PROJECT- 2

Interactive Quiz Application

Project done by:-
Indira Karunanidhi Mudaliar
Intern ID: ITID4514

Objective

The goal of this project was to develop an interactive quiz application that allows users to answer multiple-choice questions and receive a final score based on their responses.

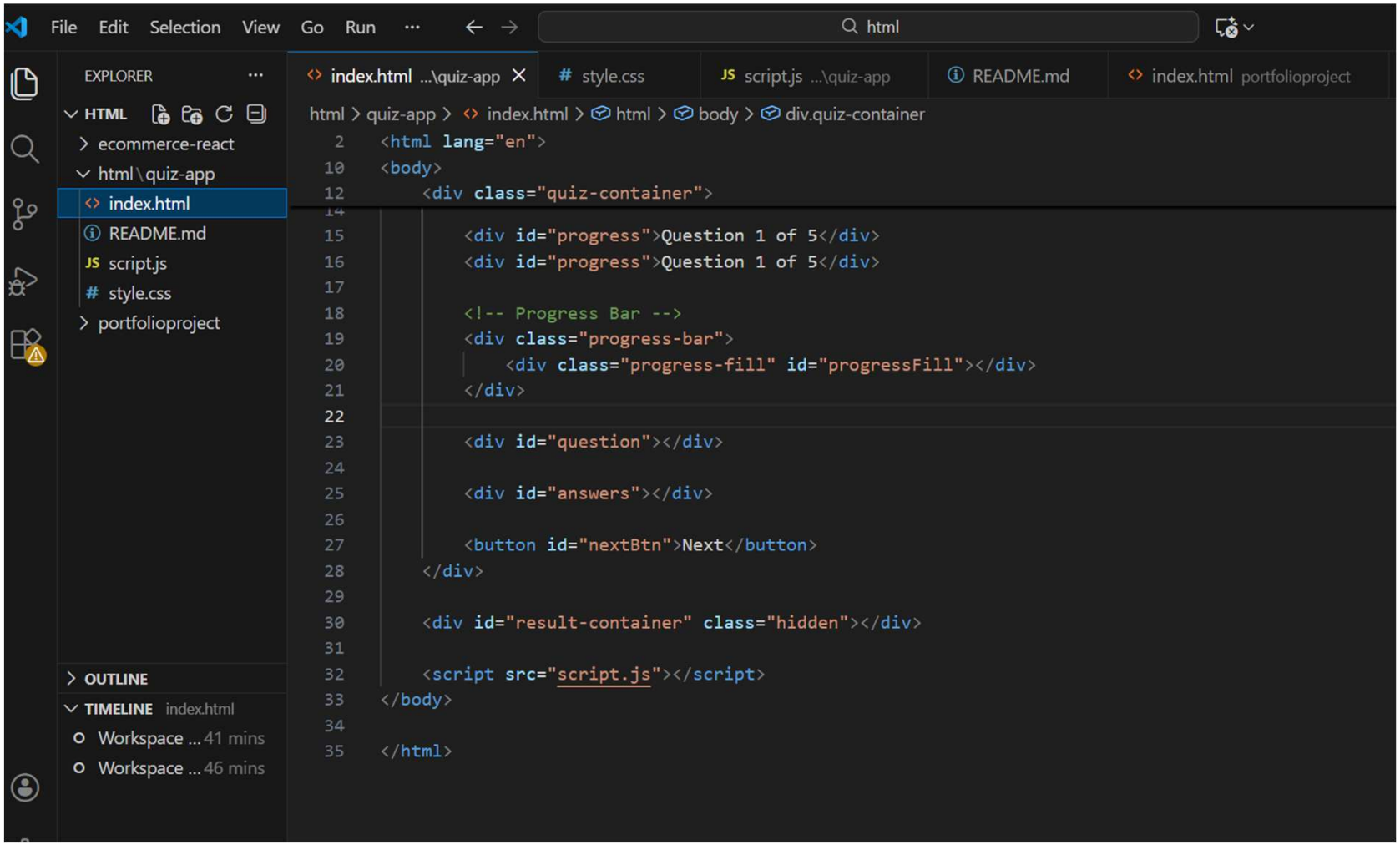
Technologies Used

HTML
CSS
JavaScript

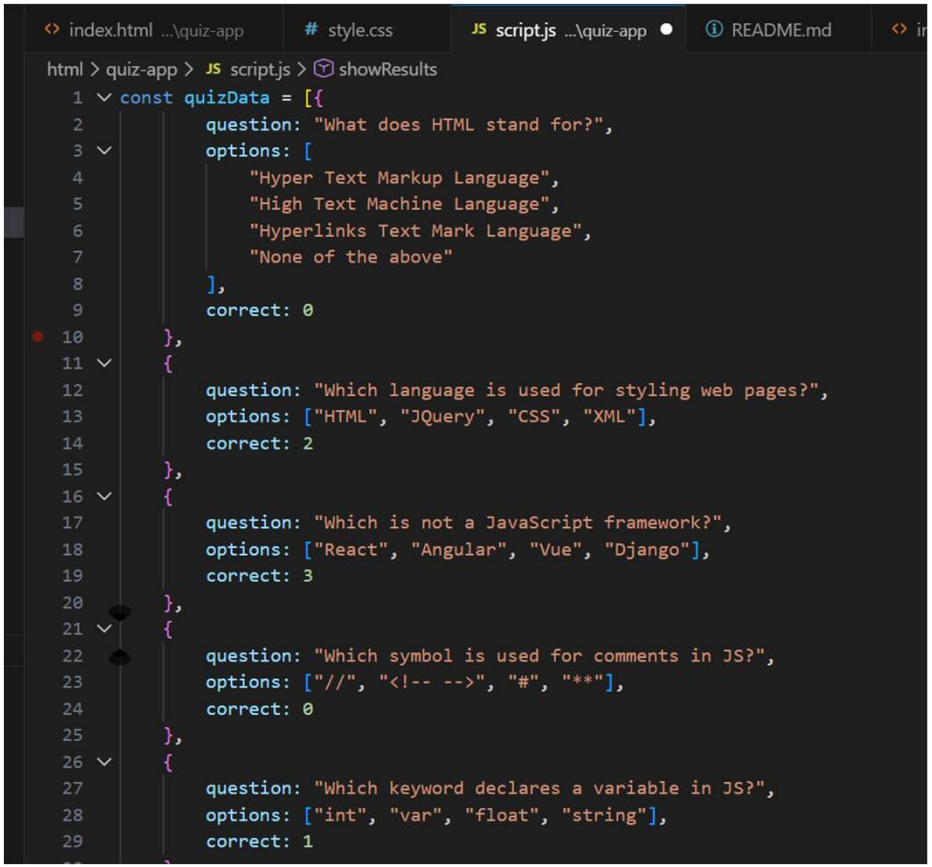
Key Features

Multiple-choice questions
Instant user interaction
Score calculation and display at the end
User-friendly and interactive interface

