Diploma Engineering

Laboratory Manual

(Responsive Web Page Design)
(4330705)

[Computer Engineering semester III]

Enrolment No	
Name	
Branch	
Academic Term	
Institute	

Directorate Of Technical Education
Gandhinagar–Gujarat

DTE’s Vision:

- To provide globally competitive technical education;
- Remove geographical imbalances and inconsistencies;
- Develop student friendly resources with a special focus on girls’ education and support to weaker sections;
- Develop programs relevant to industry and create a vibrant pool of technical professionals.

Institute’s Vision:

To cater skilled engineers having potential to convert global challenges into opportunities through embedded values and quality technical education.

Institute’s Mission:

- Impart quality technical education and prepare diploma engineering professionals to meet the need of industries and society.
- Adopt latest tools and technologies for promoting systematic problem-solving skills to promote innovation and entrepreneurship.
- Emphasize individual development of students by inculcating moral, ethical and life skills.

Department’s Vision:

Develop globally competent Computer Engineering Professionals to achieve excellence in an environment conducive for technical knowledge, skills, moral values and ethical values with a focus to serve the society.

Department’s Mission:

- To provide state of the art infrastructure and facilities for imparting quality education and computer engineering skills for societal benefit.
- Adopt industry-oriented curriculum with an exposure to technologies for building systems & application in computer engineering.
- To provide quality technical professional as per the industry and societal needs, encourage entrepreneurship, nurture innovation and life skills in consonance with latest interdisciplinary trends.

Certificate

This is to certify that Mr./Ms ……………………………………………
Enrollment No. ………….…………… of ……..………. Semester of…………………………………….…...... of Institute. (GTU Code :602) has completed the term work satisfactorily in Subject *Responsive Web Page Design – **4330705*** for the academic year: ………………. Term: ......……..
as prescribed in the curriculum.

Place: …………………

Date: …………………..

Subject Faculty **Head of the Department**

Preface

The primary aim of any laboratory/Practical/field work is enhancement of required skills as well as creative ability amongst students to solve real time problems by developing relevant competencies in psychomotor domain. Keeping in view, GTU has designed competency focused outcome-based curriculum -2021 (COGC-2021) for Diploma engineering programmes. In this more time is allotted to practical work than theory. It shows importance of enhancement of skills amongst students and it pays attention to utilize every second of time allotted for practical amongst Students, Instructors and Lecturers to achieve relevant outcomes by performing rather than writing practice in study type. It is essential for effective implementation of competency focused outcome- based Green curriculum-2021. Every practical has been keenly designed to serve as a tool to develop & enhance relevant industry needed competency in each and every student. These psychomotor skills are very difficult to develop through traditional chalk and board content delivery method in the classroom. Accordingly, this lab manual has been designed to focus on the industry defined relevant outcomes, rather than old practice of conducting practical to prove concept and theory.

By using this lab manual, students can read procedure one day in advance to actual performance day of practical experiment which generates interest and also, they can have idea of judgement of magnitude prior to performance. This in turn enhances predetermined outcomes amongst students. Each and every Experiment /Practical in this manual begins by competency, industry relevant skills, course outcomes as well as practical outcomes which serve as a key role for doing the practical. The students will also have a clear idea of safety and necessary precautions to be taken while performing experiment.

This manual also provides guidelines to lecturers to facilitate student-centered labactivitiesfor each practical/experiment by arranging and managing necessary resources in order that the students follow the procedures with required safety and necessary precautions to achieve outcomes. It also gives an idea that how students will be assessed by providing Rubrics.

Today's modern industry uses many frameworks for a front-end web design and Angular is one of them for developing dynamic web applications. It covers all the basics of frontend web application development using the Angular framework in order to provide developers insights into real-world challenges and scenarios they face throughout their day-to-day development process, as well as provides tips and best practices for becoming a web application developer.

Bootstrap is a free and open-source web development framework. Bootstrap is a sleek, intuitive, and powerful, mobile first front-end framework for faster and easier web development. It’s designed to ease the web development process of responsive, mobile-first websites Bootstrap provide in build classes and files to make more fun in designing and give new look more and design. It is used to convert html page design into RWD (Responsive

Web Design) which is easy to learn and have many designs related functionalities. It uses HTML, jQuery and JavaScript to provide cross platform web design solution.

This course will give basic knowledge and skills for client-side web UI frameworks, in particular Bootstrap. You will learn about grids and responsive design, Bootstrap CSS and JavaScript components. Thus this course aims to help the developers to build the websites faster without worrying about the basic commands and functions.

Although we try our level best to design this lab manual, but always there are chances of improvement. We welcome any suggestions for improvement.

Programme Outcomes (POs) :

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the *engineering* problems.

2. **Problem analysis**: Identify and analyse well-defined *engineering* problems using codified standard methods.

3. **Design/ development of solutions:** Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.

4. **Engineering Tools, Experimentation and Testing:** Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.

5. **Engineering practices for society, sustainability and environment:** Apply appropriate technology in context of society, sustainability, environment and ethical practices.

6. **Project Management:** Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.

7. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes *in field of engineering.*

Practical Outcome - Course Outcome matrix

CO1	Prepare environment for Bootstrap framework for first time use.
CO2	Build different web pages layouts adhering to all platforms and sizes.
CO3	Apply reusable bootstrap components to design effective user-friendly web pages.
CO4	Develop interactive features rich web pages using Bootstrap jQuery plug-ins.

S. No.	Practical Outcome/Title of experiment	CO1	CO2	CO3	CO4
1.	Install Bootstrap framework and understand various tags, attributes of HTML and other necessary files to make responsiveweb page.	√	-	-	-
2.	Design web page that shows department name, college name atcenter of web page by using bootstrap framework and without using bootstrap framework.	√	-	-	-
3.	Display student information content on responsive web page byusing container and container-fluid classes.	-	√	-	-
4.	Create responsive web page of your class time table by using bootstrap grid system.	-	√	-	-
5.	Shows at least five to six co-curricular/Extra activities of studentthat includes multiple images and short description of eachactivity on responsive web page using responsive images withdifferent styles and responsive tables with 3 to 4 different stylessuch as hover state when mouse over, different color of eachrow, table with striped row etc.	-	√	-	-
6.	Use bootstrap typography to create responsive web page on given blog topic.	-	√	-	-
7.	Design responsive web page for student registration form using bootstrap form layout, form control, bootstrap buttons.	-	√	-	-
8.	Design responsive web page that shows odd (sem1, sem3, sem5) and even (sem2, sem4, sem6) semester consider as menu, courses of each semester as submenu using button groups and button toolbar component.	-	-	√	-

9.	Use different bootstrap input groups components to create responsive webpage for job application or any other kind of application.	-	-	√	-
10.	Use Navigation tabs/Pills to create responsive web page for summarize of all individual units of any one course.	-	-	√	-
11.	Use Nav bar component to create responsive fixed to top menu design includes logo, menu, drop down menu, form input elements such as sign-up button, search mechanism etc And fixed to bottom menu design contains footer information.	-	-	√	-
12.	Use media, rounded media, Nested media object to create responsive web page for all family members in hierarchical order.	-	-	√	-
13.	Design a smooth page transition between homepage, about and contact us page using bootstrap transition plugin.	-	-	-	√
14.	Design a webpage with different modal dialog for “Save record confirmation”, “Delete record confirmation” using model dialog plugin of bootstrap.	-	-	-	√
15.	Design news story page to demonstrate usage of Scrollspy for multiple section, Tooltip for different photos, Collapsible and popover plugins of bootstrap.	-	-	-	√
16.	Design animated photo gallery page using Carousel bootstrap plugin with minimum seven photos.	-	-	-	√

Industry Relevant Skills

The following industry relevant skills are expected to be developed in the students by performance of experiments of this course.

Use Bootstrap Framework to build aesthetic responsive web pages that workuniformly in different devices (desktops, tablets and mobile) and operativeplatforms.

Guidelines to Course Faculty

1. Couse faculty should demonstrate experiment with all necessary implementation strategies described in curriculum.
2. Couse faculty should explain industrial relevance before starting of each experiment.
3. Course faculty should involve& give opportunity to all students for hands on experience.
4. Course faculty should ensure mentioned skills are developed in the students by asking.
5. Utilise 2 hrs of lab hours effectively and ensure completion of write up with quiz also.
6. Encourage peer to peer learning by doing same experiment through fast learners.

Instructions for Students

1. Organize the work in the group and make record of all observations.
2. Students shall develop maintenance skill as expected by industries.
3. Student shall attempt to develop related hand-on skills and build confidence.
4. Student shall develop the habits of evolving more ideas, innovations, skills etc.
5. Student shall refer technical magazines and data books.
6. Student should develop habit to submit the practical on date and time.
7. Student should well prepare while submitting write-up of exercise.

Resources/Equipment Required for all Practicals:

Sr.No.	Equipment/ Software Resources	Specification
1	Computer System	Intel I3 processor with minimum 4 GB RAM, 40GB HDD, Windows 7 or above Operating system.
2	Visual Code	Open source software from Microsoft
3	Node JS and NPM Package Manager	Open source software
4	Browser	Microsoft Edge, Google Chrome etc

References Links:

1. https://www.w3schools.com/bootstrap4/bootstrap_containers.asp
2. https://getbootstrap.com/docs/5.0/layout/containers/
3. https://www.geeksforgeeks.org/containers-in-bootstrap-with-examples/
4. https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-containers.php
5. https://www.youtube.com/watch?v=ceeRn8MQzPw
6. https://www.youtube.com/watch?v=fPdJ3I-jemc

7. https://www.youtube.com/watch?v=tSHhXUCCNTA
8. https://www.youtube.com/watch?v=PwEvMY7nnY8
9. Scrollspy · Bootstrap (getbootstrap.com)
10. Tooltips · Bootstrap (getbootstrap.com)
11. Bootstrap Collapse (w3schools.com)
12. (1500) Bootstrap 4 Scrollspy Tutorial - YouTube
13. (1500) Bootstrap 5 popover component. - YouTube
14. Carousel · Bootstrap (getbootstrap.com)
15. (1486) Bootstrap 4 Tutorial in Hindi Part 12 : Carousel in bootstrap 4 in Hindi | Image Slider in Bootstrap - YouTube

- **CONTINUOUS ASSESSMENT RUBRICS (15 Marks):**

- **Laboratory Work and Questionnaire Component (15 Marks):**

Component	Criteria	Percentage	Marks	Assessment
Laboratory Work and Questionnaire	**Excellent**	**91%-100%**	**14-15**	Demonstrates exceptional proficiency in laboratory work and questionnaire assessments, consistently applying skills and understanding effectively.
	Proficient	**71%-90%**	**11-13**	Shows a strong command of bothlaboratory work and questionnaireassessments, with minor areas forimprovement.
	Satisfactory	**51%-70%**	**8-10**	Achieves a satisfactory level ofperformance in laboratory work andquestionnaire assessments, withroom for improvement in some areas.
	Needs Improvement	**31%-50%**	**5-7**	Demonstrates limited proficiency in both laboratory work andquestionnaire assessments, withsignificant areas for improvement.
	Inadequate	**0%-30%**	**0-4**	Fails to meet acceptable standardsin both laboratory work andquestionnaire assessments;significant improvement is required.

