Grid Maker Project Document

Sammuel Farias

October 2025

1 Team Members

- Sammuel Farias (GitHub: @samfarias)

2 Project Description

The Grid Maker is an interactive website that allows users to create and manipulate a grid of cells. By clicking menu buttons, users can add or remove rows and columns on the fly. They can also choose a color from a dropdown menu to fill or clear the cells in various ways: one cell at a time, all uncolored cells, or the entire grid. This functionality demonstrates core web development concepts such as HTML for structure (home and grid pages, the table, menu buttons), CSS for styling, and JavaScript for DOM manipulation and event handling (adding/removing rows and columns, applying & clearing colors).

3 Requirements

3.1 Business Requirements (BR)

- BR #1: As a user, I want to be able to create and modify the dimensions of a grid by adding or removing rows and columns.
- BR #2: As a user, I want to be able to color the grid cells to create a design, including selecting a color and applying it to specific cells or all cells.
- BR #3: As a user, I want to be able to easily navigate the site, moving between the homepage and the grid tool.

3.2 Functional Requirements (FR)

- FR #1: Users shall be able to add rows to the grid.
- FR #2: Users shall be able to add columns to the grid.
- FR #3: Users shall be able to remove rows from the grid.
- FR #4: Users shall be able to remove columns from the grid.
- FR #5: Users shall be able to select a color from a dropdown menu. The options are SELECT, Red, Blue, Green, and Yellow.
- FR #6: Users shall be able to color a single cell by clicking on it to change its color to the selected color.
- FR #7: Users shall be able to color all uncolored (white) cells with the selected color.
- FR #8: Users shall be able to color all cells with the selected color.
- FR #9: Users shall be able to clear all cells' color and restore them to their initial color (white).
- FR #10: Users shall be able to navigate from the homepage (`index.html`) to the Grid Maker page (`grid.html`) via a link.
- FR #11: Users shall be able to navigate from the Grid Maker page (`grid.html`) back to the homepage (`index.html`) using a button.

3.3 Non-Functional Requirements (NFR)

- NFR #1: The website must be deployed to GitHub Pages.
- NFR #2: The code must be clean, well-formatted, commented, and easy to read.
- NFR #3: The project must use HTML, CSS, and JavaScript for DOM manipulation and event handling.
- NFR #4: All new cells added to the grid must default to an initial color (white).
- NFR #5: If a user clicks a cell without selecting a color, an alert must prompt them to select one.

4 Software Architecture and Diagram

The application is a purely client-side website with no backend. It consists of two HTML pages, their corresponding stylesheets, and a single JavaScript file for all dynamic functionality.

- `index.html`: The homepage, which introduces the project and links to the grid maker tool. It is styled by `styles.css`.
- `grid.html`: The main tool page, which contains the action buttons, color selector, and the grid table. It is styled by `gridstyles.css` and all its interactivity is powered by `script.js`.
- `script.js`: This file handles all DOM manipulation and user events. It contains functions to add/remove rows and columns, select colors, and apply colors to cells in various ways (single, uncolored, all, clear).

File Architecture

- `index.html` (Homepage)
  - Loads `styles.css`
  - Links to `grid.html`
- `grid.html` (Grid Maker Page)
  - Loads `gridstyles.css`
  - Loads `script.js`
  - Links to `index.html`

5 Epics

5.1 Epic #1 - Grid Structure Management

- Requirement Reference: BR #1
- Description: As a user, I want to be able to add and remove rows and columns to create a grid of my desired size.

5.2 Epic #2 - Grid Coloring

- Requirement Reference: BR #2
- Description: As a user, I want to be able to select a color and apply it to the grid, either to individual cells, all cells, only uncolored cells, or clear all colors.

5.3 Epic #3 - Site Navigation

- Requirement Reference: BR #3
- Description: As a user, I want to be able to navigate between the homepage and the grid maker tool.

6 User Stories & Acceptance Criteria

6.1 User Story #1 - Add Row

- Epic Reference: Epic #1
- Description: As a user, I can add rows to the grid.
- Acceptance Criteria:
  - Given I am on the grid page, when I click the "Add Row" button, a new row is added to the bottom of the grid.
  - If the grid is empty (0 rows, 0 cols), clicking "Add Row" creates a 1x1 grid.

