Web Dev I Lab Assignment 4

Prompt Quizzer – A Quiz Game

Problem Statement

Create a simple JavaScript quiz game that executes entirely within the browser console. The game should use a predefined set of questions and corresponding answers stored in an array. The program will loop through each question using a for loop, prompt the user for input with prompt(), and provide immediate feedback using alert(). User input should be processed with toLowerCase() and trim() to ensure accurate comparison regardless of case or extra spaces. The program must keep track of the user's score throughout the quiz and display the final score at the end. This assignment reinforces the use of arrays, loops, conditionals, functions, and basic input/output in JavaScript, without requiring any HTML or DOM manipulation.

Objective:

Create a basic JavaScript quiz game that runs in the browser console, taking user input with prompt() and giving feedback with alert(). The quiz should loop through predefined questions, compare answers using toLowerCase() and trim(), track the score, and display the final result.

Learning Outcomes

Upon completion of this assignment, the student will be able to:

1. Use core JavaScript features: arrays, loops, conditionals, and functions.
2. Understand how to create and use arrays to store multiple values, such as quiz questions and answers.
3. Learn to use loops, specifically for loops, to iterate through a set of items systematically.

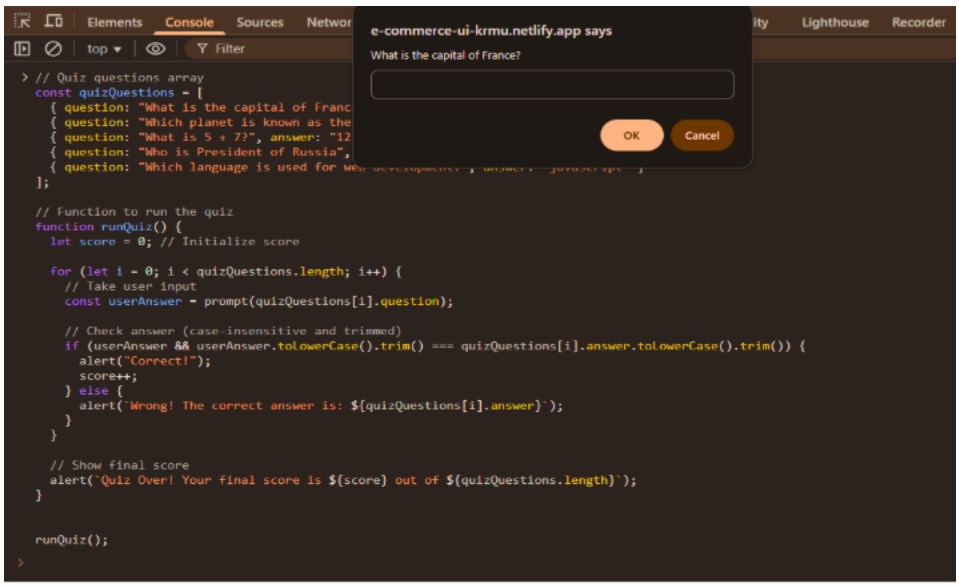
4. Practice using functions to organize code and make it reusable.
5. Develop skills in taking user input using prompt() and providing output using alert().
6. Apply string methods like toLowerCase() and trim() for accurate input validation and comparison.
7. Gain experience implementing conditional statements (if-else) to handle different program outcomes.
8. Learn to maintain and update a variable to track user performance, such as a score counter.
9. Strengthen logical thinking and problem-solving skills by combining loops, conditionals, and functions in a practical scenario.

