Streamline Your CSS Workflow with Bootstrap

Demo - 1

edureka!

Problem Statement:

As a web developer, you have been tasked with creating a responsive website using Bootstrap's container and grid system. However, you are facing challenges in understanding the concept of the container and grid system and how to effectively use them to create a responsive layout. Additionally, you are finding it difficult to customize the grid system to meet specific design requirements. This is hindering your ability to deliver a high-quality website within the given timeline.

Solution:

Task in hand to create a responsive website using Bootstrap's container and grid system:

Container:

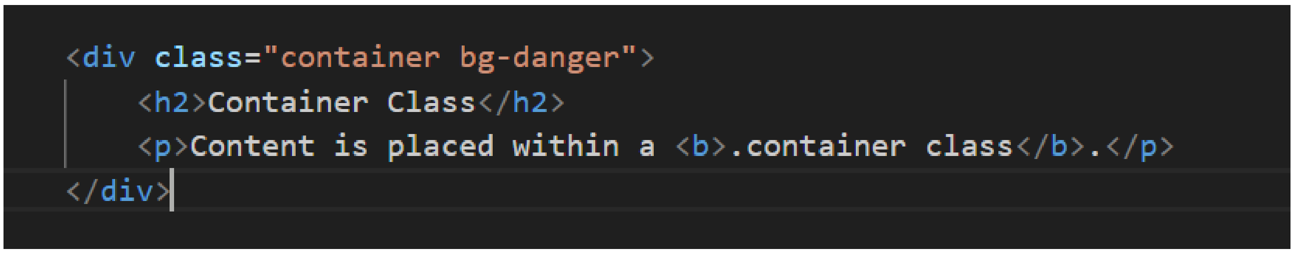
It is a fundamental layout element that helps structure the content within a fixed-width container. It ensures that the content is displayed consistently across different screen sizes and devices. The container acts as a wrapper for your web page's content and helps maintain a structured and visually appealing layout.

There are two main types of containers in Bootstrap:

`.container` and `.container-fluid`.

1. **.container:**
   The .container class represents a fixed-width container that adapts its width based on the screen size. It ensures that the content remains centred and maintains a consistent maximum width, making it easier to read and navigate the webpage. The `.container` class has a predefined width and adds horizontal padding to the content.

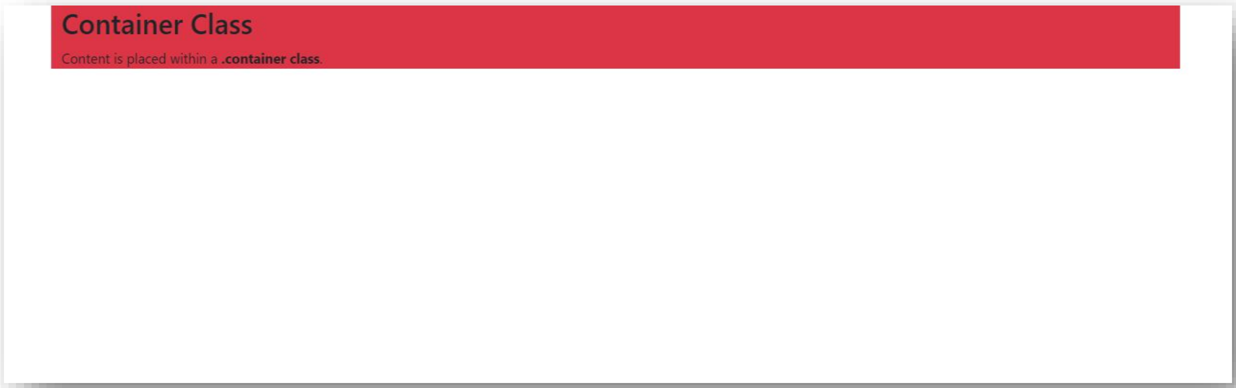
Code:



```
<div class="container bg-danger">
    <h2>Container Class</h2>
    <p>Content is placed within a <b>.container class</b>.</p>
</div>
```

This code snippet utilizes the .container class to generate a centred container and applies the **`.bg-danger`** class to give it a red background color. The container contains an **h2** heading and a paragraph that clarifies the purpose of the **.container** class. Using these Bootstrap classes guarantees that the content is centred on the page and has a unique red background color.

