

Front-End UI/UX Mini Project Report

Project Title: Interactive Quiz Application

- **Submitted By:**
 - **Team Members**
 - Aaron V Shibu (2462002)
 - Alan (2462021)
 - Albert B Varghese (2462024)
 - **College-E-mailid:**
 - aaron.v@btech.christuniversity.in
 - alan.j@btech.christuniversity.in
 - albert.b@btech.christuniversity.in
- **Course:** UI/UX Design Fundamentals
- **Instructor Name:** Mrs Nagaveena
- **Institution:** Christ University
- **Date of Submission:** 25/09/2025

2. Abstract

This project is an interactive, web-based quiz application built using HTML, CSS, and JavaScript (with jQuery). The primary goal was to create an engaging and user-friendly quiz experience. The application presents users with a series of multiple-choice questions, provides immediate feedback on their answers, and displays a final score upon completion. The focus was on creating a clean, responsive interface with intuitive navigation and clear visual cues to enhance the user experience.

3. Objectives

- To design and develop a user-friendly and interactive quiz interface.
- To implement core quiz functionalities including question display, option selection, and scoring.
- To create a visually appealing and responsive layout that works across different devices.
- To provide immediate visual feedback to the user for correct and incorrect answers.
- To manage the quiz flow with features like a countdown timer and progress tracking.

4. Scope of the Project

- The project is focused on the front-end implementation of a quiz application.
- It utilizes client-side JavaScript for all logic and does not involve any server-side integration or databases.
- The quiz content (questions and answers) is hardcoded into a JavaScript array.
- The application is designed to be viewed on modern web browsers on desktop, tablet, and mobile devices.

5. Tools & Technologies Used

Tool/Technology	Purpose
HTML5	Markup and content structure.
CSS3	Styling, layout, and responsiveness.
JavaScript (jQuery)	DOM manipulation and application logic.
Bootstrap	CSS framework for styling components.

Google Fonts Custom typography for the interface.

VS Code Code editor.

Chrome DevTools Testing and debugging.

6. HTML Structure Overview

- The structure is a single-page application with three main div containers representing different screens: #start-screen, #quiz-screen, and #results-screen.
- JavaScript is used to toggle the visibility of these screens to create the application flow.
- Semantic HTML is used where appropriate, with clear and descriptive IDs for elements that are manipulated by JavaScript.

7. CSS Styling Strategy

- An internal CSS stylesheet is used for all custom styling.
- The design employs modern CSS properties like linear-gradient for backgrounds and box-shadow for depth.
- CSS animations (@keyframes) are used to create a fade-in effect for questions, making the transition smoother.
- The styling strategy focuses on a clean and visually appealing aesthetic, with a color palette that enhances readability and user engagement.
- Pseudo-elements (::after) are used to display feedback icons on the option buttons.

8. Key Features

Feature	Description
Interactive Quiz Flow	A multi-screen experience that guides the user from a start screen, through the questions, to the results.
Countdown Timer	Each question has a 15-second timer to add a level of challenge.
Progress Bar	Visually indicates the user's progress through the quiz.
Instant Feedback	Options are immediately highlighted as correct or incorrect upon selection.

Responsive Design	The layout adapts to various screen sizes, ensuring a good user experience on mobile and desktop devices.
--------------------------	---

9. Challenges Faced & Solutions

Challenge	Solution
Resetting CSS animations on new questions	Used JavaScript to temporarily remove the animation property and then re-apply it after a reflow to restart the animation.
Managing the quiz state and flow	Implemented functions to control the visibility of different screens and used a variable to track the current question index.
Handling user interactions and timers	Used jQuery event listeners for button clicks and setInterval/clearInterval to manage the countdown timer.

10. Outcome

- Successfully developed a fully functional and interactive quiz application.
- Achieved a clean, responsive, and visually engaging user interface.
- Gained practical experience in manipulating the DOM with JavaScript and jQuery.
- Learned how to manage application state and create a seamless single-page user experience.

11. Future Enhancements

- Load quiz questions from an external API or JSON file to make the content more dynamic.
- Add different types of questions, such as fill-in-the-blanks or true/false.
- Implement user authentication and store high scores in a database to create a leaderboard.
- Introduce categories for quizzes, allowing users to choose topics.

