**🧾 Project Report: Interactive Quiz Application**

**📌 Project Title:**

**Interactive Quiz App using HTML, CSS, and JavaScript**

**🎯 Objective:**

To design and develop an **interactive, responsive quiz web application** that allows users to test their knowledge in a fun and engaging way. This project demonstrates real-time user interaction, conditional rendering of elements, score calculation, and DOM manipulation using JavaScript.

**🛠️ Tools & Technologies Used:**

* **HTML5** – For creating the structure of the quiz layout.
* **CSS3** – For designing a responsive and visually appealing user interface.
* **JavaScript (Vanilla)** – For dynamic content, question logic, answer validation, and score tracking.

**🌟 Features:**

* Multiple-choice questions with one correct answer.
* Instant feedback on user selections (e.g., color highlighting).
* Score tracking and display of final results.
* "Next" button to navigate through questions.
* Restart/Play Again functionality.
* Optional: Timer, progress bar, or categories.

**🧩 Application Flow:**

1. **Quiz Initialization**:
   * Questions are stored in an array of objects.
   * The first question loads when the page starts or the user clicks “Start Quiz.”
2. **User Interaction**:
   * User selects one answer from the options.
   * The app checks the selected answer and updates the score.
   * Correct/Incorrect feedback is shown instantly.
   * "Next" button loads the next question.
3. **Quiz Completion**:
   * Once all questions are answered, the final score is shown.
   * Option to restart the quiz from the beginning.

**📄 Code Structure:**

**1. HTML (index.html):**

Creates the main container, question area, options, and controls.

html

CopyEdit

<div class="quiz-container">

<h1>Quiz Time!</h1>

<div id="question">Question text here</div>

<div id="options">

<!-- Options inserted dynamically -->

</div>

<button id="next-btn">Next</button>

<div id="result"></div>

</div>

**2. CSS (style.css):**

Adds styling for quiz elements, responsive layout, and button effects.

css

CopyEdit

.quiz-container {

max-width: 600px;

margin: auto;

padding: 20px;

background: #f4f4f4;

border-radius: 10px;

box-shadow: 0 0 10px rgba(0,0,0,0.1);

}

.option {

padding: 10px;

margin: 10px 0;

background: #e0e0e0;

cursor: pointer;

border-radius: 5px;

}

.option.correct {

background-color: #a3e635;

}

.option.incorrect {

background-color: #f87171;

}

**3. JavaScript (script.js):**

Handles logic for questions, answers, score, and event listeners.

javascript

CopyEdit

const questions = [

{

question: "What is the capital of India?",

options: ["Mumbai", "Delhi", "Kolkata", "Chennai"],

answer: "Delhi"

},

// Add more questions

];

let currentQuestion = 0;

let score = 0;

function loadQuestion() {

// Display question and options

}

function checkAnswer(selectedOption) {

// Validate answer and update score

}

function showResult() {

// Show final score

}

**🧪 Testing:**

Tested for:

* Accurate score calculation.
* Proper question navigation.
* Responsive UI across devices.
* Edge cases like clicking multiple times or skipping answers.

**🚀 Future Enhancements:**

* Add categories or difficulty levels.
* Store high scores using **localStorage**.
* Add a **countdown timer** per question.
* Fetch questions dynamically using **APIs** (e.g., Open Trivia DB).
* Add animations and audio feedback.

**📚 Learning Outcomes:**

* Gained hands-on experience with JavaScript DOM manipulation.
* Understood user interaction handling and conditional rendering.
* Learned to separate logic, UI, and styling for better maintainability.
* Built a complete mini web app using only HTML, CSS, and JS.

**📷 Screenshots: *(Optional)***

* Question screen
* Answer selection
* Final result display