Mini Project Report on
Web Technology Workshop 2 (CSE4948)

[Quiz Application]

Submitted by

Name: Vardaan Choudhary **Reg. No.: 2141003041**

B. Tech. CSIT 6th Semester (Section D)

INSTITUTE OF TECHNICAL EDUCATION AND RESEARCH
(FACULTY OF ENGINEERING)
SIKSHA 'O' ANUSANDHAN (DEEMED TO BE UNIVERSITY), BHUBANESWAR, ODISHA

Declaration:

I, the undersigned student of B. Tech. of **CSIT-D** Department hereby declare that I own the full responsibility for the information, results etc. provided in this PROJECT i.e., **" Quiz Application", Siksha 'O' Anusandhan(Deemedto be University), Bhubaneswar** for the partial fulfillment of the subject **Web Technology Workshop-2 (CSE 4948).**

I have taken care in all respect to honor the intellectual property right and have acknowledged my contribution and further declare that in case of any violation of intellectual property right or copyright I, as the candidate, will be fully responsible for the same.

Name: Vardaan Choudhary

Reg. No.: 2141003041

Abstract:

The code implements a quiz functionality using HTML, CSS, and JavaScript. An array named quizData stores question objects with properties like the question text, answer options (a, b, c, d), and the correct answer identifier.

The HTML structure defines the quiz container, question display, answer options with radio buttons and labels, and a submit button. CSS styles the overall layout, buttons, and text.

JavaScript functions handle various functionalities:

- loadQuiz: Loads the current question data from the quizData array and displays it.
- deselectAnswers: Deselects all radio buttons (answer options).
- getSelected: Returns the ID of the selected answer option (radio button).
- Event listener for the submit button:
    - Gets the selected answer ID using getSelected.
    - Checks if an answer is selected and compares it to the correct answer for the current question.
    - Increments score if the answer is correct.
    - Increments the question index to move to the next question.
    - If all questions are answered, displays the final score and a reload button.

Contents

Problem Statement

Problem Statement

Create a web-based quiz application that allows users to test their knowledge on a variety of topics. The quiz should present multiple-choice questions with clear options and provide immediate feedback on their answers. It should also track the user's score and display it at the end of the quiz.

Here are some specific functionalities to consider:

- The quiz should display one question at a time.
- Users should be able to choose an answer using radio buttons.
- Upon submitting an answer, the application should check if the chosen option is correct.
- If the answer is correct, the user's score should increase.
- After completing all questions, the application should display the final score and provide an option to restart the quiz.

Additional Considerations:

- The quiz should have a clean and user-friendly interface.
- The application should be responsive and work well on different screen sizes.
- You may want to consider adding features like a timer for each question or displaying the correct answer after an incorrect selection.

Implementation:

I.Code

HTML:-

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Quiz App</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<div class="quiz-container">
<div id="quiz">
<div class="quiz-header">
<h2 id="question">Question Text</h2>
<ul>
<li>
<input type="radio" name="answer" id="a" class="answer">
<label for="a" id="a_text">Answer</label>
</li>
<li>
<input type="radio" name="answer" id="b" class="answer">
<label for="b" id="b_text">Answer</label>
</li>
<li>
<input type="radio" name="answer" id="c" class="answer">
<label for="c" id="c_text">Answer</label>
</li>
<li>
<input type="radio" name="answer" id="d" class="answer">
<label for="d" id="d_text">Answer</label>
</li>
</ul>
</div>
<button id="submit">Submit</button>
</div>
</div>
<script src="script.js"></script>
</body>
</html>
```

CSS:-

```
body {

    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
```

```
    margin: 0;
}

.quiz-container {
    background: #fff;
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    width: 400px;
    max-width: 100%;
}

h2 {
    margin-bottom: 20px;
}

ul {
    list-style: none;
    padding: 0;
}

li {
    margin-bottom: 10px;
}

button {
    background: #007bff;
    color: #fff;
    border: none;
    padding: 10px 20px;
    border-radius: 5px;
    cursor: pointer;
    font-size: 16px;
}

