**FSD Laboratory 02**

**Aim:**  Design and develop a responsive web page using Bootstrap front end framework.

**Objectives:**

1. To understand HTML tags
2. To learn the styling of web pages using CSS
3. To learn Bootstrap Front End Framework.

**Theory:**

**1. Bootstrap Grid System.**

The Bootstrap Grid System is a powerful layout system that uses a series of containers, rows, and columns to layout and align content. Here’s an overview of how it works:

**Core Concepts**

# Container

The container is the basic building block of the Bootstrap grid system. It can be used to hold the rows and columns. Bootstrap provides two types of containers:

* .container: A fixed-width container.
* .container-fluid: A full-width container that spans the entire width of the viewport.

# Rows

Rows are used to create horizontal groups of columns. Each row is created using the .row class. Rows must be placed within a container.

# Columns

Columns are used to create vertical divisions within a row. Bootstrap uses a 12-column grid system. You can specify the number of columns an element should span using classes like .col-\* where \* represents the number of columns (from 1 to 12).

# Breakpoints

Bootstrap's grid system is responsive and uses a series of breakpoints to adapt the layout for different screen sizes. The breakpoints are:

* xs (extra small): <576px
* sm (small): ≥576px
* md (medium): ≥768px
* lg (large): ≥992px
* xl (extra large): ≥1200px
* xxl (extra extra large): ≥1400px

# Basic Example

Here's a basic example of using the Bootstrap grid system:

html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Bootstrap Grid Example</title>

<link href="https://maxcdn.bootstrapcdn.com/bootstrap/5.0.0/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<div class="container">

<div class="row">

<div class="col-md-4">Column 1</div>

<div class="col-md-4">Column 2</div>

<div class="col-md-4">Column 3</div>

</div>

</div>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/5.0.0/js/bootstrap.min.js"></script> </body>

</html>

In this example:

* A .container class creates a fixed-width container.
* A .row class creates a horizontal group of columns.
* Three .col-md-4 classes are used to create three equal-width columns that span 4 out of 12 columns each.

# Responsive Layout

You can create responsive layouts by using different column classes for different breakpoints. For example:

html

<div class="container">

<div class="row">

<div class="col-sm-12 col-md-6 col-lg-4">Column 1</div>

<div class="col-sm-12 col-md-6 col-lg-4">Column 2</div>

<div class="col-sm-12 col-lg-4">Column 3</div>

</div>

</div>

In this example:

* On extra small and small devices (<768px), all three columns span the full width (12 columns).
* On medium devices (≥768px), the first two columns span 6 columns each (half width), and the third column spans the full width.
* On large devices (≥992px), all three columns span 4 columns each (one-third width).

# Offsetting Columns

You can offset columns to create space around them: html

<div class="container">

<div class="row">

<div class="col-md-4 offset-md-4">Centered Column</div> </div>

</div>

In this example:

• The column is centered by offsetting it by 4 columns on the left and spanning 4 columns.

# Nesting Columns

You can nest columns within columns to create more complex layouts: html

<div class="container">

<div class="row">

<div class="col-md-8">

<div class="row">

<div class="col-md-6">Nested Column 1</div>

<div class="col-md-6">Nested Column 2</div>

</div>

</div>

<div class="col-md-4">Column</div>

</div>

</div>

In this example:

* The first column spans 8 columns and contains a nested row with two columns each spanning 6 columns.
* The second column spans 4 columns.

The Bootstrap Grid System provides a flexible and powerful way to create responsive layouts using these basic concepts and classes.

**2. Bootstrap .container and .container-fluid class.**

The .container and .container-fluid classes in Bootstrap are fundamental for creating responsive and flexible layouts. They serve as the main building blocks for the grid system by providing different types of containers for holding rows and columns.

# .container

The .container class creates a fixed-width container that changes its width based on the current breakpoint. The container is responsive and will adapt to the screen size at various breakpoints.

* **Fixed Width**: The width of the container changes according to the breakpoint, providing a margin on both sides of the container.
* **Responsive**: The container adjusts its width based on the screen size to ensure the content is appropriately spaced and readable.

