## 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 Tarandeep Singh (2310992228)

**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 - 18 |
| 5. | References | 19 |

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 – Tarandeep Singh (2310992228)**

**Page created for EVALUATION-1**

**4. Results**

**4.1 Homepage**

This is the default page users will see when they enter our website.

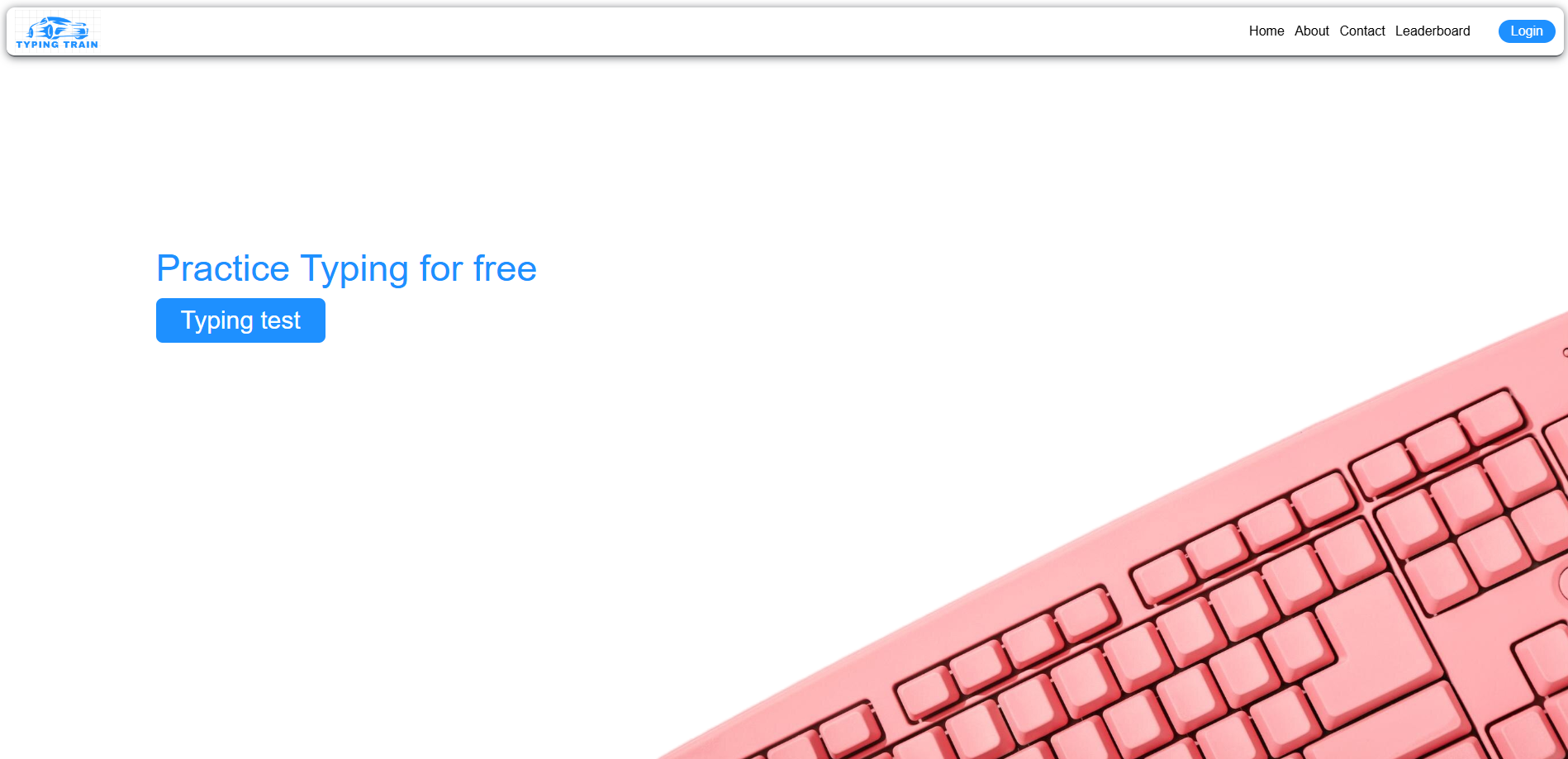


Figure 1 (Homepage)