Output:

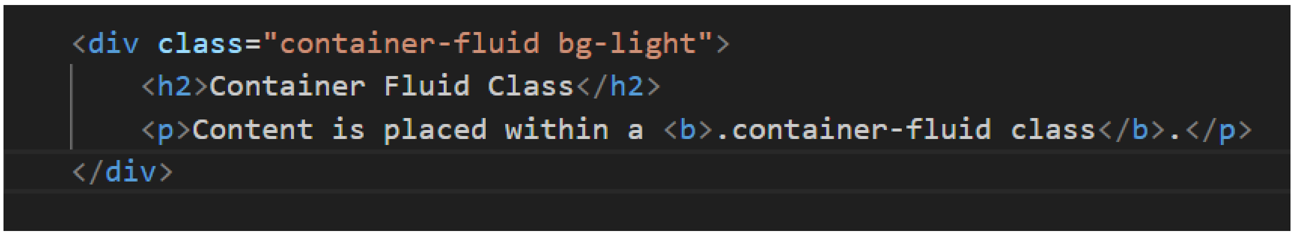


Container Class

Content is placed within a **.container class**.

2. **.container-fluid:**
   The **`.container-fluid`** class represents a full-width container that spans the entire viewport's width. It expands to fill the available screen space and is useful when you want your content to occupy the entire width of the user's screen.

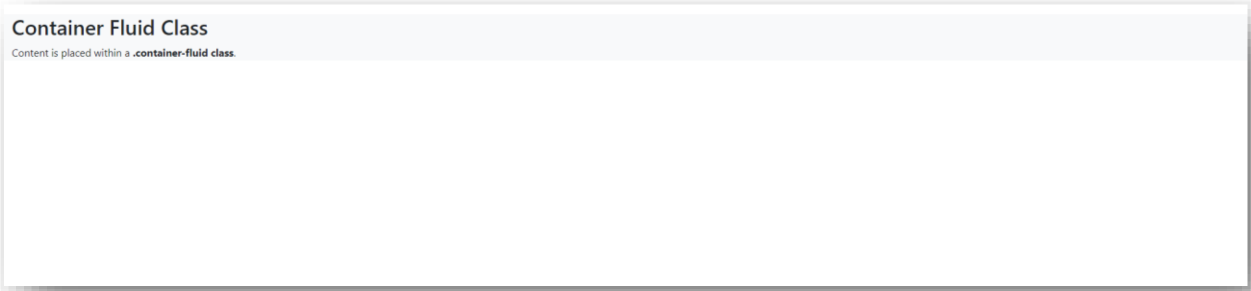
Code:



```html
<div class="container-fluid bg-light">
    <h2>Container Fluid Class</h2>
    <p>Content is placed within a <b>.container-fluid class</b>.</p>
</div>
```

The provided code snippet creates a full-width container using the **`.container-fluid`** class and gives it a light background color using the **`.bg-light`** class. Inside the container, there's an **`h2`** heading and a paragraph explaining the usage of the **`.container-fluid`** class. The combination of these Bootstrap classes ensures that the content spans the entire width of the viewport and has a visually appealing background color.

Output:



Container Fluid Class

Content is placed within a **.container-fluid class**.

Grid System:

The grid system in Bootstrap is a powerful layout mechanism that allows you to create responsive and organized web page layouts by dividing the page into rows and columns. It provides a flexible way to arrange content and maintain consistent alignment across different screen sizes and devices. The grid system is based on a combination of **`.container`**, and **`.row`** elements and columns defined by classes like **`.col-*`**. The grid system consists of 12 columns, each divided into equivalent widths.

Structure of Grid System:

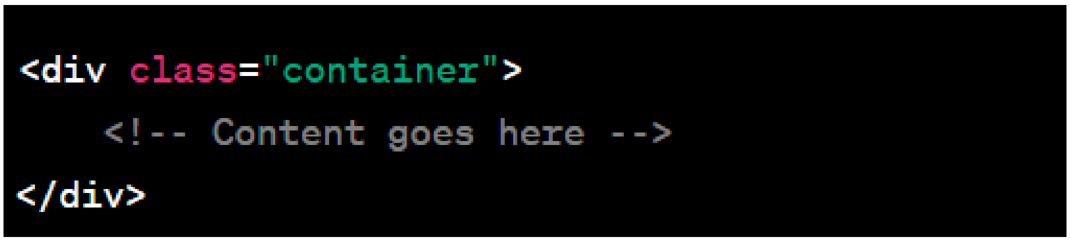
The basic structure of the Bootstrap grid system involves using containers, rows, and columns to create organized and responsive layouts. Here's an overview of each component's role in the grid system:

1. **Containers:**
   They are the outermost elements that hold your content. They provide a structured layout and help center the content on the page. Bootstrap offers two container classes:

   `.container:` Creates a fixed-width container with responsive padding. The width adjusts based on the screen size.

