

MINOR PROJECT

ON

TYPING TEST Website

COMPUTER ENGINEERING

2022 – 2025

Submitted to:-
Mr. Jatin Verma

Submitted by:-
Saurabh Kumar (10622560)



**Delhi Skill and
Entrepreneurship University**
UGC Recognized

**GURU NANAK DEV INSTITUTE OF
TECHNOLOGY**

Rohini Sector 15 Delhi 110089

Acknowledgement

I express my heartfelt gratitude to everyone who contributed to the successful completion of this project, "**Typing Speed Test Application.**"

First and foremost, I would like to thank my institution and the department for providing me with the opportunity to work on this project and for offering the necessary guidance and resources.

I am deeply indebted to my mentor/guide, **Mr. Jatin Verma**, whose continuous support, expertise, and valuable feedback were pivotal in shaping this project. Your guidance throughout every phase of this work has been invaluable.

I would also like to thank my friends and peers for their encouragement and support. Their constructive criticism and feedback helped me enhance the quality of this project.

This project has been a valuable learning experience, and I am truly thankful to everyone who helped me along the way.

Name: Saurabh Kumar (10622560)

Declaration

I hereby declare that the project titled "Typing Speed Test Application" submitted as part of my minor project work is a result of my independent effort and research. The work presented here is original and has not been submitted in any form to any other institution or organization for any purpose.

I have developed this project entirely on my own, using resources and tools available in the public domain. Any references or external materials used in this project have been properly acknowledged.

I take full responsibility for the content of this project and ensure that it meets the ethical and academic standards required for its submission.

Saurabh kumar(10622560)

Abstract:

The Typing Speed Test web application is designed to test the typing speed of users, calculate accuracy, and generate a downloadable PDF certificate with the results. Built with HTML, CSS, and JavaScript, this tool enables users to test their typing ability within a customizable time frame. The system highlights typed words dynamically, provides instant feedback, and allows users to download a professional certificate summarizing their performance.

Table of Contents:

1. Introduction
2. Objectives
3. Tools and Technologies
4. Features
5. Code Walkthrough
 - HTML
 - CSS
 - JavaScript
6. Working Flow of the Application
7. Testing and Debugging
8. Results
9. Conclusion
10. Future Scope
11. Images of Website
12. Coding

1. Introduction

The **Typing Speed Test Application** is a web tool designed to measure a user's typing speed in **words per minute (WPM)** and typing accuracy. Users can enter their name, select a timer (60, 90, or 120 seconds), and start typing the given sample text.

The application provides real-time feedback by highlighting correct words and leaves incorrect ones unhighlighted. After the test, it displays the results, including WPM, accuracy, and mistakes. Users can also download a **personalized certificate** in PDF format.

Built with **HTML, CSS, and JavaScript**, this tool is simple, efficient, and user-friendly.

2. Objectives

The **Typing Speed Test** project aims to achieve the following key objectives:

1. **Provide an Interactive Typing Test Tool:**

The primary goal of this application is to offer users an engaging and user-friendly platform to measure their typing speed. The tool is designed to be easy to use, with minimal setup required. Users simply input their name, choose a timer duration, and start typing a given text. The application tracks and evaluates their performance in real-time.

2. **Calculate Typing Speed and Accuracy:**

The application calculates two essential metrics of typing performance:

- **Typing Speed (WPM):** This measures how many words the user can type per minute, giving a clear indication of their typing efficiency.
- **Accuracy:** The application calculates the accuracy based on the proportion of correctly typed words compared to the total words typed. This helps users understand not only how fast they type but also how accurately they type.

3. **Highlight Mistakes and Correct Words in Real-Time:**

The application provides real-time feedback while the user is typing. Correctly typed words are highlighted, allowing users to track their progress. On the other hand, incorrect words are not highlighted, which helps users identify areas where they need improvement. This

interactive feedback mechanism encourages users to type more accurately and efficiently.