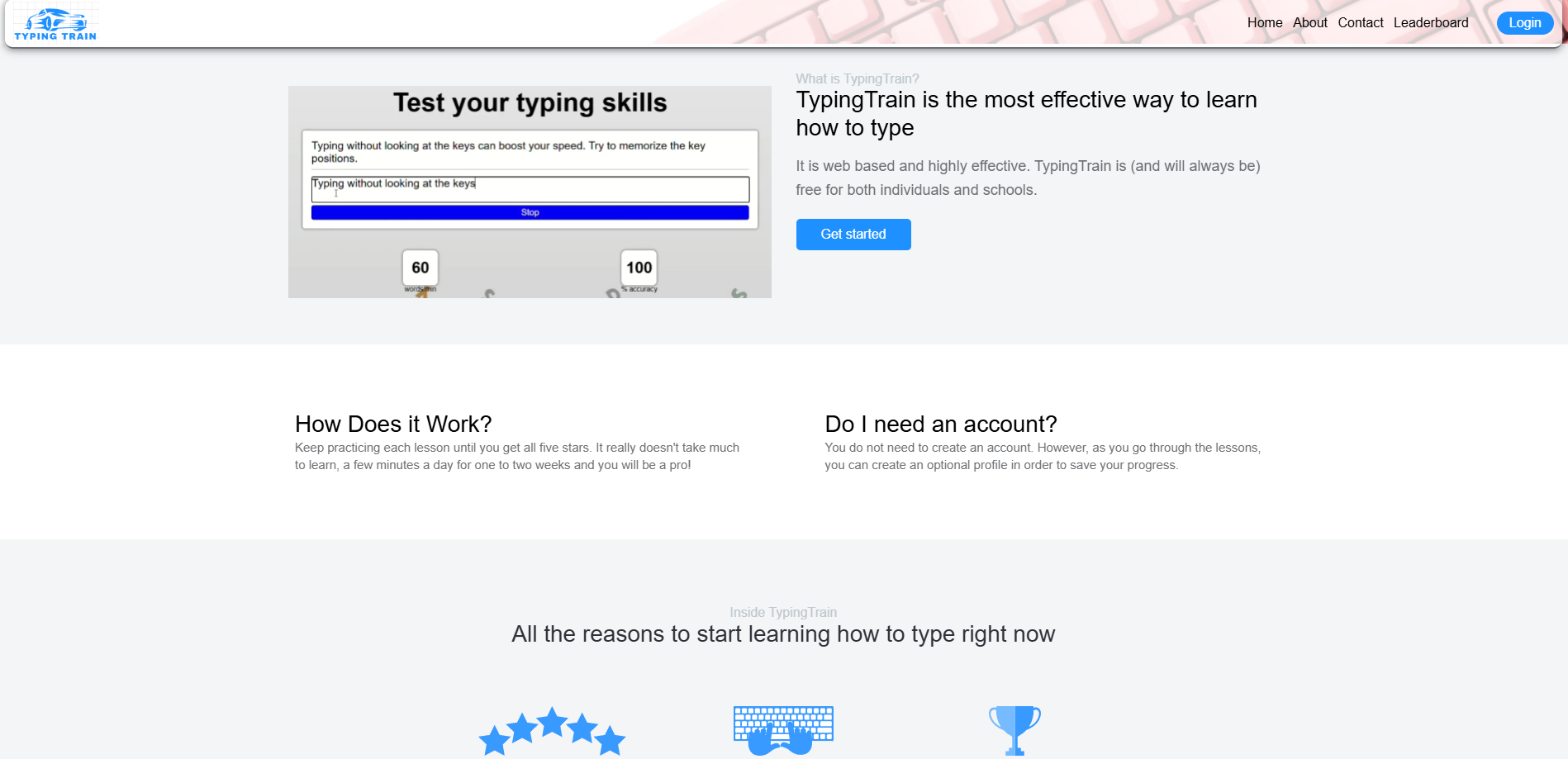


Figure 2 (Homepage)

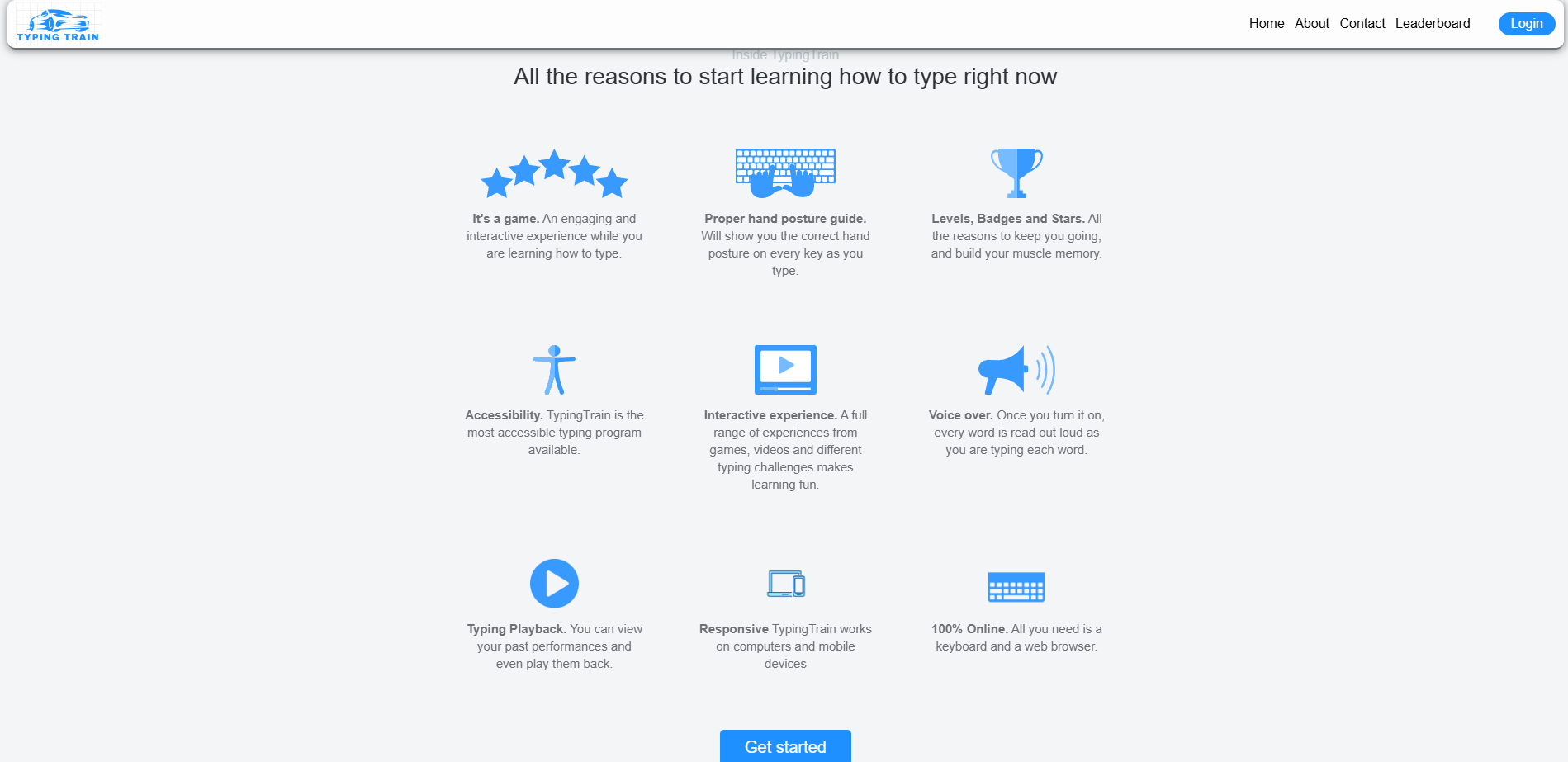


Figure 3 (Homepage: Features)

We have tried to make the Homepage simple and responsive so that our website can look user friendly and work on different devices.

* Header & Navigation: Links to Home, About, Contact, Leaderboard, and a Login button.
* Main Section: "Practice Typing for free" message with a Typing Test button.
* About Section: Video and description introducing Typing Train.
* How It Works: Explanation of practice and progress-tracking features.
* Features Section: Highlights of Typing Train’s benefits (e.g., interactivity, accessibility) with icons.
* Subscribe Section: Email input for subscription to updates.
* Styling: CSS classes control layout, typography, spacing, and responsive elements.

