A FIELD PROJECT REPORT ON

**TYPING SPEED TEST**

## Submitted

*In partial fulfilment of the requirements for the award of the degree*

**BACHELOR OF TECHNOLOGY**

**In**

## COMPUTER SCIENCE and ENGINEERING

By

|  |  |
| --- | --- |
| KUMARI | (231FA04908) |
| GANESH MULLURI | (231FA04955) |
| RAHUL DAS | (231FA04972) |
| SOMANADH | (231FA04974) |
|  |  |

Under the Guidance of

## MR. T. NARASIMHA RAO

## Assistant Professor, CSE



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**SCHOOL OF COMPUTING AND INFORMATICS**

# VIGNAN'S FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH

# (Deemed to be University)

**Vadlamudi, Guntur -522213, INDIA.**

**MARCH, 2025**



**CERTIFICATE**

This is to certify that the field project entitled “TYPING SPEED TEST” being submitted by (KUMARI ,231FA04908), (GANESH ,231FA04955), (RAHUL DAS ,231FA04972), and (SOMANADH ,231FA04974) in partial fulfilment of requirements for the degree of **Bachelor of Technology (B.Tech.) in Computer Science and Engineering** at the Department of Computer Science and Engineering, Vignan’s Foundation for Science, Technology and Research (Deemed to be University), Vadlamudi, Guntur District, Andhra Pradesh, India.

This is a bonafide work carried out by the aforementioned students under my guidance and supervision.

|  |  |
| --- | --- |
|  | **Guide** |

**Project Review Committee**

**HoD, CSE**



**DECLARATION**

**Date:**

We hereby declare that the work presented in the field project titled “TYPING SPEED TEST” is the result of our own efforts and investigations.

This project is being submitted under the supervision of **MR. T. Narasimha Rao, Assistant Professor, Department of Computer Science and Engineering** in partial fulfillment of the requirements for the Bachelor of Technology (B.Tech.) degree in Computer Science and Engineering at the Department of Computer Science and Engineering, Vignan’s Foundation for Science, Technology and Research (Deemed to be University), Vadlamudi, Guntur, Andhra Pradesh, India.

|  |  |  |
| --- | --- | --- |
| KUMARI | (231FA04908) | Signature |
| GANESH | (231FA04955) | Signature |
| RAHUL DAS | (231FA04972) | Signature |
| SOMANADH | (231FA04974) | Signature |

**Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| **1**  **1.1)**  **1.2)**  **1.3)**  **1.4)** | **Introduction**  **Problem Definition**  **Existing System**  **Proposed System**  **Literature Review** | **5-6** |
| **2**  **2.1)**  **2.2)** | **System Requirements**  **Hardware and Software Requirements**  **Software Requirements (SRS)** | **7-8** |
| **3**  **3.1)**  **3.2)** | **System Design**  **Modules of System**  **UML Diagrams** | **9-12** |
| **4**  **4.1)**  **4.2)** | **Implementation**  **Sample Code**  **Test Cases** | **13-30** |
| **5**  **5.1)** | **Results**  **Output Screens** | **31-33** |
| **6** | **Conclusion** | **34** |
| **7** | **References** | **35** |

**1.Introduction**

Typing speed is very important in web productivity. A Typing Speed Test measures words per minute (WPM) accurately. The objective of this project is to create an interactive computer program where the user can practice and test their typing ability.

The program provides example text, tracks speed, accuracy, and error in real time and offers improvement feedback. Error detection, countdown timer, and scoring features make it an intriguing tool for students, professionals, and anyone who wants to enhance their typing efficiency.

**1.1 Problem Definition: -**

With the arrival of the digital age, typing accuracy and speed are necessary for productivity and communication. However, most individuals have low typing speed and are constantly in error, which makes them inefficient. The current tools lack real-time feedback, history of errors, and personalized improvement tools.

This project will create an interactive Typing Speed Test application, which computes WPM and accuracy and gives feedback in real-time. With error checking in real-time, countdown timers, and scoring features, the application will effectively improve users' typing skills.

**1.2 Existing System: -**

This project provides a valuable learning experience in building interactive web applications. The typing speed test will be implemented with the core technologies of HTML, CSS, and JavaScript. The existing system will have the following key features:

- A passage of text that users will type.

- A countdown timer that measures the duration of the test. - Real-time tracking of typing speed (words per minute) and accuracy.

- A final score display that shows both speed and accuracy once the test is completed.

**1.3 Proposed System: -**

To overcome the issues of slow typing speeds and frequent errors, this project aims to create an interactive Typing Speed Test program. The system will display practice text to users and will track their typing speed in words per minute (WPM), verifying accuracy and errors in real time.

