## Front End Engineering-II

Project Report

Semester-III (Batch-2023)

**Typing speed enhancing website**

A red and white sign

Description automatically generated with low confidence

**Supervised By: Submitted By:**

Ms. Meenu Supreem Singh (2310992221)

**Department of Computer Science and Engineering**

## Chitkara University Institute of Engineering & Technology,

## Chitkara University, Punjab

**Abstract**

Typing Train is an interactive web-based application designed to help users improve their typing skills through a structured, level-based approach. Developed using HTML, CSS, JavaScript, and React, the application provides a responsive and engaging experience. The platform features a welcoming homepage, as well as Contact Us, About Us, and Login pages for a complete user experience. Users can access multiple typing levels, each offering a dedicated page where they can practice typing exercises tailored to enhance both speed and accuracy.

This project aims to create a valuable tool for users seeking to develop better typing proficiency, with each level progressively challenging their skills. Future improvements may include advanced typing analytics, user progress tracking, and additional levels for sustained practice. The Typing Train application not only offers a functional interface but also supports a fun and productive way for users to achieve mastery in typing.

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| S.NO. | HEADING | PAGE NO. |
| 1. | Introduction | 3 |
| 2. | Problem Statement | 4-5 |
| 3. | Methodology | 6-7 |
| 4. | Results | 8 - 10 |
| 5. | References | 11 |

1. **Introduction**
   1. **Background**

Typing is an essential skill in today's digital age, where the majority of communication and work tasks are carried out on computers and other digital devices. With the increasing reliance on technology, the ability to type quickly and accurately has become a fundamental requirement in both personal and professional environments. Traditional typing practice tools often lack engagement and fail to provide effective feedback, leading to slow progress for learners. Recognizing this gap, the "Typing Train" project was initiated to create a modern, interactive typing practice platform that leverages the latest web technologies to offer a superior user experience.

* 1. **Objectives**

The primary objectives of "Typing Train" are:

* + 1. **To Create an Interactive Typing Platform:** Develop a user-friendly web application that allows users to practice typing through various exercises and tests.
    2. **To Improve Typing Speed and Accuracy:** Provide real-time feedback and detailed performance metrics to help users enhance their typing skills.
    3. **To Engage and Motivate Users:** Incorporate interactive elements and responsive design to make the learning process enjoyable and engaging.
  1. **Significance**

The significance of "Typing Train" lies in its ability to provide an effective learning tool for users of all ages and skill levels. Unlike traditional typing programs, which often use repetitive and uninspired exercises, "Typing Train" offers a dynamic and motivating environment where users can track their progress and compete with themselves to achieve better results. The use of HTML, CSS, and JavaScript ensures that the application is lightweight, easily accessible, and runs smoothly across different devices and browsers, making it a versatile solution for improving typing proficiency.

1. **Problem Definition and Requirements**
   1. **Problem Statement**

Many people struggle with typing efficiency, which can hinder their productivity in both academic and professional settings. Existing typing tools often lack the engaging features needed to maintain user interest and do not provide sufficient feedback for meaningful improvement. "Typing Train" addresses these issues by offering an interactive platform that delivers immediate, personalized feedback and supports users in developing their typing skills through consistent practice.

* 1. **Software Requirements**

2.2.1 Frontend Development

* HTML5, CSS3: For structuring and styling the web pages, JavaScript: For implementing dynamic and interactive features.
* JavaScript: For adding dynamic and interactive elements.
* React: For building reusable UI components and managing the state of the application.

2.2.2 Development Tools

* Visual Studio Code: As the primary code editor.
* Git/GitHub: For version control and collaboration.
* Vite: For faster development and build setup for React applications.
  1. **Data Sets**

2.3.1 **Typing Content and Practice Levels**

* Typing Exercises: A diverse set of typing texts and levels to challenge users at various skill levels, with progressively increasing difficulty.
* Practice Sessions: Timed typing exercises and accuracy challenges to help users improve their skills.

2.3.2 **User Data**

* User Profiles: Information about each user's progress, typing speed, and accuracy.
* Performance Tracking: Data on user progress to provide feedback and suggest areas for improvement.

