High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components for **MediaQueryLab - Responsive Design Tester**. The project is an educational playground for testing and visualizing CSS media queries and responsive layouts, built using React and Tailwind CSS.

---

1. System Architecture Overview

Architecture Description:
MediaQueryLab is a single-page React application. The system is modular, with a clear separation between the user interface, media query editor, preview renderer, and state management.

Module	Role
UI Shell	Provides layout, navigation, and theming
Media Query Editor	Allows users to input and edit CSS media queries
Preview Renderer	Renders a live preview of user content with applied media queries
State Management	Manages user input, preview state, and settings
Utility Services	Handles parsing, validation, and formatting of CSS/media queries

---

2. Component Interactions

Source Component	Target Component	Interaction Description
UI Shell	All	Hosts and arranges all main components
Media Query Editor	State Management	Updates state with user-edited media queries
State Management	Preview Renderer	Supplies current CSS/media queries for rendering
Preview Renderer	UI Shell	Displays responsive preview within main layout

Sequence Flow:

1. User edits media queries in the Editor.
2. State Management updates the current configuration.
3. Preview Renderer receives updates and re-renders the preview.
4. UI Shell maintains overall layout and navigation.

---

3. Data Flow Overview

Data Source	Data Destination	Data Type/Description

User Input (Editor)	State Management	CSS/media query strings, layout settings
State Management	Preview Renderer	Parsed/validated CSS, responsive layout data
Utility Services	Editor/Renderer	CSS parsing, validation results

---

4. Technology Stack

Layer/Area	Technology/Framework
Frontend Framework	React
Styling/UI	Tailwind CSS
Language	JavaScript (ES6+)
State Management	React Context/State Hooks
Build Tooling	Vite or Create React App

---

5. Scalability & Reliability

- **Scalability:**
  Designed as a client-side SPA; can be deployed statically and scales horizontally via CDN/static hosting.
- **Reliability:**
  Stateless architecture ensures high reliability; no backend dependencies.
- **Security:**
  No sensitive data handled; input is sandboxed to prevent code injection.

End of Document