**Key Features:**

**Real-Time Typing Test:** Tracks WPM, accuracy, and mistakes while the typist’s type.

**Error Detection and Feedback:** This phase identifies errors and gives immediate feedback.

**Countdown Timer:** Includes a time limit to enhance speed and concentration.

**Score Tracking:** Tracks user performance for analysis of progress.

The intuitive interface promises a comfortable and smooth typing experience. By integrating these features, the system will help users improve their typing speed by continuous practice and feedback, thus serving as an effective tool for students, professionals, and anyone who wants to make their typing better.

**1.4 Literature Review: -**

Typing Speed Test project commenced with the vision of developing a web application that would assist users in enhancing their typing speed and accuracy. The thought was generated by the growing demand for digital literacy and the growing significance of typing skills in everyday life and professional life. We felt there was a need for interactive and user-friendly typing tests, and thus we started developing our project.

In the development stage, our intention was to make the experience smooth and seamless, along with the aspect that the design would be functional but engaging. **We incorporated a timer, a word per minute (WPM) calculation, and an error count option**, thereby allowing users to effectively monitor their progress.

One of the biggest challenges that were faced was the need to provide real-time accuracy in capturing user input and determining the appropriate typing speed. There was a need to test and fine-tune the system constantly to limit errors and create a smooth experience. Additionally, ensuring compatibility between different browsers and devices required plenty of testing and fine-tuning.

**In addition**, we aimed to add a leaderboard functionality that allowed users to compare their score with other users. This was a significant improvement because it added a competitive aspect to the project. Moreover, the use of local Storage to save leaderboard data was an improvement because it allowed for the saving of persistent data **without the use of a database.**

Overall, the project has been a thrilling experience that has allowed us to develop our technical skills and ability to work in teams. We have gained vast knowledge in web development, project planning, and effective solving of technical problems.

**2.SYSTEM REQURIMENTS**

**2.1 Hardware and SOFTWARE REGURIMENTS FOR TYPING SPEED TEST:**

To develop a **Typing Speed Test Project**, you will need to consider both **hardware** and **software** system requirements. Below is a general guide to the system requirements you may need.

|  |  |  |
| --- | --- | --- |
| **Component** | **Minimum Requirement** | **Recommended** |
| **Processor(CPU)** | 1.5 GHz Dual-Core Processor | 2.0 GHz Quad-Core or better |
| **Ram** | 2 GB | 4 GB or more |
| **Storage** | 200 MB free disk space for project files and dependencies | 1 GB or more |
| **Display** | 1366x768 resolution | 1920x1080 (Full HD) or higher |
| **Keyboard and Mouse** | Required for testing and user input | Required for testing and user input |
| **Internet Connection** | Required for online typing tests and real-time scoring (if web-based) | Required for online typing tests and real-time scoring (if web-based) |

**Software Requirements:**

These depend on whether the project is web-based, desktop-based, or mobile-based. Below are some general suggestions for a **web-based typing speed test**.

**Browser Compatibility:**

1. Google Chrome
2. Mozilla Firefox
3. Microsoft Edge
4. Safari

**2.2 SOFTWARE REGURIMENTS SPECIFICATION FOR TYPING SPEED TEST:**

A **Software Requirements Specification (SRS)** is a comprehensive description of the functionality, performance, and constraints of a software application. Below is an example of a **SRS document** for a **Typing Speed Test** project.

|  |  |
| --- | --- |
| **Component** | **Details** |
| **Frontend technologies** | **HTML** - Structures the content and elements of web pages. **CSS** - Enhances the visual appearance, ensuring a professional and responsive design. **JavaScript** - Adds interactivity (e.g., measuring typing speed, calculating errors, updating the leader board). |
| **Browser compatibility** | Designed to run smoothly on **Google Chrome, Mozilla Firefox, and Microsoft Edge** for better JavaScript execution, enhanced security, and improved performance. |
| **Text Editor** | Recommended: **VS Code, Sublime Text** (Lightweight, syntax highlighting, debugging features). Any other code editor can also be used based on preference. |

**3.System Design**

The Typing Speed Test system follows a **modular design**, separating frontend components for better maintainability. It includes **user interaction modules** for the test interface, **data processing modules** for calculating WPM and errors, and a **leaderboard module** for storing and displaying results. The system ensures a smooth and responsive experience across different devices and browsers.

**3.1. System of Modules**

**1. User Interface (UI) Module:**

**Components:**

index.html → Main interface with the typing area, stats display, and buttons.

style.css → Styling for the layout, text effects, and button designs.

Functionality:

Displays the typing area with a dynamically loaded paragraph.

Shows real-time updates for time left, errors, WPM (Words Per Minute), and CPM (Characters Per Minute).

