

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>QCM - Quiz</title>
  <script src="https://cdn.tailwindcss.com"></script>
  <link
href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;
700&display=swap" rel="stylesheet">
  <style>
    body {
      font-family: 'Inter', sans-serif;
      background-color: #f3f4f6; /* Light gray background */
      color: #1f2937; /* Dark gray text */
    }
    .quiz-container {
      max-width: 700px;
      margin: 2rem auto;
      padding: 2rem;
      background-color: #ffffff;
      border-radius: 0.75rem; /* Rounded corners */
      box-shadow: 0 10px 15px -3px rgba(0, 0, 0, 0.1), 0 4px 6px
-2px rgba(0, 0, 0, 0.05);
    }
    .question-card {
      background-color: #f9fafb; /* Lighter gray for question
card */
      padding: 1.5rem;
      border-radius: 0.5rem;
      margin-bottom: 1.5rem;
      border: 1px solid #e5e7eb; /* Light border */
    }
    .option-button {
      display: block;
      width: 100%;
      text-align: left;
      padding: 0.75rem 1.25rem;
      margin-bottom: 0.75rem;
      border: 1px solid #d1d5db; /* Gray border */
      border-radius: 0.5rem;
      background-color: #ffffff;
      color: #374151; /* Medium gray text */
      transition: background-color 0.2s, border-color 0.2s;
    }
    .option-button:hover {
      background-color: #eff6ff; /* Light blue on hover */

```

```

        border-color: #93c5fd; /* Blue border on hover */
    }
    .option-button.selected {
        background-color: #bfdbfe; /* Darker blue when selected */
        border-color: #60a5fa; /* Stronger blue border */
        color: #1e40af; /* Dark blue text */
        font-weight: 500;
    }
    .nav-button {
        padding: 0.75rem 1.5rem;
        border-radius: 0.5rem;
        font-weight: 500;
        transition: background-color 0.2s;
    }
    .nav-button.primary {
        background-color: #2563eb; /* Blue */
        color: #ffffff;
    }
    .nav-button.primary:hover {
        background-color: #1d4ed8; /* Darker blue */
    }
    .nav-button.secondary {
        background-color: #e5e7eb; /* Light gray */
        color: #374151; /* Medium gray text */
    }
    .nav-button.secondary:hover {
        background-color: #d1d5db; /* Darker gray */
    }
    .results-container {
        text-align: center;
        padding: 2rem;
        background-color: #f9fafb;
        border-radius: 0.5rem;
    }
    .progress-bar-container {
        width: 100%;
        background-color: #e5e7eb;
        border-radius: 0.25rem;
        height: 0.75rem; /* 12px */
        margin-bottom: 1.5rem;
        overflow: hidden;
    }
    .progress-bar {
        height: 100%;
        background-color: #2563eb; /* Blue */
        border-radius: 0.25rem;
        transition: width 0.3s ease-in-out;
    }

```

```

/* Custom modal for alerts */
.custom-modal {
  position: fixed;
  left: 0;
  top: 0;
  width: 100%;
  height: 100%;
  background-color: rgba(0,0,0,0.5);
  display: flex;
  justify-content: center;
  align-items: center;
  z-index: 1000;
  opacity: 0;
  visibility: hidden;
  transition: opacity 0.3s, visibility 0.3s;
}
.custom-modal.show {
  opacity: 1;
  visibility: visible;
}
.custom-modal-content {
  background-color: white;
  padding: 2rem;
  border-radius: 0.5rem;
  text-align: center;
  box-shadow: 0 4px 6px rgba(0,0,0,0.1);
  max-width: 90%;
  width: 400px;
}
.custom-modal-button {
  padding: 0.5rem 1rem;
  border-radius: 0.375rem;
  background-color: #2563eb;
  color: white;
  font-weight: 500;
  margin-top: 1rem;
  cursor: pointer;
}
</style>
</head>
<body class="flex items-center justify-center min-h-screen">

  <div class="quiz-container">
    <h1 class="text-3xl font-bold text-center mb-8 text-blue-600">QCM Challenge</h1>

    <div id="quizArea">
      <div class="progress-bar-container">

```

```

        <div id="progressBar" class="progress-bar"></div>
    </div>

    <div id="questionCard" class="question-card">
        <h2 id="questionText" class="text-xl font-semibold
mb-4">Question will load here...</h2>
        <div id="optionsContainer" class="space-y-3">
            </div>
    </div>

    <div class="flex justify-between items-center mt-8">
        <button id="prevButton" class="nav-button
secondary">Previous</button>
        <span id="questionCounter" class="text-sm
text-gray-600"></span>
        <button id="nextButton" class="nav-button
primary">Next</button>
    </div>
</div>

    <div id="resultsArea" class="results-container hidden">
        <h2 class="text-2xl font-bold mb-4">Quiz Completed!</h2>
        <p id="scoreText" class="text-lg mb-2"></p>
        <p id="customMessage" class="text-lg font-medium
text-blue-700 mb-6"></p>
        <button id="restartButton" class="nav-button
primary">Restart Quiz</button>
    </div>
</div>

    <div id="customModal" class="custom-modal">
        <div class="custom-modal-content">
            <p id="customModalMessage"></p>
            <button id="customModalCloseButton"
class="custom-modal-button">OK</button>
        </div>
    </div>

