

Lesson 06 Demo 05

Demonstrating Grid-Columns Layout in Tailwind CSS

Objective: To illustrate the process of configuring grid-based layouts in Tailwind CSS by specifying the number of columns, adjusting column widths using col-span, and managing spacing with gaps to achieve a well-structured and responsive design

Tools required: Visual Studio Code

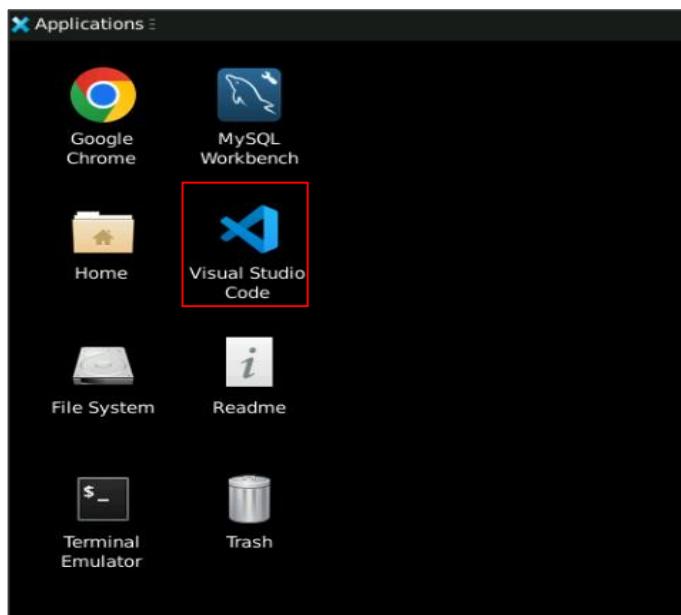
Prerequisites: HTML Basics

Steps to be followed:

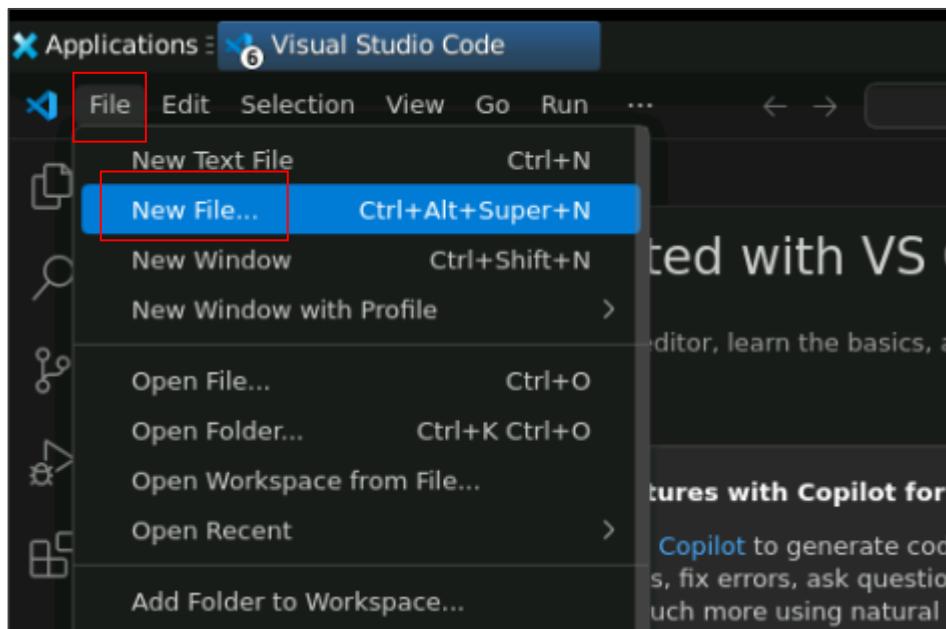
1. Open Visual Studio Code and create index.html
2. Implement the grid layout