**3. Proposed Design / Methodology**

**3.1 Platform Design**

**3.1.1 User-Centric Interface:** A clean, minimalistic UI is crafted with HTML, CSS, and JavaScript, ensuring intuitive use across various devices. The React framework is leveraged to build responsive layouts and dynamic components, providing a smooth and engaging typing experience for users.

**3.1.2 User Authentication and Secure Access:** The platform includes a secure login and registration system for users. Session management is implemented to ensure user data is protected. Future enhancements may include encrypted session storage and account recovery options for improved security and user convenience.

**3.2 Methodology**

**3.2.1 Frontend Development:** The frontend is developed with React using a component-based architecture for modular and reusable code. This design enhances scalability and maintainability. Key pages like the homepage, typing levels, and practice screens use React components, making the interface interactive and adaptable.

**3.2.2 Data Flow and Component Communication:** React’s state management is utilized to track typing progress and manage interactions within each level. Props are used to pass data between components, allowing for a level-based system where each typing level dynamically loads relevant typing exercises and updates user performance metrics.

**3.2.3 User Data Management:** Local storage and session storage are used to track user typing progress, ensuring that data such as typing speed and accuracy are maintained across sessions. Future integrations with a database may enhance this feature, allowing for real-time data persistence and user progress tracking.

**3.3 Tracking and Feedback**

**3.3.1 Typing Progress Tracking:** The platform includes real-time tracking of typing speed, accuracy, and completion time for each exercise, helping users monitor and improve their performance. This data is displayed at the end of each typing session, giving users feedback to guide their improvement.

**3.3.2 Feedback Mechanism:** A feedback system allows users to rate each typing level, providing insights into their experience. This feedback helps improve typing exercises, making the platform more adaptive to user needs and fostering continuous improvement.

**Group Member – Supreem Singh (2310992221)**

**Page created for EVALUATION-1**

**4. Results**

**4.1 Login page**

Login page is responsive and static and made using HTML and CSS.