Continuous Assessment Sheet

Enrolment No: **Name**

Name: **Term:**

Date :

SrNo	Practical Outcome/Title of experiment	Page	Date	Marks (25)	Sign
1	Install Bootstrap framework and understand various tags,attributes of HTML and other necessary files to make responsiveweb page.				
2	Design web page that shows department name, college name atcenter of web page by using bootstrap framework and withoutusing bootstrap framework.				
3	Display student information content on responsive web page byusing container and container-fluid classes.				
4	Create responsive web page of your class time table by using bootstrap grid system.				
5	Shows at least five to six co-curricular/Extra activities of studentthat includes multiple images and short description of eachactivity on responsive web page using responsive images withdifferent styles and responsive tables with 3 to 4 different stylessuch as hover state when mouse over, different color of eachrow, table with striped row etc.				
6	Use bootstrap typography to create responsive web page on given blog topic.				
7	Design responsive web page for student registration form using bootstrap form layout, form control, bootstrap buttons.				
8	Design responsive web page that shows odd (sem1, sem3, sem5) and even (sem2, sem4, sem6) semester consider as menu, courses of each semester as submenu using button groups and button toolbar component.				
9	Use different bootstrap input groups components to create responsive webpage for job application or any other kind of application.				

10.	Use Navigation tabs/Pills to create responsive web page for summarize of all individual units of any one course.				
11.	Use Nav bar component to create responsive fixed to top menu design includes logo, menu, drop down menu, form input elements such as sign-up button, search mechanism etc And fixed to bottom menu design contains footer information.				
12.	Use media, rounded media, Nested media object to create responsive web page for all family members in hierarchical order.				
13.	Design a smooth page transition between homepage, about and contact us page using bootstrap transition plugin.				
14.	Design a webpage with different modal dialog for “Save record confirmation”, “Delete record confirmation” using model dialog plugin of bootstrap.				
15.	Design news story page to demonstrate usage of Scrollspy for multiple section, Tooltip for different photos, Collapsible and popover plugins of bootstrap.				
16.	Design animated photo gallery page using Carousel bootstrap plugin with minimum seven photos.				

Practical No.1: Install Bootstrap framework and understand various tags, attributes of HTML and other necessary files to make responsive web page.

A. Objective:

- Setting up environment using Bootstrap framework to make responsive web pages of website.
- To build webpage that automatically adapt and look good on different devices and screen sizes.
- To work with bootstrap packages and accelerate the development process by leveraging ready-to-use elements and avoiding the need to build them from scratch.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the digital electronics engineering problems.
2. **Engineering Tools, Experimentation and Testing:** Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

1. Familiar with Bootstrap Framework and utilize various Bootstrap packages as per requirement.
2. Design basic layout of responsive web page of web site or web application.

D. Expected Course Outcomes(Cos)

Prepare environment for Bootstrap framework for first time use.

E. Practical Outcome(PRo)

Setup environment for Bootstrap framework and utilize various tags, attributes and other necessary resources to create first responsive, Platform-independent web page.

F. Expected Affective domain Outcome(ADos)

1. Follow Coding standards and practices.
2. Maintain tools and equipment.
3. Follow safety practices.
4. Follow ethical practices

G. Prerequisite Theory:

JQuery:jQuery is a JavaScript library that simplifies the process of HTML document manipulation, event handling, animation, AJAX requests, and more. It provides an easy-to-use and concise syntax, making it a popular choice for developers to enhance interactivity and create dynamic webpages.

jQuery uses a concise and intuitive syntax that simplifies common JavaScript tasks. The syntax revolves around the use of the $ symbol as a shorthand for the jQuery object, allowing you to easily select and manipulate HTML elements.

jQuery Syntax:

The basic syntax of jQuery involves selecting HTML elements and performing actions on them. It follows a pattern of selecting elements using CSS-like selectors and then applying methods or functions to manipulate those elements.

It is important to include the jQuery library in your project before using the syntax.You can download latest Jquery from following website.

https://releases.jquery.com/

Or

https://getbootstrap.com/

Or

https://code.jquery.com

The general syntax is as follows:

$(selector).action();

In this syntax:

- $ is the jQuery selector or function.
- selector refers to the HTML element(s) you want to target.
- action() represents the method or function you want to apply to the selected element(s).

Here's an overview of the jQuery syntax with an example:

1. Selecting Elements:
   To select HTML elements, you use the $ symbol followed by parentheses and a CSS-style selector inside the parentheses. This selector helps identify the elements you want to work with.
   Example: To select all paragraphs (<p> elements) on a webpage, you can use the following jQuery code:
   $('p')
2. Chaining Methods:
   jQuery allows you to chain multiple methods together, which means you can perform multiple operations on the selected elements in a single line of code. Each method is separated by a dot (.).

Example: To hide all paragraphs on a webpage, you can chain the .hide() method after selecting the paragraphs using the previous example:

```
$('p').hide();
```

3. Manipulating Elements:
   jQuery provides numerous methods to manipulate HTML elements. These methods allow you to modify element properties, content, styles, and more.
   Example: To change the text content of all paragraphs to "Hello, World!", you can use the .text() method:

   ```
   $('p').text('Hello, World!');
   ```

4. Event Handling:
   jQuery simplifies event handling by providing methods to attach event handlers to elements.
   Example: To execute a function when a button with the ID "myButton" is clicked, you can use the .click() method:

   ```
   $('#myButton').click(function() {
     // Your code here
   });
   ```

   These are just a few examples of the jQuery syntax. jQuery provides a comprehensive set of methods and functionalities that can be combined and customized to suit your specific needs in web development.

Consider example of HTML file with the following structure:

```
<html>
<head>
<title>jQuery Example</title>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
   $(document).ready(function() {
     // jQuery code here
   });
</script>
</head>
<body>
<h1>Hello, jQuery!</h1>
```

```
<button id="myButton">Click Me</button>
</body>
</html>
```

In this example, we have included the jQuery library by adding the <script> tag with the source URL to the jQuery CDN (Content Delivery Network). We have also wrapped our jQuery code inside the $(document).ready() function, which ensures that the code is executed only when the document (HTML) has finished loading.

Let's illustrate some basic jQuery actions with the example HTML file:

1. Selecting Elements and Modifying Content:
   To select the <h1> element and change its text content, you can use the following jQuery code:

```
$(document).ready(function() {
  $('h1').text('Welcome to jQuery!');
});
```

2. Handling Events:
   To add an event handler to the button element and perform an action when clicked, you can use the following jQuery code:

```
$(document).ready(function() {
  $('#myButton').click(function() {
alert('Button clicked!');
  });
});
```

In this code, we select the button element with the ID "myButton" using the $('#myButton') selector. We attach a click event handler to it using the .click() method and define the action to be performed, which in this case is displaying an alert message.

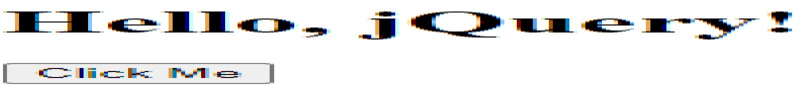
Now your final code of jQuery with HTML are given below.

```
<html>
<head>
<title>jQuery Example</title>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
  $(document).ready(function() {
```

```
 $('h1').text('Welcome to jQuery!');
 $('#myButton').click(function() {
alert('Button clicked!');
 });
});
</script>
</head>
<body>
<h1>Hello, jQuery!</h1>
<button id="myButton">Click Me</button>
</body>
</html>
```

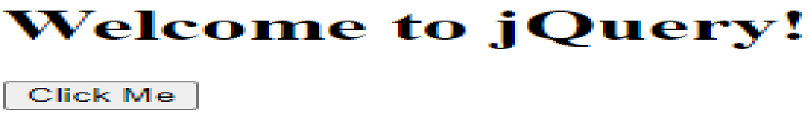
Output:

Following output is generated when you have not added text() method code in script tag.



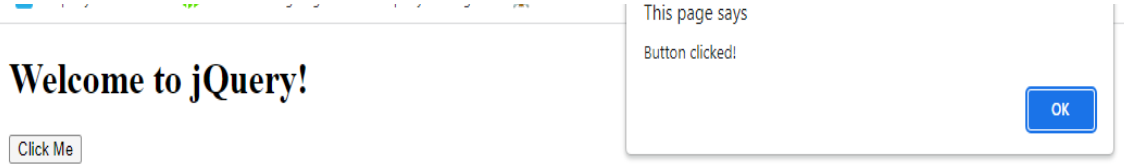
Hello, jQuery!

Following output is generated when you have added text() method code in script tag.



Welcome to jQuery!

Following output is generated when you are clicked on Click Me button.



Welcome to jQuery!

This page says

Button clicked!

Bootstrap framework:

The Bootstrap framework is a popular open-source front-end development framework that provides a collection of pre-designed and reusable components, styles, and JavaScript plugins. It was originally created by Twitter and is now maintained by a community of developers.