Step 1: Open Visual Studio Code and create index.html

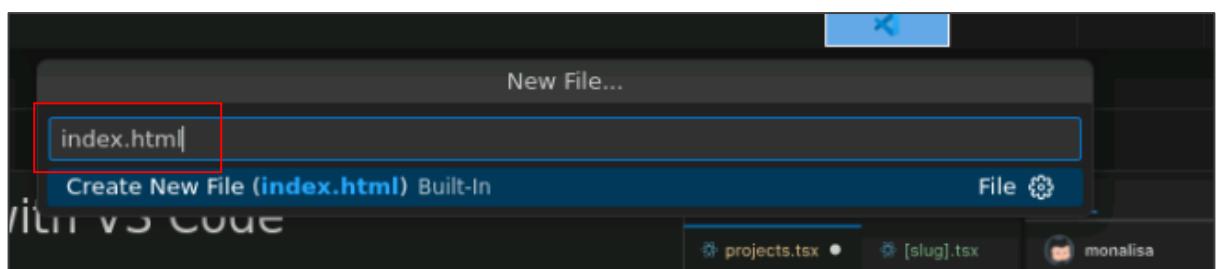
1.1 Open Visual Studio Code



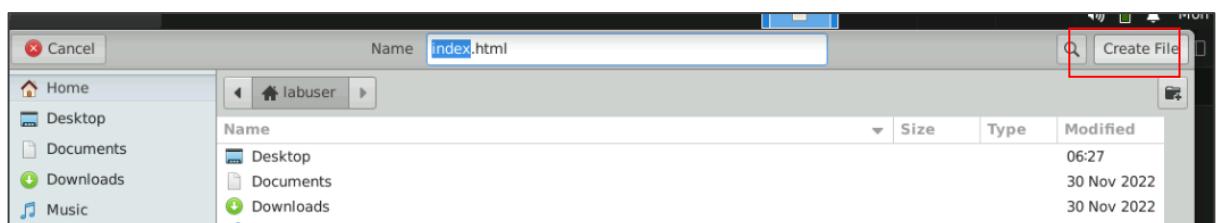
1.2 Click on **File** and then select the **New File...** option



1.3 Create an **index.html** file and press enter



1.4 Click on **Create File**

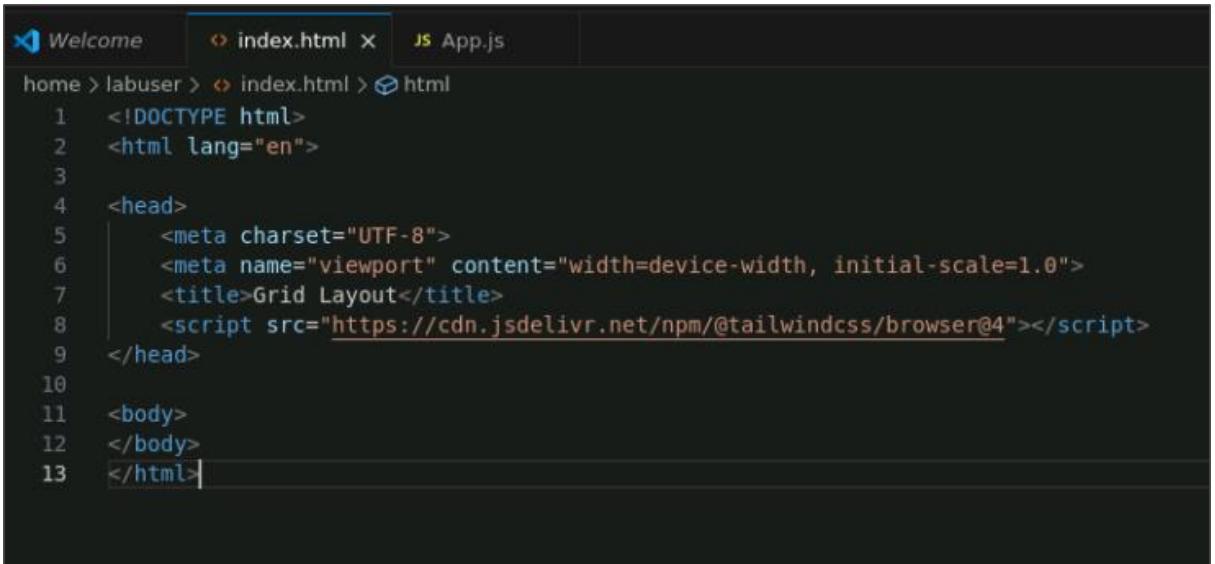


1.5 Insert the following base HTML structure into index.html:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Grid Layout</title>
  <script src="https://cdn.jsdelivr.net/npm/@tailwindcss/browser@4"></script>
</head>

