Advance CSS Grid for Web/App Frontend

**Building Block for Endless Possibility in Web/App UI Layout**

1. **Introduction**

Let go straight to the point, if you are here reading this post, thank you very much! and yes, we are on the same both, as a Web/App frontend developer we are constantly facing the problems of how to code a user interface that will render perfectly like the UI wireframe. There are many systems out there that are built on top of CSS/JS and packages in a library of components to help developer ease some pain in web/app UI development.

Today I would like to share with you the glory of a new Grid system that is available for a while and get developer around the world are so thankful to get our hands on.

1. **Why the Grid?**

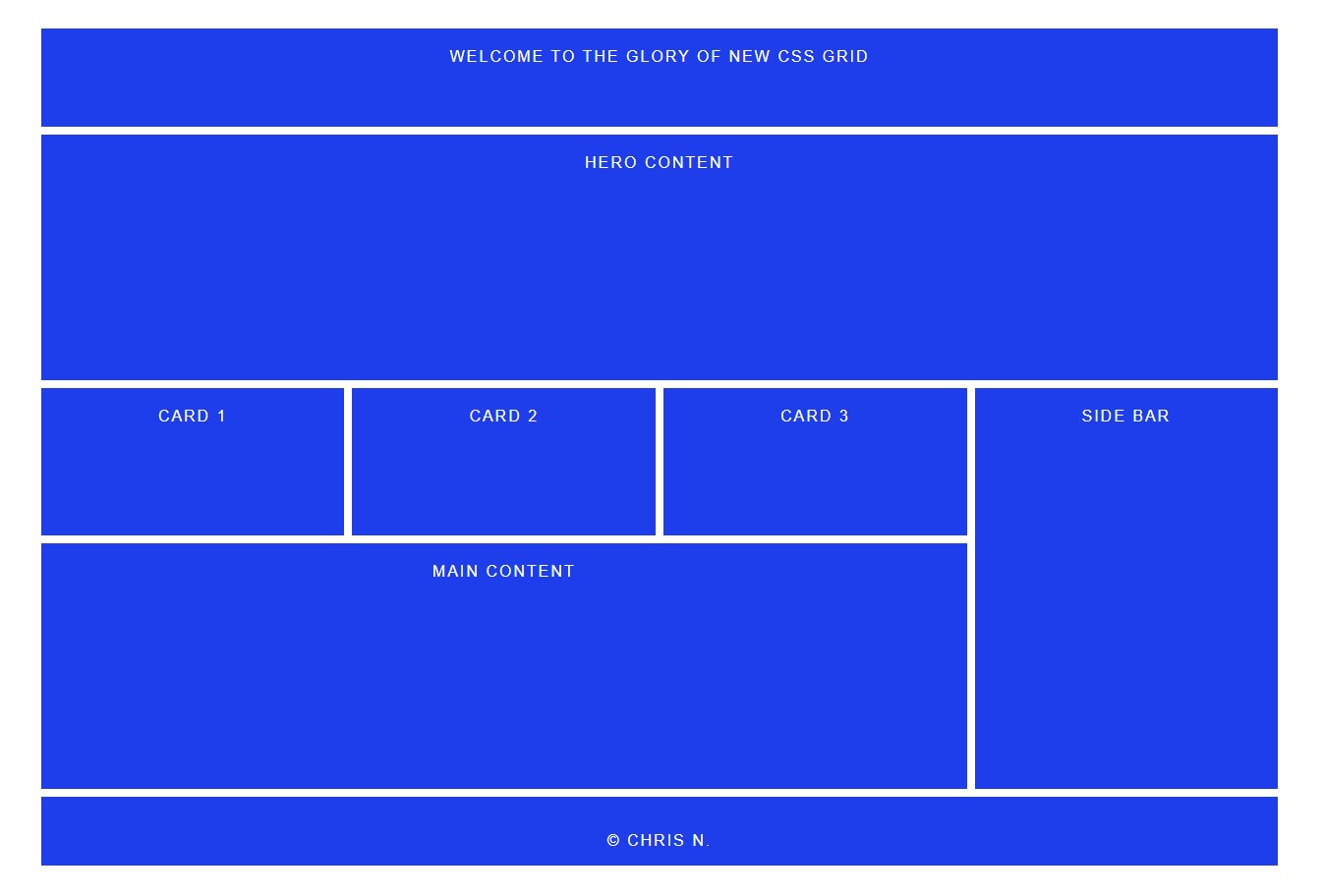
As a UI programmer I need a better way to code a design in less time, less code, and as painless of possible.