Bootstrap is primarily used for building responsive and mobile-first websites and web applications. It offers a set of tools and resources that simplify the development process and enable developers to create consistent, visually appealing, and user-friendly interfaces.

The framework is built on HTML, CSS, and JavaScript and provides a responsive grid system, CSS classes, and a wide range of components such as navigation bars, buttons, forms, modals, carousels, and more. These components are designed to be easily customizable and can be combined to create complex layouts and functionality.

One of the key features of Bootstrap is its focus on responsiveness. It includes a responsive grid system that allows developers to create fluid and flexible layouts that automatically adapt to different screen sizes and devices. This ensures that websites built with Bootstrap look good and provide a seamless user experience on desktops, tablets, and mobile devices.

Bootstrap also emphasizes cross-browser compatibility, handling browser inconsistencies and ensuring a consistent experience across different web browsers. It provides a consistent and unified design language, making it easier to create cohesive and professional-looking interfaces.

The framework is highly extensible and allows developers to customize its components, styles, and functionality to suit their specific project requirements. It also integrates well with other JavaScript libraries and frameworks, such as jQuery and AngularJS, providing additional flexibility and functionality.

Overall, Bootstrap is a widely used and versatile framework that simplifies front-end development, enables responsive design, ensures cross-browser compatibility, and promotes consistency in UI design. Its extensive documentation and large community support make it a valuable tool for developers seeking to create modern and efficient web projects.

To create a basic responsive web page using HTML, you will need to include several essential tags and attributes. Here's an overview of the necessary tags for building a responsive web page.

1. : This declaration specifies that the document is an HTML5 document.
2. <html>: The root element of the HTML document.
3. <head>: Contains meta-information about the web page, such as character encoding, viewport settings, title, and external stylesheets or scripts.
4. <meta charset="UTF-8">: Sets the character encoding of the document to UTF-8, which supports a wide range of characters.
5. <meta name="viewport" content="width=device-width, initial-scale=1.0">: Defines the viewport settings for responsive behavior. The width=device-width ensures that the page width adapts to the device's screen width, and initial-scale=1.0 sets the initial zoom level.

6. <title>: Specifies the title of the web page, which appears in the browser's title bar or tab.
7. <style>: Allows you to add custom CSS styles directly in the HTML file. You can define your own styles within the <style> tags.
8. <body>: Contains the main content of the web page.
9. <header>: Represents the introductory or navigational section of the web page. It typically contains the site logo, navigation menu, and other header elements.
10. <nav>: Defines a navigation menu. You can add your navigation links and structure within this tag.
11. <main>: Represents the main content area of the web page. It holds the primary content of the page, such as articles, sections, or other relevant content.
12. <footer>: Represents the footer section of the web page. It usually contains copyright information, links to additional resources, or contact details.

    You can add your content, structure, and styling within the respective sections of the HTML file to create a complete responsive web page. Remember to include appropriate CSS file to make your page responsiveand visually appealing across different devices along with manage various actions handling using JS file.

```
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Responsive Web Page</title>
<style>
body {
font-family: Arial, sans-serif;
margin: 0;
padding: 0;
   }
header {
background-color: #333;
color: #fff;
padding: 10px;
text-align: center;
   }
nav {
background-color: #555;
color: #fff;
padding: 10px;
text-align: center;
   }
main {
```

```
padding: 20px;
text-align: center;
   }
footer {
background-color: #333;
color: #fff;
padding: 10px;
text-align: center;
   }
   @media only screen and (min-width: 768px) {
nav {
display: flex;
justify-content: space-around;
     }
   }
</style>
</head>
<body>
<header>
<h1>My Responsive Web Page</h1>
</header>
<nav>
<a href="#">Home</a>
<a href="#">About</a>
<a href="#">Services</a>
<a href="#">Contact</a>
</nav>
<main>
<h2>Welcome to my website!</h2>
<p>This is a basic responsive web page design.</p>
</main>
<footer>
<p>&copy; 2023 My Website. All rights reserved.</p>
</footer>
</body>
</html>
```

It's important to note that the example provided is a basic template, and you can extend and modify it as per your specific requirements. Additionally, you can add additional HTML elements and CSS classes to create more complex and interactive web pages using Bootstrap framework.

Download BOOTSTRAP Framework:

To download the Bootstrap framework, you can follow these steps:

1. Visit the official Bootstrap website: Go to the official Bootstrap website at https://getbootstrap.com/.
2. Download Bootstrap: On the Bootstrap website, click on the "Download" button located in the top navigation menu. Alternatively, you can directly access the download page at https://getbootstrap.com/docs/5.5/getting-started/download/.

3. Choose the download method: Bootstrap offers two download options: downloading the pre-compiled CSS and JavaScript files or using the Bootstrap source files to customize and compile your own version.
    - Pre-compiled files: If you want to use the pre-compiled CSS and JavaScript files, click on the "Download" button under the "Compiled CSS and JS" section. This option is suitable if you don't need to modify Bootstrap's source code.
    - Source files: If you prefer to customize and compile your own version of Bootstrap, click on the "Download source" button under the "Source files" section. This option allows you to modify the SCSS files and recompile Bootstrap to fit your specific needs.
4. Extract the downloaded files: Once the download is complete, extract the downloaded ZIP file to a folder on your computer.

After downloading and extracting the Bootstrap files, you can include them in your project by linking the Bootstrap CSS and JavaScript files in your HTML file.
You can either reference the files locally from your project directory or use a CDN (Content Delivery Network) to include them remotely.

1>USE CDN (Content Delivery Network) inclusion:
Here's an example of including Bootstrap CSS and JavaScript files using CDN links in your HTML file:

```
<html>
<head>
<meta charset="UTF-8">
<title>Bootstrap CDN Inclusion</title>
<!-- Include Bootstrap CSS from a CDN -->
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.5.0/dist/css/bootstrap.min.css
">
</head>
```

```
<body>
<!-- Your HTML content here -->

<!-- Include Bootstrap JS from a CDN -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.5.0/dist/js/bootstrap.min.js">
</script>
</body>
</html>
```

In this example, the Bootstrap CSS file is included using the <link> tag with the href attribute pointing to the Bootstrap CSS file on the CDN. The Bootstrap JavaScript file is included using the <script> tag with the src attribute pointing to the Bootstrap JavaScript file on the CDN.

- Use Bootstrap files hosted on a CDN by directly linking to the remote files.
- This method provides the advantage of using cached files, potentially improving page load times.

2>USE Local file inclusion:

- Download the Bootstrap CSS and JavaScript files from the official website and save them in your project directory.
- Reference the local files in your HTML using relative paths.
- Example:

  ```
  <html>
  <head>
  <meta charset="UTF-8">
  <title>Bootstrap Local File Inclusion</title>
  <!-- Include Bootstrap CSS -->
  <link rel="stylesheet" href="bootstrap.min.css">
  </head>
  <body>
  <!-- Your HTML content here -->

  <!-- Include Bootstrap JS -->
  <script src="bootstrap.min.js"></script>
  </body>
  </html>
  ```

Choose the method that best suits your project's needs and preferences.

Now we are going to design our first responsive web page using bootstrap framework and necessary attributes and tags.

```
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<title>Bootstrap 4 Responsive Web Page</title>
<!-- Include Bootstrap CSS -->
<link rel="stylesheet" href="bootstrap.min.css">
</head>
<body>
<header>
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
<a class="navbar-brand" href="#">My Website</a>
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#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 <span class="sr-only">(current)</span></a>
</li><li class="nav-item">
<a class="nav-link" href="#">About</a>
</li><li class="nav-item">
<a class="nav-link" href="#">Services</a>
</li><li class="nav-item">
<a class="nav-link" href="#">Contact</a>
</li></ul>
</div>
</nav>
</header>
<main class="container">
<h1>Welcome to My Website</h1>
<p>This is a simple responsive web page created.</p>
<div class="alert alert-info" role="alert">
    This is an info alert. You can use alerts to display important messages to your users.
</div><button class="btn btn-primary">Click Me</button>
</main>
```

```
<!-- Include Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.2/dist/js/bootstrap.min.js">
</script>
</body>
</html>
```

In this example:

- The web page includes the necessary HTML structure and includes the Bootstrap CSS file via the <link> tag.
  - ✓ The <header> section contains a responsive navbar created using the navbar class and its related components, such as navbar-brand, navbar-toggler, and navbar-nav.

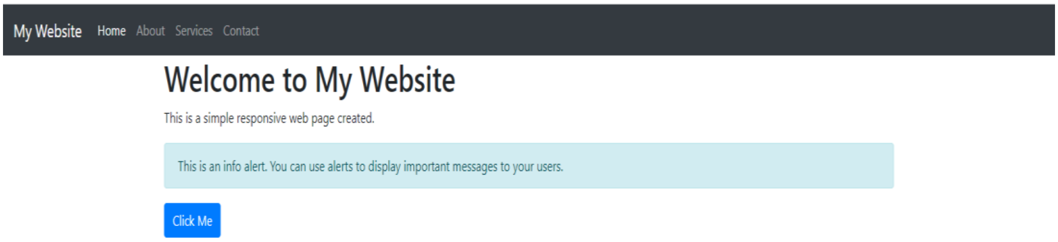
    Let's take a closer look at the components used to build a responsive navbar:
  - ✓ navbar: This class is applied to the <nav> element to create the navbar. It provides the basic structure and styling for the navigation bar.
  - ✓ navbar-brand: This class is applied to the <a> element within the navbar to represent your website or brand logo. It is typically placed at the left side of the navbar. You can customize the styling of the navbar brand as per your design requirements.
  - ✓ navbar-toggler: This class is applied to the button element that triggers the collapsing of the navbar content on smaller screens. It is used to create the "hamburger" icon or toggle button. When clicked, it expands or collapses the navbar content.
  - ✓ navbar-nav: This class is applied to the <ul> element that contains the navigation links or items. It represents the unordered list of navigation items within the navbar. You can customize the styling of the navbar links using additional classes like nav-item and nav-link.
  - ✓ In this example, the navbar has a dark background (bg-dark) and is styled to be responsive using the navbar-expand-lg class. The navbar-brand is set to "Your Website" and acts as a link to the home page. The navbar-toggler button is created with the hamburger icon, and it triggers the collapse of the navbar content on smaller screens. The navigation items are contained within the navbar-nav class, and each item is styled as a link using the nav-link class.

