Product Requirements & Specification Document

Project Name

MediaQueryLab - Responsive Design Tester

Description

MediaQueryLab is an educational web playground for testing and visualizing CSS media queries and responsive layouts. Built with React and Tailwind CSS, it enables users to write, preview, and interactively adjust media queries to observe their effects in real-time.

---

1. Goals & Objectives

Goal	Description
Education	Help users learn and experiment with CSS media queries
Visualization	Provide real-time preview of responsive layouts
Usability	Offer an intuitive, interactive interface
Accessibility	Ensure basic accessibility for all users

---

2. Target Users

User Type	Needs/Use Cases
Students	Learn and practice responsive design
Developers	Test and debug media queries quickly
Educators	Demonstrate responsive concepts in real-time

---

3. Core Features

Feature	Description
Live Code Editor	Write/edit HTML and CSS (with media queries)
Real-Time Preview	See immediate visual feedback of code changes
Device Viewport Controls	Adjust viewport width (slider, presets: mobile/tablet/desktop)
Tailwind CSS Integration	Use Tailwind utility classes in code editor
Example Templates	Load sample responsive layouts for quick start
Error Handling	Display syntax errors in code editor
Reset/Clear Functionality	Reset editor to default state

4. Non-Goals

- No user authentication or persistent storage
- No backend or server-side processing
- No export/import of projects

---

5. Functional Requirements

ID	Requirement
FR1	Users can write/edit HTML and CSS (including media queries) in the editor
FR2	The preview updates in real-time as code changes
FR3	Users can adjust viewport width via slider and preset buttons
FR4	Tailwind CSS classes are supported in the editor and preview
FR5	Users can load example templates into the editor
FR6	Syntax errors are highlighted and displayed
FR7	Users can reset/clear the editor

---

6. Non-Functional Requirements

ID	Requirement
NFR1	Responsive and performant UI
NFR2	Works in latest versions of major browsers
NFR3	Basic accessibility (keyboard navigation)
NFR4	Minimal, clean, and intuitive design

---

7. Technical Specifications

Aspect	Specification
Framework	React (functional components, hooks)
Styling	Tailwind CSS
Editor	Use a lightweight code editor (e.g., react-simple-code-editor)
Live Preview	Render user HTML/CSS in an isolated iframe
Viewport Control	Adjustable via slider (min: 320px, max: 1920px) and preset buttons
Error Handling	Use CSS/HTML parsers for basic syntax validation
Example Templates	Store as static JSON or JS objects

---

8. User Interface Overview

---

9. Success Metrics

Metric	Target/Description
Usability	Users can preview media queries easily
Performance	Preview updates within 200ms of changes
Accessibility	All controls keyboard-accessible
Error Feedback	Syntax errors shown within 1s

---

10. Milestones & Timeline

Milestone	Description	Target Date
UI Skeleton	Layout, editor, preview, controls	Week 1
Live Preview Integration	Real-time rendering	Week 2
Error Handling	Syntax validation	Week 2
Example Templates	Load and reset functionality	Week 3
Testing & Polish	Accessibility, bug fixes	Week 3

---

11. Out of Scope

- User authentication
- Project saving/loading
- Advanced CSS preprocessors (Sass, Less)
- Third-party plugin support

---

12. Appendix

Example Template (Pseudocode)

```
{
  name: "Simple Responsive Card",
  html: `<div class="p-4 bg-white rounded shadow">...</div>`,
  css: `@media (max-width: 640px) { ... }`
}
```

End of Document