Provides buttons for resetting the test and trying again.

**2. Typing Logic Module**

File: app.js

**Functionality:**

Tracks user input and compares it with the displayed paragraph.

Highlights correct (green) and incorrect (red) characters in real-time.

Manages backspace corrections, ensuring proper index tracking.

**3. Timer Module**

File: app.js

**Functionality:**

Initiates a countdown timer (60 seconds by default).

Updates the time left dynamically on the screen.

Ensures that the test automatically stops when time runs out.

Prevents multiple timer instances when restarting the test.

**4. Random Paragraph Generator Module**

File: app.js

**Functionality:**

Stores multiple paragraphs inside an array.

Selects a random paragraph each time the user starts or restarts the test.

Ensures diversity in typing tests to avoid repetition.

**5. Performance Evaluation Module**

File: app.js

**Functionality:**

Calculates WPM (Words Per Minute) using the standard formula:

WPM=[(Correct characters - Errors)/5]÷[Elapsed Time (minutes)/60].

Calculates CPM (Characters Per Minute) based on typed characters.

Displays errors in real-time as users type.

Ensures WPM and CPM values never go negative.

**6. Leaderboard Module**

File: app.js (Local Storage Feature)

**Functionality:**

Stores top 10 WPM scores in localStorage.

Updates the leaderboard dynamically after each test.

Retrieves and sorts previous scores to display the fastest typists.

**7. Event Handling & Reset Module**

File: app.js

**Functionality:**

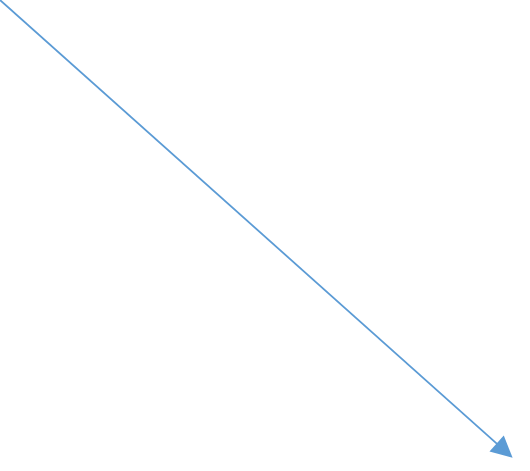
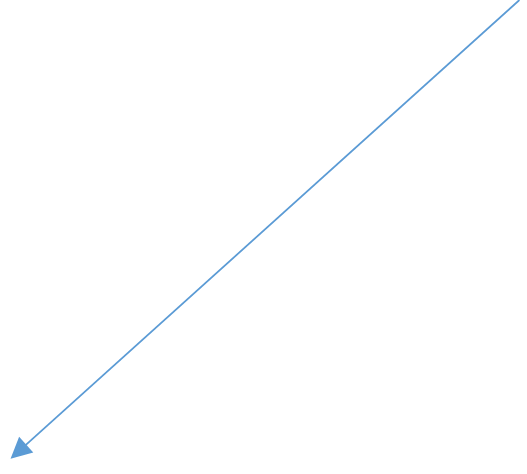
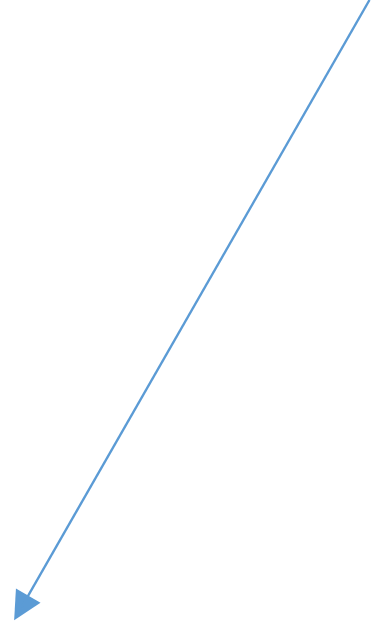
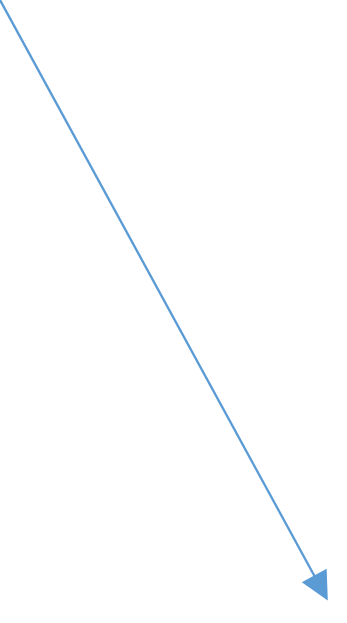
Handles keyboard input events to detect character matches and errors.

