**FSD Laboratory 02**

**Name: Harshil Patel**

**Batch:A2**

**Roll no: PA 26**

**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 and responsive layout system used in web development. It's a part of the Bootstrap framework, which is a popular front-end framework for building responsive and mobile-first websites and web applications.

The Bootstrap grid system is based on a 12-column layout. It allows you to divide your webpage into rows and columns, making it easy to create complex and flexible layouts. Here are the key components and classes of the Bootstrap grid system:

Container: To create a grid layout, you start with a container. Bootstrap provides two types of containers: .container and .container-fluid. The .container class creates a fixed-width container, while the .container-fluid class creates a full-width container that spans the entire viewport.

<div class="container">

<!-- Your content goes here -->

</div>

Row: Inside the container, you create rows to define horizontal groups of columns. Rows are defined using the .row class.

<div class="container">

<div class="row">

<!-- Columns go here -->

</div>

</div>

Columns: Columns are placed inside rows and define the actual content areas. You can specify how many columns each element should occupy using classes like .col-xs-, .col-sm-, .col-md-, and .col-lg-, where xs, sm, md, and lg represent different screen sizes (extra small, small, medium, and large). The total number of columns within a row should add up to 12.

<div class="container">

<div class="row">

<div class="col-md-6">

<!-- Content for the first half of the row -->

</div>

<div class="col-md-6">

<!-- Content for the second half of the row -->

</div>

</div>

</div>

Offset: You can also add offsets to columns using classes like .offset-md-2. This allows you to create spacing between columns.

<div class="container">

<div class="row">

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

<!-- Content -->

</div>

<div class="col-md-4 offset-md-4">

<!-- Content with a 4-column offset -->

</div>

</div>

</div>

Nesting: You can nest rows and columns within other columns to create more complex layouts.

<div class="container">

<div class="row">

<div class="col-md-6">

<!-- Content -->

</div>

<div class="col-md-6">

<div class="row">

<div class="col-md-6">

<!-- Nested content -->

</div>

<div class="col-md-6">

<!-- Nested content -->

</div>

</div>

</div>

</div>

</div>

The Bootstrap grid system makes it easy to create responsive layouts that adapt to various screen sizes. It's a versatile and widely used tool in web development for creating structured and visually appealing designs.

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

In Bootstrap, the container and container-fluid classes are used to define the layout and width of the content container within your web page. These classes are essential for creating responsive and structured layouts in Bootstrap-based web projects. Here's an explanation of each class:

container Class:

The .container class is used to create a fixed-width container for your website's content.

It centers the content horizontally within the viewport and adds responsive padding on the left and right sides.

The width of the .container class is adaptive and changes based on the screen size.

Example:

<div class="container">

<!-- Your content goes here -->

</div>

container-fluid Class:

The .container-fluid class is used to create a full-width, fluid container for your content.

It spans the entire width of the viewport, making it suitable for backgrounds or full-width elements.

The content within a .container-fluid extends to the edges of the screen.

Example:<div class="container-fluid">

<!-- Your content goes here -->

</div>

When to Use Which Class:

Use .container when you want to create a centered and fixed-width content container. This is suitable for most page content, like text, images, and forms.

Use .container-fluid when you need a full-width container, such as for a header or background element that should stretch across the entire viewport width.

It's common to use a combination of both classes within your Bootstrap layout to achieve the desired effect. For example, you might use a .container for the main content section of your page and a .container-fluid for the header or footer.

Remember that Bootstrap's grid system (rows and columns) is typically placed inside these container elements to create structured layouts. This combination of containers and the grid system helps you create responsive and visually appealing web designs.

FAQ:

1. What is a responsive website?

A responsive website is a website that is designed and developed to adapt and respond to various screen sizes and devices. This means that the layout and content of the website automatically adjust to provide an optimal viewing experience on a wide range of devices, including desktop computers, laptops, tablets, and smartphones.

Key characteristics and features of a responsive website include:

Fluid Layouts: Responsive websites use fluid grid systems that allow content to expand or contract based on the screen size. This ensures that elements on the page are not fixed to specific pixel dimensions and can adjust proportionally.

Media Queries: CSS media queries are used to apply different styles and layouts to different screen sizes or device types. Media queries allow you to define breakpoints where the design should change to accommodate various screen widths.

Flexible Images and Media: Images and media elements are coded to scale and resize appropriately, preventing them from overflowing or becoming too small on smaller screens. This often involves setting max-width properties to prevent images from exceeding the width of their containers.

Navigation Adjustments: Navigation menus may change in appearance and behavior on smaller screens to ensure they are easy to navigate with touch gestures or limited screen space. This could involve creating mobile-friendly navigation menus or collapsing menus into a hamburger icon.

Typography Optimization: Text elements adjust in size and line spacing to maintain readability on different screen sizes. This ensures that text is legible without requiring users to zoom in.

Touch-Friendly Interfaces: Buttons, links, and interactive elements are designed to be touch-friendly on mobile devices, with sufficient spacing and size to prevent accidental clicks.

Viewport Meta Tag: The viewport meta tag is included in the HTML to instruct the browser to scale the page according to the device's viewport width, helping to prevent zooming issues.

Cross-Browser Compatibility: Responsive websites are designed to work across different web browsers and operating systems to provide a consistent user experience.

2. How Bootstrap helps to design a responsive website?

Output: Screenshots of the output to be attached.

Bootstrap is a popular front-end framework that greatly simplifies the process of designing and developing responsive websites. It provides a set of pre-built CSS classes, JavaScript components, and responsive grid system that makes it easier for developers to create responsive and mobile-friendly web designs. Here's how Bootstrap helps in designing a responsive website:

Responsive Grid System:

Bootstrap's grid system is one of its core features. It uses a 12-column grid layout, which allows developers to create flexible and responsive layouts easily.

You can use classes like .container, .container-fluid, .row, and .col-\* to structure your content within responsive grids.

Grid classes adapt to different screen sizes, making it straightforward to create layouts that work well on various devices.

Pre-designed CSS Components:

Bootstrap provides a wide range of pre-designed CSS components, such as buttons, forms, navigation bars, alerts, modals, and more.

These components are designed to be responsive by default, which means they adapt to different screen sizes and maintain a consistent look and feel across devices.

Responsive Typography:

Bootstrap includes typography styles that automatically adjust font sizes and line heights to ensure readability on various screen sizes.

Text elements are designed to scale gracefully, ensuring that text remains legible on both small and large screens.

Mobile-First Approach:

Bootstrap follows a mobile-first approach, which means that it prioritizes the design and functionality for mobile devices by default.

This approach ensures that your website looks and works well on smaller screens and then progressively enhances the design for larger screens.

Responsive Navigation:

Bootstrap provides responsive navigation components, including responsive navigation bars and navigation menus.

The navigation components are designed to collapse into a mobile-friendly menu (often referred to as a "hamburger" menu) on smaller screens.

Device-specific CSS Classes:

Bootstrap includes CSS classes that allow you to target specific screen sizes and apply different styles or behavior using media queries.

Classes like .d-none, .d-md-block, and .d-lg-flex enable you to control the visibility and layout of elements on different devices.

Customization Options:

While Bootstrap provides a lot of ready-to-use components and styles, it's highly customizable. You can easily override Bootstrap's default styles to match your branding and design requirements.

Community and Documentation:

Bootstrap has a large and active community of developers and designers, making it easier to find solutions to common design and development challenges.

Bootstrap's documentation is comprehensive and provides clear instructions and examples for using its features.

**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.). 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.