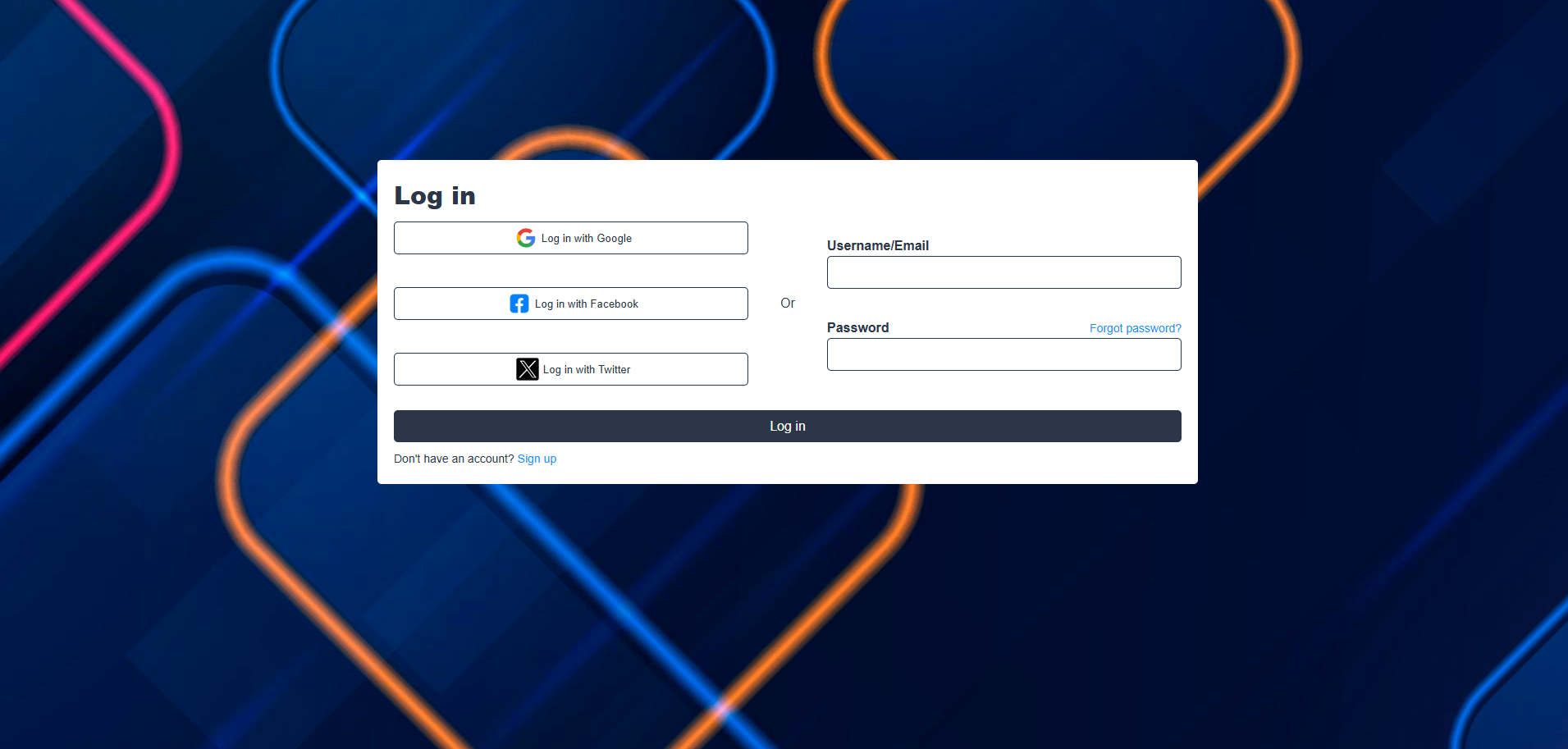


Figure 1 (Login page)

* Login Structure: The page includes social login buttons (Google, Facebook, Twitter) and a form for email/username and password.
* Responsive Design: The layout adjusts to a vertical stack on smaller screens (below 1000px) using flexbox.
* Styling: Buttons and input fields have consistent borders, padding, and hover effects.
* Forgot Password: A clickable "Forgot password?" link is included beside the password field.
* Login Button: A full-width login button with hover effects is provided for form submission.

**Group Member – Supreem Singh (2310992221)**

**Page created for EVALUATION-2**

**4. Results**

**4.1 Login page**

Login page is responsive and static and made using HTML and CSS.

Login logic is implemented using javascript.