If I am a development manager, I will try to find too that help the team reduce the learning curve, improve productivities, having less code to manage, and simple to debug.

Let get right to it!

1. **Basic Wireframe**

Wireframe Layout Concept



1. Time to Code

**Step 1.**

To get started, we need a working directory, the index.html file, and the styles.css file so create a directory:

From command line type the following:

$ mkdir CssGrid

$ cd CssGrid

$ touch index.html

$ touch styles.css

**Step 2.**

We need a text editor or any code editor of your choice. I am currently using vs code.

From command line, make sure to cd inside the CssGrid folder:

Start vs code type

$ code . Enter

Now from vs code open the index.html file to add the HTML code and type in all the HTML code below (To learn more about HTML please to <https://www.w3schools.com/>)

<!DOCTYPE *html*>

<html *lang*="en">

<head>

  <meta *charset*="UTF-8">

  <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">

  <meta *http-equiv*="X-UA-Compatible" *content*="ie=edge">

  <link *rel*="stylesheet" *href*="styles.css">

  <title>New Grid</title>

</head>

<body>

  <div *class*="container">

    <div *class*="header">Welcome to the Glory of New CSS Grid</div>

    <div *class*="hero">Hero Content</div>

    <div *class*="card-1">Card 1</div>

    <div *class*="card-2">Card 2</div>

    <div *class*="card-3">Card 3</div>

    <div *class*="main-content">Main content</div>

    <div *class*="side-col">Side Bar</div>

    <div *class*="footer">

      <p>&copy; Chris N.</p>

    </div>

  </div>

</body>

</html>

The line of code highline in white background above is the html link to the css code that will render the page layout.

**Step 3:**

This block of code setup the grid system and basic page width and margin.

I have commented every line code to make it easy to understand.

/\* Code comment look like this, you don’t need to type them in your code \*/

Now open the styles.css file and type in the code here:

*.container* {

  width: 90%; /\* Set container width to 90% of viewport \*/

  margin: 30px auto; /\* Set 30px for margin top/bottom \*/

  display: grid; /\* Activate the grid system \*/

  grid-template-rows: 100px 250px 150px 250px 70px; /\* Set height for each row \*/

  grid-template-columns: repeat(4, 1fr); /\* set 4 column, 1fr = 1 faction of 4 \*/

  grid-gap: 8px; /\* Set grid gap to 8px \*/

}

This block set typography for elements inside each box on the grid

*.container* > div { /\* Set parameters for all the content inside the boxes \*/

  padding: 20px; /\* Set padding inside each box \*/

  background-color: #1e3deb; /\* Set background color for all boxes \*/

  color: white; /\* Set font color \*/

  font-size: 16px; /\* Set font size to 16px \*/

  font-family: sans-serif; /\* Set font-family \*/

  text-transform: uppercase; /\* covert font to upper case \*/

  letter-spacing: 2px; /\* Set spacing between letters \*/

  text-align: center; /\* Center align all text \*/

}

This block of code is for positioning all the boxes in their position

*.header* {

  grid-column: 1 / span 4; /\* start column 1 and span across 4 columns \*/

}

*.hero* {

  grid-column: 1 / -1; /\* start column 1 and span across 4 columns \*/

}

*.card-1*, *.card-2*, *.card-3* {

  grid-row: 3 / 4; /\* Start at row 3 of 4 columns \*/

}

*.side-col* {

  grid-column: 4 / 5; /\* Start at column 4 of 5 row \*/

  grid-row: 3 / span 2; /\* Start at row 3 and span down across 2 rows \*/

}

*.main-content* {

  grid-column: 1 / span 3; /\* Start at column 1 and span across 3 columns \*/

}

*.footer* {

  grid-column: 1 / -1; /\* Start at column 1 and span across the grid \*/

}

Save the code and check the result.

Thanks for taking the time to code and test this concept, hope this help!

Chris N.