Listens for clicks on the "Try Again" button to restart the test.

Clears and resets stats (WPM, CPM, Errors, Timer) when a new test starts.

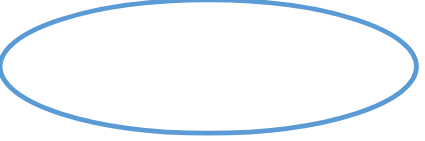
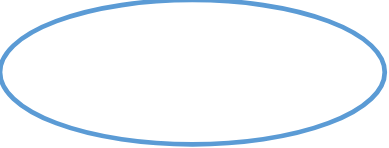
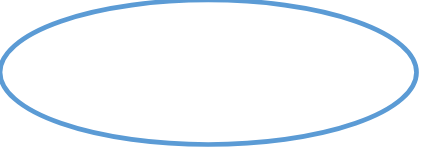
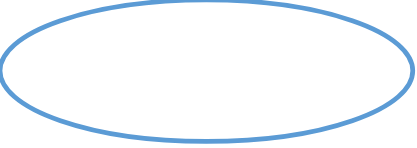
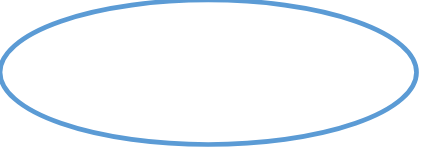
**3.2: UML DIAGRAMS:** 

USER



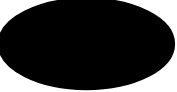
**-**





**View Leader board View WPM, Errors Type Paragraph Enter Name Start Test**

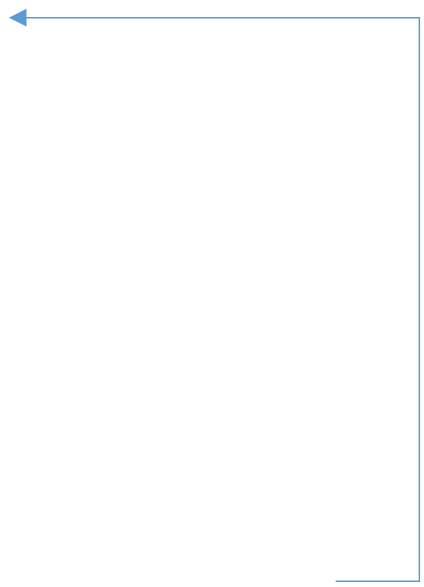
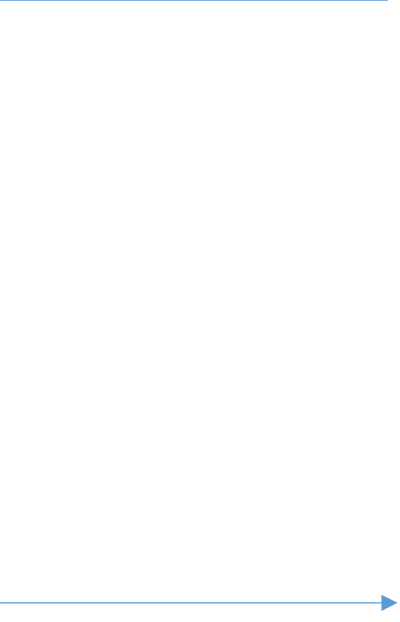
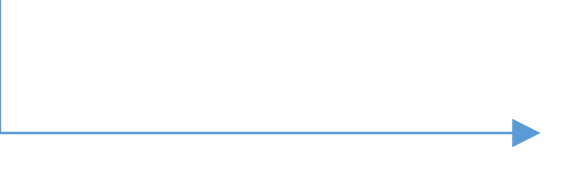
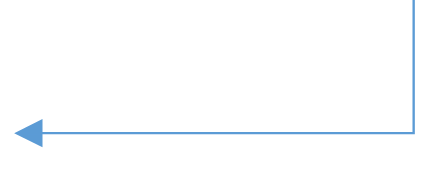
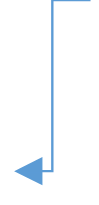
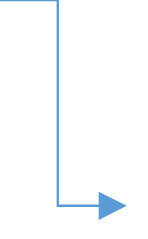
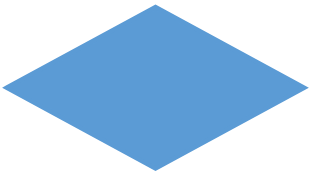
**Fig. 3.2.1: Use Case Diagram**







**User Start Test**



**Sequence Diagram**

**State Diagram**

**Fig. 3.2.2: Activity Diagram**

Display Typing Area

Typing in progress?