4. **Generate a Downloadable Performance Certificate:**

Upon completing the typing test, the application generates a **personalized certificate** in the form of a PDF. This certificate includes the user's name, typing speed (WPM), accuracy, and the number of mistakes made during the test. The downloadable certificate serves as a record of the user's typing performance, which can be saved or printed for personal use or shared as proof of typing proficiency.

Through these objectives, the Typing Speed Test project aims to offer users a comprehensive tool for both self-assessment and improvement of their typing skills, while also providing motivation through a certificate of achievement.

3. Tools and Technologies

1. **HTML (HyperText Markup Language):**

HTML is used to create the structure and content of the application. It defines elements like input fields, buttons, and the text area for typing. The application is built around a clean, simple layout that allows users to input their name, select a timer, and start typing the provided text. HTML also includes the results display, where the user's typing speed and accuracy are shown after completing the test. It forms the backbone of the user interface, ensuring accessibility and basic functionality.

2. **CSS (Cascading Style Sheets):**

CSS is responsible for the styling and visual presentation of the application. It makes the page visually appealing and user-friendly by defining the layout, colors, fonts, and positioning of elements. Key features like centering the content on the screen, styling buttons, and text areas, and providing hover effects are handled by CSS. It also helps in adding dynamic effects, such as highlighting correct or incorrect words in real-time as the user types. Overall, CSS ensures the application is aesthetically pleasing and easy to interact with.

3. **JavaScript:**

JavaScript is the core programming language that drives the functionality of the application. It handles the timer logic, starting and stopping the countdown based on the selected time. It also validates the user's input by comparing it to the original text, highlighting correct words and showing mistakes. JavaScript is responsible for calculating the user's typing speed (WPM), accuracy, and mistakes, and then displaying the results. Additionally, JavaScript is used to generate a downloadable certificate containing the user's performance metrics after completing the test.

4. **jsPDF Library:**

The **jsPDF** library is used to generate a downloadable PDF certificate that records the user's performance in the typing test. It allows the creation of a customized PDF document with text, such as the user's name, words per minute (WPM), accuracy, and mistakes. The library provides features to format text, adjust font sizes, and position content in the PDF, ensuring that the certificate is professional and easy to read. Once the test is completed, the user can download the certificate as a PDF file for future reference or sharing.

4. Features

1. **Custom Timer:**

The application offers flexible timer options, allowing users to choose the duration for their typing test. They can select between 60, 90, or 120 seconds based on their preference. This feature caters to users of different skill levels, enabling them to test their speed within a comfortable time frame. The custom timer ensures that the test is personalized and provides a more tailored experience. It also adjusts the results based on the time spent typing, giving accurate performance metrics.

2. **Word Highlighting:**

As users type, the application highlights the words they have typed correctly in real-time. This provides immediate visual feedback, allowing users to see their progress. Incorrectly typed words are displayed normally, with no highlighting. This helps users easily identify mistakes and focus on correcting them. The feature also serves as a motivational tool, as users can track their accuracy while typing. The dynamic highlighting makes the typing test more interactive and engaging for the user.

3. **Real-Time Feedback:**

The application provides instant feedback on the user's performance during the typing test. As the user types, the system compares the input against the original text and visually indicates correct and incorrect words. This real-time feedback helps the user to adjust their typing speed and accuracy as they progress. The feedback is immediate and helps users improve by identifying areas where they made mistakes. This feature enhances the learning experience and provides clear insights into the user's typing abilities.

4. **Dynamic Timer:**

The dynamic timer starts as soon as the user begins typing, providing a countdown based on the selected timer option (60, 90, or 120 seconds). The timer runs continuously until the time is up or the user finishes typing the entire text. This feature ensures that the test is timed accurately, adding a sense of urgency and focus. It helps users manage their time effectively, pushing them to maintain a consistent speed throughout the test. The timer display updates in real-time, providing the user with an ongoing sense of how much time is remaining.

