

Typing Practice Application - Product Specification

Version 1.0

Project Name: Typing Practice App

Author: Bruce

Date: April 14, 2025

Version: 1.0

1. Objective

To design and develop a browser-based typing practice application that offers multiple interactive typing modes, real-time feedback, and performance statistics to help users improve typing speed and accuracy.

2. Core Features

2.1 Game Modes

- Timer Mode: Preset durations (15s, 30s, 50s) or custom; ends when time is up.
- Word Count Mode: Targets (25, 50, 100) or custom; ends when word count is reached.
- Instant Death Mode: Ends on first incorrect keystroke.
- Free-for-All Mode: User types indefinitely until quitting manually.

2.2 Typing Engine

- Active character highlighting
- Feedback for correct/incorrect inputs
- Input restricted to provided text
- Restart/reset functionality

2.3 Performance Metrics

- WPM (Words Per Minute)
- Accuracy (%)
- Word Count (if applicable)
- Time Remaining (for timer mode)

2.4 Audio Feedback

- Sound on correct input
- Optional sound on mistake
- Toggle button to mute/unmute

2.5 Theme Selector

- At least three themes: Light, Cyberpunk, Minimal
- Dynamic UI theme changes
- Uses CSS variables for styling

3. User Interface Components

- Mode selector (buttons or dropdown)
- Typing display area
- Stats panel
- Start, Restart, Sound Toggle, Theme Selector
- Inputs for custom time/word count

4. Technical Specifications

- Frontend: HTML, CSS, JavaScript (Vanilla)
- Audio: HTML5 <audio> + JS Audio API
- Theme: CSS variables + class switching

- Timer: setInterval & Date.now()
- Mobile responsive

5. Development Tasks and Milestones

1. Implement core typing engine and keystroke detection
2. Develop mode logic controllers (Timer, Word Count, Instant Death, Free-for-All)
3. Integrate real-time statistics (WPM, Accuracy, Word Count, Timer)
4. Add audio feedback and sound toggle
5. Design and implement theme selection with at least three visual themes using CSS variables:
 - Define themes (Light, Cyberpunk, Minimal)
 - Build theme selector UI
 - Apply CSS variables dynamically
 - Ensure consistent styling during gameplay
6. Build UI layout: mode selector, typing display, stats panel, and controls
7. Perform browser testing and finalize UX polish

6. Notes

- Modular and reusable game logic
- Accessibility to be considered in future versions
- Code should be well-commented and maintainable

End of Document