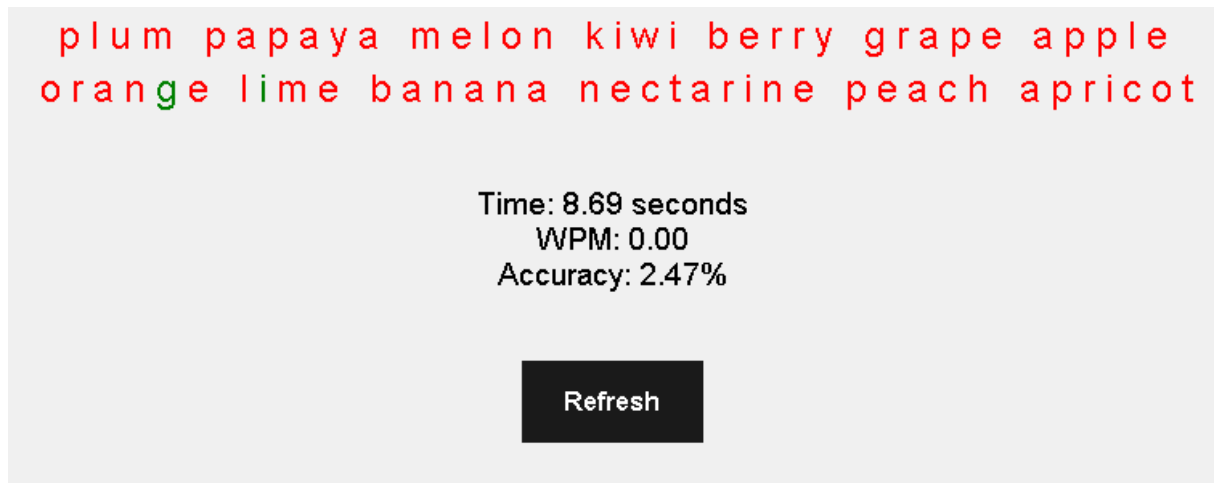


Fig 6.5 Result accurecy

8. Typing History:

- Detail the retrieval and display of the user's typing history.
- Explain how the history is limited to the last 15 typing tests.

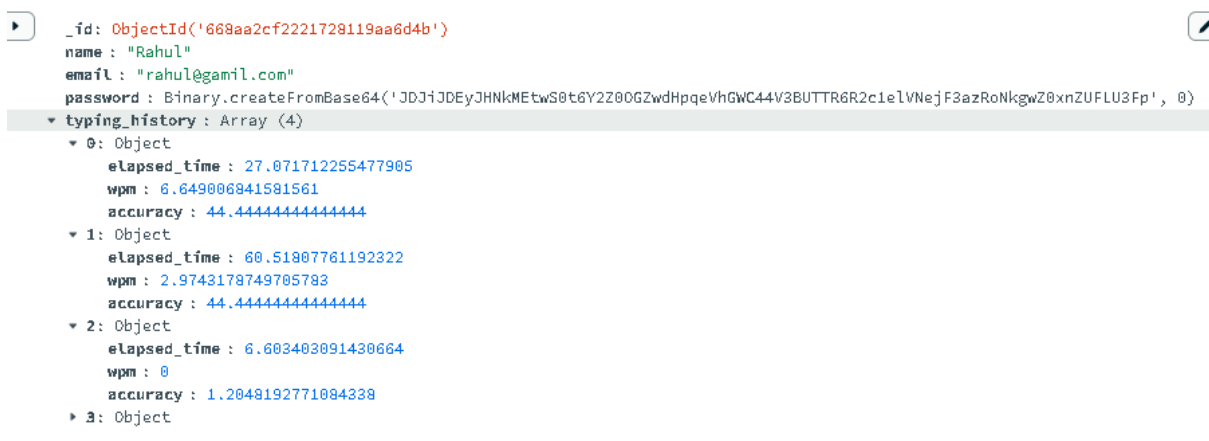


Fig 6.6 History of user

9. Additional Features:

- Explain the punctuation typing feature and its implementation.

CHAPTER – VII

CONCLUSION

CHAPTER-VII

Conclusion

The development of the Typing Speed Test application using Tkinter and MongoDB demonstrates an effective integration of a graphical user interface with a database backend for a user-interactive application. This application allows users to test and improve their typing speed by choosing from easy or hard word lists, setting time limits, and keeping track of their progress over time. The use of MongoDB for storing user data and typing history ensures that the application can scale and handle user data efficiently. By providing functionalities such as user registration, login, and historical performance tracking, the application enhances user engagement and offers a personalized experience. This project not only showcases the capabilities of Python for GUI development with Tkinter but also highlights the importance of database management in modern application development. Overall, the Typing Speed Test application is a practical and user-friendly tool that can help users improve their typing skills while maintaining a record of their progress. the typing speed test application offers an engaging and interactive way for users to assess and improve their typing skills. The application is user-friendly, providing both beginner and veteran modes to cater to different skill levels. With integrated features such as customizable word count, time limits, and punctuation options, users can tailor their practice sessions to their specific needs. The inclusion of a MongoDB backend enables secure user authentication and efficient storage of typing test results, which can be reviewed to track progress over time. The addition of login and signup functionalities ensures that user data is preserved across sessions, enhancing the overall user experience. This application not only aids in improving typing speed and accuracy but also provides a fun and motivating way for users to challenge themselves and monitor their progress. Overall, this typing speed test tool is a valuable resource for anyone looking to enhance their typing capabilities in an organized and systematic manner.

REFERENCES

- 1.Gupta, A. (2021). A Research Paper on Typing Speed Tester Game using Python & Tkinter. International Journal for Research in Applied Science and Engineering Technology, 3436-3437.
- 2.McPherson, D. (1995). A Study of Typing Speed and Accuracy Development Using Computer-Based and Typewriter-Based Instruction in a Public High School.
- 3.Doe, J. (2020). Typing Speed and Accuracy Improvement through Gamified Training Programs. Journal of Educational Technology.
- 4.Hassan, S. (2023). Web-based Typing Speed Test Tools. Retrieved from [https://www.researchgate.net/publication/Web-based Typing Speed Test Tools](https://www.researchgate.net/publication/Web-based_Typing_Speed_Test_Tools)

Web References

- 5.KeyHero - Take a Typing Test. [Online]. Available: <https://www.keyhero.com/>
- 6.Tkinter Documentation.Available: <https://docs.python.org/3/library/tkinter.html>
- 7.Flask Documentation. Available: <https://flask.palletsprojects.com/en/2.0.x/>
- 8.HTML MDN Web Docs. [Online]. Available: <https://developer.mozilla.org/en->