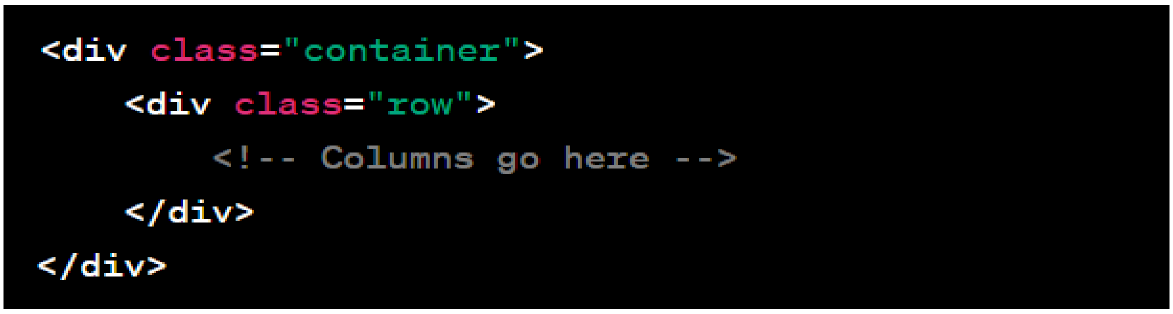
   `.container-fluid:` Creates a full-width container that spans the entire viewport width.

   Example:



```
<div class="container">
    <!-- Content goes here -->
</div>
```

2. **Rows:**
   Inside the container, rows help organize and group columns together. Rows ensure proper alignment and spacing between columns by clearing any horizontal margins.
   Example:



```
<div class="container">
    <div class="row">
        <!-- Columns go here -->
    </div>
</div>
```

3. **Columns:**
   They are the building blocks of the grid system. They define the horizontal layout of your content. Bootstrap provides classes like **`.col-*,`** where ***** indicates the number of columns the element should span. You can adjust

column widths based on screen sizes using responsive classes like **`.col-md-*`.**

Example:

```
<div class="container">
    <div class="row">
        <div class="col-md-6">Column 1</div>
        <div class="col-md-6">Column 2</div>
    </div>
</div>
```

In this example, both columns span 6 columns out of the available 12 columns on medium-sized screens or larger.

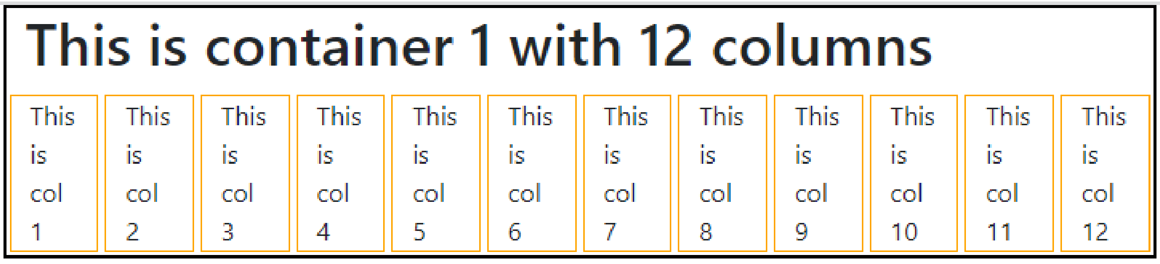
Let’s explore some more examples:

Code Example 1:

```
<body>
    <div class="container">
        <h1>This is container 1 with 12 columns</h1>
        <div class="row">
            <div class="col">This is col 1 </div>
            <div class="col">This is col 2</div>
            <div class="col">This is col 3</div>
            <div class="col">This is col 4</div>
            <div class="col">This is col 5</div>
            <div class="col">This is col 6</div>
            <div class="col">This is col 7</div>
            <div class="col">This is col 8</div>
            <div class="col">This is col 9</div>
            <div class="col">This is col 10</div>
            <div class="col">This is col 11</div>
            <div class="col">This is col 12</div>
        </div>
    </div>
</body>
```

The above code snippet shows the functionality of Grid System where a row contains 12 columns along with some text.