Check typed char

Highlight Correct

INC error count

Calculate WPM and Errors

Show Results

Yes

Yes

NO



user

Typing test

Leader board



Start Test



Display Paragraph

Type characters





Show errors and wpm

User

Typing test

Leader board

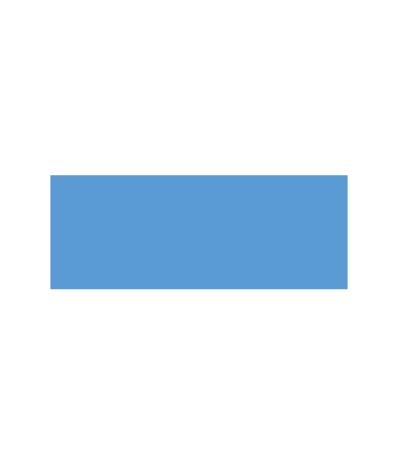
View Top Scores

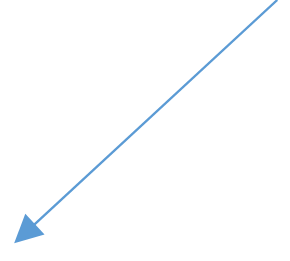
**Fig. 3.2.3:** **Sequence Diagram**



idle

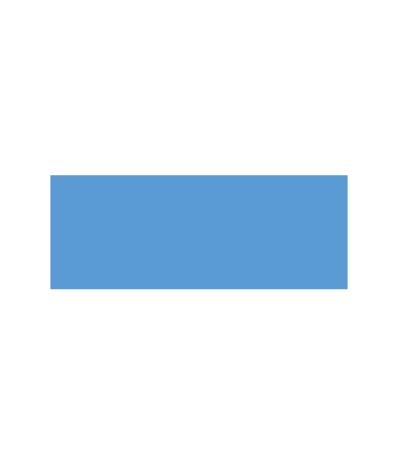


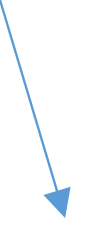




User Starts Test

Typing

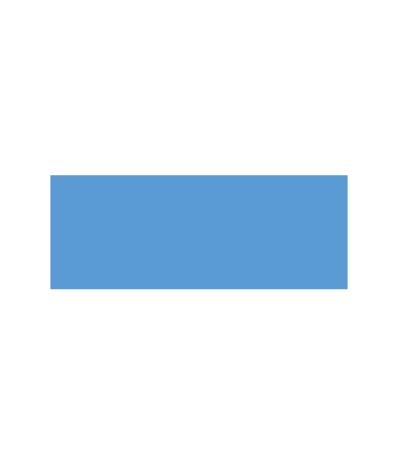


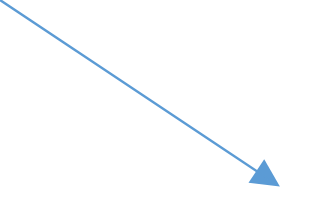


Completed

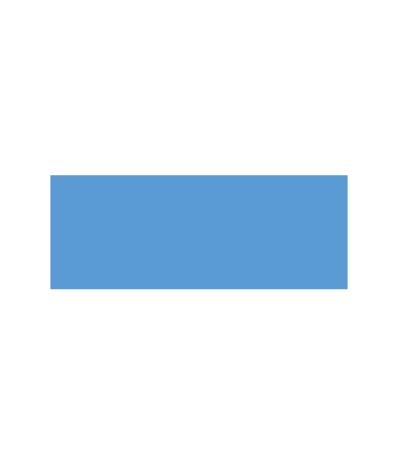
Time Runs out/paragraph completed

User Clicks Try again





Reviewing



User Checks Results

**Fig. 3.2.4: State Diagram**

**4.Implementation**

**4.1 Sample Codes:**

**1.index.html:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Typing Speed Test</title>

<script src="https://cdn.tailwindcss.com"></script>

<style>

body {

background: linear-gradient(to right, #0f2027, #203a43, #2c5364);

color: white;

font-family: 'Poppins', sans-serif;

text-align: center;

padding: 20px;

}

.container {

max-width: 800px;

margin: auto;

background: rgba(255, 255, 255, 0.1);

padding: 20px;

border-radius: 10px;

box-shadow: 0px 5px 15px rgba(0, 0, 0, 0.3);

}

textarea {

width: 100%;

height: 100px;

font-size: 16px;

</style>

</head>

<body>

<div class="container">

<h1 class="text-3xl font-bold mb-4">Typing Speed Test</h1>

<p id="text" class="text-lg mb-4">The quick brown fox jumps over the lazy dog.</p>

<textarea id="input" placeholder="Start typing here..." class="border border-gray-400"></textarea>

