Hiragana Quiz

Subject Name: **Front End Engineering**

Subject Code: **CS186**

Cluster **: iGamma**

Department : **DCSE**

Group :**G19**



Submitted By :

Shreya Banta

2110992008

G19

Submitted To:

Ms. Pritpal Kaur

INTRODUCTION

The Hiragana Quiz Application is an innovative and interactive tool designed to facilitate the learning and mastery of the Hiragana script, an essential component of the Japanese language. Hiragana is the foundation of written Japanese and is used for native Japanese words, verb endings, and various grammatical elements . The application combines elements of gamification with language learning, creating an enjoyable and motivating experience for users. Through the Hiragana Quiz Application, users can not only test their knowledge of Hiragana characters but also track their progress through streaks and receive instant feedback on their answers.

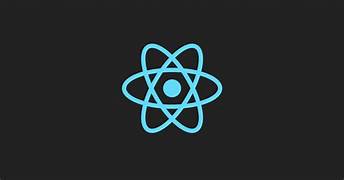


TECHNOLOGIES USED

CSS (Cascading Style Sheet) 

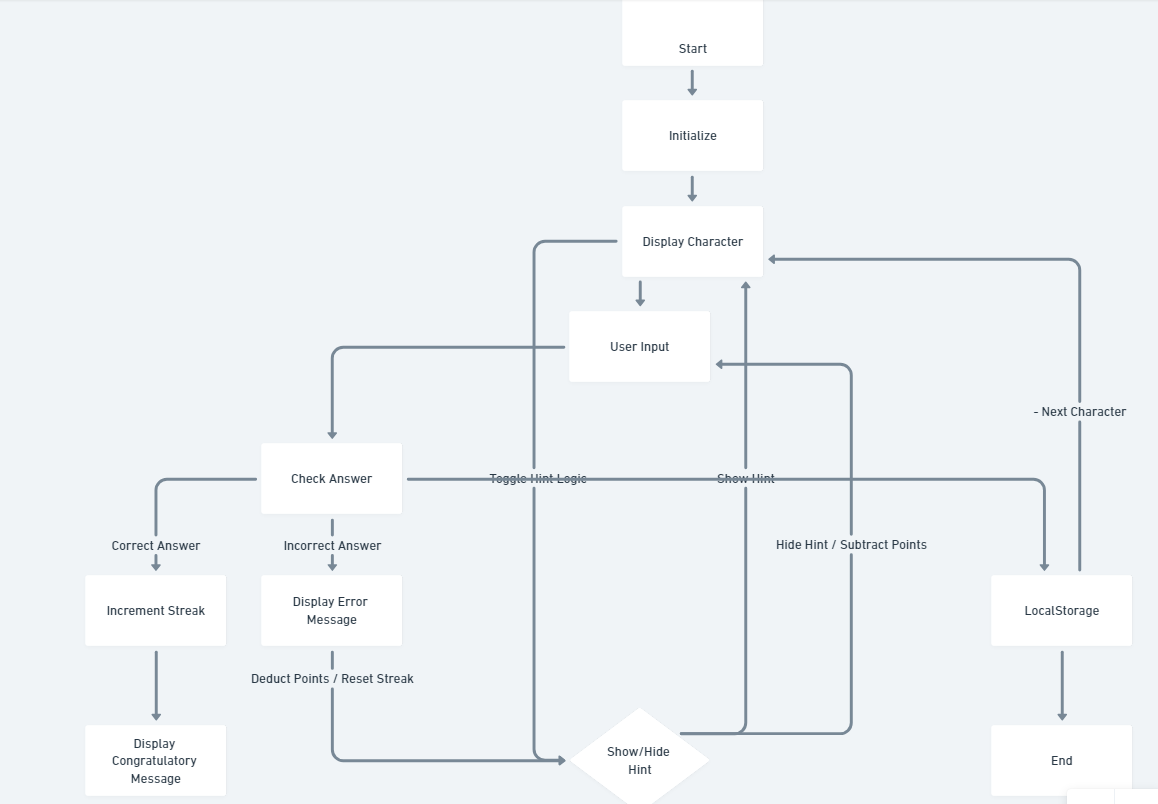
HTML (Hyper Text Markup Language ) 

JavaScript 



React

FLOW CHART



Working Of The Project

1. When the application loads, a random Hiragana character is displayed.

2. Users input the Romanized form in the text field and submit their answer.

3. If the answer is correct, the user's streak is increased and the next character is presented. If the answer is incorrect, the user's streak is reset, and the correct answer is shown.