- The <main> section holds the main content of the web page.The container class is a fundamental component of the Bootstrap framework that helps create a responsive layout for the content of a web page. It provides a fixed-width container that adapts its size and behavior based on the screen size or viewport.

- Inside the <main>, there's an <h1> heading, a <p> paragraph, an info alert component, and a <button> styled using the btn and btn-primary classes from Bootstrap.The alert class in Bootstrap is used to create various types of alert messages to provide information, feedback, or notifications to the user. The alert-info class specifically represents an alert message with an informational or general-purpose context.
- The Bootstrap JavaScript files and jQuery are included at the end of the <body> for Bootstrap's interactive components to work.

This example demonstrates a basic responsive web page layout using Bootstrap Framework. The page will adjust its appearance and layout automatically based on different screen sizes and devices as shown below.

<u>Output 1:</u>

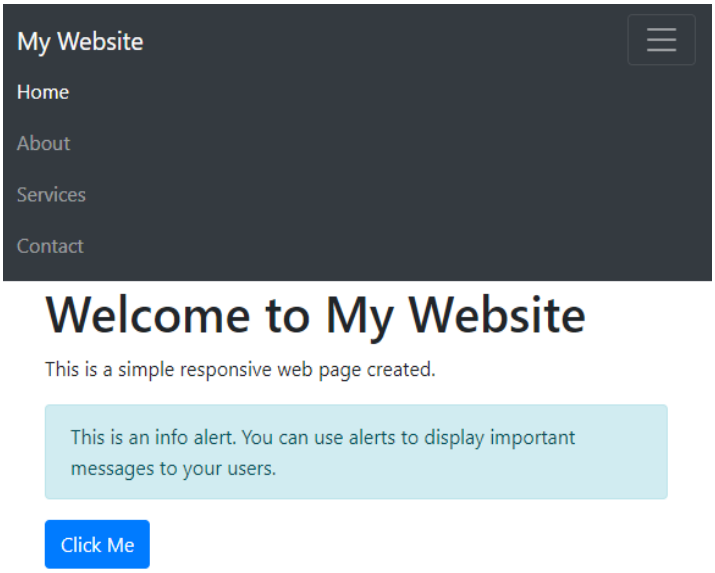


Welcome to My Website

This is a simple responsive web page created.

This is an info alert. You can use alerts to display important messages to your users.

<u>Output 2:</u>



Welcome to My Website

This is a simple responsive web page created.

This is an info alert. You can use alerts to display important messages to your users.

H. Output Source code:

1. Snapshot of Directory Structure of successfully Bootstrap Framework installed.

OUTPUT

Signature

Practical No.2: Design web page that shows department name, college name at center of web page using bootstrap framework and without using bootstrap framework.

A. Objective:

To demonstrate how to use various classes of bootstrap framework fordisplay Information or output on the web page.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the digital electronics engineering problems.
2. **Engineering Tools, Experimentation and Testing:** Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.

C. Expected Skills to be developed based on competency:

Familiar and Utilize variousclasses of Bootstrap framework to view information on web page.

D. Expected Course Outcomes(Cos)

Prepare environment for Bootstrap framework for first time use.

E. Practical Outcome(PRo)

Display department name, college name at center of web page using bootstrap framework and without using bootstrap framework.

F. Expected Affective domain Outcome(ADos)

1. Follow Coding standards and practices.
2. Maintain tools and equipment.
3. Follow safety practices.
4. Follow ethical practices

Prerequisite Theory:

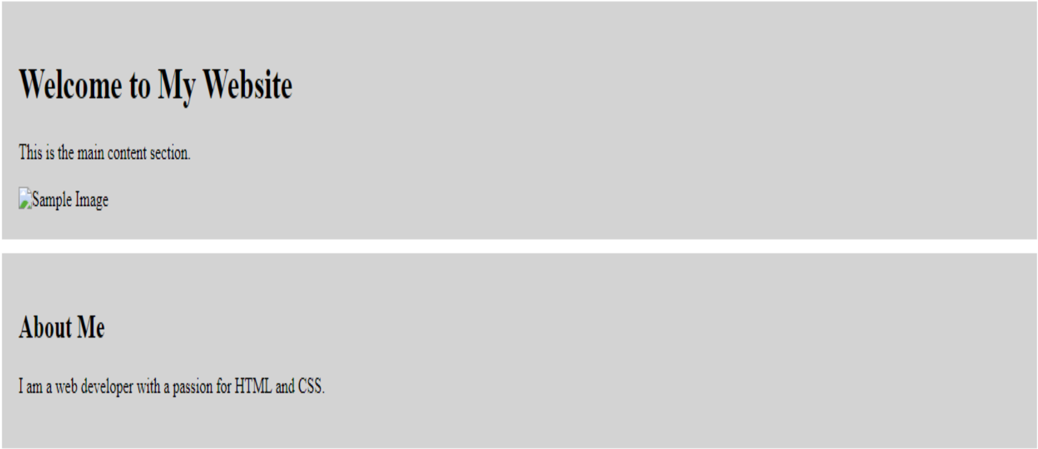
Div tag: The <div> tag is one of the most commonly used elements in HTML. It is a container or a division that allows you to group and organize other HTML elements within it. The <div> tag itself does not have any inherent meaning or specific styling, but it serves as a generic block-level container that can be styled and manipulated using CSS or JavaScript.

1. Grouping content: The **<div>** tag is often used to group related elements together. It provides a way to create logical sections or divisions within a web page. For example, you can use **<div>** tags to group a set of paragraphs, images, headings, or any other elements that are semantically related.

2. Styling and layout: The **<div>** tag can be styled using CSS to define its appearance, positioning, and layout properties. You can apply styles, such as background color, borders, padding, and margins, to create visual separation and structure within your web page.
3. JavaScript manipulation: The **<div>** tag can be targeted and manipulated using JavaScript. You can add event listeners, dynamically modify its content, or change its attributes using JavaScript functions.
4. Semantic meaning: While the **<div>** tag itself does not provide any semantic meaning, it can be used as a wrapper or container to group together other HTML elements and give them semantic context. By applying appropriate class names or IDs, you can convey the purpose or role of the content within the **<div>**.

Here's an example of how the <div> tag can be used in HTML:

```
<html>
<head>
<style>
   .container {
background-color: lightgray;
padding: 20px;
margin-bottom: 10px;
   }
</style>
</head>
<body>
<div class="container">
<h1>Welcome to My Website</h1>
<p>This is the main content section.</p>
<img src="image.jpg" alt="Sample Image">
</div>

<div class="container">
<h2>About Me</h2>
<p>I am a web developer with a passion for HTML and CSS.</p>
</div>
</body>
</html>
```



Welcome to My Website

This is the main content section.

Sample Image

About Me

I am a web developer with a passion for HTML and CSS.

In the example above, two <div> elements with the class name "container" are used to group different sections of a web page. The CSS styles applied to the "container" class give them a light gray background, padding, and a margin-bottom for spacing. Within each <div> container, you can place various HTML elements, such as headings, paragraphs, or images, to structure and organize your content.

CSS Property : justify-content

The justify-content property in CSS is used to define how flex items are distributed along the main axis (horizontal axis) of a flex container. It controls the alignment and spacing of the items. Here are the possible values for justify-content:

1. flex-start (default): The flex items are aligned to the start of the container. This means they will be positioned at the beginning of the main axis.
   ```
   .container {
   justify-content: flex-start;
   }
   ```
2. flex-end: The flex items are aligned to the end of the container. They will be positioned at the end of the main axis.
   ```
   .container {
   justify-content: flex-end;
   }
   ```
3. center: The flex items are centered within the container along the main axis.
   ```
   .container {
   justify-content: center;
   }
   ```
4. space-between: The space between the flex items is distributed evenly along the main axis. The first item is positioned at the start of the container, and the last item is positioned at the end.

```
.container {
justify-content: space-between;
}
```

5. space-around: The space around the flex items is distributed evenly along the main axis. This means that there is equal space between each pair of adjacent items, as well as at the start and end of the container.

   ```
   .container {
   justify-content: space-around;
   }
   ```

Here's an example that demonstrates the different values of justify-content:

```
<html>
<head>
<style>
   .container {
display: flex;
justify-content: center;
height: 200px;
background-color: lightgray;    }
   .item {
width: 50px;
height: 50px;
background-color: teal;    }
</style>
</head>
<body>
<div class="container">
<div class="item">ABC</div>
<div class="item">DEF</div>
<div class="item">GHI</div>
</div></body>
</html>
```

<u>Text-Align Property:</u>The text-align CSS property is used to specify the horizontal alignment of text within its containing element. It applies to block-level elements, inline-block elements, and table cells. Here are the possible values for text-align:

1. left (default): Text is aligned to the left edge of the container.
2. right: Text is aligned to the right edge of the container.
3. center: Text is centered within the container.
4. justify: Text is spread out evenly across the container, creating equal space between words and adjusting the line breaks to align with both the left and

right edges. This property is commonly used in paragraphs or blocks of text.