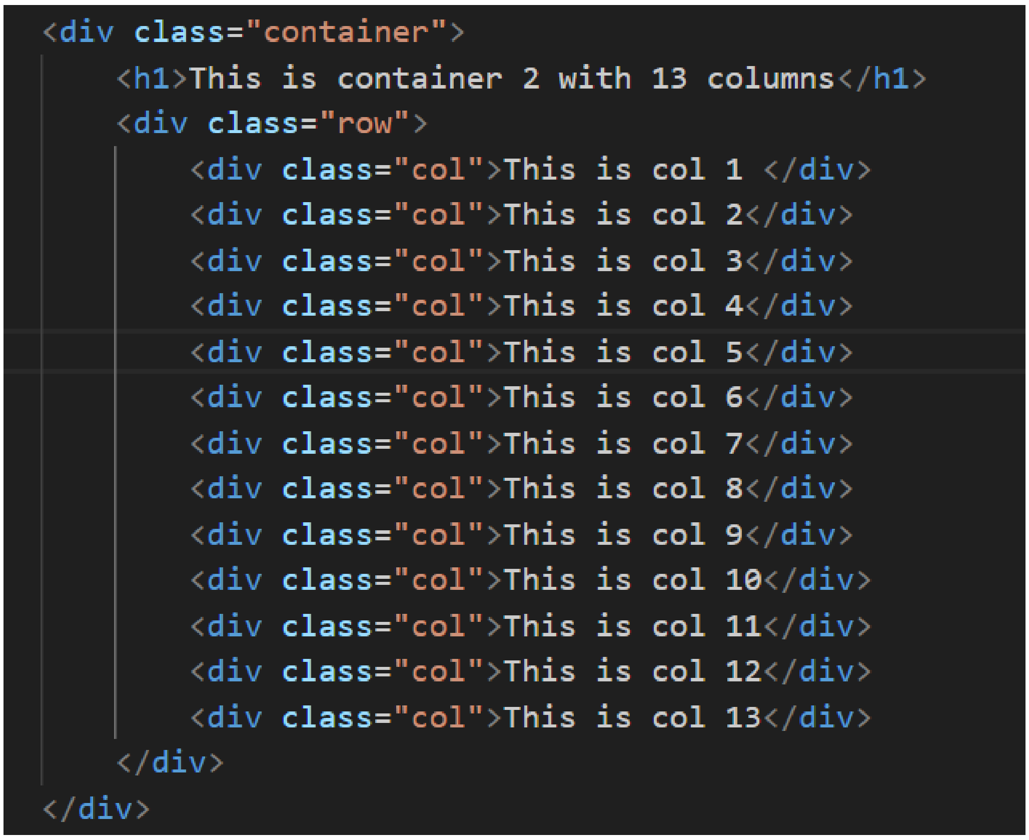
Output:



This is container 1 with 12 columns

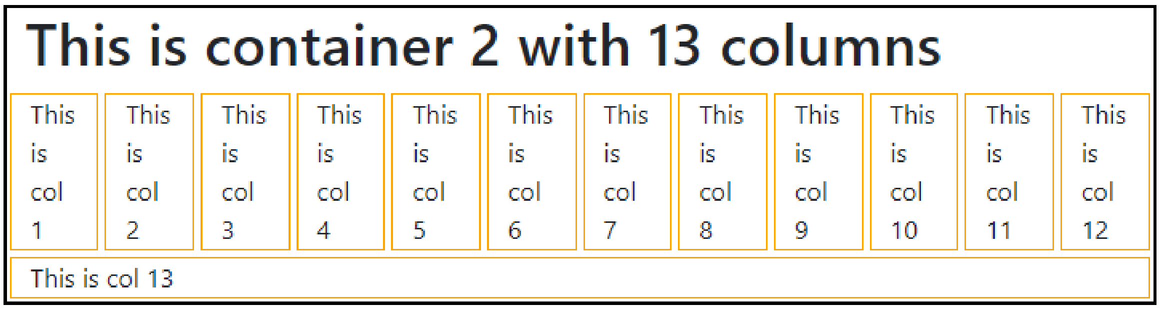
```
This  This  This  This  This  This  This  This  This  This  This  This
is    is    is    is    is    is    is    is    is    is    is    is
col   col   col   col   col   col   col   col   col   col   col   col
1     2     3     4     5     6     7     8     9     10    11    12
```

