

Training Day 12 Report:

Date: 16 June 2024

Location: Science & Technology Entrepreneurs' Park

Project Title: *Bootstrap 5 Grid System and Layout*

On Day 12, we learned about one of the most important features of Bootstrap called the **Grid System**. The grid system is used to create layouts for web pages. It helps us arrange content in rows and columns, making the webpage look organized and clean on all devices like mobiles, tablets, and computers.

Project Objectives:

- Understand the Bootstrap 5 grid system and how it works.
 - Learn about rows and columns.
 - Create different column layouts using grid classes.
 - Use responsive column classes to change layout on different screen sizes.
 - Practice with examples using colors and padding to visualize the grid.
-

What is Bootstrap Grid?

Bootstrap grid uses a system based on **flexbox**, which divides the page into 12 columns. We place content inside these columns. The total columns in a row should add up to 12 or less. If the total goes over 12, the columns will move to a new line automatically.

This system is responsive, which means columns adjust their width automatically based on the device screen size.

Basic Structure of Bootstrap Grid:

1. **Container:** Holds rows and columns.
 2. **Row:** Used to create horizontal groups of columns.
 3. **Column:** The blocks inside rows that hold content.
-

How to Use the Grid:

```
<div class="container">
  <div class="row">
    <div class="col">Column 1</div>
    <div class="col">Column 2</div>
    <div class="col">Column 3</div>
  </div>
</div>
```

In the above example, the row has three columns, each sharing equal space.

Column Width Classes:

Bootstrap has classes like `.col-1`, `.col-2`, ..., `.col-12` which specify how many columns a block should take. For example, `.col-6` takes half the row width (6 out of 12 columns).

Example:

```
<div class="row">
  <div class="col-4">4 columns</div>
  <div class="col-8">8 columns</div>
</div>
```

This will create two columns; the first takes 1/3 of the width, the second takes 2/3.

Responsive Columns:

Bootstrap grid can change depending on screen size using prefixes:

- `col-sm-` for small devices like phones
- `col-md-` for medium devices like tablets
- `col-lg-` for large devices like desktops
- `col-xl-` for extra large screens

Example:

```
<div class="row">
  <div class="col-sm-6 col-md-4">Column</div>
  <div class="col-sm-6 col-md-8">Column</div>
</div>
```

This means on small screens, both columns take half the width each; on medium screens, the first column takes 4 parts and second takes 8 parts.

Practice Example:

We tried a layout with four columns using `.col-sm-3`:

```
<div class="container-fluid mt-3">
  <div class="row">
    <div class="col-sm-3 p-3 bg-primary text-white">Column 1</div>
    <div class="col-sm-3 p-3 bg-dark text-white">Column 2</div>
    <div class="col-sm-3 p-3 bg-primary text-white">Column 3</div>
    <div class="col-sm-3 p-3 bg-dark text-white">Column 4</div>
  </div>
</div>
```

This divides the page into four equal columns on small and bigger screens.

Nesting Columns:

We also learned about **nested columns**, where a column can have another row inside it to create smaller grids.

Example:

```
<div class="row">
  <div class="col-8">
    Main content here
    <div class="row">
```

```
<div class="col-6">Nested col 1</div>
  <div class="col-6">Nested col 2</div>
</div>
</div>
<div class="col-4">Sidebar</div>
</div>
```

This helps in making complex layouts.

Offset and Order:

Bootstrap allows us to move columns around using **offset** and **order** classes.

- `.offset-*` adds space before a column.
- `.order-*` changes the position of a column.

Example:

```
<div class="row">
  <div class="col-4 offset-4">Centered column</div>
</div>
```

This moves the column to the center by adding empty space to the left.

Summary of Grid Features:

- Grid divides page into 12 columns.
 - Use containers to hold rows and columns.
 - Rows contain columns.
 - Columns can have fixed width or auto width.
 - Responsive classes adjust layout on different screens.
 - Nesting allows more control.
 - Offset and order can shift column position.
-

Sample Code We Used:

```
<!DOCTYPE html>
<html>
<head>
  <title>Bootstrap Grid Practice</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>

  <div class="container mt-4">
    <div class="row">
      <div class="col-md-3 bg-warning p-3">Column 1</div>
      <div class="col-md-6 bg-success text-white p-3">Column 2 (larger)</div>
      <div class="col-md-3 bg-info p-3">Column 3</div>
    </div>
  </div>
```

```
<div class="container-fluid mt-4">
  <div class="row">
    <div class="col-sm-3 p-3 bg-primary text-white">Col 1</div>
    <div class="col-sm-3 p-3 bg-dark text-white">Col 2</div>
    <div class="col-sm-3 p-3 bg-primary text-white">Col 3</div>
    <div class="col-sm-3 p-3 bg-dark text-white">Col 4</div>
  </div>
</div>

</body>
</html>
```

What I Learned:

- Bootstrap's grid system helps make responsive layouts.
- Rows and columns are the main building blocks.
- Columns must add up to 12 or less.
- Different classes for small, medium, and large screens make the site flexible.
- Nesting columns can create more complex designs.
- Offset and order classes can control column position.

Conclusion:

Day 12 was very helpful because the grid system is the core of Bootstrap layouts. By learning about rows, columns, and responsive classes, we can now design websites that work well on any screen size. The practice exercises helped me understand how columns behave and how to organize content properly using Bootstrap 5. I am excited to learn more about Bootstrap components in the next sessions.