12. Sample Code

```
</head>
<body>

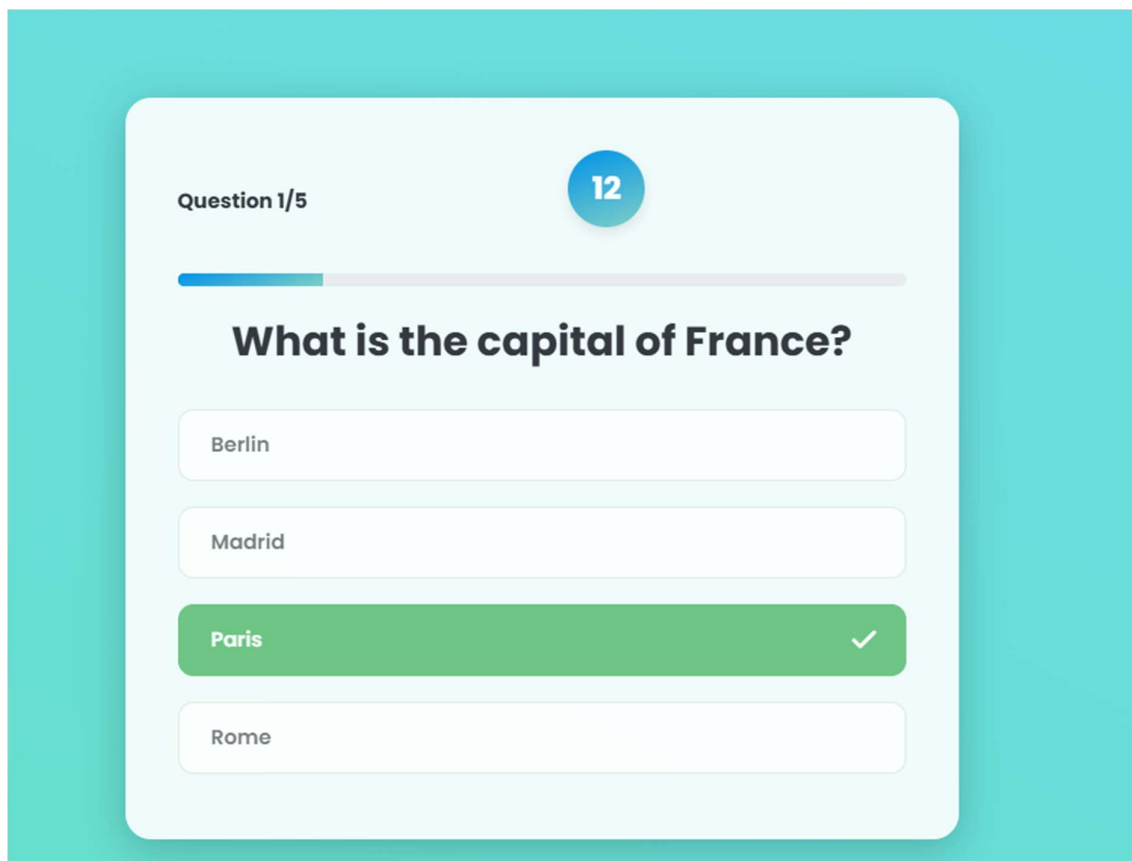
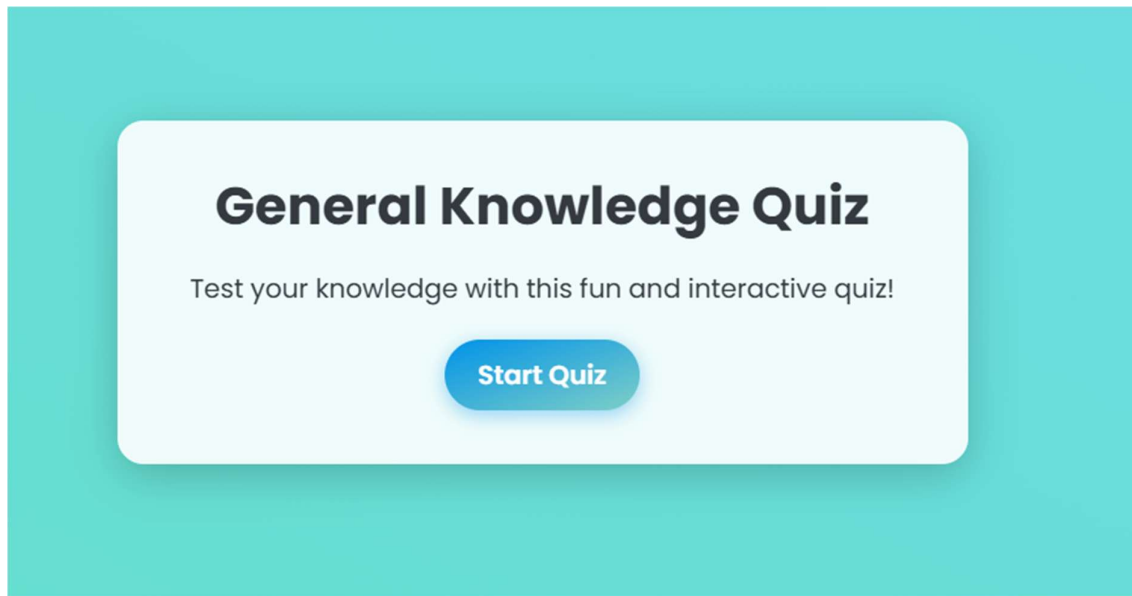
  <div class="container">
    <div class="quiz-container text-center">
      <!-- Start Screen -->
      <div id="start-screen">
        <h1 class="mb-4">General Knowledge Quiz</h1>
        <p class="lead mb-4">Test your knowledge with this fun and interactive quiz!</p>
        <button id="start-btn" class="btn btn-custom btn-lg">Start Quiz</button>
      </div>

      <!-- Quiz Screen -->
      <div id="quiz-screen" class="d-none">
        <div class="d-flex justify-content-between align-items-center mb-3">
          <div id="progress-text" class="font-weight-bold">Question 1/5</div>
          <div class="timer" id="timer">15</div>
        </div>
        <div class="progress mb-4" style="height: 10px; border-radius: 5px;">
          <div id="progress-bar" class="progress-bar" role="progressbar" style="width: 20%;" aria-valuenow="20" aria-valuemin="0" aria-valuemax="100"></div>
        </div>
        <h2 id="question" class="mb-4 font-weight-bold">Question text goes here</h2>
        <div id="options" class="list-group">
          <!-- Options will be dynamically inserted here -->
        </div>
      </div>

      <!-- Results Screen -->
      <div id="results-screen" class="d-none">
        <h2>Quiz Complete!</h2>
        <p class="lead">You've reached the end of the quiz.</p>
        <h3 class="my-4">Your Score: <span id="score"></span></h3>
        <button id="restart-btn" class="btn btn-custom">Play Again</button>
      </div>
    </div>
  </div>
```

```