Code Example 2:



```
<div class="container">
    <h1>This is container 2 with 13 columns</h1>
    <div class="row">
        <div class="col">This is col 1 </div>
        <div class="col">This is col 2</div>
        <div class="col">This is col 3</div>
        <div class="col">This is col 4</div>
        <div class="col">This is col 5</div>
        <div class="col">This is col 6</div>
        <div class="col">This is col 7</div>
        <div class="col">This is col 8</div>
        <div class="col">This is col 9</div>
        <div class="col">This is col 10</div>
        <div class="col">This is col 11</div>
        <div class="col">This is col 12</div>
        <div class="col">This is col 13</div>
    </div>
</div>
```

Output:

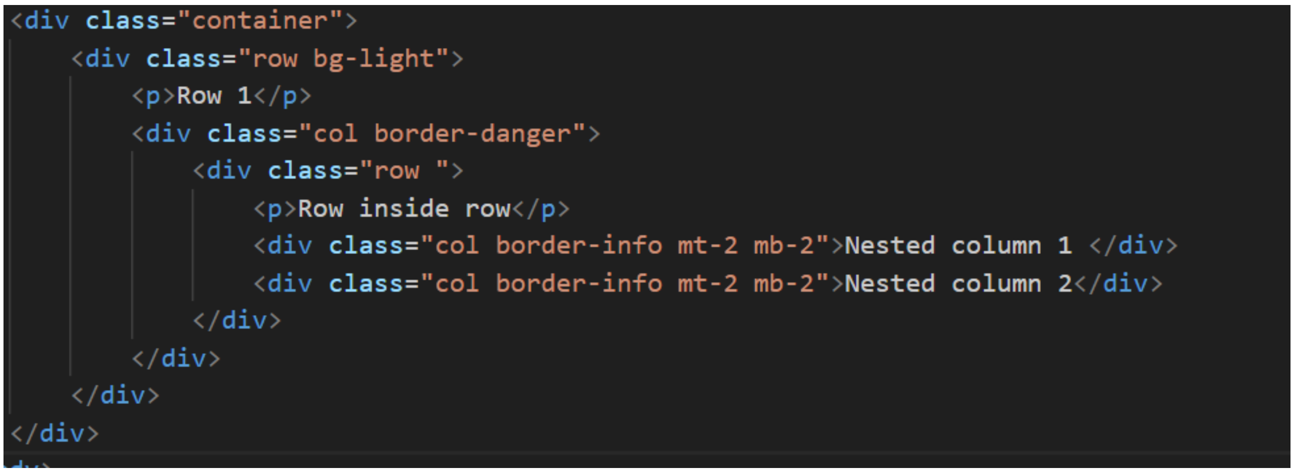


This is container 2 with 13 columns

```
This  This  This  This  This  This  This  This  This  This  This  This
is    is    is    is    is    is    is    is    is    is    is    is
col   col   col   col   col   col   col   col   col   col   col   col
1     2     3     4     5     6     7     8     9     10    11    12
This is col 13
```

The output depicts the functionality of Grid System, if you are trying to add few more columns in a row, it will fall into the next line.

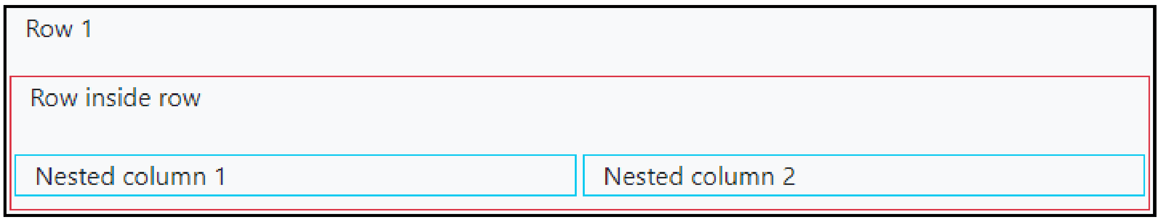
Code Example 3:



```html
<div class="container">
    <div class="row bg-light">
        <p>Row 1</p>
        <div class="col border-danger">
            <div class="row ">
                <p>Row inside row</p>
                <div class="col border-info mt-2 mb-2">Nested column 1 </div>
                <div class="col border-info mt-2 mb-2">Nested column 2</div>
            </div>
        </div>
    </div>
</div>
```

The code depicts how to create a nested columns inside a column.

Output:



```
Row 1
Row inside row
Nested column 1                Nested column 2
```

Conclusion: In this way, you can utilise grid system components effectively.