

**Typing Test**

**INF651 – Front End Web Development I**

**Final Project Documentation**

**12/09/2025**

**Dasiksa (Cole) Ol**

# 1. Project Overview

Typing Test is an interactive web application built with HTML, CSS, and JavaScript with the purpose of demonstrating key JavaScript concepts such as event handling, DOM manipulation, conditional logic, timers, and storing data using localStorage.

Users can select a test mode, choose a timer duration, and complete a typing test that provides real-time feedback on their typing performance. At the end of each test, users will receive a results breakdown and their best performance metrics are stored for future viewing.

The site consists of three pages:

- Home – interactive typing test
- About – project description + FAQ
- Stats – saved performance statistics

## 2. Sketches / Wireframes

Home Page

The wireframe shows a web browser window with the title 'typingtest'. The navigation bar contains links for 'Home', 'About', and 'Stats'. The main content area features a control panel with three input fields labeled 'time', 'mode', and 'theme', followed by 'start' and 'restart' buttons. Below this is a typing area with six lines of 'example words'. At the bottom, a 'Result' box displays the text: 'Your speed is 67 wpm. You are a fast typer'.

About Page

typingtest

HomeAboutStats

About

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis aliquet mi sed porttitor euismod. Cras a dolor sit amet nibh ultricies egestas non sed nisl. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Phasellus ut augue eget augue efficitur facilisis non cursus risus. Maecenas scelerisque tortor ut diam aliquet varius. Donec pellentesque ipsum ipsum, a porttitor est blandit eget. Proin tincidunt enim id pretium sollicitudin. Etiam suscipit orci vitae felis dictum blandit. Proin consectetur tempus metus, eget egestas eros.

FAQ

Question 1 - aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

Question 1 - aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

Question 1 - aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

typingtest

HomeAboutStats

About

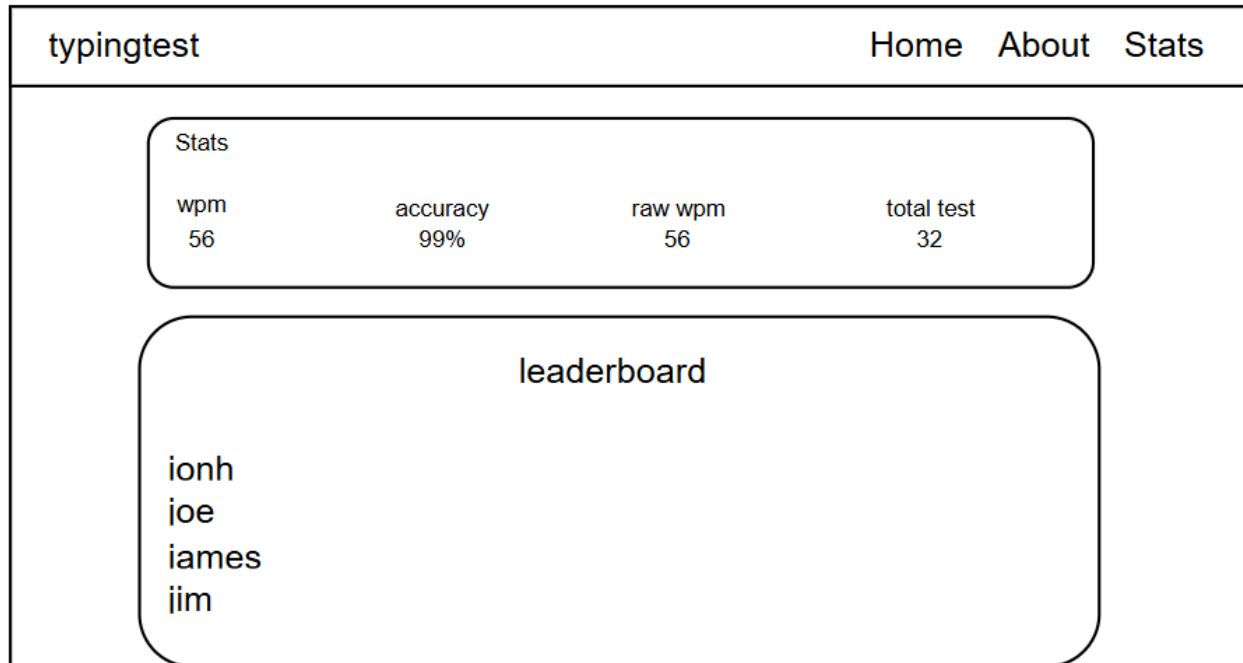
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis aliquet mi sed porttitor euismod. Cras a dolor sit amet nibh ultricies egestas non sed nisl. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Phasellus ut augue eget augue efficitur facilisis non cursus risus. Maecenas scelerisque tortor ut diam aliquet varius. Donec pellentesque ipsum ipsum, a porttitor est blandit eget. Proin tincidunt enim id pretium sollicitudin. Etiam suscipit orci vitae felis dictum blandit. Proin consectetur tempus metus, eget egestas eros.