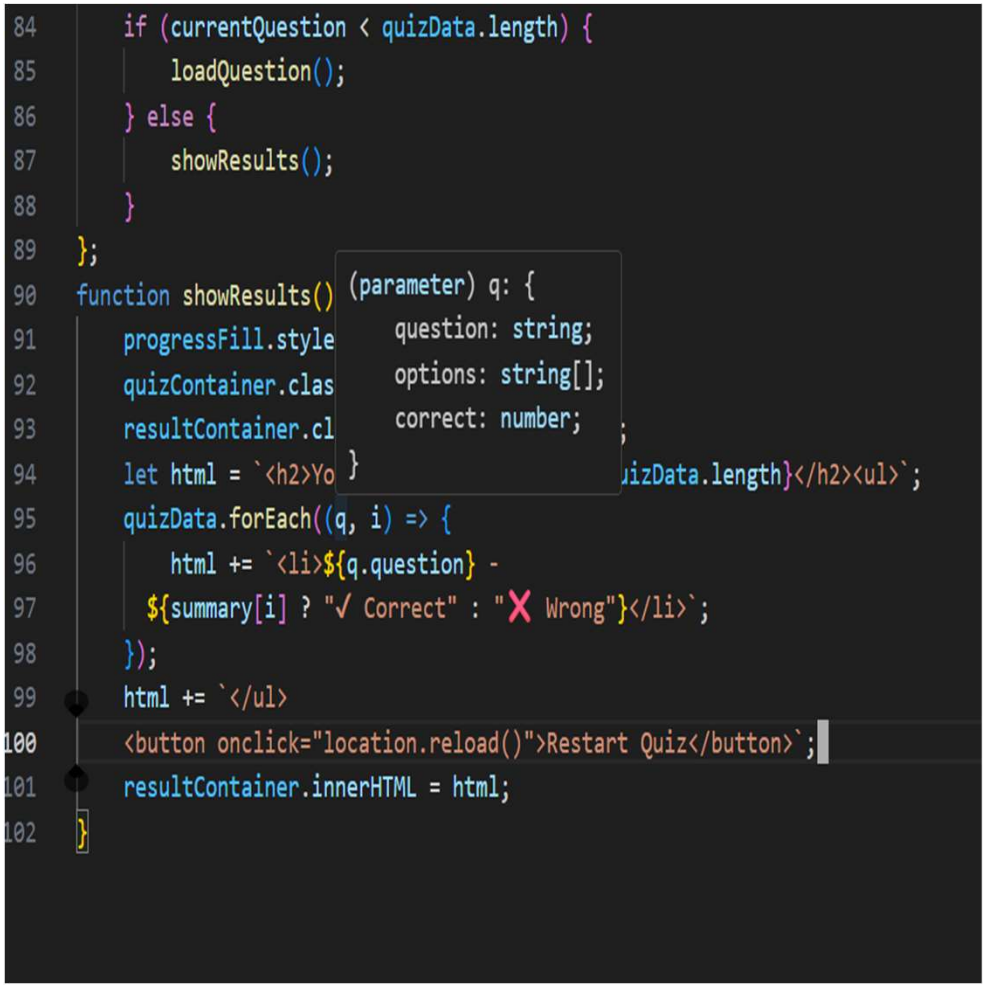
Contents of index.html



Contents of script.js



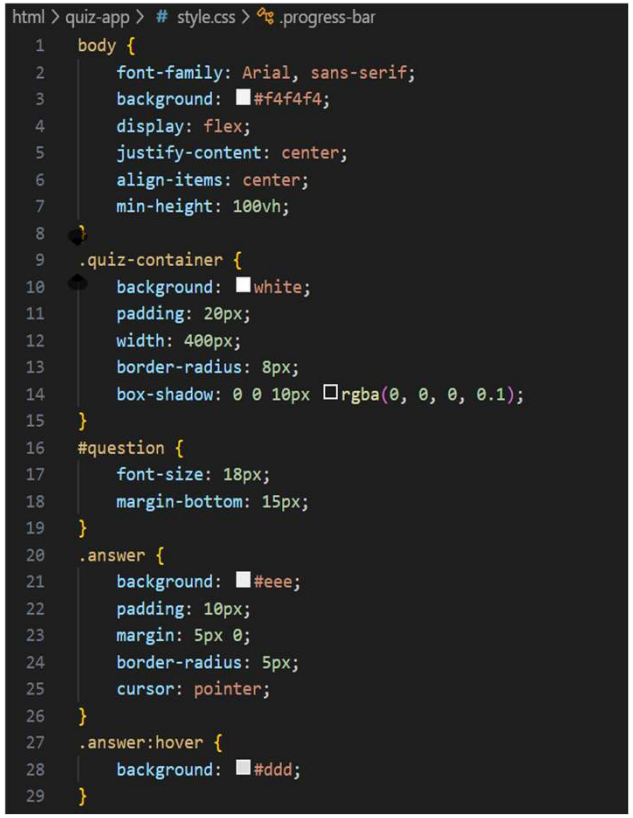
```
        answersEl.appendChild(div);
    });
    const progressPercent =
        (currentQuestion / quizData.length) * 100;
    progressFill.style.width = `${progressPercent}%`;
    nextBtn.textContent =
        currentQuestion === quizData.length - 1 ? "View Results" : "Next";
}
function selectAnswer(element, index) {
    if (selectedAnswer !== null) return;
    selectedAnswer = index;
    const correctIndex = quizData[currentQuestion].correct;
    document.querySelectorAll(".answer").forEach(btn => btn.style.pointerEvents = "none");
    if (index === correctIndex) {
        element.classList.add("correct");
        score++;
        summary.push(true);
    } else {
        element.classList.add("wrong");
        document.querySelectorAll(".answer")[correctIndex].classList.add("correct");
        summary.push(false);
    }
}
nextBtn.onclick = () => {
    if (selectedAnswer === null) return alert("Please select an answer");
    currentQuestion++;
    if (currentQuestion < quizData.length) {
```