    <script>
        // --- START: YOUR QUESTION ---
        const quizQuestions = [
            {
                question: "wach ma3naha RP:",
                options: ["Recette principal.", "Recette pourboire.",
"Achraf sans slip.", "Manich 3arf."],
                correctAnswer: "Recette principal."
            }
        ]
        // You can add more questions here if you want, following

```

the same format:

```
// {
//   question: "Your next question here",
//   options: ["Option A", "Option B", "Option C",
"Option D"],
//   correctAnswer: "The correct option text"
// }
];
// --- END: YOUR QUESTION ---

const quizArea = document.getElementById('quizArea');
const questionCard = document.getElementById('questionCard');
const questionText = document.getElementById('questionText');
const optionsContainer =
document.getElementById('optionsContainer');
const prevButton = document.getElementById('prevButton');
const nextButton = document.getElementById('nextButton');
const questionCounter =
document.getElementById('questionCounter');
const progressBar = document.getElementById('progressBar');

const resultsArea = document.getElementById('resultsArea');
const scoreText = document.getElementById('scoreText');
const customMessage =
document.getElementById('customMessage');
const restartButton =
document.getElementById('restartButton');

// Custom Modal elements
const customModal = document.getElementById('customModal');
const customModalMessage =
document.getElementById('customModalMessage');
const customModalCloseButton =
document.getElementById('customModalCloseButton');

let currentQuestionIndex = 0;
let userAnswers = new Array(quizQuestions.length).fill(null);

function loadQuestion(index) {
  if (index < 0 || index >= quizQuestions.length) {
    console.error("Invalid question index:", index);
    return;
  }
  const questionData = quizQuestions[index];
  questionText.textContent = questionData.question;
  optionsContainer.innerHTML = ''; // Clear previous options

  questionData.options.forEach(option => {
```

```

        const button = document.createElement('button');
        button.textContent = option;
        button.classList.add('option-button');
        button.addEventListener('click', () =>
selectOption(button, option));
        optionsContainer.appendChild(button);

        if (userAnswers[index] === option) {
            button.classList.add('selected');
        }
    });

    updateNavigationButtons();
    updateProgressBar();
    updateQuestionCounter();
}

function selectOption(button, option) {
    // Deselect other options for the current question
    const buttons =
optionsContainer.querySelectorAll('.option-button');
    buttons.forEach(btn => btn.classList.remove('selected'));

    // Select the clicked option
    button.classList.add('selected');
    userAnswers[currentQuestionIndex] = option;
}

function updateNavigationButtons() {
    prevButton.disabled = currentQuestionIndex === 0;
    prevButton.classList.toggle('opacity-50',
prevButton.disabled);
    prevButton.classList.toggle('cursor-not-allowed',
prevButton.disabled);

    if (currentQuestionIndex === quizQuestions.length - 1) {
        nextButton.textContent = 'Submit Quiz';
    } else {
        nextButton.textContent = 'Next';
    }
}

function updateProgressBar() {
    const progressPercentage = ((currentQuestionIndex + 1) /
quizQuestions.length) * 100;
    progressBar.style.width = `${progressPercentage}%`;
}

```

```

function updateQuestionCounter() {
    questionCounter.textContent = `Question
${currentQuestionIndex + 1} of ${quizQuestions.length}`;
}

function showCustomAlert(message) {
    customModalMessage.textContent = message;
    customModal.classList.add('show');
}

customModalCloseButton.addEventListener('click', () => {
    customModal.classList.remove('show');
});

prevButton.addEventListener('click', () => {
    if (currentQuestionIndex > 0) {
        currentQuestionIndex--;
        loadQuestion(currentQuestionIndex);
    }
});

nextButton.addEventListener('click', () => {
    if (userAnswers[currentQuestionIndex] === null) {
        showCustomAlert('Please select an answer before
proceeding. ');
        return;
    }

    if (currentQuestionIndex < quizQuestions.length - 1) {
        currentQuestionIndex++;
        loadQuestion(currentQuestionIndex);
    } else {
        // Submit the quiz
        calculateResults();
    }
});

function calculateResults() {
    let score = 0;
    quizQuestions.forEach((question, index) => {
        if (userAnswers[index] === question.correctAnswer) {
            score++;
        }
    });

    const percentage = (score / quizQuestions.length) * 100;

```

```

        scoreText.textContent = `Your score: ${score} out of
        ${quizQuestions.length} (${percentage.toFixed(1)}%)`;
        customMessage.textContent = "chet achraf malgri ana machi
informaticien"; // Your custom message

        quizArea.classList.add('hidden');
        resultsArea.classList.remove('hidden');
    }

    restartButton.addEventListener('click', () => {
        currentQuestionIndex = 0;
        userAnswers.fill(null);
        resultsArea.classList.add('hidden');
        quizArea.classList.remove('hidden');
        loadQuestion(currentQuestionIndex);
        nextButton.textContent = 'Next'; // Reset button text
    });

    // Initial load
    if (quizQuestions.length > 0) {
        loadQuestion(currentQuestionIndex);
    } else {
        questionCard.innerHTML = '<p class="text-center
text-red-500">No questions loaded. Please add questions to the
`quizQuestions` array in the script.</p>';
        prevButton.classList.add('hidden');
        nextButton.classList.add('hidden');
        questionCounter.classList.add('hidden');
        progressBar.parentElement.classList.add('hidden');
    }
</script>

</body>
</html>

```