5. **Certificate Generation:**

Once the typing test is completed, users can download a personalized certificate that summarizes their performance. The certificate includes key details like typing speed (WPM), accuracy, and the number of mistakes made during the test. The jsPDF library generates the certificate in a professional format, allowing users to save or print it for their records. This feature adds a sense of achievement, as users can keep track of their typing progress. It also serves as a motivational tool for users to improve their typing skills over time.

5. Code Walkthrough

HTML

The HTML structure of the project is simple and effective:

- **Text Input (`textarea`):** For typing the test sentence.
- **Buttons:**
 - Start button to begin the test.
 - A hidden button to download the certificate after the test ends.
- **Timer and User Info:** Displays the time remaining, user's name, and test results.

CSS

The CSS file is responsible for the aesthetics:

- **Centered Layout:** The `.centerDiv` ensures the content is centered both horizontally and vertically.
- **Text Area Styling:** The text area for typing is styled with a light green background, rounded corners, and a border to make it visually appealing.
- **Button Styling:** Styled buttons with hover effects to improve user interaction.
- **Highlighting Text:** The `.highlighted` class is used to highlight correct words in sky blue, making the typing experience more interactive.

JavaScript

JavaScript handles the core functionality:

- **playGame()**: Initializes the test by setting the text, enabling the text area, and starting the timer.
- **startTimer()**: This function starts the countdown timer and updates the display every second.
- **updateTyping()**: It checks each typed word against the original text, dynamically highlighting the correct ones and removing highlights from the incorrect ones.
- **endGame()**: Ends the test when the time runs out or when the user finishes typing all the words.
- **generatePDF()**: Uses the jsPDF library to generate a certificate with the user's name, the correct words typed, and the test results.

6. Working Flow of the Application

1. **User Inputs Name:** The user enters their name and selects the timer duration.
2. **Test Starts:** When the "Start" button is clicked, the test begins, and the timer starts counting down.
3. **Typing Validation:** As the user types, the application compares the input with the provided text and highlights the correct words.
4. **Timer Countdown:** The countdown timer is updated every second.
5. **End of Test:** When the time runs out or the user completes typing the full text, the test ends.
6. **Result Display:** The user sees their results: correct words, mistakes, and typing speed (WPM).
7. **Certificate:** The "Download Certificate" button becomes visible, allowing the user to download their certificate in PDF format.

7. Testing and Debugging

1. **Functional Testing:**

Functional testing ensured that all key features, such as the timer, typing validation, and result calculation, worked correctly. The timer started once the user began typing and stopped at the right time. Validation of typed words was accurate, highlighting correct and incorrect words. The result calculation correctly determined words per minute (WPM), accuracy, and mistakes. Edge cases like fast or slow typing were also tested to confirm the system's robustness.

2. **Performance Testing:**

Performance testing was performed across various devices to ensure the application worked smoothly on desktops, tablets, and smartphones. The application was tested on different browsers to ensure compatibility, including Chrome, Firefox, Safari, and Edge. The loading time was checked to ensure quick initialization and smooth performance. The timer and result generation were tested for any delays in processing. Ensuring the app runs seamlessly on all devices helped improve the user experience.

3. **Debugging:**

Debugging involved resolving errors, such as ensuring the test doesn't start without a username. The username input was validated to ensure the test only begins when a name is provided. Additionally, checks were added to ensure the timer couldn't start without selecting a time duration. The result generation, including the PDF certificate, was debugged to ensure accuracy in performance metrics. Debugging helped resolve issues quickly, ensuring the application functioned as expected.

8. Results

The Typing Speed Test application effectively performs as designed, offering a seamless user experience. Once the user completes the test, the application accurately calculates the typing speed in words per minute (WPM), alongside the number of correct words and mistakes made. The results are displayed on the screen, providing users with a clear overview of their typing performance. The downloadable certificate generated also reflects these results accurately, showing typing speed, accuracy, and mistakes. These results give users an objective measure of their typing skills, and they can use them to track improvements over time. The overall functionality of the application ensures users receive valid and accurate feedback.