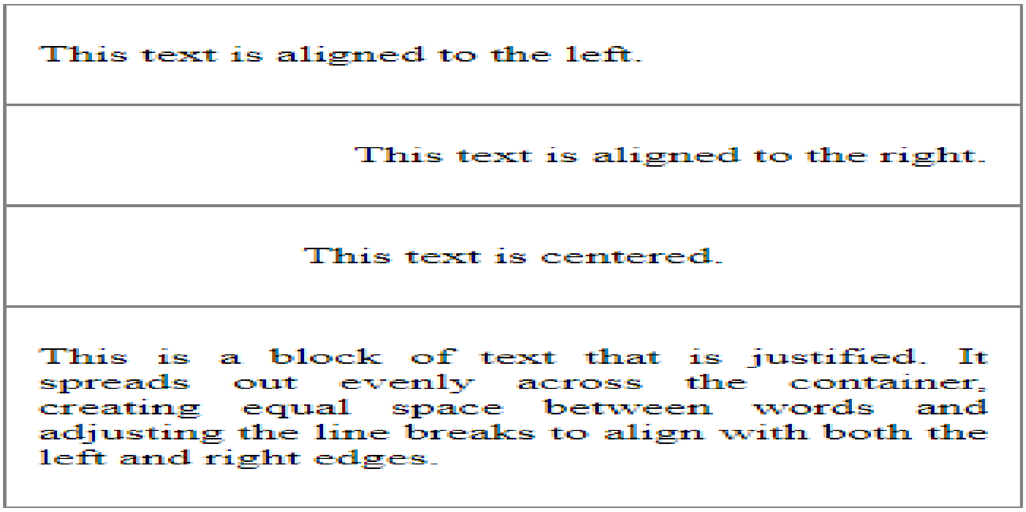
5. start: This value depends on the writing direction. In left-to-right writing, it behaves the same as left. In right-to-left writing, it behaves the same as right.
6. end: This value depends on the writing direction. In left-to-right writing, it behaves the same as right. In right-to-left writing, it behaves the same as left.

Here's an example that demonstrates the different values of text-align:

```
<html>
<head>
<style>
   .container {
width: 300px;
margin: 0 auto;
border: 1px solid gray;
padding: 10px;
   }
   .left-align {
text-align: left;
   }
   .right-align {
text-align: right;
   }
   .center-align {
text-align: center;
   }
   .justify-align {
text-align: justify;
   }
</style>
</head>
<body>
<div class="container left-align">
<p>This text is aligned to the left.</p></div>
<div class="container right-align">
<p>This text is aligned to the right.</p></div>
<div class="container center-align">
<p>This text is centered.</p></div>
<div class="container justify-align">
```

```
<p>This is a block of text that is justified. It spreads out evenly across the container, creating equal space between words and adjusting the line breaks to align with both the left and right edges.</p>
</div></body>
</html>
```

In the above example, the text-align property is applied to different <div> containers to showcase the different alignment options. The paragraphs within each container inherit the alignment from their respective parent containers.



This text is aligned to the left.

This text is aligned to the right.

This text is centered.

This is a block of text that is justified. It spreads out evenly across the container, creating equal space between words and adjusting the line breaks to align with both the left and right edges.

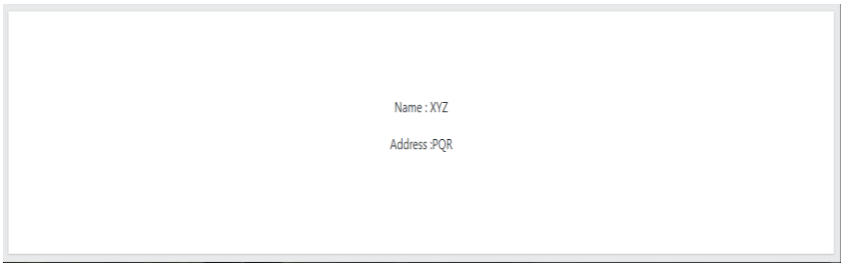
Classes : d-flex, flex-column, justify-content-center, and align-items-center

It can be used together to create a flex container and center its content vertically and horizontally. Here's a description of each class and an example to illustrate their usage:

1. **d-flex**: This class is often used in frameworks like Bootstrap and represents the display property **flex**. It enables the container to become a flex container.
2. **flex-column**: This class is used to make the flex container arrange its items vertically (from top to bottom) instead of horizontally (from left to right).
3. **justify-content-center**: This class is used to align the flex items along the vertical axis (main axis) of the container and centers them vertically. In a flex container with **flex-direction: row**, this will center the items horizontally.
4. **align-items-center**: This class is used to align the flex items along the horizontal axis (cross axis) of the container and centers them horizontally. In a flex container with **flex-direction: row**, this will center the items vertically.

Here's an example that demonstrates the usage of these classes:

```
    <divclass="d-flex flex-column justify-content-center align-items-center"style="width: 100%; height: 100vh;">
        <p>Name : XYZ</p>
        <p>Address :PQR</p>
    </div>
```



Name : XYZ

Address :PQR

G. Program Code & Output Source code:

SOURCE CODE

OUTPUT

Signature

Practical No.3: Display student information content on responsive web page by using container and container-fluid classes.

A. Objective:

To demonstrate how to design responsive web page using bootstrap framework fordisplay Information or output using container and container-fluid classes.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the *engineering* problems.
2. **Problem analysis**: Identify and analyse well-defined *engineering* problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
5. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes *in field of engineering.*

C. Expected Skills to be developed based on competency:

Applycontainer and container-fluid classes of Bootstrap framework to manage information on web page as per needs of website.

D. Expected Course Outcomes(Cos)

Build different webpages layouts adhering to all platforms and sizes.

E. Practical Outcome(PRo)

Display Student information content of web page usingcontainer and container-fluid classes of bootstrap framework.

F. Expected Affective domain Outcome(ADos)

1. Follow Coding standards and practices.
2. Maintain tools and equipment.
3. Follow safety practices.
4. Follow ethical practices

G. Prerequisite Theory:

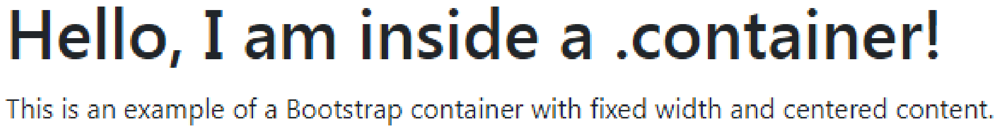
In Bootstrap, the .container and .container-fluid classes are used to create responsive container elements that help to structure and organize content on a web page. Both classes create a fixed-width container, but they differ in how they handle padding and responsiveness.

Container Class:

The .container class creates a responsive fixed-width container that is centered on the page. It sets a maximum width based on the user's device or screen size, ensuring that the content doesn't become too stretched or too narrow on different screen sizes.

Example:

```
<html>
<head>
<title>Bootstrap .container Example</title>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.cs
s"></head>
<body>
<div class="container">
<h1>Hello, I am inside a .container!</h1>
<p>This is an example of a Bootstrap container with fixed width and
centered content.</p></div>
</body>
</html>
```



Hello, I am inside a .container!

This is an example of a Bootstrap container with fixed width and centered content.

In this example, the content inside the .container will have a fixed width and be centered horizontally on the page.

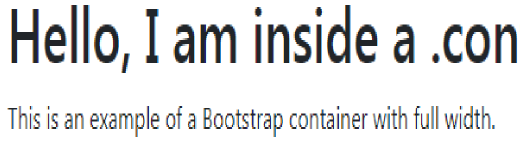
container-fluid class

The The .container-fluid class creates a full-width container that spans the entire width of the viewport, regardless of the screen size. It automatically adjusts its width according to the available screen space, making it ideal for content that needs to take up the entire width of the screen.

Example:

```
<html>
<head>
<title>Bootstrap .container-fluid Example</title>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.cs
s">
</head>
```

```
<body>
<div class="container-fluid">
<h1>Hello, I am inside a .container-fluid!</h1>
<p>This is an example of a Bootstrap container with full width.</p>
</div></body>
</html>
```



Hello, I am inside a .container-fluid!

This is an example of a Bootstrap container with full width.

In this example, the content inside the .container-fluid will take up the entire width of the screen.

It's important to note that you should use .container for most of your content layout needs, and use .container-fluid only when you need a full-width element, such as a full-width hero banner or a background image that spans the entire width of the screen.

Both container and container-fluid classes are essential tools for creating responsive and visually appealing layouts in Bootstrap-based websites. The choice between them depends on the specific design requirements of your project.

H. Program Code & Output Source code:

SOURCE CODE

OUTPUT

Signature

Practical No.4: Create responsive web page of your class time table by usingbootstrap grid system.

A. Objective:

Tofamiliar with various classes and sub classes of grid system of bootstrap framework.

To display contents of information at various positions that should be responsive in nature using various bootstrap grid classes.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the *engineering* problems.
2. **Problem analysis**: Identify and analyse well-defined *engineering* problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
5. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes *in field of engineering.*

C. Expected Skills to be developed based on competency:

Utilize specific grid system classes of Bootstrap framework to manage information on web page that should be display in responsive base.

D. Expected Course Outcomes(Cos)

Build different webpages layouts adhering to all platforms and sizes.

E. Practical Outcome(PRo)

Design time table structure specific information on web page that should be responsivebase.

F. Expected Affective domain Outcome(ADos)

1. Follow Coding standards and practices.
2. Maintain tools and equipment.
3. Follow safety practices.
4. Follow ethical practices

G. Prerequisite Theory:

H. **Prerequisite Theory:**we are going to use column, reordering column and Nesting column as below.

Column Class: In Bootstrap, the "column" class is a fundamental component used for creating responsive grid layouts. It is a key part of the grid system provided by Bootstrap, which allows you to create flexible and dynamic layouts for your web pages.

The "column" class is typically applied to HTML elements such as <div> to define their behavior within the grid system. By combining columns with other Bootstrap classes and utilities, you can create a variety of responsive layouts that adapt to different screen sizes and devices.

the "column" class is used to create responsive grid layouts. It allows you to divide the horizontal space within a row into 12 equal parts, which can then be assigned to different columns. This class is a fundamental component of Bootstrap's grid system, enabling you to create flexible and responsive designs.

To use the column class, you need to include it within a row. Here's an example of how you can create a row with three equal-width columns:

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

```
Column 1                    Column 2                    Column 3
```

In this example, each column has the "col" class applied to it. Since we haven't specified any specific width for the columns, they will automatically divide the available horizontal space equally.