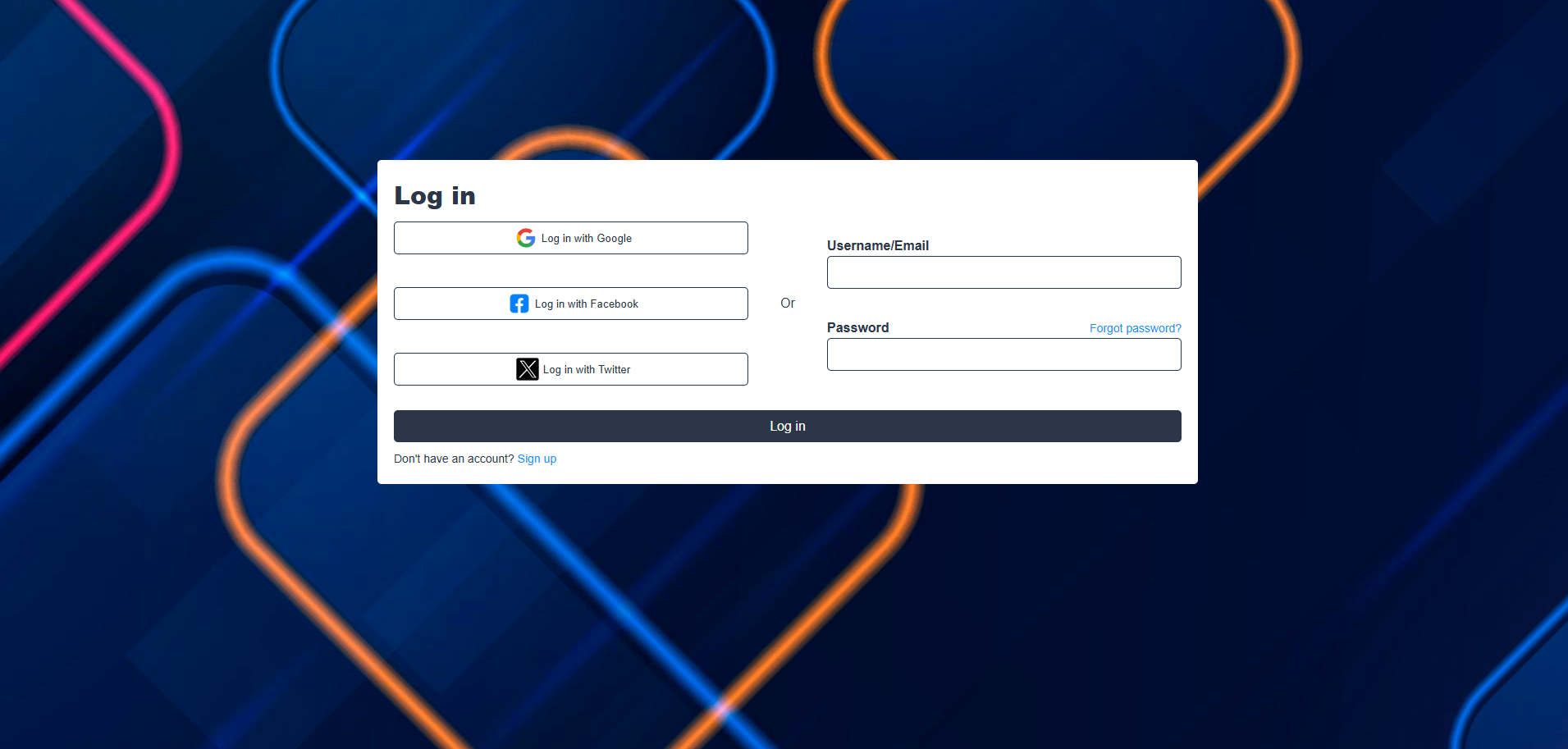


Figure 1 (Login page)

* Form Submission: The login form prevents default submission, capturing the username and password fields for validation.
* User Validation: The credentials are compared with sample hardcoded values (admin for both username and password).
* Login Success: If credentials match, a success message appears, and a flag (isLoggedIn) is stored in localStorage, followed by a redirect to index.html.
* Login Failure: If the credentials don't match, an error message is shown, prompting for valid credentials.
* Local Storage: On successful login, localStorage.setItem stores a flag (isLoggedIn) indicating the user's logged-in status.

**Group Member – Supreem Singh (2310992221)**

**Page created for EVALUATION-3**

**4. Results**

**4.1 Login page**

Login page is responsive and static and made using HTML and CSS.

Login logic is implemented using javascript.