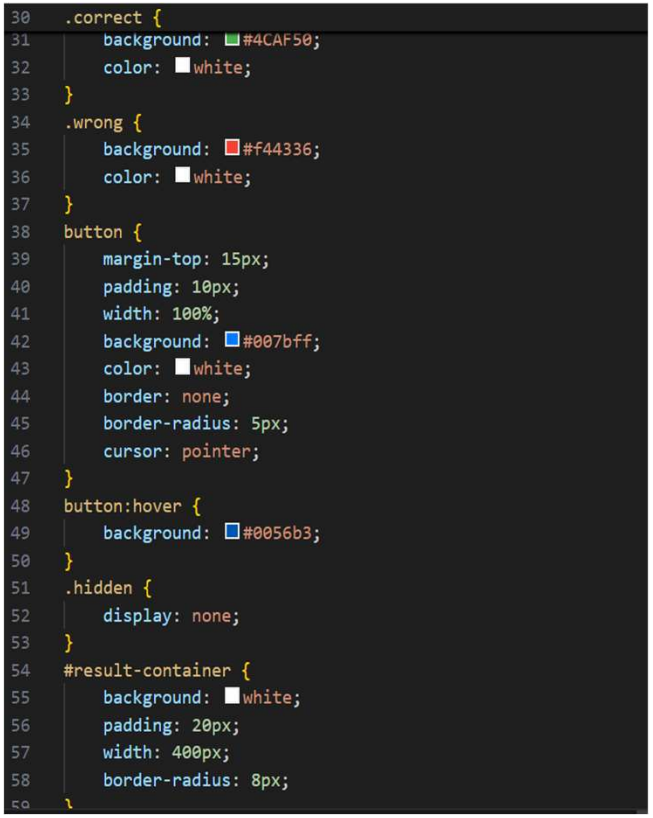
```
    if (currentQuestion < quizData.length) {
        loadQuestion();
    } else {
        showResults();
    }
};
function showResults()
    progressFill.style
    quizContainer.clas
    resultContainer.cl                                    ;
    let html = `<h2>Yo                                    izData.length}</h2><ul>`;
    quizData.forEach((q, i) => {
        html += `<li>${q.question} -
      ${summary[i] ? "✓ Correct" : "❌ Wrong"}</li>`;
    });
    html += `</ul>
    <button onclick="location.reload()">Restart Quiz</button>`;
    resultContainer.innerHTML = html;
}
```

```
(parameter) q: {
    question: string;
    options: string[];
    correct: number;
}
```

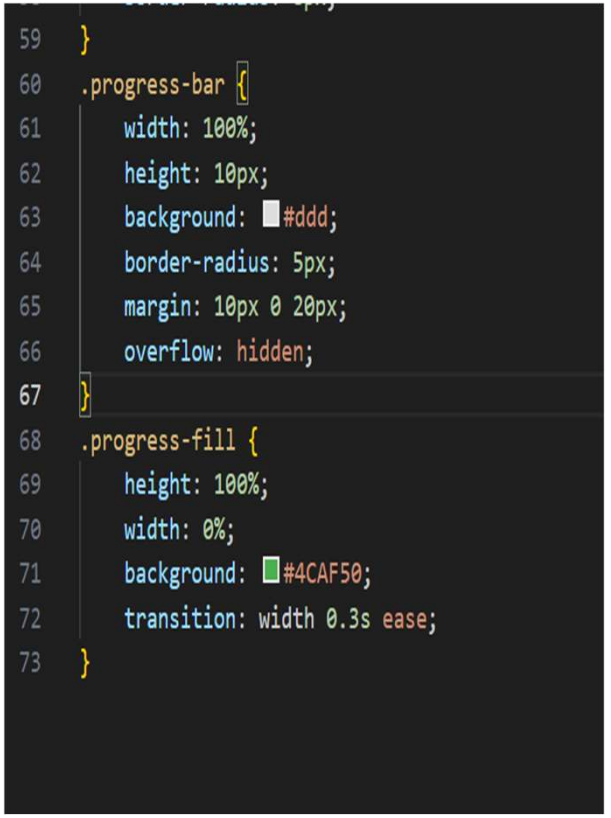
Contents of style.css



```
html > quiz-app > # style.css > .progress-bar
body {
    font-family: Arial, sans-serif;
    background: #f4f4f4;
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
}
.quiz-container {
    background: white;
    padding: 20px;
    width: 400px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
#question {
    font-size: 18px;
    margin-bottom: 15px;
}
.answer {
    background: #eee;
    padding: 10px;
    margin: 5px 0;
    border-radius: 5px;
    cursor: pointer;
}
.answer:hover {
    background: #ddd;
}
```



```
.correct {
    background: #4CAF50;
    color: white;
}
.wrong {
    background: #f44336;
    color: white;
}
button {
    margin-top: 15px;
    padding: 10px;
    width: 100%;
    background: #007bff;
    color: white;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}
button:hover {
    background: #0056b3;
}
.hidden {
    display: none;
}
#result-container {
    background: white;
    padding: 20px;
    width: 400px;
    border-radius: 8px;
```



```
}
.progress-bar {
    width: 100%;
    height: 10px;
    background: #ddd;
    border-radius: 5px;
    margin: 10px 0 20px;
    overflow: hidden;
}
.progress-fill {
    height: 100%;
    width: 0%;
    background: #4CAF50;
    transition: width 0.3s ease;
}
```