If you want to create columns with different widths, you can combine the "col" class with additional classes like "col-sm", "col-md", "col-lg", or "col-xl".Bootstrap provides breakpoint classes like xs, sm, md, lg, and xl to control the column widths at different screen sizes. These classes define different breakpoints for various screen sizes, allowing you to create responsive designs. For instance, the following code snippet creates two columns where the first column occupies two-thirds of the row on medium-sized screens and the second column occupies one-third of the row:

```
<div class="row">
```

```
<div class="col-md-8">
   Two-thirds width on medium screens
</div>
<div class="col-md-4">
   One-third width on medium screens
</div>
</div>
```

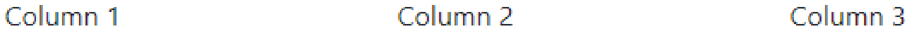
Two-thirds width on medium screens

One-third width on medium screens

By utilizing the column classes in Bootstrap, you can easily create flexible and responsive layouts for your web pages.Here's another example of how the column class can be used:

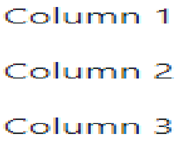
```
<div class="container">
<div class="row">
<div class="col-sm-6 col-md-4">
<p>Column 1</p></div>
<div class="col-sm-6 col-md-4">
<p>Column 2</p></div>
<div class="col-sm-6 col-md-4">
<p>Column 3</p>
</div></div>
</div>
```

Output1 :



Column 1 Column 2 Column 3

Output 2:



Column 1

Column 2

Column 3

In this example, we have a container element that holds a row. Inside the row, we have three columns, each specified with the col-sm-6 col-md-4 classes.

The classes col-sm-6 and col-md-4 define the width of the columns at different viewport sizes. In this case, col-sm-6 means that each column will occupy 6 out of 12 columns (i.e., 50% width) on small screens and larger. And col-md-4 specifies that on medium-sized screens and above, each column will occupy 4 out of 12 columns (i.e., 33.33% width).

By using this approach, the columns will automatically stack vertically on smaller screens, while maintaining the desired layout on larger screens. This responsiveness ensures that your content looks good and adapts well to different screen sizes.

Please note that the specific class names (col-sm-6, col-md-4) may vary depending on the version of Bootstrap you are using. It's always a good idea to consult the official Bootstrap documentation for the version you're working with to ensure you have the correct class names and syntax for your grid system.

column offset class

The column offset class in Bootstrap allows you to create empty space or gaps between columns within a row. It enables you either to shift columns horizontally to the right, effectively creating an offsetor to create a layout where columns are not directly adjacent to each other, providing more control over the positioning of content. The offset classes can be used alongside column classes to achieve the desired layout.

To use the column offset class in Bootstrap, you apply it alongside the column class within a row.To apply the column offset in Bootstrap, you can use **the offset-*-*** class, where the asterisks represent the screen size breakpoint and the number of columns to offset

Here's an example of how you can use column offset classes in Bootstrap:

```
<div class="container">
<div class="row">
<div class="col-md-4">
<!-- Content for the first column -->
</div>
<div class="col-md-4 offset-md-3">
<!-- Content for the second column with an offset of 3 columns -->
</div></div>
</div>
```

In the example above, we have a container with a row that contains two columns. The first column has a width of 4 columns (col-md-4). The second column has a width of 4 columns as well (col-md-4), but it also has an offset of 3 columns (offset-md-3).

With the offset-md-3 class, the second column is shifted 3 columns to the right, leaving an empty space of 3 columns before it. This creates a visual gap between the columns, resulting in a more flexible and customized layout.

You can use different offset classes, such as offset-sm-*, offset-md-*, offset-lg-*, and so on, to control the offset at different screen sizes. Adjust the offset value according to the desired number of columns you want to shift the column by.

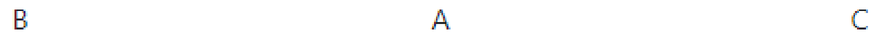
By using column offset classes in Bootstrap, you have more control over the positioning and spacing of columns within a row, allowing you to create unique and visually appealing grid layouts.

Reorder column class

The reorder column class in Bootstrap allows you to change the order of columns within a row based on different screen sizes. This feature is particularly useful for creating responsive layouts where the order of content needs to be adjusted on various devices.

To use the reorder column class in Bootstrap, you can apply the order-*-* class to the columns within a row. The asterisks represent the screen size breakpoint and the desired order value. Here's an example to help illustrate its usage:

```
<div class="container">
<div class="row">
<div class="col-md-4 order-md-2">
<!-- Content for the first column with a different order on medium devices and above -->A
</div>
<div class="col-md-4 order-md-1">
<!-- Content for the second column with a different order on medium devices and above -->B
</div>
<div class="col-md-4 order-md-3">
<!-- Content for the third column with a different order on medium devices and above -->C
</div></div>
</div>
```



B A C

In the example above, we have a container that contains a row with three columns. Each column has an order-md-* class, specifying the desired order on medium-sized devices and above. The order-md-2 class assigned to the first column means it will appear second in the layout on medium devices and above. The second column has an order-md-1 class, ensuring it appears first in the layout, and the third column has an order-md-3 class to indicate it should appear third in the layout on medium devices and above.

By using the reorder column class, you can change the order of columns, rearranging their positions within a row based on different screen sizes. This allows you to create responsive designs where content is displayed optimally across various devices and screen resolutions.

Bootstrap provides order classes for different screen size breakpoints, including xs, sm, md, lg, and xl. By utilizing these classes, you can customize the order of columns based on specific breakpoints, providing a more tailored user experience for different devices.

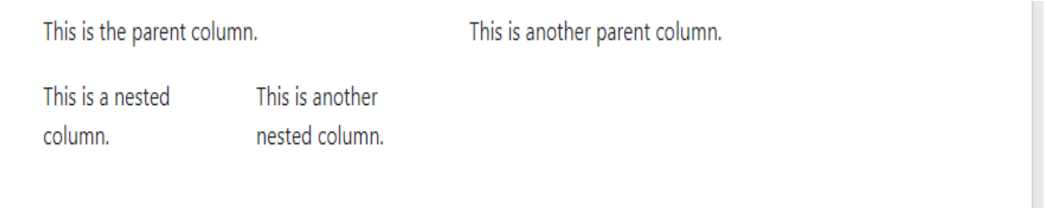
Nesting Columns :

In the Bootstrap framework, the nesting of column classes refers to the ability to create more complex and intricate grid layouts by combining multiple columns within a parent column. This feature allows for the creation of nested grids and helps achieve finer control over the positioning and alignment of content.

To nest columns in Bootstrap, you simply place a new set of columns within an existing column. The nested columns should be contained within a parent column, which acts as a container for the nested content. This nesting can be done at any level, allowing for multiple levels of hierarchy within the grid structure.

Here's an example of how nesting column classes works in Bootstrap:

```
<div class="container">
<div class="row">
<div class="col-md-6">
<!-- Parent column -->
<p>This is the parent column.</p>
<div class="row">
<div class="col-md-6">
<!-- Nested column 1 -->
<p>This is a nested column.</p>
</div>
<div class="col-md-6">
<!-- Nested column 2 -->
<p>This is another nested column.</p>
</div>
</div>
</div>
<div class="col-md-6">
<!-- Another parent column -->
<p>This is another parent column.</p>
</div></div>
</div>
```



```
This is the parent column.                    This is another parent column.

This is a nested      This is another
column.               nested column.
```

In the example above, we have a container, a row, and two parent columns within the row. The first parent column is assigned a width of 6 columns (col-md-6). Within this parent column, we have a nested row, which contains two nested columns, each assigned a width of 6 columns as well. The second parent column occupies the remaining 6 columns of the row.

By nesting columns, you can create more complex layouts that adapt to different screen sizes and provide greater flexibility in organizing your content. It's important to remember to maintain the 12-column structure of Bootstrap, as columns should always add up to 12 within a parent column.
The nested column feature in Bootstrap allows you to create sophisticated and responsive grid layouts, making it easier to design and structure your web pages.

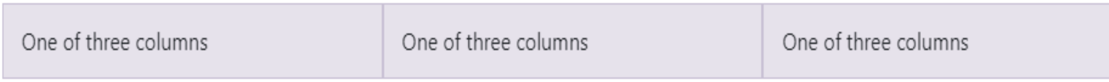
Responsive Grid System: The Bootstrap grid system is a powerful and widely used CSS grid layout framework that provides a responsive grid structure for building flexible and responsive web layouts. The grid system consists of a series of containers, rows, and columns.

Containers: Bootstrap grid layouts begin with a container. Containers are used to wrap the entire content of your webpage and provide a consistent width for the layout. There are two types of containers in Bootstrap: .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.

Rows: Inside the container, you can have one or more rows. Rows are horizontal groups that contain columns. They ensure proper alignment and spacing of columns within the grid layout. Rows are created using the .row class.

Columns: Columns are the building blocks of the grid layout. They are contained within rows and determine the width and placement of content. Bootstrap grid system uses a 12-column grid. You can define the width of a column by specifying the number of columns it should span out of the total 12. For example, .col-6 will create a column that spans half of the row, while .col-3 will create a column that spans one-fourth of the row.

The .col class is used to define columns, and you can combine it with additional classes to control the column behavior in different screen sizes. For example, .col-6 will apply the column style for all screen sizes, while .col-md-6 will apply it only for medium-sized screens and above.Here is example of equal width grid system.