<!-- jQuery and Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.bundle.min.js"></script>
<script>
  $(document).ready(function() {
    const quizData = [
      {
        question: "What is the capital of France?",
        options: ["Berlin", "Madrid", "Paris", "Rome"],
        answer: "Paris"
      },
      {
        question: "Which planet is known as the Red Planet?",
        options: ["Earth", "Mars", "Jupiter", "Venus"],
        answer: "Mars"
      },
      {
        question: "Who wrote the play 'Romeo and Juliet'?",
        options: ["Charles Dickens", "William Shakespeare", "Jane Austen", "Mark Twain"],
        answer: "William Shakespeare"
      },
      {
        question: "What is the largest ocean on Earth?",
        options: ["Atlantic Ocean", "Indian Ocean", "Arctic Ocean", "Pacific Ocean"],
        answer: "Pacific Ocean"
      },
      {
        question: "What is the chemical symbol for water?",
        options: ["O2", "H2O", "CO2", "NaCl"],
        answer: "H2O"
      }
    ]
  })
</script>
```

13. Screenshots of Final Output



14. Conclusion

This project was a valuable exercise in front-end web development. It provided hands-on experience in combining HTML, CSS, and JavaScript to create a functional and interactive web application. The process reinforced the importance of a well-structured HTML document, the power of CSS for creating modern user interfaces, and the essential role of JavaScript in bringing a web page to life. The final result is a polished and engaging quiz application that successfully meets all the initial objectives.

15. References

- [MDN Web Docs](#)
- [jQuery API Documentation](#)
- [Bootstrap Documentation](#)