# High Level Design Document

## Introduction

This High Level Design (HLD) document outlines the architecture and core components for **Gridify - CSS Grid Visualizer**. Gridify is an interactive, open-source tool built with React and Tailwind CSS, enabling users to visually design and preview CSS Grid layouts for educational and development purposes.

## 1. System Architecture Overview

**Architecture Summary:**
Gridify is a single-page application (SPA) structured around modular React components, styled with Tailwind CSS. The system is entirely client-side, requiring no backend.

| Module/Component | Role/Responsibility |
|---|---|
| UI Controls | Input grid parameters (rows, columns, gaps, etc.) |
| Grid Preview Canvas | Render live CSS Grid layout based on user input |
| Code Generator | Generate and display corresponding CSS/HTML code |
| State Management | Manage and propagate grid configuration state |
| Export/Share Utility | Allow users to export or share grid configurations |

## 2. Component Interactions

| Source Component | Target Component | Interaction Description |
|---|---|---|
| UI Controls | State Management | Updates grid configuration state on user input |
| State Management | Grid Preview Canvas | Triggers re-render with updated grid settings |
| State Management | Code Generator | Supplies current config for code generation |
| Export/Share Utility | State Management | Reads current state for export/share actions |

**Sequence Flow:**

1. User updates grid settings via UI Controls.
2. State Management updates and notifies dependent components.
3. Grid Preview Canvas and Code Generator re-render accordingly.
4. Export/Share Utility accesses current state for output.

## 3. Data Flow Overview

| Data Source | Data Destination | Data/Action Transferred |
|---|---|---|

| | | |
|---|---|---|
| User Input (UI) | State Management | Grid parameters (rows, columns, gaps, etc.) |
| State Management | Grid Preview Canvas | Grid configuration object |
| State Management | Code Generator | Grid configuration object |
| State Management | Export/Share Utility | Grid configuration object |

## 4. Technology Stack

| Layer/Area | Technology/Framework |
|---|---|
| Frontend Framework | React (JavaScript) |
| Styling | Tailwind CSS |
| State Management | React Context/State |
| Build Tooling | Vite or Create React App |
| Deployment | Static hosting (e.g., Vercel, Netlify, GitHub Pages) |

## 5. Scalability & Reliability

- **Scalability:**
  As a client-side SPA, Gridify scales horizontally via static hosting; no backend bottlenecks.
- **Reliability:**
  Minimal points of failure; all logic runs in-browser. No persistent data or authentication required.
- **Security:**
  No sensitive data handled; standard web security practices apply.

**End of Document**