**4.2 Typing testing page**

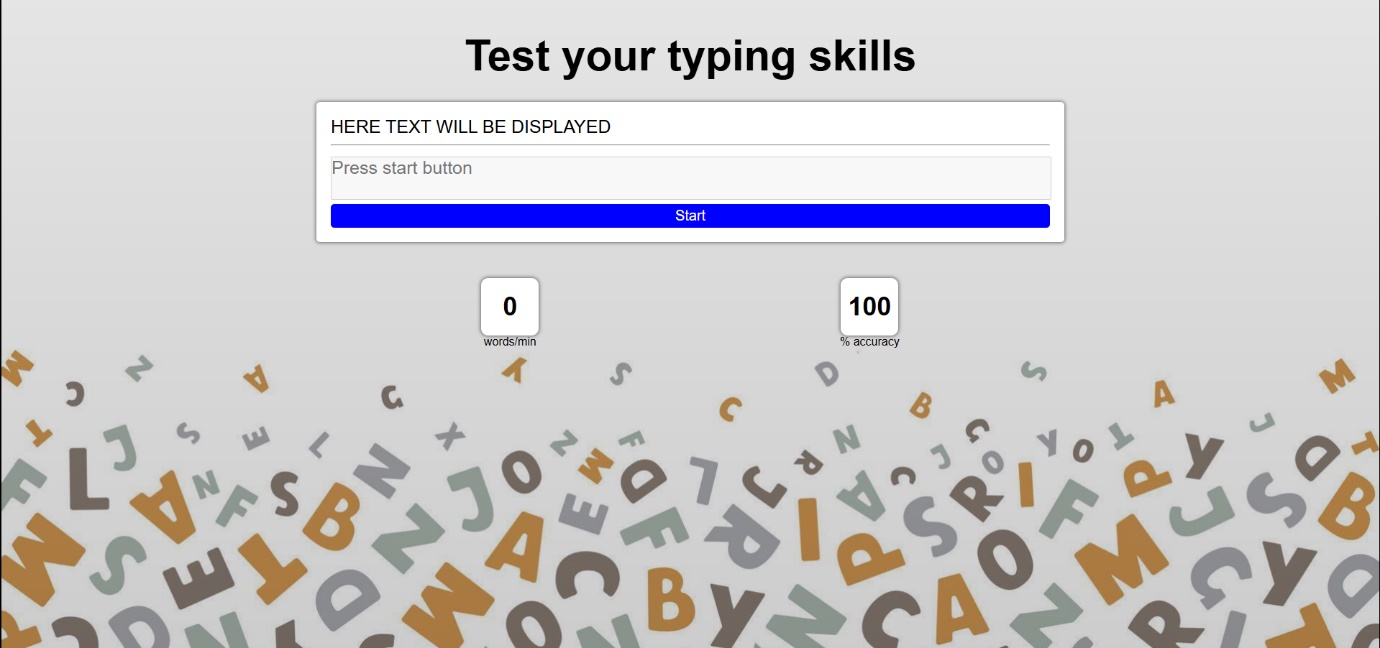
 This is the page on which users can practice typing :-

Figure 4 (Typing testing page)

It is just UI, no logic is implemented yet.

* Main Section: Displays a header for testing typing skills.
* Content Box: An empty box (id="content\_box") prepared to show text for the test.
* Input Field: A disabled textarea (id="input\_field") where users will type test content after pressing "Start."
* Start Button: A button to start the typing test.
* Result Boxes: Displays typing speed (id="speed") in words per minute and accuracy (id="accuracy") percentage, set to default values.
* Styling: CSS used for layout, typography, and box alignment.

**Group Member – Tarandeep Singh (2310992228)**

**Page created for EVALUATION-2**

**4. Results**

**4.1 Homepage**

This is the default page users will see when they enter our website.

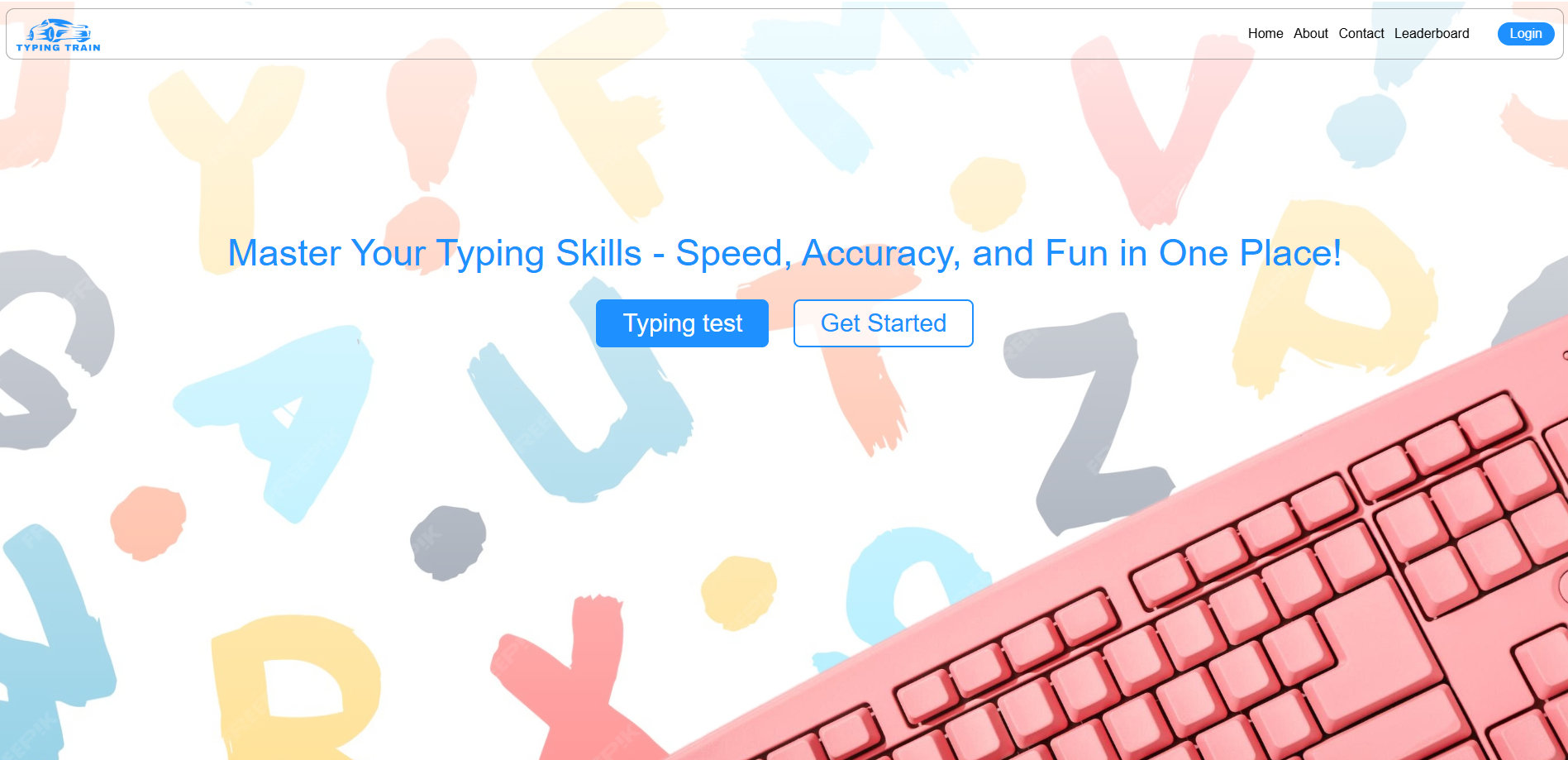


Figure 4 (Homepage)