- ## Implementation Details

  The quiz logic was implemented using JavaScript arrays and objects to store questions, options, and correct answers. Event listeners were used to capture user responses. The application dynamically displays questions and tracks the user's score. At the end of the quiz, the total score is calculated and shown to the user. CSS was used to style buttons, questions, and layout for better usability.

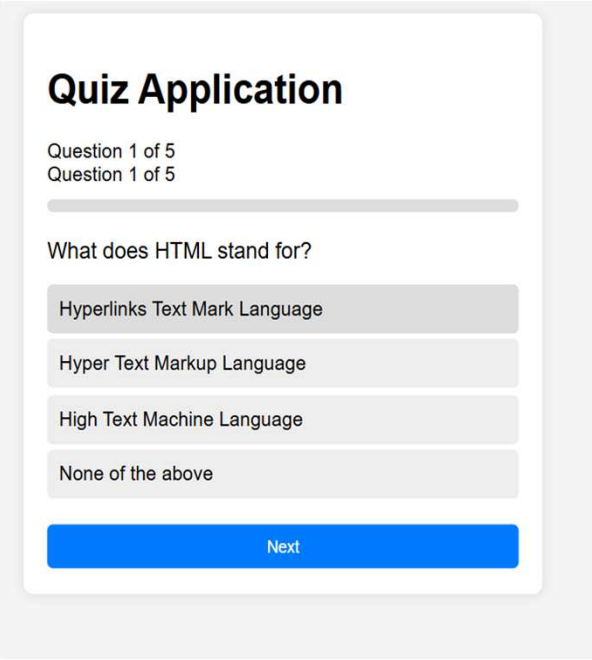
Challenges & Solutions

Handling user input validation and preventing multiple submissions were challenging. This was solved by disabling options after selection and managing quiz flow using conditional logic.

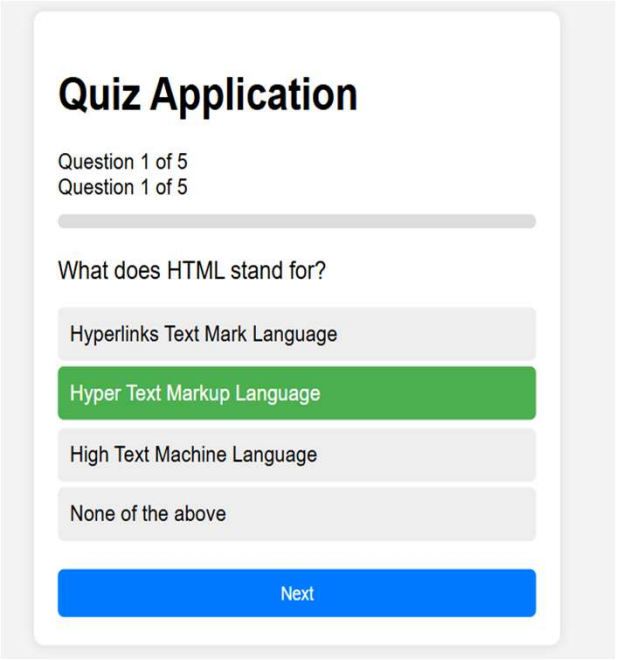
Outcome

This project strengthened my skills in JavaScript logic building, event handling, and state management in web applications.