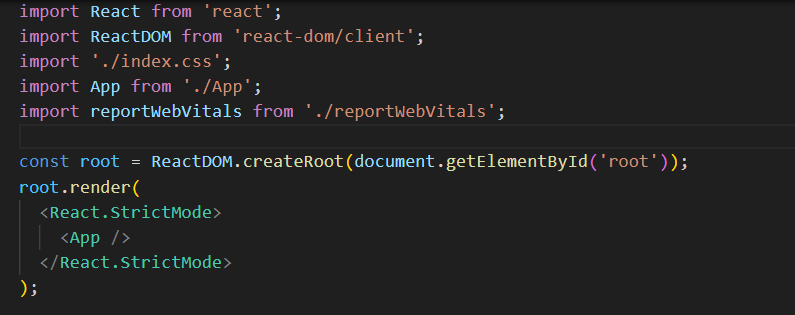
4. Users can click the "Show Hint" button to reveal a hint, but this deducts one point from their streak.

5. The user's streak and max streak are displayed on the screen.



🡨------------------------------Documentation------------------------------🡪

**Index.js**



The "index.js" file is the entry point for your React application. It initializes the application by rendering the root component, "App," into the root HTML element on the web page.

Import Statements: import ReactDOM from "react-dom/client"; import App from "./App"; import { StrictMode } from "react";

Root Element:

The `createRoot` method is used to create a root element for rendering the React application. It selects the HTML element with the id "root" as the container for the application.

const root = ReactDOM.createRoot(document.getElementById("root"));

Rendering the Application:

The `render` method is used to render the "App" component within a `<StrictMode>`. `<StrictMode>` is a wrapper component provided by React that performs additional runtime checks and helps identify potential issues in your application.

root.render(

<StrictMode>

<App />

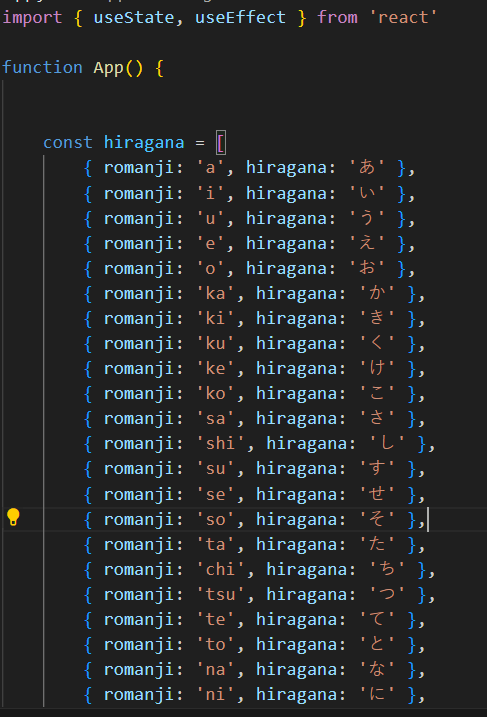
</StrictMode>

);

1. **App.js**

App.js is the main component of a React application. It serves as the entry point of the application and is responsible for rendering the user interface, handling user interactions, and managing the state of the Hiragana quiz.

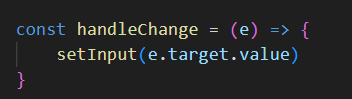
* Rendering User Interface



* ***Managing State***

The component uses the useState hook to manage various state variables. These variables include the user's input, the current character being displayed, the streak, maximum streak, error messages, and whether hints should be shown. These states are crucial for tracking and updating the user's progress and interactions with the quiz.