9. Conclusion

The Typing Speed Test application is an effective tool for individuals looking to assess and enhance their typing skills. It provides a straightforward user interface and delivers accurate results, making it easy for users to measure their typing speed and accuracy. The feature of downloading a certificate based on the results makes the experience more engaging and rewarding. Whether for personal improvement or as a fun challenge, this application is a valuable resource. It not only helps users evaluate their current typing abilities but also motivates them to practice and improve over time. Overall, the project successfully fulfills its objectives and adds value for anyone looking to become a better typist.

10. Future Scope

The Typing Speed Test application has a lot of potential for future expansion and enhancement.

1. **Add Multiple Texts:**

In the future, users could be given a choice of multiple texts to type, ranging from simple to complex. This would allow users to challenge themselves further, as they could select texts based on difficulty or topics of interest, making the test more versatile and engaging.

2. **Include Leaderboards:**

To encourage competition and interaction, a leaderboard could be integrated. This would allow users to compare their results with others, adding a social element to the test. Leaderboards could rank users based on their typing speed or accuracy, motivating them to improve and achieve higher rankings.

3. **Multilingual Support:**

Adding multilingual support would make the application accessible to a broader audience. By offering typing tests in various languages, users from different regions could participate. This feature would increase the utility of the tool for people worldwide, helping them improve typing skills in their native or other languages.

11. Images of Website

Typing Speed Test

Enter Your Name:

Select Timer: 60 Sec ▼

Click 'Start' and type the text that appears above...

Start

Time: 0:00

Typing Speed Test

Enter Your Name:

Select Timer: 60 Sec ▼

After all, you're only an immortal until someone manages to kill you. We were like deaf people trying to dance to a beat we couldn't hear. Time plays like an accordion in the way it can stretch out and compress itself in a thousand melodic ways. Life is beautiful, as long as it consumes you. When it is rushing through you, destroying you, life is glorious. As long as people have been on this earth, the moon has been a mystery to us.

Click 'Start' and type the text that appears above...

Done

Time: 0:00

Typing Speed Test

Enter Your Name:

Select Timer: 90 Sec ▼

You typed 0 words correctly with 1 mistakes.

Click 'start' and type the text that appears above...

Start

Download Certificate

Time: 0:00

12. Coding

→ index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Typing Speed Test</title>

<link rel="stylesheet" href="style.css">

<script src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.5.1/jspdf.umd.min.js"></script>

</head>

<body>

<div class="mainDiv">

<div class="centerDiv">

<h1>Typing Speed Test</h1>

<label for="username">Enter Your Name: </label>

**<input type="text" id="username" placeholder="Your Name">

**

<div>

<label for="timeSelect">Select Timer: </label>

<select id="timeSelect">

<option value="60">60 Sec</option>

<option value="90">90 Sec</option>

<option value="120">120 Sec</option>

</select>

</div>

<h2 id="msg"></h2>

<textarea id="mywords" cols="100" rows="10" placeholder="Click 'Start' and type the text that appears above..."></textarea>

<button id="btn" class="mainBtn">Start</button>

```
<br>
<button id="downloadCertificate" class="hiddenBtn">Download
Certificate</button>
<p id="timerDisplay">Time: 0:00</p><br>
</div>
</div>

<script src="script.js"></script>
</body>

</html>
```

→ **styles.css**

/* Resetting margin, padding, and setting box-sizing */

```
* {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
}
```

/* Main container styling */

```
.mainDiv {  
  width: 100%;  
  height: 100vh;  
  position: relative;  
  background-color: rgb(255, 206, 187);  
}
```

/* Center content using absolute positioning */

```
.centerDiv {  
  position: absolute;  
  top: 50%;  
  left: 50%;  
  transform: translate(-50%, -50%);  
  text-align: center;  
}
```

