Product Requirements & Specification Document

Project Name

Gridify - CSS Grid Visualizer

Description

Gridify is an interactive, open-source web tool for designing and previewing CSS Grid layouts. Built with React and Tailwind CSS, it enables users to visually create, modify, and export CSS Grid code for educational and prototyping purposes.

---

1. Goals & Objectives

Goal	Description
Visual Grid Design	Allow users to create and modify CSS Grid layouts interactively.
Real-time Preview	Instantly preview grid changes and content placement.
Code Export	Enable users to export generated CSS/HTML code.
Educational Focus	Help users learn CSS Grid concepts visually.
Open Source	Codebase is public and contributions are encouraged.

---

2. Target Users

- Web developers (beginners to intermediate)
- Students learning CSS Grid
- Educators and content creators

---

3. Core Features

Feature	Description
Grid Editor	Visual interface to set grid rows, columns, and gaps.
Item Placement	Drag-and-drop or form-based placement of grid items.
Live Preview	Real-time rendering of the grid and its items.
Code Panel	Displays generated HTML and CSS (Tailwind and standard CSS).
Export Functionality	Copy or download the generated code.
Responsive Controls	Adjust grid for different breakpoints.
Presets	Predefined grid templates for quick start.

Theming	Light/dark mode toggle.

---

4. Non-Functional Requirements

Requirement	Specification
Performance	Instant feedback (<100ms UI response)
Accessibility	Keyboard navigation, ARIA labels
Browser Support	Latest Chrome, Firefox, Edge, Safari
Responsiveness	Usable on desktop and tablet devices
Open Source	MIT License, public GitHub repository

---

5. Technical Specifications

Aspect	Specification
Framework	React (functional components, hooks)
Styling	Tailwind CSS
State Mgmt	React Context or useState (no external state libs)
Build Tool	Vite or Create React App
Code Export	Generates both Tailwind and standard CSS
Testing	Basic unit tests (Jest, React Testing Library)

---

6. User Flows

6.1. Create Grid Layout

1. User sets grid rows/columns (number, size, gap).
2. User adds grid items.
3. User positions items via drag-and-drop or form.
4. Preview updates in real-time.
5. User views and exports code.

6.2. Load Preset

1. User selects a preset template.
2. Grid and items update accordingly.
3. User customizes as needed.

---

7. UI Components

Component	Purpose

GridCanvas	Visual grid editor and item placement
ControlsPanel	Grid settings (rows, columns, gap, presets)
ItemList	List and manage grid items
CodePanel	Shows generated code, export options
ThemeToggle	Switch between light/dark modes

---

8. Milestones & Deliverables

Milestone	Deliverable
MVP	Basic grid editor, live preview, code export
Presets & Theming	Preset templates, light/dark mode
Accessibility & Testing	Keyboard support, basic tests
Documentation	User guide, contribution guidelines

---

9. Out of Scope

- Support for CSS Grid subgrids
- Advanced animation or transitions
- Mobile phone optimization (tablet/desktop only)

---

10. Example Pseudocode

```
// Example: Adding a grid item
function addGridItem() {
  setItems([...items, { id: uuid(), row: 1, col: 1, content: "Item" }]);
}
```

---

11. Success Metrics

Metric	Target
Usability	Users can create/export a grid in <5 min
Performance	UI updates in <100ms
Accessibility	Passes basic a11y checks
Community	At least 5 external contributors in 6 months

---

12. Risks & Mitigations

Risk	Mitigation
Complexity creep	Strict MVP scope, phased features
Accessibility gaps	Early a11y testing, ARIA best practices
Browser inconsistencies	Test on all major browsers

---

13. References

- CSS Grid Layout MDN
- Tailwind CSS Grid Docs

End of Document