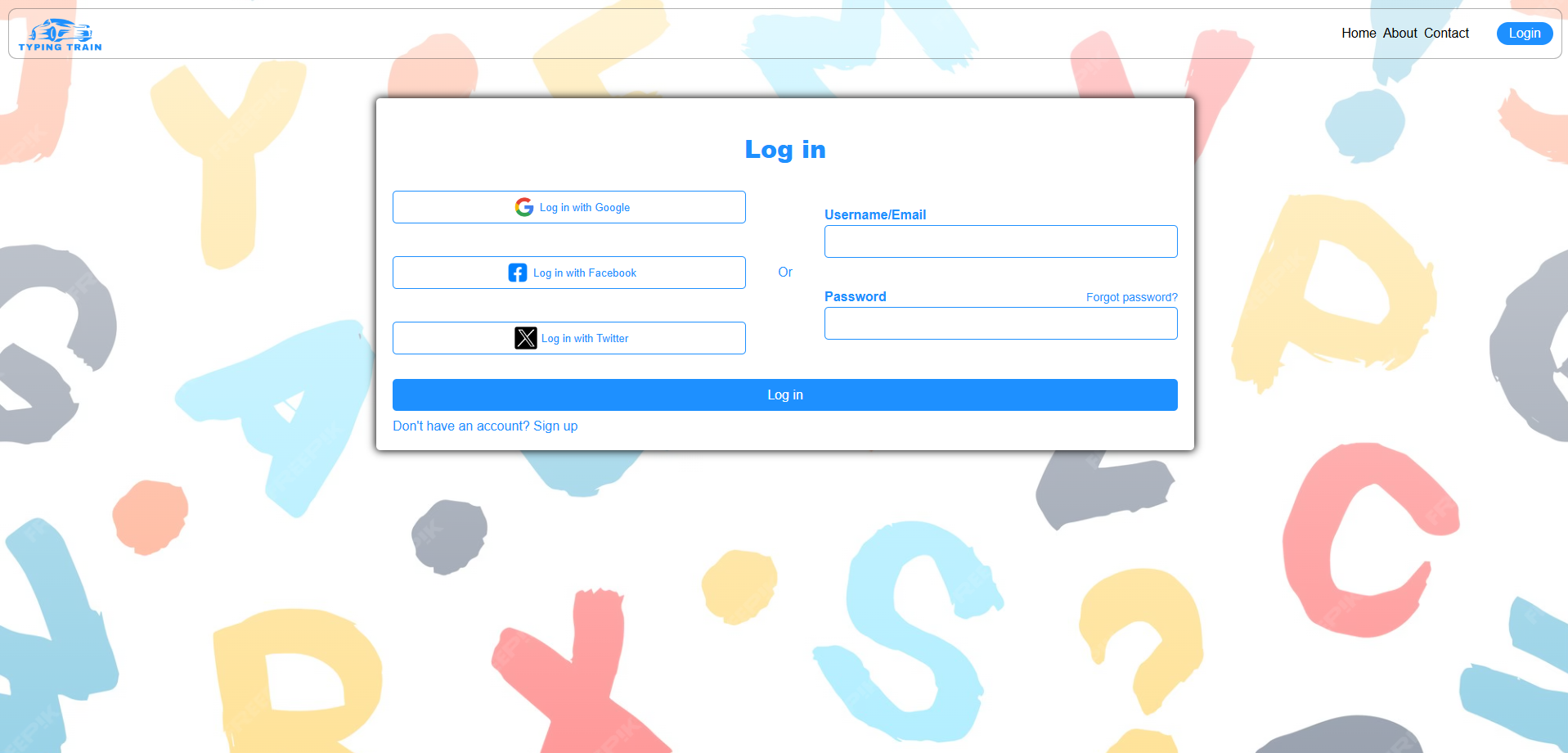


Figure 1 (Login page)

* State Management: Uses useRef to manage references to input fields (user\_id, user\_pass) for form validation.
* Form Handling: The handleSubmit function handles form submission and login validation.
* Routing with useNavigate: After successful login, useNavigate is used to redirect to the homepage (/).
* Reusable Button Component: The Button component is reused for different login methods (Google, Facebook, Twitter), making it modular.
* CSS Modules: Scoped CSS using LoginPage.module.css for styling, avoiding global style conflicts.

**References**

**1. Web Development and Frameworks**

* React Documentation: https://reactjs.org/docs/getting-started.html
  + Purpose: Use this to reference React's component-based approach and best practices for building interactive UIs. You can also cite it to support your choice of React for creating reusable components and responsive design.
* MDN Web Docs on HTML, CSS, JavaScript: <https://developer.mozilla.org/>
  + Purpose: This source provides guidelines on best practices in front-end web development, covering accessibility, responsive design, and UI/UX principles.

**2. Web Accessibility and User Experience**

* **Web Content Accessibility Guidelines (WCAG)**: [W3C WCAG Documentation](https://www.w3.org/WAI/standards-guidelines/wcag/)
  + Purpose: Referencing WCAG helps ensure the Typing Train site is accessible to users of all abilities. This source provides guidelines on color contrast, keyboard navigation, and readability, ensuring an inclusive user experience.

**3. Typing Practice and Gamification in Education**

* **Typing.com Blog and Resources:** [**Typing.com**](https://www.typing.com/)
  + *Purpose*: Referencing WCAG helps ensure the Typing Train site is accessible to users of all abilities. This source provides guidelines on color contrast, keyboard navigation, and readability, ensuring an inclusive user experience.

**5. Github Link**

* **https://github.com/Tarandeep9988/FEE-PROJECT**