/* Styling for main heading */

```
h1 {  
  text-transform: capitalize;  
  font-size: 2.5rem;  
  margin-bottom: 30px;  
  text-shadow: 1px 2px 3px #a8e9ff;  
}
```

/* Styling for secondary heading */

```
h2 {  
  margin-bottom: 25px;  
  user-select: none;  
  font-size: 1.2rem;  
}
```

```
/* Text area styling */
textarea {
  width: 100%;
  height: 300px;
  background: #e6ffa5;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
  border-radius: 10px;
  border: 3px solid #f30909;
  padding: 10px;
  font-size: 1.5rem;
  margin-bottom: 20px;
  outline: none;
  overflow: hidden; /* Hide scrollbars if text overflows */
}
```

```
/* Styling for main button */
.mainBtn {
  border-radius: 40px;
  padding: 10px 40px;
  font-family: 'Arial Narrow Bold', sans-serif;
  font-size: 1.1rem;
  background: #fdb3e3;
  border: 5px solid #0dddfa7;
  cursor: pointer;
  margin-top: 10px;
}
```

```
/* Styling for hidden button */
.hiddenBtn {
  display: none; /* Hidden by default */
  margin-top: 15px;
  border-radius: 20px;
  padding: 10px 20px;
  font-size: 1rem;
  background: #27ae60;
  border: none;
  color: white;
  cursor: pointer;
}
```

```
/* Show hidden button when the 'show' class is added */
```



```
.hiddenBtn.show {  
  display: inline-block;  
}
```

```
/* Timer display styling */  
#timerDisplay {  
  font-size: 1.5rem;  
  margin-top: 20px;  
  color: #000000;  
}
```

```
/* Highlighted text styling */  
.highlighted {  
  background-color: skyblue;  
  transition: background-color 0.1s ease;  
}
```

→ **script.js**

```
const sentences = [
```

```
  "After all, you're only an immortal until someone manages to kill you. We were like  
  deaf people trying to dance to a beat we couldn't hear. Time plays like an accordion in  
  the way it can stretch out and compress itself in a thousand melodic ways. Life is  
  beautiful, as long as it consumes you. When it is rushing through you, destroying you,  
  life is glorious. As long as people have been on this earth, the moon has been a mystery  
  to us.",
```

```
];
```

```
const msg = document.getElementById('msg');  
const typedWords = document.getElementById('mywords');  
const btn = document.getElementById('btn');  
const downloadBtn = document.getElementById('downloadCertificate');  
const userNameInput = document.getElementById('username');  
const timeSelect = document.getElementById('timeSelect');  
const timerDisplay = document.getElementById('timerDisplay');
```

```
let startTime, originalText, userName, timerInterval;  
let isTyping = false;  
let totalTime = 0; // Total time in seconds
```

```
const playGame = () => {  
  userName = userNameInput.value.trim();  
  if (!userName) return alert("Please enter your name!");  
  
  originalText = sentences[0].split(" ");  
  msg.innerHTML = originalText.map(word => `<span>${word}</span>`).join(" ");  
  typedWords.value = "";  
  typedWords.disabled = false;  
  btn.innerText = "Done";  
  timerDisplay.innerText = `Time: 0:00`;  
  
  totalTime = parseInt(timeSelect.value);  
  
  typedWords.addEventListener('input', startTimerOnce);  
}
```

```
const startTimerOnce = () => {
```

```

    if (!isTyping) {
      isTyping = true;
      startTimer();
      typedWords.removeEventListener('input', startTimerOnce);
    }
    updateTyping();
  }

  const startTimer = () => {
    let elapsed = 0;
    timerInterval = setInterval(() => {
      elapsed++;
      const remainingTime = totalTime - elapsed;
      timerDisplay.innerText = `Time: ${Math.floor(remainingTime / 60)}:${('0' +
        (remainingTime % 60)).slice(-2)}`;

      if (remainingTime <= 0 || isTextComplete()) {
        endGame();
      }
    }, 1000);
  }

  const endGame = () => {
    clearInterval(timerInterval);
    const typedText = typedWords.value.trim().split(/\s+/);
    let correctWords = 0, mistakes = 0;

    typedText.forEach((word, index) => {
      if (word === originalText[index]) {
        correctWords++;
      } else {
        mistakes++;
      }
    });

    msg.innerText = `You typed ${correctWords} words correctly with ${mistakes}
mistakes.`;
    typedWords.disabled = true;
    btn.innerText = "Start";
    downloadBtn.classList.add('show');
    downloadBtn.onclick = () => generatePDF(correctWords, mistakes);
  }