•

Here is an example:

html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Bootstrap .container Example</title>

<link href="https://maxcdn.bootstrapcdn.com/bootstrap/5.0.0/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<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>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/5.0.0/js/bootstrap.min.js"></script> </body>

</html>

# .container-fluid

The .container-fluid class creates a full-width container that spans the entire width of the viewport, regardless of the screen size.

* **Full Width**: The container always takes up the full width of the viewport, providing no margin on the sides.
* **Responsive**: The content inside the container will still respond to the grid system and breakpoints.

Here is an example:

html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Bootstrap .container-fluid Example</title>

<link href="https://maxcdn.bootstrapcdn.com/bootstrap/5.0.0/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<div class="container-fluid">

<div class="row">

<div class="col">Column 1</div>

<div class="col">Column 2</div>

<div class="col">Column 3</div>

</div>

</div>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/5.0.0/js/bootstrap.min.js"></script> </body>

</html>

# Comparison

* **Usage**: Use .container for a fixed-width layout that adapts to different breakpoints, providing consistent margins. Use .container-fluid for a layout that spans the entire width of the viewport.
* **Behavior**: .container provides padding on the left and right, whereas .container-fluid does not, making it ideal for edge-to-edge content.

# Breakpoints and Container Widths

For the .container class, the width varies depending on the breakpoint:

* xs (extra small): Auto (100% width)
* sm (small): 540px
* md (medium): 720px
* lg (large): 960px
* xl (extra large): 1140px
* xxl (extra extra large): 1320px

For the .container-fluid class, the width is always 100%.

By using these container classes, you can ensure that your content is well-structured and responsive, providing an optimal viewing experience across different devices and screen sizes.

**FAQ:**

**1. What is a responsive website?**

A responsive website is designed to provide an optimal viewing experience across a wide range of devices, from desktop computers to mobile phones. The main goal is to ensure that the site is easily readable and navigable, with minimal resizing, panning, and scrolling required.

# Key Features of Responsive Web Design

1. **Fluid Grid Layouts**:
   * Uses a flexible grid system to adjust the layout based on the screen size. Instead of fixed-width layouts, elements are sized in relative units like percentages, which allows them to scale proportionally.
2. **Flexible Images**:
   * Images and other media are designed to scale and adapt to different screen sizes. This can be achieved using CSS techniques like max-width: 100%, which ensures that images don’t exceed the width of their container.
3. **Media Queries**:
   * CSS media queries are used to apply different styles based on the characteristics of the device, such as its width, height, or orientation. This allows for different layouts and styles to be applied depending on the screen size.

css

@media (max-width: 768px) {

.container {

padding: 10px;

}

}

@media (min-width: 769px) {

.container {

padding: 20px;

}

}

1. **Responsive Typography**:
   * Font sizes and line heights are adjusted according to the screen size to maintain readability. This often involves using relative units like em or rem instead of fixed units like px.
2. **Adaptive Navigation**:
   * Navigation menus are designed to adapt to different screen sizes. For example, a horizontal menu on a desktop may turn into a dropdown or a hamburger menu on mobile devices.
3. **Touchscreen-Friendly**:
   * Responsive design also takes into account touch interactions, ensuring that buttons and links are easy to tap on touchscreens. o

# Benefits of Responsive Design

* **Improved User Experience**: Provides a seamless and consistent experience across different devices and screen sizes.
* **Cost-Effective**: Avoids the need to create and maintain separate versions of a website for different devices (e.g., a desktop version and a mobile version).
* **SEO Benefits**: Google prefers responsive designs because they offer a single URL for all devices, making it easier for search engines to index and rank the content.
* **Future-Proofing**: A responsive design is adaptable to new devices and screen sizes, making it more future-proof.

•

# Examples of Responsive Design

1. **Fluid Grid Layout**: A layout that adjusts columns and elements proportionally based on the screen size.
2. **Responsive Images**: Images that scale up or down based on the size of the viewport to fit within their container.
3. **Media Queries**: CSS rules that apply different styles based on the device’s screen size or other features.

By implementing responsive design principles, websites can offer a more user-friendly experience, adapt to various devices, and maintain a cohesive look and feel across all platforms.

**2. How Bootstrap helps to design a responsive website?**

Bootstrap simplifies designing responsive websites by providing a comprehensive framework that includes pre-defined CSS classes, JavaScript components, and a responsive grid system. Here’s how Bootstrap helps in creating responsive designs:

# Responsive Grid System

Bootstrap’s grid system is based on a 12-column layout that adjusts according to the screen size. The grid is flexible and allows for complex layouts to be easily managed across different devices.

