Step-by-Step Guide for Building a Quiz Application

1. Define Your Goal

Understand the core functionalities of the application:

- Add questions to the quiz.
- Randomize the order of questions.
- Render quiz questions dynamically on the page.
- Collect user answers.
- Validate answers and calculate the score.
- Display the score and unanswered questions.

---

2. Plan the Structure

Break the application into smaller components:

1. **Data Storage**: Use an array for `questions` and an object for `userAnswers`.
2. **Methods**: Define functions for adding questions, randomizing them, rendering the quiz, collecting answers, scoring the quiz, and displaying results.
3. **Event Handling**: Handle user interactions like quiz submission and option selection.

---

3. Start Writing Code

Begin with the basics and build incrementally:

A. Set Up a Quiz Object

Create a JavaScript object (`quizApp`) to manage the quiz and its associated actions:

- Properties:
  - `questions`: Array to store quiz questions.
  - `userAnswers`: Object to store answers provided by the user.
- Methods:
  - `addQuestion()`
  - `randomizeQuestions()`
  - `renderQuiz()`
  - `collectAnswers()`
  - `calculateScore()`
  - `displayScore()`

B. Implement Core Methods

Write the methods in a modular way:

- `addQuestion`: Add a new question to the `questions` array.
- `randomizeQuestions`: Shuffle the questions array to randomize the order.
- `renderQuiz`: Dynamically generate HTML to display the questions and their options.
- `collectAnswers`: Capture the selected options for each question.
- `calculateScore`: Compare user answers to correct answers and calculate the score.
- `displayScore`: Show the user's score and any unanswered questions.

C. Attach Event Listeners

Handle the interactions:

- **Quiz Submission**: Use `addEventListener` on the submit button to collect answers, calculate the score, and display the results.

D. Test and Debug

- Test each method individually in the browser console.
- Validate inputs for edge cases (e.g., unanswered questions).

---

4. Order of Implementation

Follow this sequence:

Initialize the Quiz Object:

```
const quizApp = { questions: [], userAnswers: {} };
```

1.
2. **Add Core Methods to the Object**:
    - `addQuestion()`
    - `randomizeQuestions()`
    - `renderQuiz()`
    - `collectAnswers()`
    - `calculateScore()`
    - `displayScore()`
3. **Create Event Handlers**:
    - Write the logic for quiz submission and attach it to the "Submit" button.

4. **DOM Manipulation**:
    - Use `document.createElement` and `appendChild` to render questions and options.
    - Dynamically update the DOM for results display.
5. **Test the Flow**:
    - Add questions.
    - Render the quiz.
    - Submit answers.
    - Validate and display the score.

---

5. Add Features Incrementally

Once the basics work, enhance the application:

- **Feedback**: Provide immediate feedback for correct/incorrect answers.
- **Timer**: Add a timer for the quiz.
- **Styling**: Apply CSS classes for better UI and user experience.
- **Validation**: Ensure all questions are answered before submission.

---

6. Checklist for Completion

- Questions are added correctly to the quiz.
- Questions are randomized before rendering.
- User answers are collected and stored properly.
- The score is calculated accurately.
- The application handles unanswered questions gracefully.
- The quiz and results render dynamically.

By following this roadmap, you will systematically build the Quiz Application with clarity and focus.