```

```

    isTyping = false;
  }

  const isTextComplete = () => {
    return typedWords.value.trim() === originalText.join(" ");
  }

  const updateTyping = () => {
    const typedText = typedWords.value.trim().split(/\s+/);
    const spans = msg.querySelectorAll('span');

    typedText.forEach((word, index) => {
      if (spans[index]) {
        if (word === originalText[index]) {
          spans[index].classList.add('highlighted');
        } else if (word) {
          spans[index].classList.remove('highlighted');
        }
      }
    });

    if (isTextComplete()) {
      endGame();
    }
  }

  btn.onclick = () => btn.innerText === 'Start' ? playGame() : endGame();

  /*const generatePDF = (correctWords, mistakes) => {
    const { jsPDF } = window.jspdf;
    const doc = new jsPDF();
    doc.setFontSize(22).text('Typing Test Certificate', 20, 30);
    doc.setFontSize(16).text(`Name: ${userName}`, 20, 50)
      .text(`Correct Words: ${correctWords}`, 20, 70)
      .text(`Mistakes: ${mistakes}`, 20, 90)
      .text(`Total Time: ${timeSelect.value / 60} minutes`, 20, 110);
    doc.save('TypingTestCertificate.pdf');
  }*/

  const generatePDF = (correctWords, mistakes) => {
    const { jsPDF } = window.jspdf;
    const doc = new jsPDF();

```

```
// Header
doc.setFont("Helvetica", "bold");
doc.setFontSize(14);
doc.text("Typing Test completed | " + new Date().toLocaleDateString() + " | Typing Master", 20, 20);
```

```
// Title
doc.setFontSize(16);
doc.text("TYPING TEST - " + (correctWords / originalText.length >= 0.75 ? "PASSED" : "NOT PASSED"), 20, 30);
```

```
// User Info
doc.setFont("Helvetica", "normal");
doc.setFontSize(12);
doc.text(`User: ${userName}`, 20, 40);
doc.text("Test name: Custom Typing Test", 20, 50);
doc.text("Date: " + new Date().toLocaleString(), 20, 60);
```

```
// Test Results
doc.setFont("Helvetica", "bold");
doc.text("TEST RESULTS", 20, 80);
doc.setFont("Helvetica", "normal");
const grossSpeed = Math.round((correctWords + mistakes) / (totalTime / 60));
const netSpeed = Math.round(correctWords / (totalTime / 60));
const accuracy = Math.round((correctWords / (correctWords + mistakes)) * 100);
```

```
doc.text(`Duration: ${totalTime / 60} min. of total ${totalTime / 60} min.`, 20, 90);
doc.text(`Gross speed: ${grossSpeed} wpm`, 20, 100);
doc.text(`Net speed: ${netSpeed} wpm`, 20, 110);
doc.text(`Accuracy: ${accuracy}%`, 20, 120);
doc.text(`Correct Words: ${correctWords}`, 20, 130);
doc.text(`Mistakes: ${mistakes}`, 20, 140);
```

```
// Footer
doc.setFont("Helvetica", "italic");
doc.text("Generated using Typing Speed Test App", 20, 160);
```

```
doc.save("TypingTestCertificate.pdf");
};
```