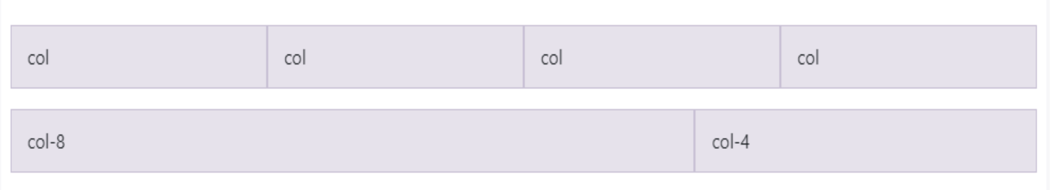


One of three columns	One of three columns	One of three columns

```
<div class="container">
<div class="row">
<div class="col">One of three columns</div>
<div class="col">One of three columns</div>
```

```
<div class="col">One of three columns</div></div>
</div>
```

Another Example:



col col col col

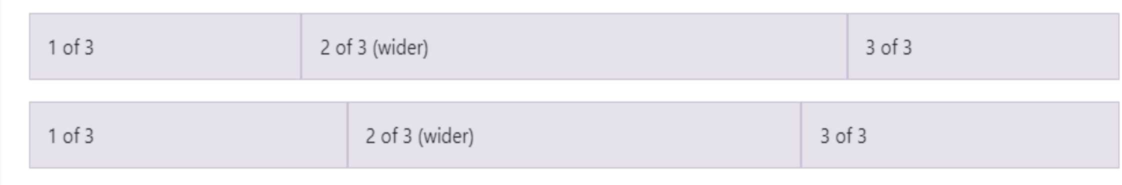
col-8 col-4

For grids that are the same from the smallest of devices to the largest, use the .col and .col-* classes. Specify a numbered class when you need a particularly sized column; otherwise, feel free to stick to .col.

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

Another Example:

```
<div class="container">
<div class="row">
<div class="col">1 of 3</div>
<div class="col-6">2 of 3 (wider)</div>
<div class="col">3 of 3</div></div>
<div class="row"><div class="col">1 of 3</div>
<div class="col-5">2 of 3 (wider)</div>
<div class="col">3 of 3</div></div>
</div>
```



1 of 3 2 of 3 (wider) 3 of 3

1 of 3 2 of 3 (wider) 3 of 3

I. Program Code & Output Source code:

SOURCE CODE

OUTPUT

Signature

Practical No.5: Shows at least five to six co-curricular/Extra activities of student that includes multiple images and short description of each activity on responsive web page using responsive images with different styles and responsive tables with 3 to 4 different styles such as hover state when mouse over, different color of each row, table with striped row etc.

A. Objective:

Use Various Table classes to design responsive web page to display data in tabular format structure.

Use Img-Fluid classes to make responsive image on web page.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the *engineering* problems.
2. **Problem analysis**: Identify and analyse well-defined *engineering* problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
5. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes *in field of engineering.*

C. Expected Skills to be developed based on competency:

UseTable and Img-Fulid Classes to managedata in tabular format on web page in responsive nature.

D. Expected Course Outcomes(Cos)

Build different WebPages layouts adhering to all platforms and sizes.

E. Practical Outcome(PRo)

Design various co-curricular/Extra activities specific information including image with different styles and various effect using table classes on web page that should be in responsive.

F. Expected Affective domain Outcome(ADos)

1. Follow Coding standards and practices.
2. Maintain tools and equipment.
3. Follow safety practices.

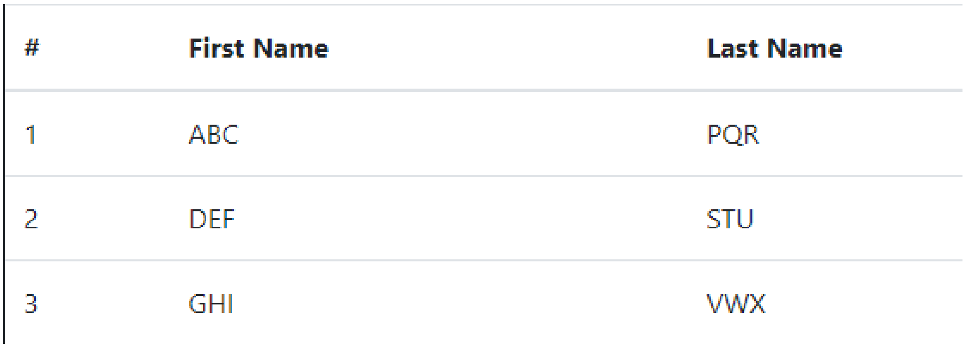
4. Follow ethical practices

G. Prerequisite Theory:

Table Class :

The Bootstrap framework provides a Table class that allows you to create responsive and visually appealing tables on your web page. The Table class offers various features and options for customizing the appearance and functionality of the table. Let me provide you with an example to demonstrate how to use the Table class in Bootstrap.

```
<html>
<head>
<link rel="stylesheet" href="bootstrap.min.css">
</head>
<body>
<table class="table">
<thead>
<tr><th>#</th><th>First Name</th>
<th>Last Name</th></tr></thead>
<tbody>
<tr><td>1</td><td>ABC</td>
<td>PQR</td></tr>
<tr><td>2</td><td>DEF</td>
<td>STU</td></tr>
<tr><td>3</td><td>GHI</td>
<td>VWX</td></tr></tbody>
</table>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"
></script>
</body>
</html>
```



#	First Name	Last Name
1	ABC	PQR
2	DEF	STU
3	GHI	VWX

In the above example, we have included the Bootstrap CSS and JavaScript files from a CDN (Content Delivery Network) to leverage the Bootstrap framework.

Within the <table> element, we have applied the "table" class to the table. This class is provided by Bootstrap and enables the default table styling.

The <thead> section contains the table header, defined using the <th> (table header) elements. In this example, we have three columns: "#", "First Name," and "Last Name."

The <tbody> section contains the table body, where the actual data rows are placed. Each row is represented by the <tr> (table row) element. Within each row, we use the <td> (table data) elements to define the cells of the table.

By using the Bootstrap Table class, the table will be styled with a clean and responsive design, adapting to different screen sizes. You can further customize the table appearance by utilizing additional Bootstrap classes, such as "table-striped" for adding zebra-striping to the table or "table-bordered" for adding borders to the cells.

Remember to include the Bootstrap CSS and JavaScript files as shown in the example for the table to render correctly.

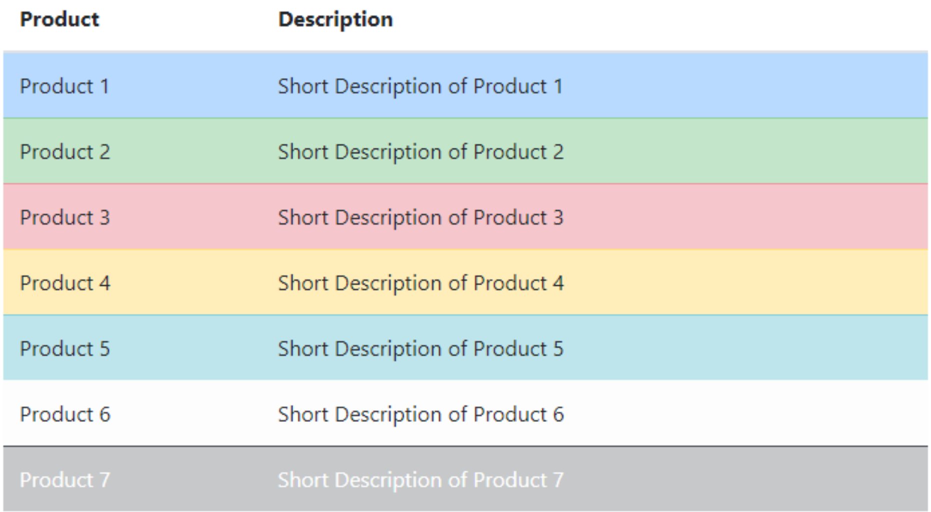
Contextual Table Classes

The table class in Bootstrap provides a set of styling options to create tables with different row colors. By using contextual classes, you can assign specific background colors to individual rows to differentiate them visually. Bootstrap provides contextual classes like table-primary, table-secondary, table-success, table-danger, table-warning, table-info, table-light, and table-dark that can be applied to table rows to achieve different color schemes.

Here's an example of a table with different row colors using Bootstrap's table classes:

```
<table class="table">
<thead>
<tr><th>Product</th>
<th>Description</th></tr>
</thead><tbody>
<tr class="table-primary">
<td>Product 1</td>
<td>Short Description of Product 1</td></tr>
<tr class="table-success">
<td>Product 2</td>
<td>Short Description of Product 2</td></tr>
<tr class="table-danger">
<td>Product 3</td>
```

```
<td>Short Description of Product 3</td></tr>
<tr class="table-warning">
<td>Product 4</td>
<td>Short Description of Product 4</td></tr>
<tr class="table-info">
<td>Product 5</td>
<td>Short Description of Product 5</td></tr>
<tr class="table-light">
<td>Product 6</td>
<td>Short Description of Product 6</td></tr>
<tr class="table-dark">
<td>Product 7</td>
<td>Short Description of Product 7</td></tr>
</tbody>
</table>
```



Product	Description
Product 1	Short Description of Product 1
Product 2	Short Description of Product 2
Product 3	Short Description of Product 3
Product 4	Short Description of Product 4
Product 5	Short Description of Product 5
Product 6	Short Description of Product 6
Product 7	Short Description of Product 7

In this example, each table row is assigned a different contextual class that determines the background color of the row. The classes used here (table-primary, table-success, table-danger, table-warning, table-info, table-light, and table-dark) correspond to different predefined color schemes in Bootstrap.

By applying these classes to the <tr> elements within the <tbody>, you can easily create a table with visually distinct rows.

Please note that Bootstrap provides several other contextual classes that you can use to define custom row colors or match your own color scheme. These classes include table-primary, table-secondary, table-success, table-danger, table-warning, table-info, table-light, and table-dark. Feel free to choose the appropriate class for your specific design needs.

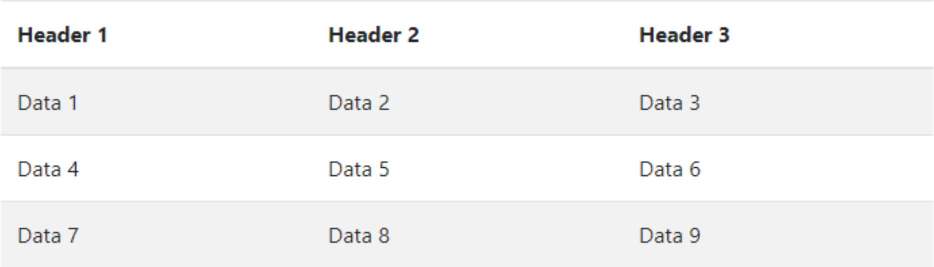
Also, ensure that you have included the Bootstrap CSS file in your HTML document for the table styles to be applied correctly.