button:hover {
    background: #0056b3;
}
```

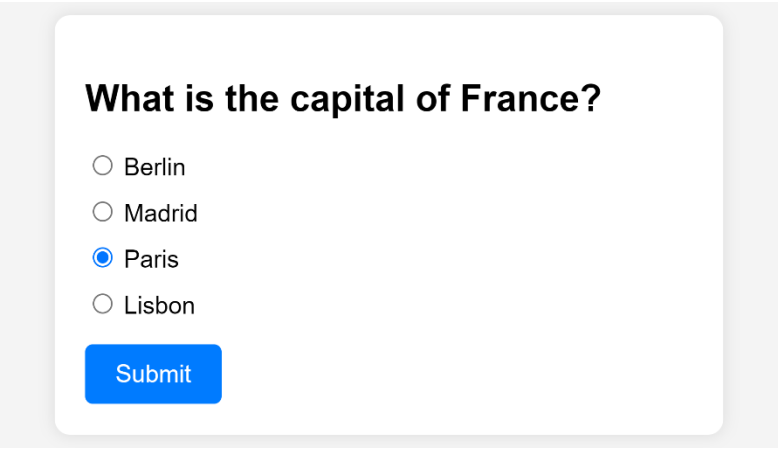
JavaScript:-

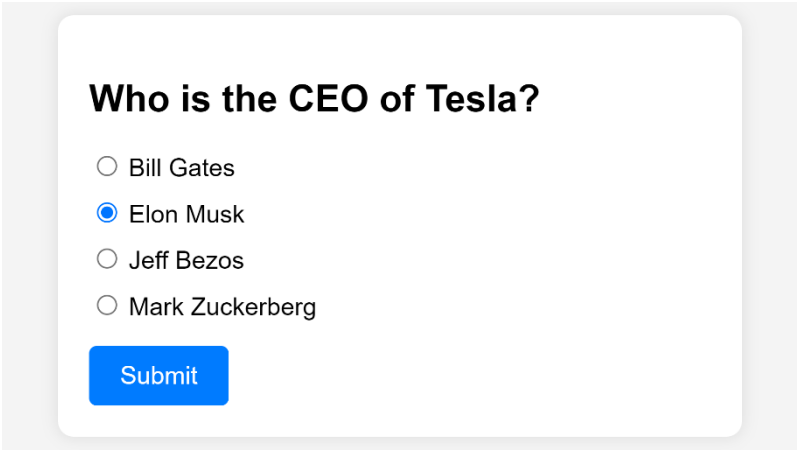
```
const quizData = [   //Question Objects

    {//object 1
        question: "What is the capital of France?",
        a: "Berlin",
        b: "Madrid",
        c: "Paris",
        d: "Lisbon",
        correct: "c",
    },
    {//object 2
        question: "Who is the CEO of Tesla?",
        a: "Bill Gates",
        b: "Elon Musk",
```

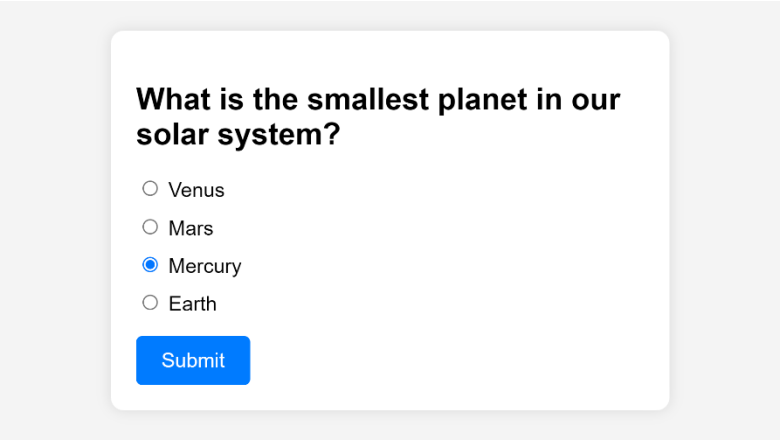
```
        c: "Jeff Bezos",
        d: "Mark Zuckerberg",
        correct: "b",
},
{//object 3
question: "What is the smallest planet in our solar system?",
a: "Venus",
b: "Mars",
c: "Mercury",
d: "Earth",
correct: "c",
},
{//object 4
question: "What is the square root of 64?",
a: "6",
b: "7",
c: "8",
d: "9",
correct: "c",
},
];

const quiz = document.getElementById('quiz');  //the whole quiz div
const answerEls = document.querySelectorAll('.answer'); //all the answer class selector elements

//this the radio field list
const questionEl = document.getElementById('question');//current question

//these are labels
const a_text = document.getElementById('a_text');//option a
const b_text = document.getElementById('b_text');//option b
const c_text = document.getElementById('c_text');//option c
const d_text = document.getElementById('d_text');//option d
const submitBtn = document.getElementById('submit'); //submit button

let currentQuiz = 0; //current quiz question index
let score = 0;//score of the quiz

loadQuiz();

function loadQuiz() {
deselectAnswers();//deselect the options

const currentQuizData = quizData[currentQuiz];

questionEl.innerText = currentQuizData.question; //current question
a_text.innerText = currentQuizData.a;
b_text.innerText = currentQuizData.b;
c_text.innerText = currentQuizData.c;
d_text.innerText = currentQuizData.d;
}

