

PROJECT ABSTRACT

Typing Speed Test and Practice Web App

A web app measuring typing speed (WPM) and accuracy with live stats, result analytics and focused practice drills. Delivered built-to-order with full source, passage datasets, report, PPT and viva kit.

01. Title

Typing Speed Test and Practice Web App

02. Introduction / Background

Typing speed quietly decides outcomes for programmers — coding rounds, documentation, viva typing — yet most students have never measured theirs beyond a vague guess. A typing-test web app makes it measurable: timed passages, live per-character feedback, and metrics computed the standard way (WPM as characters/5 per minute; accuracy as correct keystrokes over total). Beyond the headline number, result analytics — a WPM-over-time curve, character-level breakdown, consistency score and most-missed keys — turn practice into deliberate improvement.

03. Problem Statement

Students cannot improve what they cannot measure, and generic advice ('type more') lacks feedback. The project addresses this with a complete typing-test web app: timed tests with live scoring, honest analytics per attempt, a logged history for tracking progress, and practice drills generated from the user's own weak keys.

04. Objectives

- Run timed typing tests (15/60/120 s) over curated passages with live character feedback.
- Compute WPM, accuracy, consistency and character-level stats with documented definitions.
- Render result analytics: WPM curve, correct/incorrect/extra/missed counts, most-missed keys.
- Log every attempt in a history view with personal best and weekly trend.
- Generate focused practice drills from the user's weak keys plus curated packs.
- Deliver the complete student kit: source, datasets, report, PPT and viva Q&A.;

05. Proposed Methodology

The app is a single dependency-free HTML file. (1) Engine: keystroke events are compared against the passage; the timer starts on first input; WPM and accuracy update live. (2) Analytics: per-second WPM is sampled into a canvas-rendered curve; misses are counted per key and mapped to fingers. (3) Persistence: attempts are stored via the Web Storage API and rendered in the history view with trends. (4) Practice: drill passages are

generated from weak-key statistics or chosen from curated packs (top words, punctuation, numbers, code).

06. System Architecture / Working

The user picks duration and passage type; the passage renders with a blinking caret and the first keystroke starts the timer. Each keystroke is scored live against the expected character with backspace-aware error accounting. At time-up the attempt is recorded and the results screen renders the WPM curve, character breakdown and missed keys; history appends the attempt and practice mode builds drills from the weakest keys.

07. Hardware / Software Requirements

- Any modern browser on laptop/desktop (design target)
- Physical keyboard (supported input)
- HTML5, CSS3, JavaScript ES6, Canvas API, Web Storage API
- No build step; single-file deployment

08. Expected Outcomes

- A fully interactive typing test with live WPM/accuracy scoring.
- Result analytics with WPM curve and weak-key detection.
- A persistent attempt history with trend tracking.
- A complete report, PPT and viva Q&A; on metric design and real-time input handling.

09. Applications

- Typing practice for students and programmers.
- Teaching real-time event handling and canvas visualisation.
- Demonstrating metric design with documented definitions.

10. Future Scope

- Account-based cloud sync of history across devices.
- Multiplayer race mode over WebSockets.
- Adaptive difficulty that tracks the user's level.
- Mobile-optimised touch typing mode.