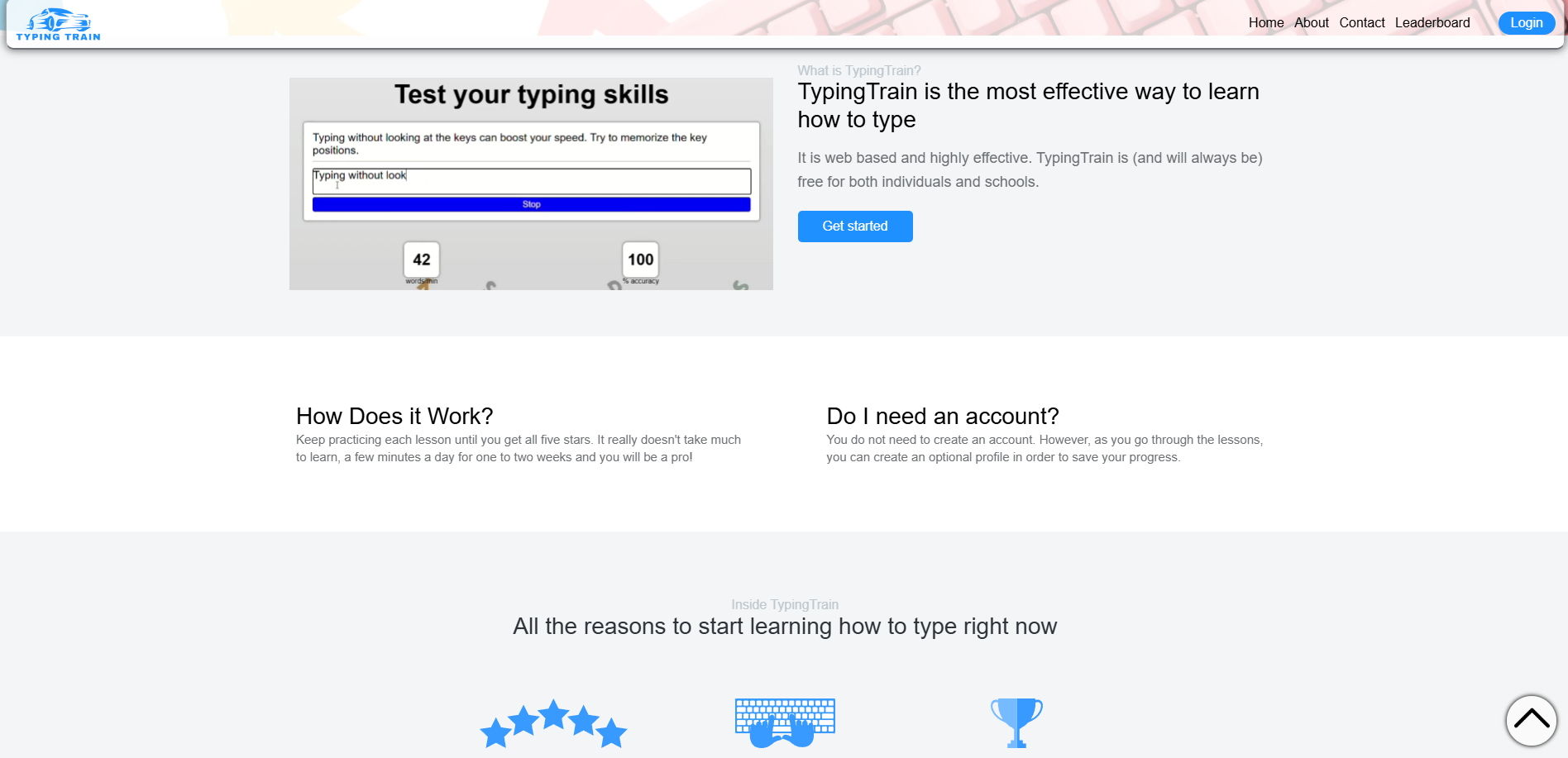


Figure 5 (Homepage)

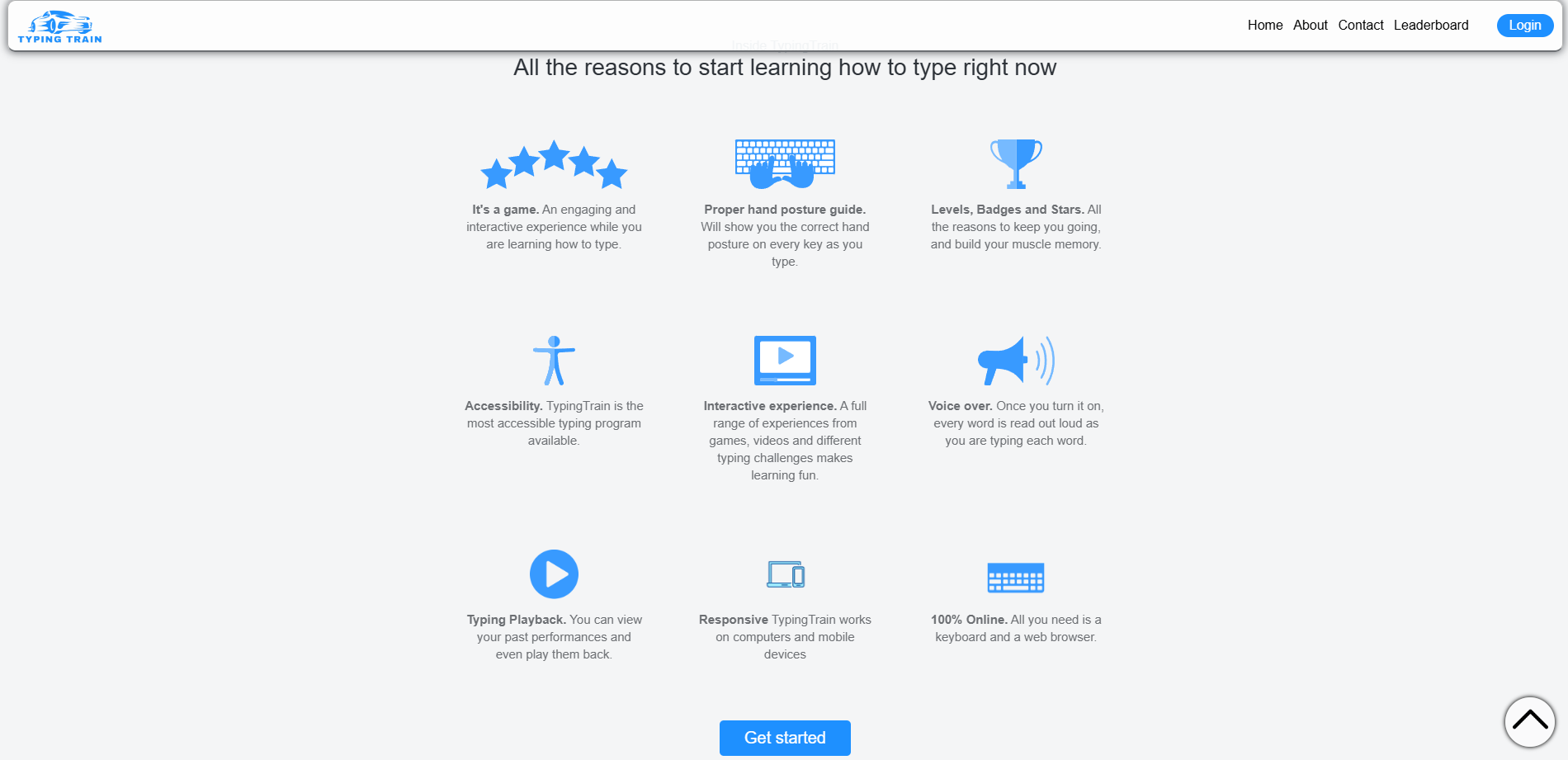


Figure (Homepage: Features)

We have tried to make the Homepage simple and responsive so that our website can look user friendly and work on different devices.