<div class="stats mt-4">

<p>Time: <span id="time">30</span> sec</p>

<p>Speed: <span id="speed">0</span> WPM</p>

<p>Accuracy: <span id="accuracy">100</span>%</p>

</div>

<button onclick="startTest()" class="bg-blue-500 px-6 py-2 mt-4 rounded">Restart</button>

</div>

<script>

let startTime, timer, textSample;

const textElement = document.getElementById("text");

const inputElement = document.getElementById("input");

const timeElement = document.getElementById("time");

const speedElement = document.getElementById("speed");

const accuracyElement = document.getElementById("accuracy");

function startTest() {

textSample = textElement.innerText;

inputElement.value = "";

startTime = null;

clearInterval(timer);

timeElement.innerText = "30";

speedElement.innerText = "0";

accuracyElement.innerText = "100";

startTimer();

}

function startTimer() {

let timeLeft = 30;

timer = setInterval(() => {

timeLeft--;

timeElement.innerText = timeLeft;

if (timeLeft <= 0) {

clearInterval(timer);

inputElement.disabled = true;

}

}, 1000);

}

inputElement.addEventListener("input", () => {

if (!startTime) {

startTime = new Date().getTime();

}

updateStats();

});

function updateStats() {

const elapsedTime = (new Date().getTime() - startTime) / 1000;

const wordsTyped = inputElement.value.trim().split(/\s+/).length;

const speed = Math.round((wordsTyped / elapsedTime) \* 60);

speedElement.innerText = isNaN(speed) ? 0 : speed;

checkAccuracy();

}

function checkAccuracy() {

let correctChars = 0;

let enteredText = inputElement.value;

for (let i = 0; i < enteredText.length; i++) {

if (enteredText[i] === textSample[i]) correctChars++;

}

const accuracy = (correctChars / enteredText.length) \* 100;

accuracyElement.innerText = isNaN(accuracy) ? 100 : accuracy.toFixed(2);

}

</script>

</body>

</html>

**2.main.html:**

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Typing Speed Test</title>

<style>

body {

background: linear-gradient(to right, #0f2027, #203a43, #2c5364);

color: #ffffff;

font-family: 'Arial', sans-serif;

line-height: 1.6;

overflow-x: hidden;

margin: 0;

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;

}

h1 {

font-size: 2.5rem;

font-weight: bold;

text-shadow: 0px 4px 6px rgba(0, 0, 0, 0.4);

margin-bottom: 1rem;

}

p {

font-size: 1.2rem;

margin-bottom: 1.5rem;

}

.content-wrapper {

max-width: 600px;

padding: 2rem;

background-color: rgba(0, 0, 0, 0.5);

border-radius: 1rem;

box-shadow: 0px 8px 15px rgba(0, 0, 0, 0.5);

text-align: center;

}

.button-container {

display: flex;

justify-content: center;

gap: 20px;

}

button {

padding: 12px 24px;

font-size: 1rem;

font-weight: bold;

color: white;

border: none;

border-radius: 50px;

cursor: pointer;

transition: transform 0.3s ease, background-color 0.4s ease, box-shadow 0.3s ease;

}

.start-button {

background-color: #e67e22;

}

.start-button:hover {

background-color: #d35400;

transform: scale(1.1);

box-shadow: 0 5px 10px rgba(255, 165, 0, 0.5);

}

.leaderboard-button {

background-color: #27ae60;

}

.leaderboard-button:hover {

background-color: #1e8449;

transform: scale(1.1);

box-shadow: 0 5px 10px rgba(0, 255, 127, 0.5);

}

</style>

</head>

<body>

<div class="content-wrapper">

<h1>Typing Speed Test</h1>

<p>Test your typing skills and improve your speed and accuracy.</p>

<div class="button-container">

<a href="index.html">

<button class="start-button">Start Test</button>

</a>

<a href="leaderboard.html">

<button class="leaderboard-button">View Leaderboard</button>

</a>

</div>

</div>

</body>

</html>

**3.styles.css:**

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: Arial, sans-serif;

}

body {

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

background: linear-gradient(160deg, #34495e, #2c3e50);

color: white;

}

.container {

width: 700px;

padding: 25px;

text-align: center;

background: rgba(0, 0, 0, 0.8);

border-radius: 12px;

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.5);

}

h1 {

font-size: 24px;

margin-bottom: 20px;

text-transform: uppercase;

text-shadow: 0 2px 4px rgba(0,0,0,0.4);

}

/\* Leaderboard Box \*/

.leaderboard-box {

margin-top: 15px;

}

table {

width: 100%;

border-collapse: collapse;

margin-bottom: 20px;

}

th, td {

padding: 12px;

text-align: center;

border-bottom: 1px solid #ddd;