* **Containers**: Use .container for fixed-width layouts or .container-fluid for full-width layouts.
* **Rows**: Create horizontal groups of columns using .row.
* **Columns**: Define the width of columns using classes like .col-md-4, .col-lg-3, etc., which adjust automatically based on the viewport size.
* **Breakpoints**: Bootstrap provides classes for different breakpoints (xs, sm, md, lg, xl, xxl) to control the layout at various screen sizes.

html

<div class="container">

<div class="row">

<div class="col-md-6 col-lg-4">Column 1</div>

<div class="col-md-6 col-lg-4">Column 2</div>

<div class="col-md-12 col-lg-4">Column 3</div>

</div>

</div>

# Responsive Utility Classes

Bootstrap offers utility classes to show or hide content based on screen size. These classes help you manage visibility without writing custom media queries.

• **.d-none, .d-sm-block, .d-md-none**: Control the display of elements based on screen sizes.

html

<p class="d-none d-md-block">Visible on medium and larger screens.</p> <p class="d-block d-md-none">Visible only on small screens.</p>

# Responsive Typography

Bootstrap includes responsive typography utilities to adjust font sizes and text alignment based on viewport size.

* **Responsive Font Sizes**: Use classes like .fs-1, .fs-2, etc., for font size adjustments.
* **Text Alignment**: Control text alignment with classes like .text-center, .text-md-left, .textlg-right

.

html

<h1 class="fs-4 fs-md-3 fs-lg-2">Responsive Heading</h1>

<p class="text-center text-md-left">Responsive Text Alignment</p>

# Responsive Images and Media

Bootstrap provides classes to make images and media elements responsive, ensuring they scale properly within their containers.

* **.img-fluid**: Makes images responsive by setting max-width: 100% and height: auto.

html

<img src="image.jpg" class="img-fluid" alt="Responsive Image">

* **Embedded Media**: Use .embed-responsive and .embed-responsive-item to create responsive video embeds.

html

<div class="embed-responsive embed-responsive-16by9">

<iframe class="embed-responsive-item" src="video.mp4" allowfullscreen></iframe> </div>

# Responsive Navigation

Bootstrap provides components for responsive navigation, including collapsible menus and hamburger icons for mobile devices.

• **Navbar Component**: Use .navbar, .navbar-toggler, and .collapse classes to create a responsive navigation bar.

html

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<a class="navbar-brand" href="#">Brand</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" datatarget="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<ul class="navbar-nav">

<li class="nav-item active">

<a class="nav-link" href="#">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Features</a>

</li>

</ul>

</div>

</nav>

# JavaScript Components

Bootstrap includes JavaScript components that are responsive out of the box, such as modals, carousels, and tooltips. These components can adapt to different screen sizes and orientations.

* **Modals**: Responsive popups that adjust size based on the viewport.
* **Carousels**: Responsive image sliders that work on both mobile and desktop. html

<div id="carouselExample" class="carousel slide">

<div class="carousel-inner">

<div class="carousel-item active">

<img src="slide1.jpg" class="d-block w-100" alt="Slide 1">

</div>

<div class="carousel-item">

<img src="slide2.jpg" class="d-block w-100" alt="Slide 2"> </div>

</div>

<a class="carousel-control-prev" href="#carouselExample" role="button" data-slide="prev">

<span class="carousel-control-prev-icon" aria-hidden="true"></span>

<span class="sr-only">Previous</span>

</a>

<a class="carousel-control-next" href="#carouselExample" role="button" data-slide="next">