FAQ

Question 1 - aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

answer - bbbbbbbbbbbbbbbbbbbbbbbbbbbb

Question 1 - aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa



### 3. Features

#### Typing Test

- Words or sentences are generated using arrays.
- Timer waits for the first keystroke before starting.
- Live stats are updated on every input event.
- Words are highlighted based on correctness.
- A blinking caret tracks where the user is typing.
- When time runs out or all words are typed, the test ends.

#### Result Summary

- Calculates: WPM, Raw WPM, Accuracy, Characters typed, Error count.
- Displays a message based on performance.

#### Stats Page

Uses localStorage to save:

- Best WPM (highest net WPM ever achieved)
- Best Accuracy
- Total number of tests

### Theme Selector

Users can switch between Light, Dark, and Dracula themes.  
The selected theme is saved and automatically applied on next visit.

### FAQ

A simple toggle feature:

- Clicking a question expands or hides the answer.

## **4. JavaScript Concepts Used**

### Event Listeners

- click for start/reset buttons and FAQ items
- input for typing
- DOMContentLoaded for initialization

### DOM Manipulation

- Creating word and character elements dynamically
- Adding/removing classes (active, done-correct, done-error, incorrect)
- Updating content inside spans for stats and results
- Showing/hiding results panel

### Timers

- setInterval() used for countdown
- Controlled start (first keystroke) and stop (reset or finish) logic

### Conditional Logic

- Determine correct/incorrect characters
- Handle active vs completed words
- Sentence mode vs word mode
- Display different messages based on WPM

### Loops

- For loops to render words and letters
- Loop through characters to check correctness

- Loop through sentences to generate paragraphs

#### Arrays & Objects

- SENTENCES and WORDS arrays generate typing material
- Local storage keys stored in STORAGE\_KEYS object

#### localStorage

Stores:

- Best WPM
- Best Accuracy
- Total Tests
- Saved theme

## **5. Challenges & Solutions**

1. Making the timer start only on the first keystroke: In the beginning the timer would start immediately after clicking Start, which is not ideal for user to click then prepare themselves for typing. Hence, why I added readyToStart and timerStarted flags to control when setInterval begins.

2. Handling incorrect character highlighting: All future words were turning red unexpectedly after test started. It took a lot of testing to find and updated logic to only evaluate characters in the current word.

3. Removing duplicate words in word mode: Random generation caused many duplicates for both words and sentences mode which ruined the UI and can mess with user experience. To fix it, I implemented a pool-removal system that selects only unique words.

## **6. Future Enhancements**

If given the opportunity to continue, I would work on:

- A history table of past WPM scores
- Additional modes (punctuation, quotes, numbers-only)
- High-score leaderboard using backend storage
- Improved paragraph generation (longer text, literary excerpts)

All features will add user engagement to the web app and allow users to explore different form of testing with a lot of commitment and competitiveness to get better.

## **7. Conclusion**

This project successfully demonstrates core JavaScript skills, including event handling, DOM manipulation, timers, arrays, conditional logic, and persistent storage. It results in a fully functional web application that is interactive, user-friendly, and extendable. The development process provided valuable experience in debugging, planning, designing interactive UI components, and organizing JavaScript logic into modular functions.