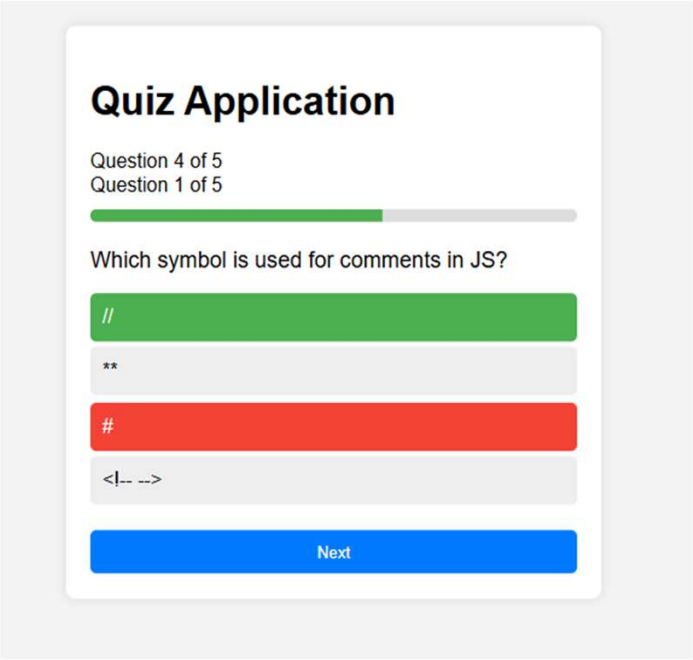
Layout of the quiz application



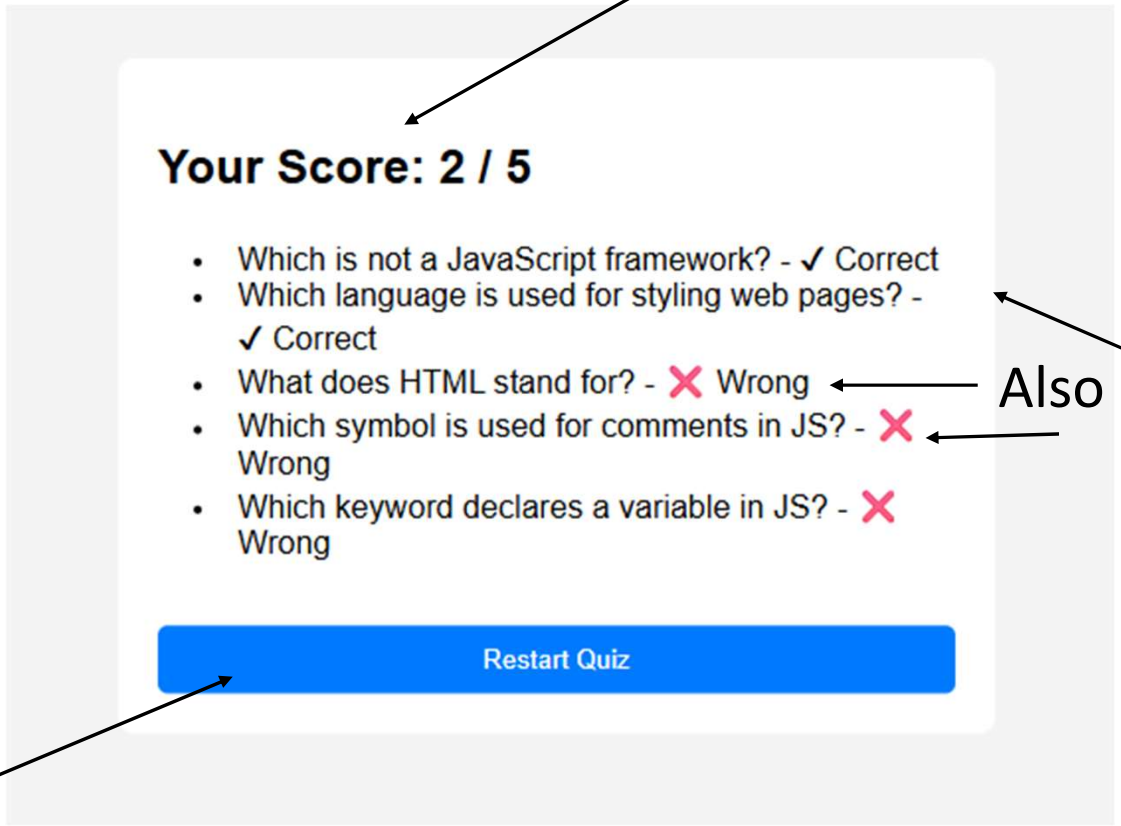
If the answer is correct



If the opted answer is wrong it also shows you the right option!



The total score



Also shows you which one is right and wrong

To restart the quiz