JavaScript Functionality

* Get Started Buttons: Redirects to levels.html.
* Subscribe Button: Shows an alert thanking the user for subscribing.
* Scroll Event: Adds a ‘scrolled’ class to the header if the user scrolls down.
* Displays the ‘to-top’ arrow with smooth scrolling functionality.

Login Button:

* Updates based on localStorage login status.
* Redirects to login.html if not logged in, or logs out if logged in, with an alert.

**4.2 Levels page**

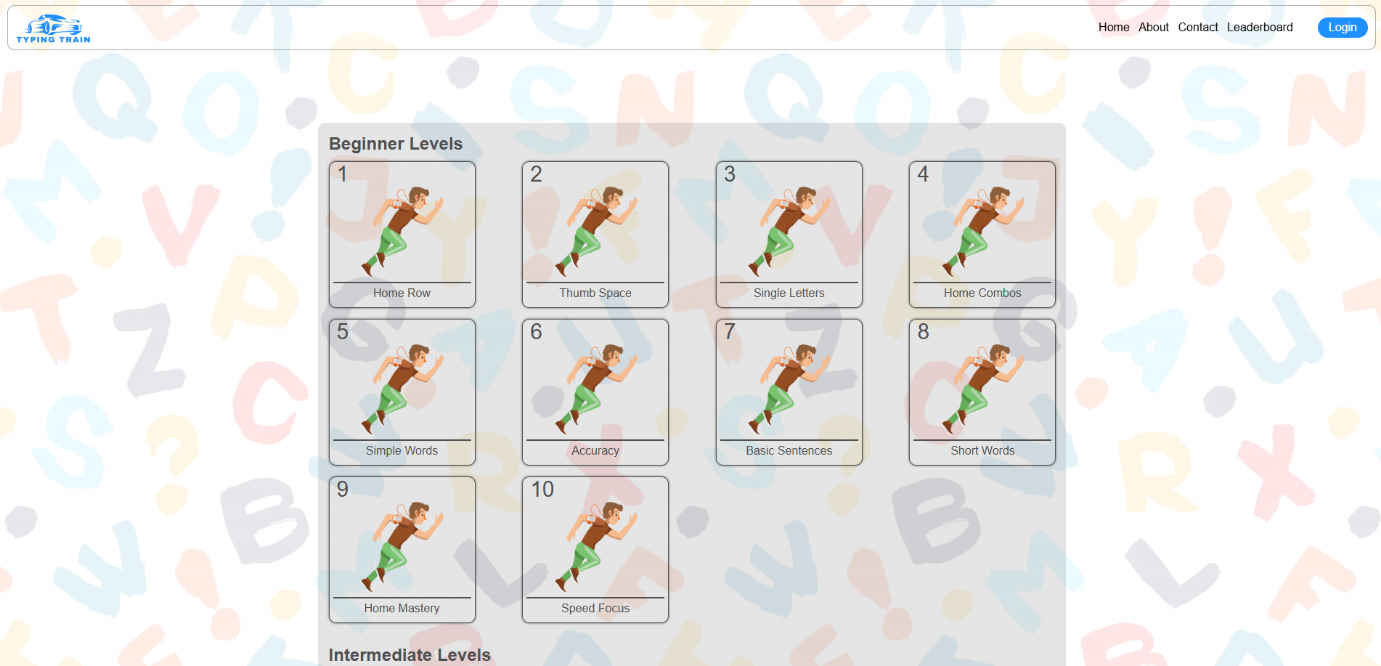
 This is the page on which users can choose different typing levels;

Figure 4 (Levels page)

* These levels are generated dynamically using JavaScript.
* Dynamic Level Generation: JavaScript generates levels (Beginner, Intermediate, Advanced) on the page using data imported from an external file, levels.js. Each level is rendered with a unique icon, name, and level number.
* Level Navigation with Local Storage: Each level box is clickable, storing specific content in localStorage and navigating the user to a typing test page.
* Scroll-Based Header Styling: Adds a .scrolled class to the header when the user scrolls, creating a visual change in the navigation bar.
* Scroll-To-Top Button: A button with an up arrow appears upon scrolling, which, when clicked, smoothly scrolls the user to the top of the page.
* Login Button State and Logic: Checks login status from localStorage to toggle the Login/Logout button text dynamically. Clicking on Logout resets login status, while Login redirects to the login page.

**4.3 Typing testing page**

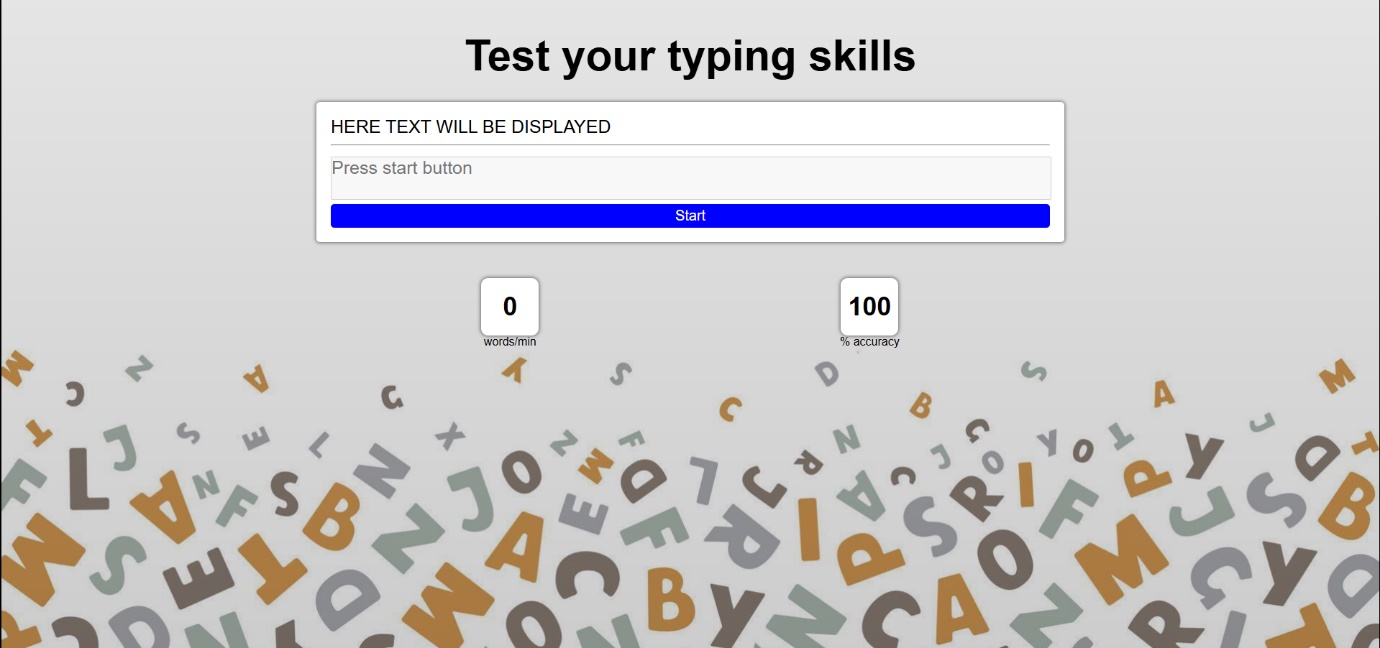
 This is the page on which users can practice typing :-

