

Task: Advanced Responsive Interactive Quiz Application:

Requirements:

HTML Structure:

Landing Page:

Implement a visually appealing and responsive design.

Include a "Start Quiz" button that triggers the quiz.

Add a section for displaying instructions and rules for the quiz.

Incorporate a high scores section that displays the top 5 quiz scores.

Quiz Page:

Design a clean and intuitive layout for the quiz questions.

Implement a progress bar that dynamically updates as the user moves through the quiz.

Integrate a timer that counts down the remaining time for the quiz.

Provide a "Pause/Resume" button for the timer.

Implement a "Back" button that allows the user to navigate to the previous question.

Include a "Submit" button that the user can click to submit their final answers.

Results Page:

Display the user's final score, along with the total number of questions.

Show the correct answers for each question, highlighting the user's answers.

Implement a "Try Again" button that allows the user to restart the quiz.

Add an option for the user to save their score to the high scores section on the landing page.

CSS Styling:

Responsive Design:

Ensure the application is fully responsive, providing a seamless experience on different screen sizes.

Optimize the layout and typography for mobile devices.

Visual Appeal:

Incorporate a modern and visually appealing design with a cohesive color scheme.

Use subtle animations and transitions to enhance the user experience.

Implement hover effects on interactive elements, such as buttons and quiz answers.

Section Distinction:

Clearly differentiate the landing page, quiz questions, and results page using distinct styles.

Utilize CSS grid or flexbox to create a clean and organized layout.

JavaScript Functionality:

Quiz Logic:

Implement a randomization mechanism for the quiz questions to ensure a unique experience for each attempt.

Provide dynamic feedback for correct and incorrect answers, such as displaying a success/error message.

Integrate a scoring system that awards points based on the user's performance.

Timer Functionality:

Implement a countdown timer that adjusts the remaining time based on the user's actions (e.g., pausing, resuming).

Trigger a warning or visual cue when the timer is close to expiring.

Automatically submit the quiz and display the results page when the timer runs out.

High Scores:

Store the high scores in the user's browser using local storage or a simple server-side implementation.

Implement a mechanism to update the high scores list when a new high score is achieved.

Display the high scores on the landing page in a visually appealing manner.

User Interactions:

Handle the "Back" button functionality, allowing the user to navigate to the previous question.

Implement the "Pause/Resume" button for the timer, updating the UI accordingly.

Ensure a smooth transition between the landing page, quiz page, and results page.