}

th {

background: #27ae60;

color: white;

font-weight: bold;

text-transform: uppercase;

}

tr:nth-child(even) {

background: rgba(255, 255, 255, 0.1);

}

tr:hover {

background: rgba(255, 255, 255, 0.2);

}

/\* Form Styling \*/

.form-container {

margin-top: 20px;

}

label {

font-size: 16px;

font-weight: bold;

display: block;

margin-bottom: 8px;

}

input {

width: 80%;

padding: 10px;

font-size: 16px;

border-radius: 6px;

border: none;

outline: none;

text-align: center;

margin-bottom: 15px;

}

button {

width: 100%;

padding: 12px;

font-size: 18px;

border: none;

background: #f39c12;

color: white;

cursor: pointer;

border-radius: 6px;

font-weight: bold;

transition: background 0.3s;

}

button:hover {

background: #e67e22;

}

**4.leaderboard.html:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Leaderboard - Typing Speed Test</title>

<link rel="stylesheet" href="styles.css"> <!-- Link to external CSS -->

</head>

<body>

<div class="container">

<h1>Typing Speed Test Leaderboard</h1>

<div class="leaderboard-box">

<table>

<thead>

<tr>

<th>Rank</th>

<th>Name</th>

<th>WPM</th>

</tr>

</thead>

<tbody id="leaderboard-list">

<!-- Leaderboard entries will go here -->

</tbody>

</table>

<div class="form-container">

<label for="player-name">Enter Your Name:</label>

<input type="text" id="player-name" placeholder="Your Name">

<button id="start-test">Start Test</button>

</div>

</div>

</div>

<script>

// Placeholder for leaderboard data (This would normally be saved in a database or localStorage)

const leaderboard = [

{ name: 'rajesh', wpm: 120 },

{ name: 'ganesh', wpm: 110 },

{ name: 'rahul', wpm: 105 },

{ name: 'kumari', wpm: 100 },

{ name: 'soujith', wpm: 95 },

{ name: 'franklin ramesh', wpm: 90 },

{ name: 'devara', wpm: 85 },

{ name: 'kalyan', wpm: 80 },

{ name: 'nawin', wpm: 75 },

{ name: 'ramesh model', wpm: 70 }

];

// Function to display leaderboard

function displayLeaderboard() {

const leaderboardList = document.getElementById('leaderboard-list');

leaderboardList.innerHTML = '';

leaderboard.forEach((entry, index) => {

const row = `<tr>

<td>${index + 1}</td>

<td>${entry.name}</td>

<td>${entry.wpm}</td>

</tr>`;

leaderboardList.innerHTML += row;

});

}

// Display leaderboard when page loads

displayLeaderboard();

// Handle test start

document.getElementById('start-test').addEventListener('click', () => {

const playerName = document.getElementById('player-name').value;

if (playerName.trim() === '') {

alert('Please enter your name!');

return;

}

localStorage.setItem('playerName', playerName);

window.location.href = 'index.html'; // Navigate to the typing test page

});

</script>

</body>

</html>

**5.test.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 Test</title>

<style>

body {

font-family: Arial, Helvetica, sans-serif;

background-color: #f4f4f4;

color: #333;

text-align: center;

padding: 20px;

}

.container {

background-color: #fff;

padding: 30px;

border-radius: 10px;

width: 80%;

max-width: 800px;

margin: 0 auto;

box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);

}

h1 {

color: #4caf50;

}

p {

font-size: 1.2em;

margin-bottom: 20px;

}

#typing-area {

margin: 20px 0;

padding: 20px;

background-color: #f0f0f0;

color: #333;

border-radius: 10px;

font-size: 1.2em;

text-align: left;

}

#typing-area span.correct {

color: #4caf50;

}

#typing-area span.incorrect {

color: #f44336;

}

input {

width: 100%;

padding: 10px;

font-size: 18px;

border-radius: 5px;

border: 1px solid #ccc;

margin-bottom: 20px;

background-color: #f9f9f9;

color: #333;

}

button {

padding: 10px 20px;

background-color: #4caf50;

color: #fff;

border: none;

border-radius: 5px;

cursor: pointer;

margin-top: 10px;

}

button:hover {

background-color: #45a049;

}

.result {

display: flex;

justify-content: space-around;

margin-top: 20px;

}

.result div {

font-size: 1.2em;

padding: 10px;

background-color: #f0f0f0;

border-radius: 5px;

}

.result div p {

margin: 0;