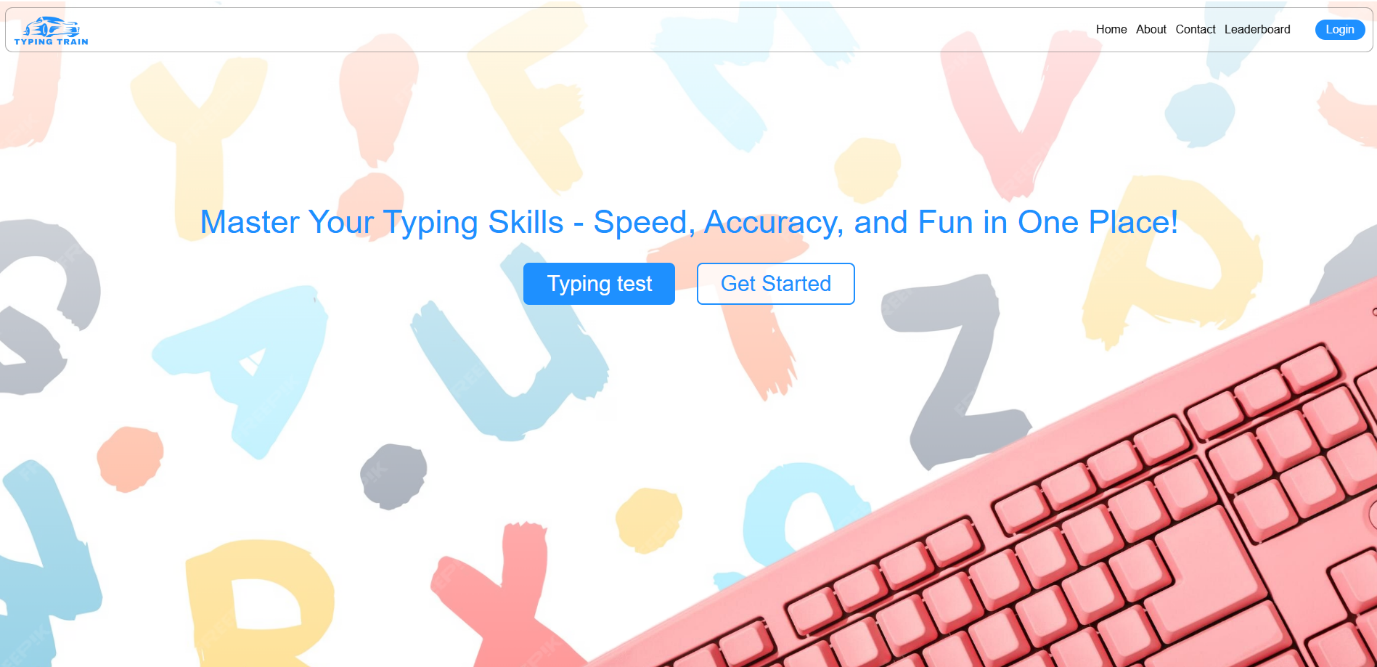
Figure 5 (Typing testing page)

* Typing testing logic is implemented.
* Test Interface: The page allows users to test their typing skills with a provided default text or content stored in local storage.
* Start/Stop Button: Toggles between starting and stopping the test.
* Dynamic Content: The text content for typing practice is retrieved from local storage using window.localStorage.getItem('content').
* Real-Time Speed Calculation: As users type, the app calculates and displays typing speed in words per minute (WPM).
* Accuracy Calculation: The app compares typed content to the original and calculates accuracy based on the number of errors.
* Visual Feedback: The input field background color turns red when errors are detected, providing immediate feedback.
* Disabling Input: The input field is disabled until the user clicks "Start," preventing accidental typing before the test begins.
* Button Behavior: The "Start" button begins the test, and "Stop" ends it. The button's text toggles between these two states.
* Local Storage: The content for the test is dynamically loaded from the local storage to allow users to practice specific text.

**Group Member – Tarandeep Singh (2310992228)**

**Page created for EVALUATION-3**

**4. Results**

**4.1 Homepage**

This is the default page users will see when they enter our website.

Figure 7 (Homepage)