<span class="carousel-control-next-icon" aria-hidden="true"></span>

<span class="sr-only">Next</span>

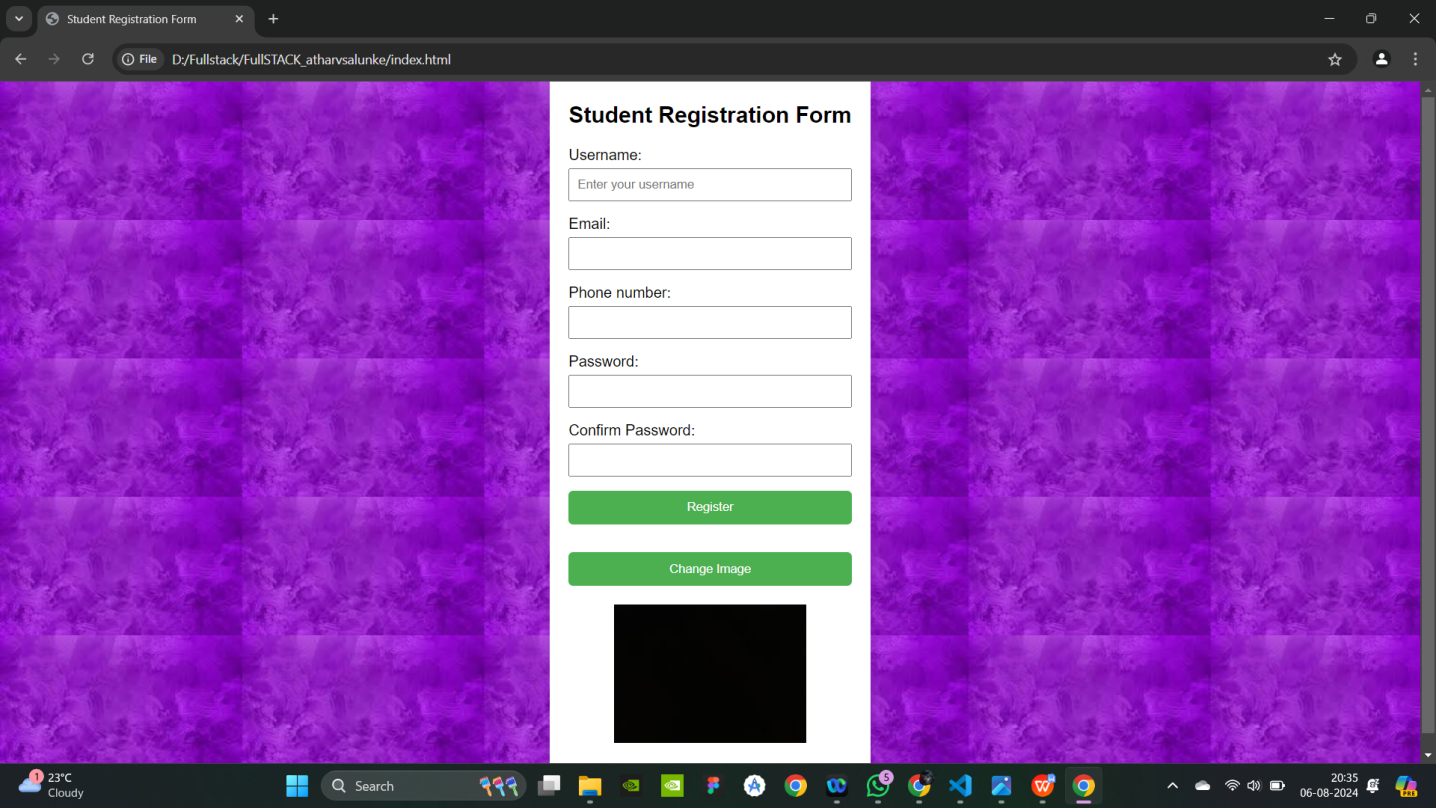
</a>

</div>

# Summary

Bootstrap streamlines the process of building responsive websites by offering a robust grid system, responsive utilities, and pre-designed components that adapt to different screen sizes. This helps developers create consistent, user-friendly interfaces across a wide range of devices without needing to write extensive custom CSS.

**Output:** Screenshots of the output to be attached.



**Problem Statement:**

Design and develop a responsive web page (For example student registration, course enrollment, library management system, online shopping system etc.) using Bootstrap front end framework. Web pages should contain HTML5 elements (Use all possible formatting for example font, colour etc.).

Web page should include various

* images,
* links within the page,

* tables
* lists
* form elements
* paragraphs
* links to other pages for navigation, ● new tabs.

Use Cascaded Style Sheets (CSS) to style the web pages designed. Make use of ● Three types of styles (Inline, Internal and External)

* border,
* margins,
* padding,
* navigation,
* dropdown list
* tag selector
* class selector
* id selector
* exploring background image and position property etc.