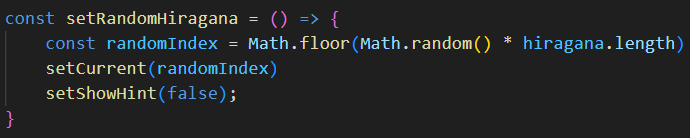
* ***Handling User Input***: The handleChange function is responsible for capturing and updating the user's input as they type their answers into the input field.



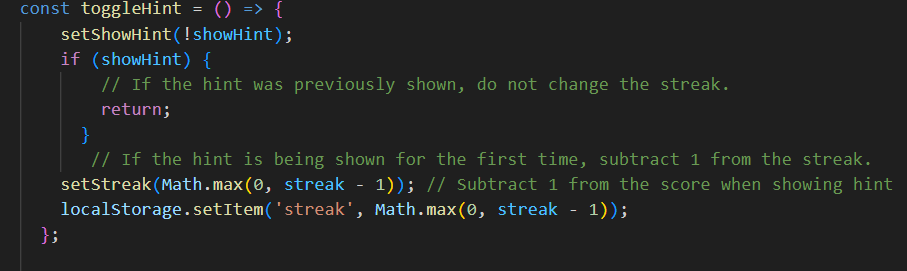
* **Handling Form Submission**: The handleSubmit function is triggered when the user submits their answer. It checks if the answer is correct, updates the streak, and provides feedback to the user. If the answer is incorrect, it displays an error message. Correct answers increase the streak, and the streak and maximum streak are stored in local storage.



* **Random Character Selection**



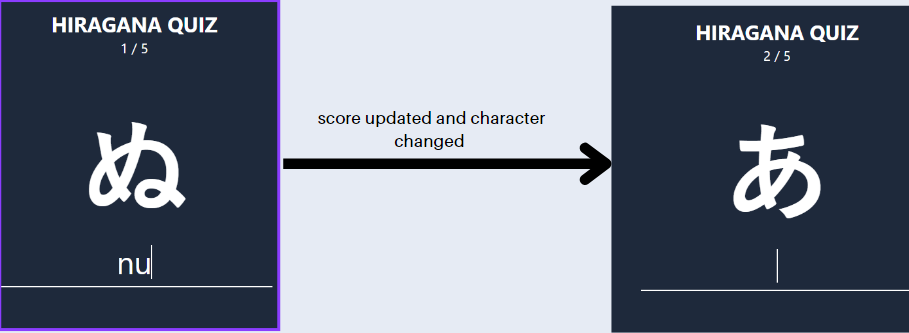
* **Showing Hints**: The toggleHint function allows the user to toggle the display of hints. It can also decrease the user's streak if they choose to view a hint.



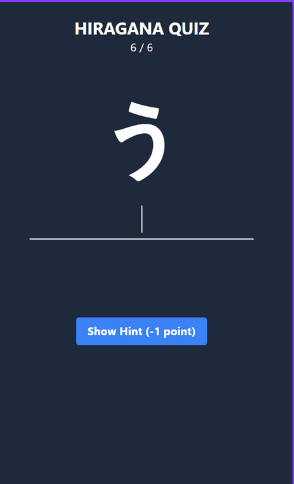
Local Storage This allows the application to remember the

user's progress between sessions

* **Rendering the User Interface**: The return statement within the render function specifies how the UI elements, including the character to be guessed, the input field, error messages, and hint buttons, are displayed to the user.



CONCLUSION



Hiragana Quiz Application represents a significant step forward in making the learning of the Japanese Hiragana characters more engaging and accessible. The project successfully addressed its primary objectives, which were to create an interactive tool for learning and practicing Hiragana, enable users to test their knowledge, offer hints and feedback, and track user progress. Looking forward, there is ample room for improvement and expansion of the Hiragana Quiz Application. Adding more levels of complexity, integrating audio pronunciation, supporting additional character sets, and incorporating user authentication for progress tracking are among the potential future enhancements.