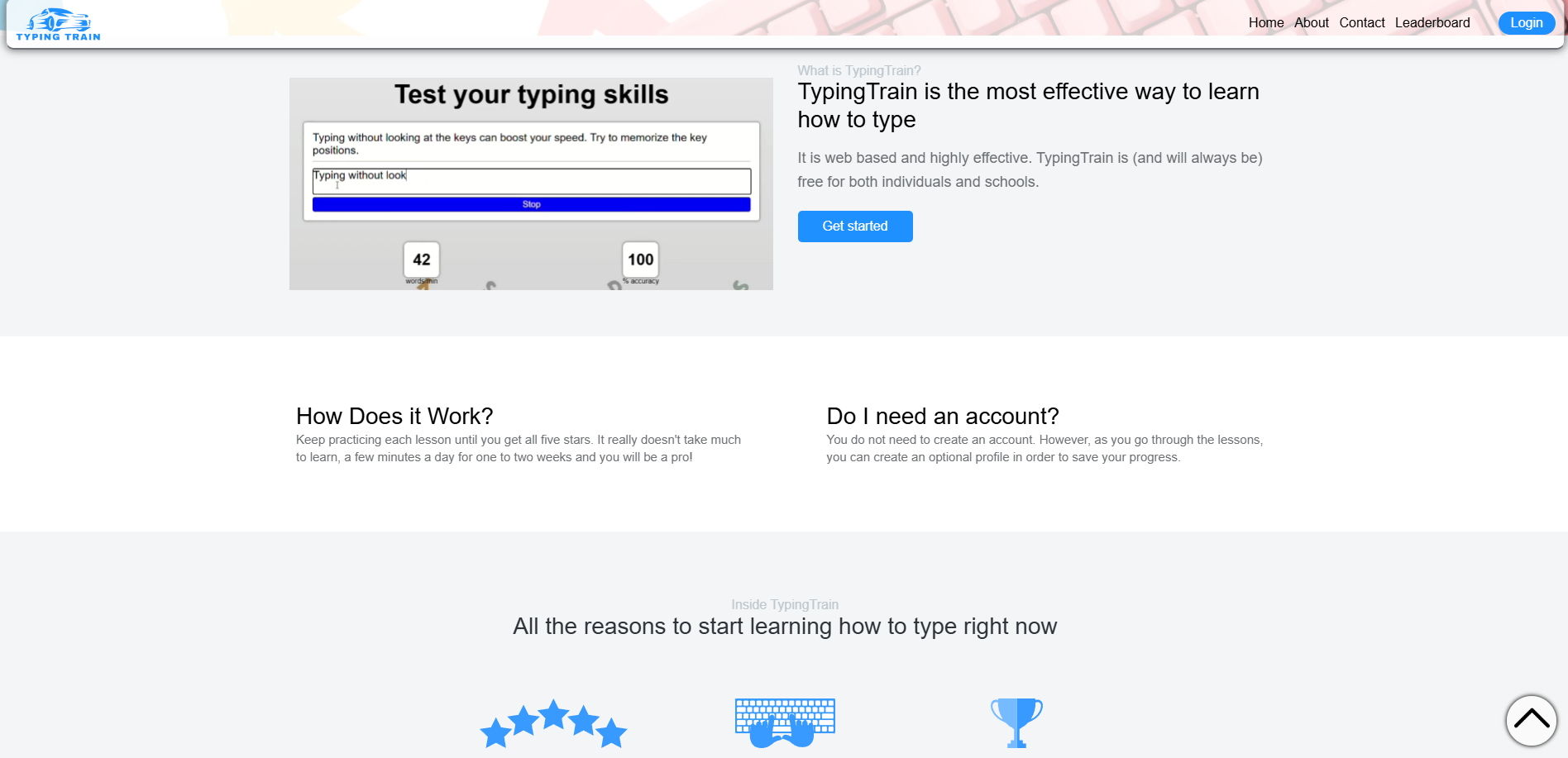


Figure 8 (Homepage)

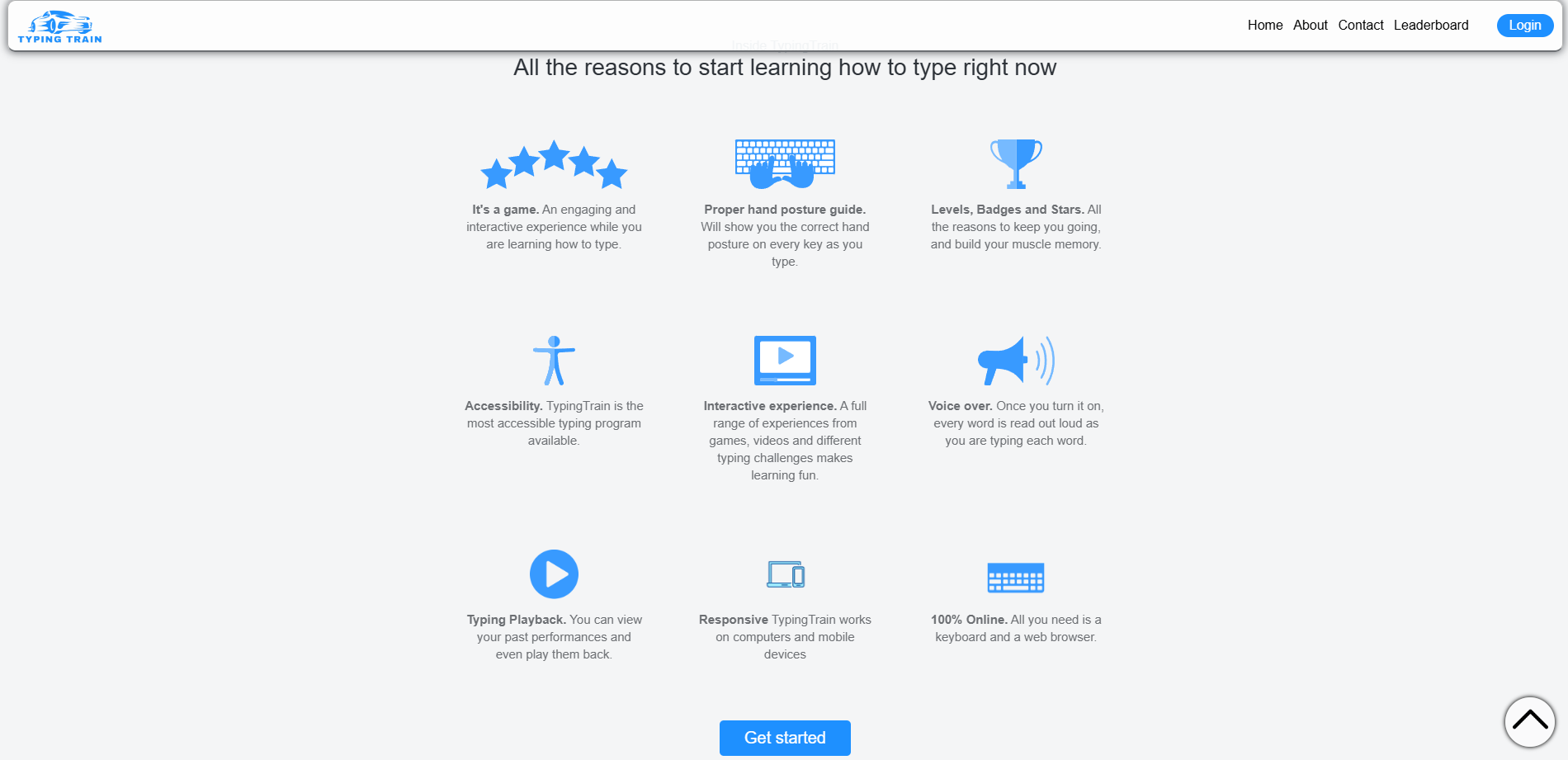


Figure 9 (Homepage: Features)

* We have tried to make the Homepage simple and responsive so that our website can look user friendly and work on different devices.
* App Component: Main entry point; imports and renders other components.
* Header Component: Contains logo, navigation links, and login button.
* MainSection Component: Displays heading, typing test link, and "Get Started" button.
* WhatIsTypingTrain Component: Displays a video and description about TypingTrain.
* HowItWorks Component: Explains how TypingTrain works and account creation.
* InsideTypingTrain Component: Lists features of TypingTrain with images and descriptions.
* Newsletter Component: Allows users to subscribe with their email.
* ToTopArrow Component: A button to scroll the page to the top.