}

</style>

</head>

<body>

<div class="container">

<h1>Typing Speed Test</h1>

<p>Start typing the text below and see your speed and accuracy!</p>

<!-- Display the paragraph for the typing test -->

<div id="typing-area"></div>

<!-- Input field to type in -->

<input type="text" id="input-field" placeholder="Start typing..." />

<div class="result">

<div>

<p>Time Left</p>

<span id="time">60</span>

</div>

<div>

<p>Errors</p>

<span id="errors">0</span>

</div>

<div>

<p>WPM</p>

<span id="wpm">0</span>

</div>

<div>

<p>CPM</p>

<span id="cpm">0</span>

</div>

</div>

<button id="try-again-button">Try Again</button>

</div>

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

</body>

</html>

**4.2 Test Cases:**

**5**. **Results**

**Functional Results:**

Typing Speed Calculation Works Accurately. The system correctly calculates Words Per Minute (WPM) based on the user's typed words and elapsed time. WPM updates dynamically without delays. **Accuracy Measurement is Precise.**

The application correctly identifies errors and calculates the accuracy percentage based on the number of correct and incorrect characters typed. Accuracy updates in real-time as the user types.

Countdown Timer Functions Properly. The timer starts at 30 seconds and counts down correctly. When the timer reaches 0, the input field is disabled to prevent further typing**. Restart Functionality Works as Expected**

* Clicking the Restart button resets the timer, WPM, accuracy, and input field.
* Users can start a new test instantly.
* Predefined Text is Displayed Correctly
* The passage provided for typing is displayed without errors.

**UI/UX Results:**

* Responsive and User-Friendly Interface
* The webpage layout adapts well to different screen sizes (desktop, tablet, and mobile).
* Typography, colors, and spacing provide a smooth reading and typing experience.
* Buttons respond with hover effects and animations, improving usability.
* Clicking buttons gives immediate feedback.
* The white text on a dark background ensures good visibility and reduces eye strain.

**Performance & Error Handling Results**

* Handles Fast & Slow Typing Without Lag
* The application updates WPM in real-time without performance issues.
* Whether a user types slowly or rapidly, the system calculates results accurately.
* Error Detection is Effective
* Incorrect characters are counted properly, and accuracy drops accordingly.
* Clipboard Pasting is Handled Properly
* The application remains stable across multiple test cases, including rapid typing,

pausing, and restarting.

**5.1 Output Screen(s):**

**5.1.1: Main page**

**5.1.2: Index page:**

**5.1.3: Leader board page**

**5.1.4: Typing page**

**6. Conclusion**

The project effectively integrates real-time tracking, accuracy calculation, and a user-friendly interface, making it both educational and interactive.

* **Key Achievements:**

**Accurate WPM & Accuracy Measurement** – The system dynamically calculates Words Per Minute (WPM) and accuracy percentage in real-time.

**Countdown Timer & Auto-Lock Mechanism** – The test automatically stops at 30 seconds, preventing further input.

**User-Friendly & Responsive UI** – The design is optimized for desktop and mobile, ensuring a seamless experience across devices.

**Performance Stability** – The system handles fast, slow, and incorrect typing e fficiently without any performance issues.

**Leaderboard Integration** – Users can store their scores locally and track their progress.

The typing speed test project successfully fulfills its goal of helping users evaluate and improve their typing speed and accuracy. by offering a real-time, interactive experience, the **system serves as a valuable learning tool**. additionally, the project provided an opportunity for the development team to gain expertise in front-end technologies and java script based interactive applications. with further enhancements, the project can evolve into a full-scale typing proficiency platform.

The Typing Speed Test project successfully provides a comprehensive typing assessment tool that enables users to evaluate and enhance their typing proficiency. By combining real-time feedback, performance tracking, and ease of use, the project has significant potential in both **educational and professional environments.**

**7**. **References**

The development and implementation of the Typing Speed Test project were guided by various technical resources, research articles, and existing online typing platforms. Below are the key references:

**[1].**Typing Academy – WPM Calculation Methods

**Source**: https://www.typingacademy.com

Used for understanding the formula and logic behind Words Per Minute (WPM) calculation.

**[2].**Keybr – Best Practices for Typing Accuracy

**Source:** https://www.keybr.com

Helped in implementing real-time accuracy tracking based on user input.

**[3].**10FastFingers – Typing Test Mechanics

**Source:** https://www.10fastfingers.com

Used as inspiration for UI/UX design, leader board system, and test mechanics.

**[4].**Mozilla Developer Network (MDN) – JavaScript & Local Storage

**Source:** https://developer.mozilla.org

Used for implementing JavaScript event handling, performance tracking, and local Storage leader board.

**[5].** W3Schools – Responsive Web Design & Tailwind CSS

**Source:** https://www.w3schools.com

Helped in designing a responsive and visually appealing interface.

**[6].** Google Fonts & UI Design Guidelines

**Source:** https://fonts.google.com

Used for selecting professional and readable fonts in the project.