<body>
</body>
</html>
```



The screenshot shows a code editor window with a dark theme. At the top, there are tabs for 'Welcome', 'index.html' (which is currently selected), and 'App.js'. Below the tabs, the file path 'home > labuser > index.html' is displayed. The main area contains the following code:

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>Grid Layout</title>
8    <script src="https://cdn.jsdelivr.net/npm/@tailwindcss/browser@4"></script>
9  </head>
10
11 <body>
12 </body>
13 </html>
```

The code is numbered from 1 to 13. The script tag's URL is underlined, indicating it is a link or a placeholder for a link.

Step 2: Implement the grid layout

2.1 Modify the index.html file to include the grid layout, using the following code:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Grid Layout</title>
  <script src="https://unpkg.com/@tailwindcss/browser@4"></script>
</head>

<body>
  <div class="container mx-auto p-4">
    <h1 class="text-5xl mb-6">Explore Grid Layout Options</h1>

    <!-- Basic 3-column grid -->
    <h2 class="text-2xl mb-2">1. Basic 3-Column Grid with 9 Elements</h2>
    <div class="grid grid-cols-3 gap-4">
      <div class="bg-blue-500 text-white p-4">Item 1</div>
      <div class="bg-blue-500 text-white p-4">Item 2</div>
      <div class="bg-blue-500 text-white p-4">Item 3</div>
      <div class="bg-blue-500 text-white p-4">Item 4</div>
      <div class="bg-blue-500 text-white p-4">Item 5</div>
      <div class="bg-blue-500 text-white p-4">Item 6</div>
      <div class="bg-blue-500 text-white p-4">Item 7</div>
      <div class="bg-blue-500 text-white p-4">Item 8</div>
      <div class="bg-blue-500 text-white p-4">Item 9</div>
    </div>

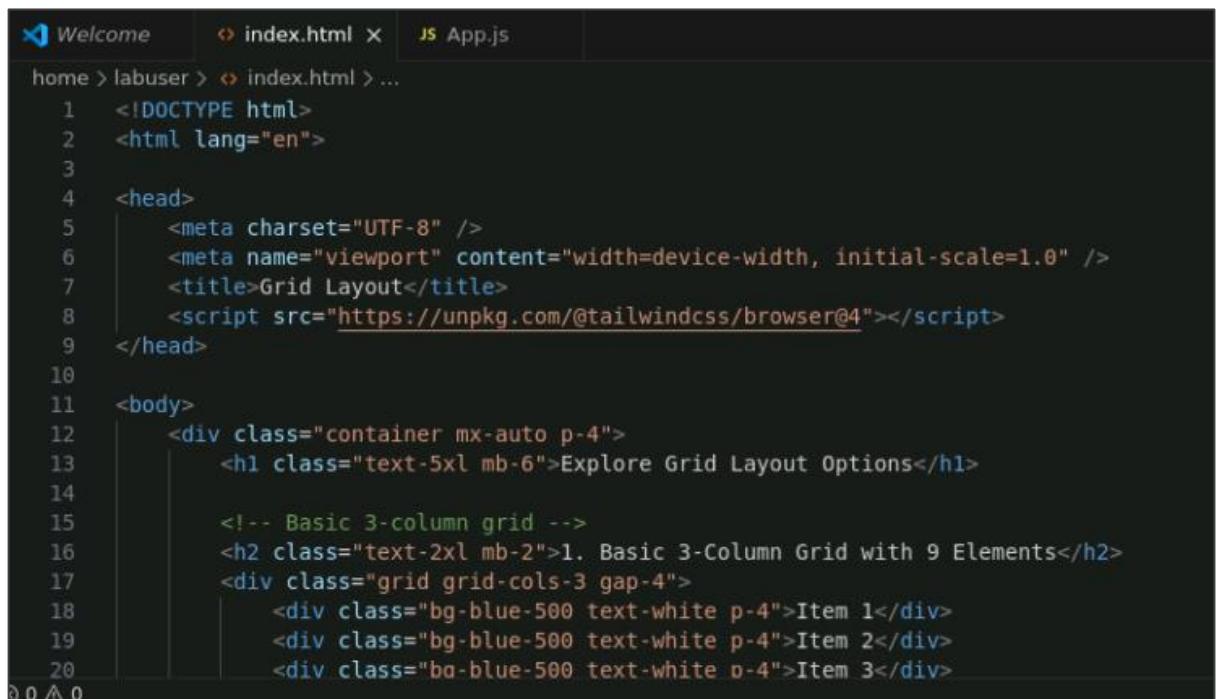
    <!-- 2-column grid with different widths -->
    <h2 class="text-2xl mt-6 mb-2">2. Two-Column Grid with Custom Widths</h2>
    <div class="grid grid-cols-3 gap-4">
      <div class="col-span-2 bg-green-500 text-white p-4">Item 1 (Spans 2 Columns)</div>
      <div class="bg-green-500 text-white p-4">Item 2</div>
    </div>
  </div>
</body>
```