**4.2 Levels page**

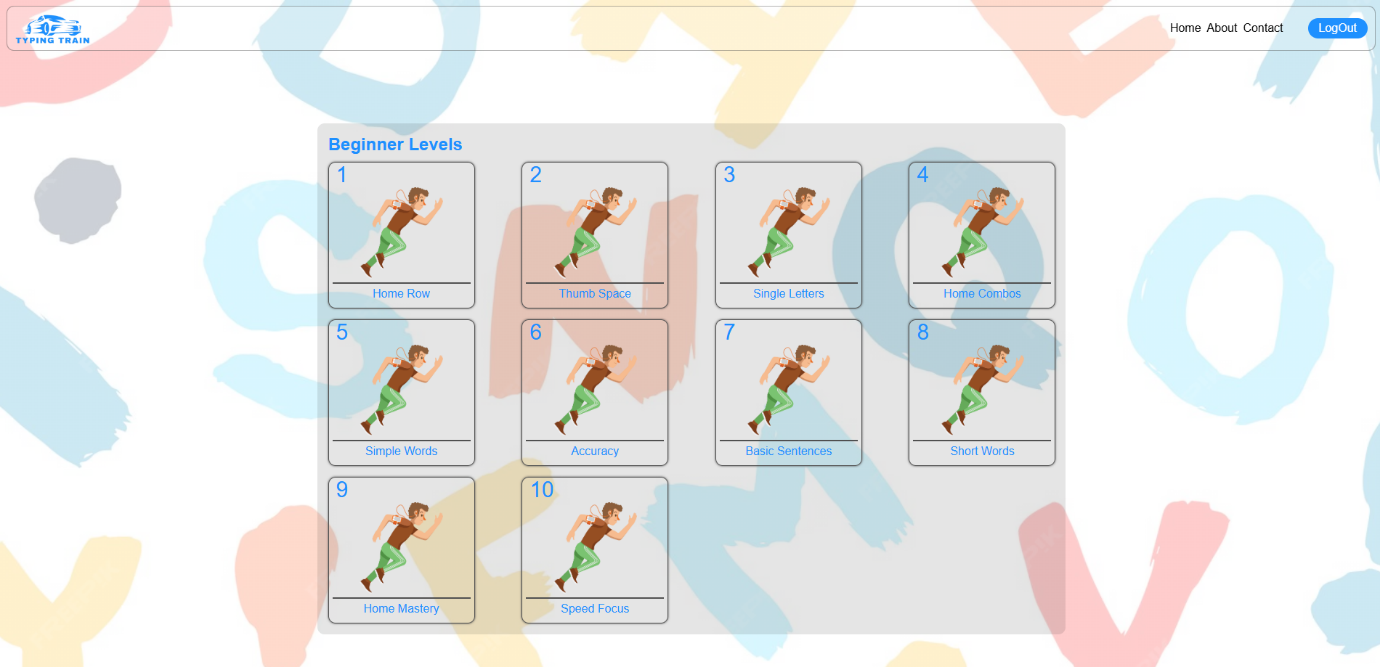
 This is the page on which users can choose different typing levels;

Figure 4 (Levels page)

* These levels are created using react components

LevelPage Component

* Displays three sections: Beginner Levels, Intermediate Levels, Advanced Levels.
* Loops through the level arrays (BeginnerLevels, IntermediateLevels, AdvancedLevels) using .map() to create LevelCard components for each level.
* Passes levelIconSrc, levelTitle, levelNumber, and levelContent as props to LevelCard.

LevelCard Component

* Displays a card with the level number, icon, and title.
* Navigates to the /typingtest page when clicked, passing levelContent as state.
* Uses useNavigate to handle navigation and startTypingTest function to trigger it.

**4.3 Typing testing page**

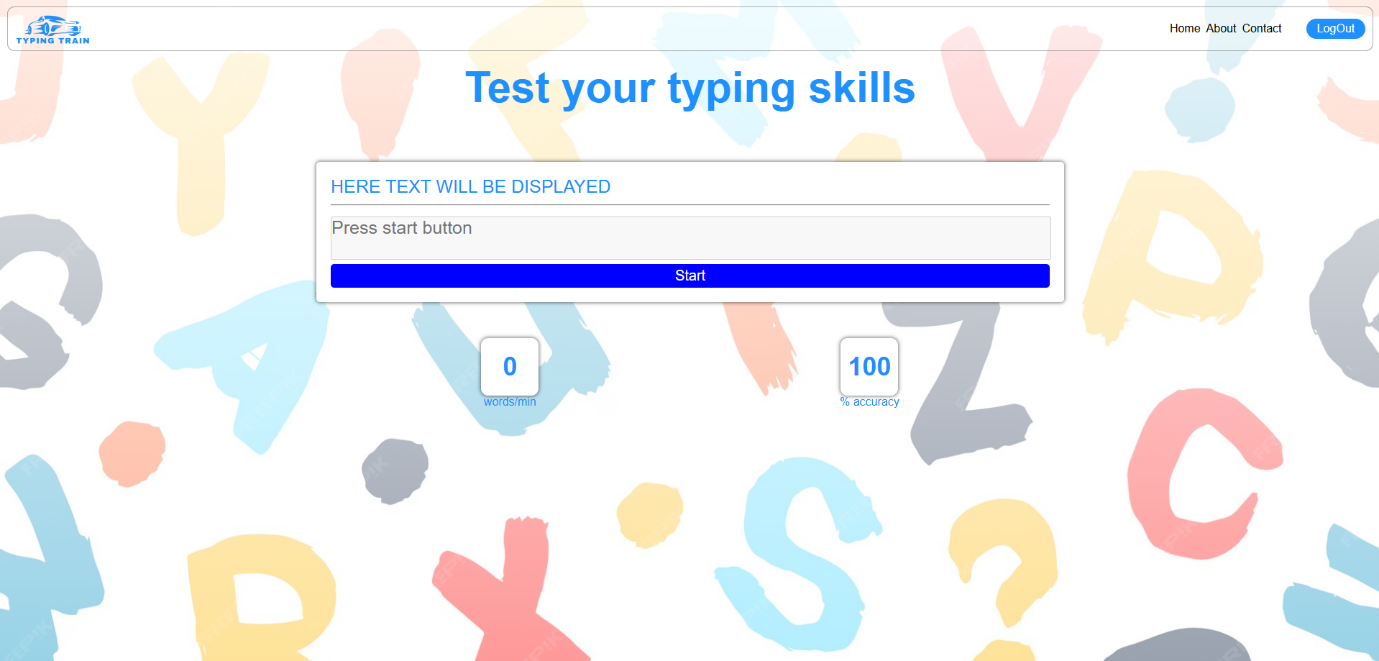
 This is the page on which users can practice typing :-

Figure 5 (Typing testing page)

React is used for its implementation.

**State Variables**:

* content: Text to be typed.
* speed: Typing speed in WPM.
* accuracy: Typing accuracy in percentage.
* startingTime: Time when test starts.
* Refs for button (start\_stop\_btn) and input field (input\_field).

**Functions**:

* getRandomPara: Returns a random typing paragraph.
* getContent: Returns content from state or random paragraph.
* startTypingTest: Starts the test, enables input field, sets content.
* stopTypingTest: Disables input field, resets button text.
* handleButtonclick: Toggles start/stop button.
* handleKeyboardInput: Calculates and updates speed and accuracy; stops test if completed.

**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**