**Create quizquestions.js**

Fetches quiz questions from a server.

Dynamically builds a quiz form in the DOM.

Adds radio buttons for each choice.

Adds a "Submit Quiz" button.

Calculates and displays the user's score when submitted.

**Step 1: Initialize quiz data**

let quizData = [];

Declares an empty array to hold the fetched quiz questions.

This is used later to store both the questions and the correct answers for grading.

**Step 2: Fetch quiz from API**

fetch('/api/quiz')

.then(res => res.json())

.then(data => {

quizData = data;

renderQuiz(data);

})

.catch(err => {

console.error('Error loading quiz:', err);

});

fetch('/api/quiz'): Sends an HTTP request to retrieve quiz data.

.then(res => res.json()): Parses the JSON response.

.then(data => {...}): Stores the quiz data and calls renderQuiz() to display it.

.catch(...): Logs any errors if the fetch fails.

**Step 3: Define the renderQuiz function**

function renderQuiz(data) {

This function creates the entire quiz layout dynamically using DOM methods.

**Step 4: Get the quiz container and clear it**

const container = document.getElementById('quiz');

container.innerHTML = ''; // clear in case of re-render

Finds the container in the HTML where the quiz will be inserted.

Clears it so you can re-render if needed (e.g., after submitting or changing data).

**Step 5: Loop through quiz questions**

data.forEach((q, index) => {

Loops through each question object in the data array.

q is the current question object, index is its position.

**Step 6: Create a div for the question**

const questionDiv = document.createElement('div');

questionDiv.classList.add('mb-4');

Creates a wrapper <div> for each question, adds spacing (mb-4 from Bootstrap).

**Step 7: Add the question text**

const question = document.createElement('h5');

question.textContent = `${index + 1}. ${q.question}`;

questionDiv.appendChild(question);

Creates a <h5> element with the question number and text.

Appends it to the question wrapper.

**Step 8: Add each choice as a radio button**

q.choices.forEach(choice => {

const wrapper = document.createElement('div');

wrapper.classList.add('form-check');

const input = document.createElement('input');

input.type = 'radio';

input.name = `question-${index}`; // Grouping radio buttons

input.value = choice;

input.classList.add('form-check-input');

input.id = `q${index}-${choice}`;

const label = document.createElement('label');

label.classList.add('form-check-label');

label.setAttribute('for', input.id);

label.textContent = choice;

wrapper.appendChild(input);

wrapper.appendChild(label);

questionDiv.appendChild(wrapper);

});

Loops through each answer choice.

Creates:

A radio input button

A <label> for accessibility

Groups the radios by name="question-${index}" so only one can be selected.

Adds Bootstrap classes for styling (form-check, etc.)

**Step 9: Add the question div to the page**

container.appendChild(questionDiv);

Appends the completed block for this question to the main quiz container.

**Step 10: Add the submit button**

const submitBtn = document.createElement('button');

submitBtn.textContent = 'Submit Quiz';

submitBtn.classList.add('btn', 'btn-primary');

submitBtn.onclick = evaluateQuiz;

container.appendChild(submitBtn);

Creates a “Submit Quiz” button.

Adds Bootstrap styles.

Hooks it up to the evaluateQuiz() function.

**Step 11: Add a div for the result**

const resultDiv = document.createElement('div');

resultDiv.id = 'result';

resultDiv.classList.add('mt-3');

container.appendChild(resultDiv);

Creates a <div> where the score will be shown after submission.

Adds vertical spacing (mt-3 = margin-top).

**Step 12: Define the evaluateQuiz function**

function evaluateQuiz() {

let score = 0;

Starts the grading process.

Initializes score to 0.

**Step 13: Loop through quiz questions again**

quizData.forEach((q, index) => {

const selected = document.querySelector(`input[name="question-${index}"]:checked`);

if (selected && selected.value === q.answer) {

score++;

}

});

For each question:

Finds the checked radio input (:checked) for the question

Compares its value to the correct answer

Increments the score if it matches

**Step 14: Show the score**

const resultDiv = document.getElementById('result');

resultDiv.innerHTML = `<h4>Your score: ${score} out of ${quizData.length}</h4>`;

Gets the result display element (#result)

Updates it with the user's score

**Could You Use RxJS Here?**

Yes, but it's not necessary unless you're:

Listening to form input changes reactively

Adding debounced autosave

Creating a live quiz with streamed data

Using Angular (which integrates RxJS)

This script is perfectly fine in plain JavaScript.