```

<!-- Centering content using place-items -->
<h2 class="text-2xl mt-6 mb-2">3. Centering Content</h2>
<div class="grid h-32 bg-red-500 text-white place-items-center">
    <p>Centered Content</p>
</div>
</div>
</body>

</html>

```



```

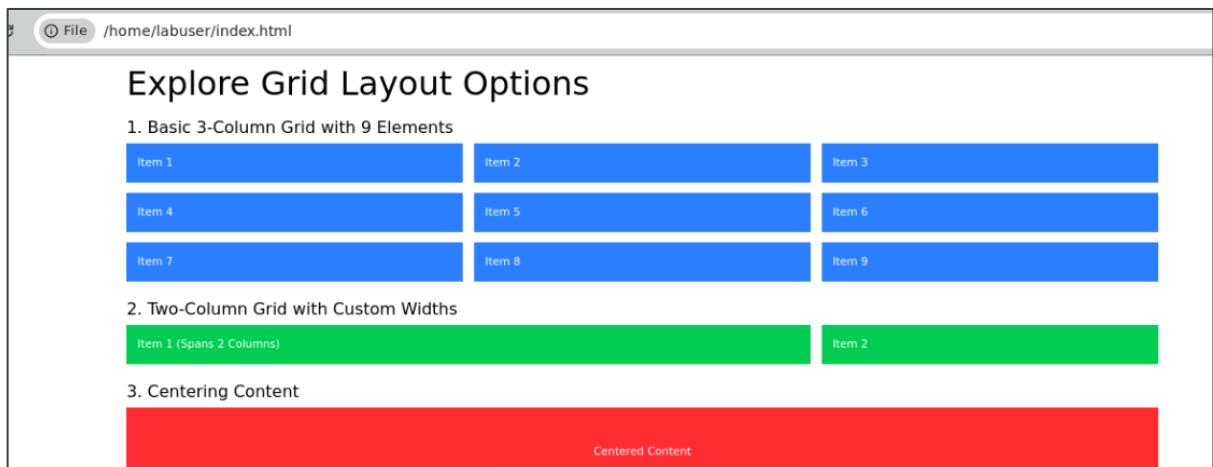
Welcome index.html App.js
home > labuser > index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3
4      <head>
5          <meta charset="UTF-8" />
6          <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7          <title>Grid Layout</title>
8          <script src="https://unpkg.com/@tailwindcss/browser@4"></script>
9      </head>
10
11     <body>
12         <div class="container mx-auto p-4">
13             <h1 class="text-5xl mb-6">Explore Grid Layout Options</h1>
14
15             <!-- Basic 3-column grid -->
16             <h2 class="text-2xl mb-2">1. Basic 3-Column Grid with 9 Elements</h2>
17             <div class="grid grid-cols-3 gap-4">
18                 <div class="bg-blue-500 text-white p-4">Item 1</div>
19                 <div class="bg-blue-500 text-white p-4">Item 2</div>
20                 <div class="bg-blue-500 text-white p-4">Item 3</div>

```

2.2 Navigate to the index.html file, right-click on the file, and select **Open With “Google Chrome”**



2.3 Verify that the displayed output correctly reflects the expected grid layout



By following these steps, you have successfully implemented a structured and responsive grid-based layout using Tailwind CSS. This approach allows developers to efficiently define columns, adjust widths using col-span, and manage spacing with gap, enhancing the organization and flexibility of web pages.