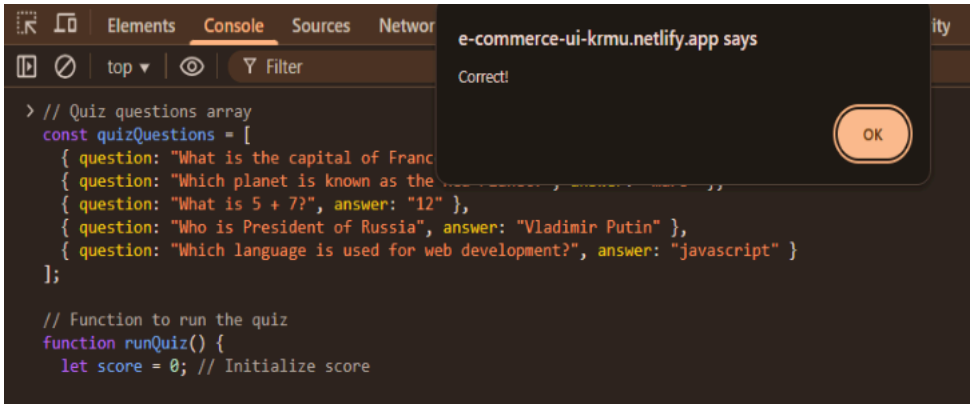
Detailed Instructions

1. **Quiz Questions Array**
   Create an array called quizQuestions to store all the quiz questions and their correct answers. Make each item an object with a question and an answer. Keep all the questions in this array so they are organized and easy to manage.
2. **Function to Run the Quiz**
   Define a function named runQuiz() to hold all the quiz logic. This function will handle showing each question, checking answers, updating the score, and giving feedback. Using a function keeps the code clean and organized.
3. **Score Initialization**
   Inside the function, declare a variable called score and set it to zero. This will track the number of correct answers while the quiz runs.
4. **Loop Through Questions**
   Use a for loop to go through each question in the array. The loop will allow each question to be asked one by one, and the current index will help access the correct question and answer.
5. **Prompt for User Input**
   Use prompt() to show the current question and collect the answer from the participant. Make sure to store the response in a variable for checking.
6. **Normalize the Input**
   Convert the participant's input to lowercase and remove extra spaces using toLowerCase() and trim(). This ensures that the answer is checked correctly even if letters are capitalized or extra spaces are entered.
7. **Check the Answer**
   Compare the participant's input with the correct answer using an if statement. If the answer is correct, increase the score by one. This step helps practice using conditional statements.

8. **Provide Immediate Feedback**
   Show an alert right after each answer. If the answer is correct, display a message saying it is correct. If the answer is wrong, display the correct answer so the participant can learn.
9. **Display the Final Score**
   After all questions have been answered, show an alert with the total score out of the total number of questions. This gives the participant a clear idea of how well they performed.
10. **Run the Quiz in the Browser Console**
    Open the browser console, paste the entire code, and press Enter to start the quiz. Answer each prompt, see the feedback, and check the final score at the end.

Expected Output





Guidelines to Students

1. **Understand the Requirements:**
   Review the project carefully. Ensure that the quiz includes multiple questions, collects user input, provides feedback, and displays the final score.
2. **Organize the Code:**
   Use arrays and functions to keep the code structured and reusable. Avoid writing all logic in a single block to maintain readability.
3. **Validate User Input:**
   Always process the input using toLowerCase() and trim() to handle different cases and extra spaces. This ensures fair and accurate evaluation.

4. **Provide Immediate Feedback:**
   Include alerts for both correct and incorrect answers. Feedback should help participants understand their mistakes instantly.
5. **Track Performance:**
   Maintain a score variable to count correct answers. Display the final score clearly at the end of the quiz.
6. **Test in the Browser Console:**
   Run the code in the browser console. Check that prompts display questions correctly, feedback is shown for each answer, and the final score is accurate.

Improvements/Adjustments

1. Add more questions to increase the challenge and variety.
2. Include multiple-choice options instead of only text input.
3. Implement a timer for each question to add a time-based challenge.
4. Customize feedback messages with hints for incorrect answers.
5. Store high scores in the browser using local storage for tracking performance.
6. Add restart option or score-based messages.

Submission Guidelines

1. Push quiz.js file to a GitHub repository.
2. Include a README.md
3. Ensure a minimum of 5 questions and working score display.
4. Share the GitHub repository link with the faculty.

5. Performance Metrics (Out of 5 Marks)

Criteria	Marks
Quiz logic using loops and conditionals	1

Input handling and answer comparison	1
Score tracking and display	1
Knowledge of browser console	1
Overall code readability and structure	1