function deselectAnswers() {//deselect all options by using Query Selector
answerEls.forEach(answerEl => answerEl.checked = false);
}
```

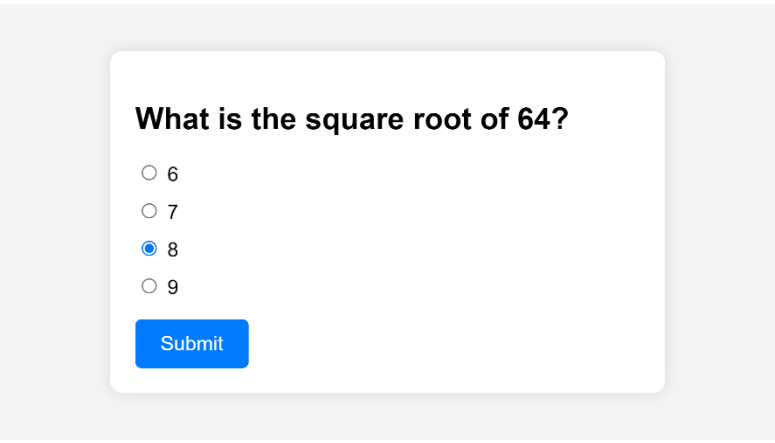
```
function getSelected() {
let answer;
answerEls.forEach(answerEl => {
if(answerEl.checked) {
answer = answerEl.id;
}
});
return answer;//return the selected options id
}

submitBtn.addEventListener('click', () => {
const answer = getSelected(); //if the answer is selected then only we can go to the next question
if(answer) {
if(answer === quizData[currentQuiz].correct) { //check if the option macthes
score++;//increase the score
}

currentQuiz++; //index++

if(currentQuiz < quizData.length) { //if we have not reached the end
loadQuiz();//then load the next question
} else {
//define the quiz div's innerHTML
quiz.innerHTML = `
<h2>You answered ${score}/${quizData.length} questions correctly</h2>
<button onclick="location.reload()">Reload</button>
`;
}
}
});
```

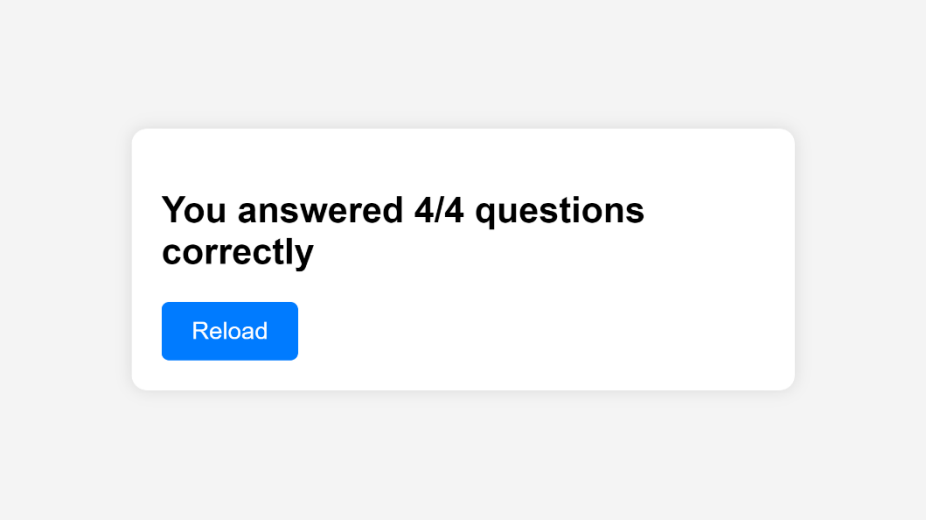
Result & Interpretation:











Conclusion:

This code successfully implements a basic, functional quiz application using HTML, CSS, and JavaScript. Users can answer multiple-choice questions, receive feedback on their selections, and see their final score at the end.

Here are some key points about the implementation:

- **Data Storage:** The quiz data is stored in a JavaScript array (`quizData`) with objects containing the question, answer choices, and the correct answer identifier.
- **User Interaction:** Users interact with the quiz by selecting answer options using radio buttons and submitting their answers with a button.
- **Functionality:** The code includes functions to:
    - Load and display the current question.
    - Deselect all answer options.
    - Retrieve the selected answer ID.
    - Check the answer and update the score.
    - Display the final score and a reload button upon completion.

Improvements and Further Development:

- **Error Handling:** The code could be improved by adding error handling to gracefully handle situations where no answer is selected before submission.
- **Responsiveness:** While the current styling might work, implementing responsive design practices would ensure the quiz looks good on various screen sizes.
- **Advanced Features:** Additional functionalities like a timer for each question, displaying the correct answer after an incorrect selection, or integrating with a backend for user accounts and quiz history could be added for a more comprehensive quiz experience.

This code provides a solid foundation for a web-based quiz application. With further development and enhancements, it can become a more engaging and informative learning tool.