Product Requirements & Specification Document

Project Name

Quizzy - Interactive JavaScript Quiz

1. Overview

Quizzy is a web-based quiz application designed to test users on JavaScript fundamentals. The app leverages React, TypeScript, and Tailwind CSS for a modern, responsive, and maintainable user experience. The quiz dynamically renders questions, tracks user progress, and provides instant scoring.

2. Goals & Objectives

- Provide an engaging, educational quiz on JavaScript basics.
- Ensure a responsive, accessible, and visually appealing UI.
- Enable dynamic question rendering and real-time scoring.
- Support easy extensibility for future question sets.

3. Target Audience

- JavaScript learners (beginners to intermediate)
- Educators and trainers
- Self-learners seeking to assess their JS knowledge

4. Core Features

Feature	Description
Dynamic Questions	Render questions from a data source; support multiple-choice format.
Scoring System	Track correct/incorrect answers; display score at end of quiz.
Progress Indicator	Show current question number and total questions.
Responsive UI	Mobile-first, accessible design using Tailwind CSS.
Feedback	Indicate correct/incorrect answers after each question (optional).
Restart Quiz	Allow users to retake the quiz.

5. Functional Requirements

ID	Requirement
FR1	Display a welcome/start screen with quiz instructions.
FR2	Fetch and render a set of 8-12 JavaScript multiple-choice questions.
FR3	Allow user to select one answer per question.
FR4	On answer selection, proceed to next question (or show feedback if enabled).
FR5	After last question, display total score and option to restart.
FR6	Ensure quiz is fully responsive and accessible (keyboard navigation, ARIA labels).

6. Non-Functional Requirements

ID	Requirement
NFR1	Use React (v18+), TypeScript, and Tailwind CSS.
NFR2	Codebase must be modular and maintainable.
NFR3	App loads in <2 seconds on standard broadband.
NFR4	No backend; all data is client-side.
NFR5	Support latest versions of Chrome, Firefox, Safari, Edge.

7. User Flow

```
flowchart TD
    Start[Start Screen] --> Quiz[Quiz Questions]
    Quiz -->|Answer| Next[Next Question]
    Next -->|Repeat| Quiz
    Quiz -->|Last Question| Score[Score Summary]
    Score -->|Restart| Start
```

8. UI Specifications

Screen	Elements
Start	App title, brief instructions, “Start Quiz” button
Quiz	Question text, 3-4 answer options (radio buttons), “Next” button, progress indicator
Score Summary	Final score, feedback message, “Restart Quiz” button

Styling:

- Use Tailwind CSS for all styling.
- Consistent color palette, clear typography, and accessible contrast ratios.

9. Data Model

```
type Question = {
  id: string;
  question: string;
  options: string[];
  answerIndex: number;
};
```

10. Technical Stack

Component	Technology
Frontend	React, TypeScript
Styling	Tailwind CSS
State Mgmt	React useState
Data	Local JSON/TS

11. Acceptance Criteria

ID	Criteria
AC1	User can complete a quiz and see their score.
AC2	UI is responsive and accessible.
AC3	All questions and answers render dynamically from data.
AC4	No errors or crashes during normal use.

12. Out of Scope

- User authentication or profiles
- Backend integration or persistent storage
- Advanced analytics or reporting

13. Milestones

Milestone	Description	Target Date
UI Prototype	Basic screens and navigation	Week 1
Core Functionality	Dynamic questions, scoring	Week 2
Styling & Accessibility	Tailwind integration, ARIA support	Week 3
Testing & QA	Bug fixes, acceptance testing	Week 4

14. Appendix

Sample Question Data:

```
{
  "id": "q1",
  "question": "What is the output of `typeof null` in JavaScript?",
  "options": ["'object'", "'null'", "'undefined'", "'number'"],
  "answerIndex": 0
}
```

End of Document