6.2 User Story #2 - Add Column

- Epic Reference: Epic #1
- Description: As a user, I can add columns to the grid.
- Acceptance Criteria:
  - Given I am on the grid page, when I click the "Add Col" button, a new column is added to the right side of every existing row.
  - If the grid is empty (0 rows, 0 cols), clicking "Add Col" creates a 1x1 grid.

6.3 User Story #3 - Remove Row

- Epic Reference: Epic #1
- Description: As a user, I can remove rows from the grid.
- Acceptance Criteria:
  - Given a grid with at least one row, when I click the "Remove Row" button, the last row is removed from the grid.
  - If no rows exist, an alert appears.

6.4 User Story #4 - Remove Column

- Epic Reference: Epic #1
- Description: As a user, I can remove columns from the grid.
- Acceptance Criteria:
  - Given a grid with at least one column, when I click the "Remove Col" button, the last cell from each row is removed.
  - If no columns exist, an alert appears.

6.5 User Story #5 - Select Color

- Epic Reference: Epic #2
- Description: As a user, I can select a color from a dropdown menu.
- Acceptance Criteria:
  - Given I am on the grid page, when I click the dropdown menu, I see options for SELECT, Red, Blue, Green, and Yellow.
  - When I select an option, that color is stored for use by other functions.

6.6 User Story #6 - Color Single Cell

- Epic Reference: Epic #2
- Description: As a user, I can color a single cell by clicking on it.
- Acceptance Criteria:
  - Given I have selected a valid color (e.g., "Red"), when I click on any cell, that cell's background color changes to "Red".
  - Given I have "SELECT" chosen as the color, when I click on any cell, an alert appears prompting me to select a color.

6.7 User Story #7 - Fill All Uncolored Cells

- Epic Reference: Epic #2
- Description: As a user, I can color all uncolored cells with the selected color.
- Acceptance Criteria:
  - Given I have selected a color and the grid contains white cells, when I click the "Fill All Uncolored" button, all cells with a "white" background change to the selected color.
  - Cells that are already colored (not white) remain unchanged.

6.8 User Story #8 - Fill All Cells

- Epic Reference: Epic #2
- Description: As a user, I can color all cells with the selected color.
- Acceptance Criteria:
  - Given I have selected a color, when I click the "Fill All" button, every cell in the grid changes to the selected color, regardless of its previous color.

6.9 User Story #9 - Clear All Cells

- Epic Reference: Epic #2
- Description: As a user, I can clear all cells' color.
- Acceptance Criteria:
  - Given a grid with colored cells, when I click the "Clear" button, every cell in the grid reverts to a "white" background color.

6.10 User Story #10 - Navigate to Homepage

- Epic Reference: Epic #3
- Description: As a user on the grid page, I can return to the homepage.
- Acceptance Criteria:
  - Given I am on the `grid.html` page, when I click the "Back to Homepage" button, I am navigated to the `index.html` page.

7 Project Schedule

Note: A Gantt chart would be inserted here. The following is a task list to build the chart from.

Phase 1: Planning and Setup

- Task: Define project requirements and user stories.
- Task: Create project document (Requirements, Epics, User Stories).
- Task: Set up GitHub repository.

Phase 2: HTML Structure and Navigation

- Task: Create `index.html` homepage.
- Task: Create `grid.html` page structure with all buttons and dropdowns.
- Task: Link `index.html` and `grid.html` together.

Phase 3: Core JavaScript Functionality

- Task: Implement `addR()` and `addC()` functions.
- Task: Implement `removeR()` and `removeC()` functions.
- Task: Implement `selectColor()` and `colorCell()` functions.
- Task: Implement `fillU()`, `fillAll()`, and `clearAll()` functions.

Phase 4: Styling and Documentation

- Task: Style `index.html` using `styles.css`.
- Task: Style `grid.html` (buttons, grid, dropdown) using `gridstyles.css`.
- Task: Write `README.md` file with project overview and features.

Phase 5: Deployment and Submission

- Task: Deploy the website to GitHub Pages.
- Task: Add GitHub Pages link to the `README.md`.
- Task: Finalize and submit the project document PDF.