Table stripped Class:

In Bootstrap, the "table-striped" class is used to add zebra-striping to a table. Zebra-striping refers to the alternating background colors of rows in a table, which makes it easier for users to read and differentiate between rows.

To apply the "table-striped" class to a table, you need to add it to the <table> element in your HTML markup. Here's an example:

```
<table class="table table-striped">
<thead>
<tr><th>Header 1</th>
<th>Header 2</th>
<th>Header 3</th></tr>
</thead>
<tbody>
<tr><td>Data 1</td>
<td>Data 2</td>
<td>Data 3</td></tr>
<tr><td>Data 4</td>
<td>Data 5</td>
<td>Data 6</td></tr>
<tr><td>Data 7</td>
<td>Data 8</td>
<td>Data 9</td></tr>
</tbody>
</table>
```



Header 1	Header 2	Header 3
Data 1	Data 2	Data 3
Data 4	Data 5	Data 6
Data 7	Data 8	Data 9

In the above example, the "table-striped" class is added to the <table> element. This will result in the table rows having alternating background colors. The first row will have the default background color, and the second row will have a slightly lighter background color. This pattern continues for subsequent rows, creating a striped effect.

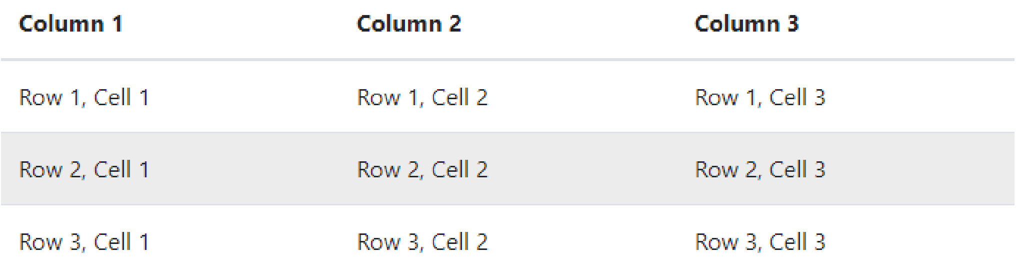
By using the "table-striped" class, you can enhance the readability and visual appeal of your tables, making it easier for users to scan and interpret the data.

Table Hover Class:

In the Bootstrap framework, the "table-hover" class is used to add a hover effect to tables. When applied, it adds a background color to the table rows when the user hovers over them with the mouse pointer, making it visually interactive and highlighting the row being hovered.

Here's an example of how to use the "table-hover" class in Bootstrap:

```
<html>
<head>
<!-- Include Bootstrap CSS -->
<link rel="stylesheet" href="bootstrap.min.css">
</head>
<body>
<div class="container">
<table class="table table-hover">
<thead>
<tr><th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th></tr>
</thead>
<tbody>
<tr><td>Row 1, Cell 1</td>
<td>Row 1, Cell 2</td>
<td>Row 1, Cell 3</td></tr>
<tr><td>Row 2, Cell 1</td>
<td>Row 2, Cell 2</td>
<td>Row 2, Cell 3</td></tr>
<tr><td>Row 3, Cell 1</td>
<td>Row 3, Cell 2</td>
<td>Row 3, Cell 3</td></tr>
</tbody></table>
</div>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js">
</script>
</body>
</html>
```



Column 1	Column 2	Column 3
Row 1, Cell 1	Row 1, Cell 2	Row 1, Cell 3
Row 2, Cell 1	Row 2, Cell 2	Row 2, Cell 3
Row 3, Cell 1	Row 3, Cell 2	Row 3, Cell 3

In this example, the "table" element has the "table-hover" class applied to it. As a result, when you hover over any row in the table body, it will change the background color of that row, indicating the hover effect. This helps in improving the user experience and makes the table more interactive.

Remember to include the Bootstrap CSS and JavaScript files in your HTML file for the Bootstrap styles and functionality to work properly.

Responsive images with different styles:

In Bootstrap, you can create responsive images that automatically adjust their size and behavior based on the screen size or device. Bootstrap provides various classes to style responsive images. Here are a few examples of different styles for responsive images in Bootstrap:

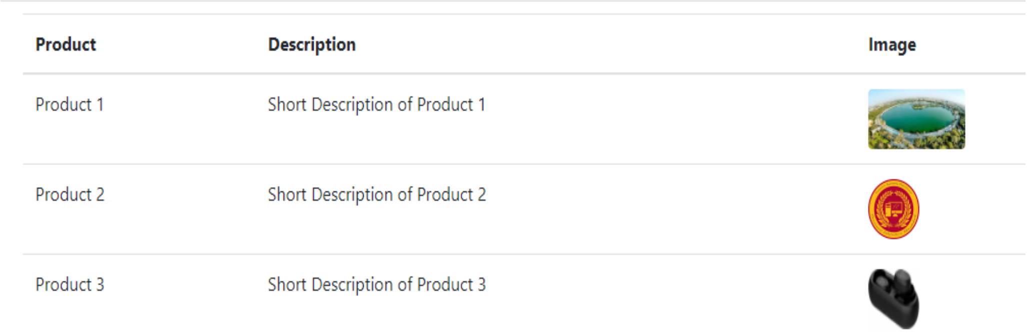
- Fluid Images:
  Fluid images automatically scale with the size of the parent container. Use the class img-fluid to make an image responsive and fluid. The image will resize proportionally as the container size changes.
  <img src="image.jpg" class="img-fluid" alt="Responsive Image">
- Rounded Images:
  To create rounded responsive images, you can combine the img-fluid class with the rounded class. This will make the image fluid and add rounded corners.
  <img src="image.jpg" class="img-fluid rounded" alt="Rounded Image">
- Circle Images:
  To create circular responsive images, use the rounded-circle class along with the img-fluid class. This will make the image circular and responsive.
  <img src="image.jpg" class="img-fluid rounded-circle" alt="Circular Image">
- Thumbnail Images:
  Bootstrap provides a thumbnail class that adds padding and a box shadow effect around the image. Combine it with the img-fluid class for responsiveness.
  <img src="image.jpg" class="img-fluid thumbnail" alt="Thumbnail Image">

- Centered Images:
  To horizontally center an image within its parent container, use the mx-auto class along with d-block. This is useful when you want to center an image within a column or a specific container.
  <img src="image.jpg" class="img-fluid mx-auto d-block" alt="Centered Image">

These are just a few examples of the different styles you can apply to responsive images in Bootstrap. By combining these classes and custom CSS, you can achieve a wide range of image styling options that adapt to different screen sizes and devices.Here's an example of a Bootstrap responsive table with images:

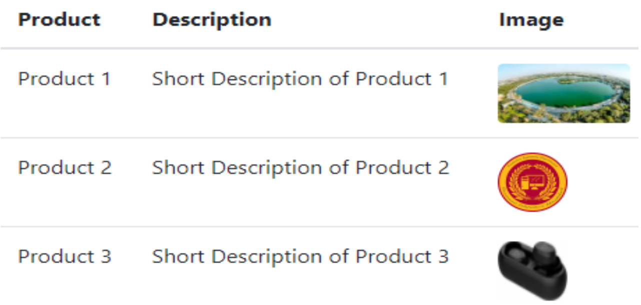
```
<div class="table-responsive">
<table class="table">
<thead>
<tr><th>Product</th>
<th>Description</th>
<th>Image</th></tr>
</thead>
<tbody>
<tr><td>Product 1</td>
<td>Short Description of Product 1</td>
<td><img src="1.jpg" class="img-fluid rounded" alt="Product 1 Image"
style="height:50px; width 50px;"></td></tr>
<tr><td>Product 2</td>
<td>Short Description of Product 2</td>
<td><img src="2.jpg" class="img-fluid rounded" alt="Product 2 Image"
style="height:50px; width: 50px;"></td></tr>
<tr><td>Product 3</td>
<td>Short Description of Product 3</td>
<td><img src="3.jpg" class="img-fluid rounded" alt="Product 3 Image"
style="height:50px; width: 50px;"></td></tr>
</tbody>
</table>
</div>
```

Output 1:



Product	Description	Image
Product 1	Short Description of Product 1	
Product 2	Short Description of Product 2	
Product 3	Short Description of Product 3	

Output 2:



Product	Description	Image
Product 1	Short Description of Product 1	
Product 2	Short Description of Product 2	
Product 3	Short Description of Product 3	

In this example, the table-responsive class is applied to a container div surrounding the table. This class enables horizontal scrolling on small screens when the table overflows horizontally.

The table class is applied to the <table> element itself, providing the base styling for the table.

Within the table rows (<tr>), the table data cells (<td>) contain the product information. The images are added within the table data cells using the <img> tag.

The img-fluid class is applied to the <img> tags, ensuring that the images are responsive and adapt to different screen sizes by automatically scaling with the parent container.

Remember to replace "1.jpg", "2.jpg", and "3.jpg" with the actual paths to your image files.

With the combination of the table-responsive class and the img-fluid class, the table and images will adjust dynamically to fit within the available space on different devices, making the table and images more accessible and user-friendly on various screen sizes.

H. Program Code & Output Source code:

SOURCE CODE

OUTPUT

Signature

Practical No.6: Use bootstrap typography to create responsive web page on given blog topic.

A. Objective:

- Styling and formatting content of web page using Bootstrap typography classes.
- To provide consistent and aesthetically pleasing typographic styles across different devices and screen sizes.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the *engineering* problems.
2. **Problem analysis**: Identify and analyse well-defined *engineering* problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
5. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes *in field of engineering.*

C. Expected Skills to be developed based on competency:

to create consistent and responsive typography in web design as per need of layout of website or web application.

D. Expected Course Outcomes(Cos)

Build different webpages layouts adhering to all platforms and sizes.

E. Practical Outcome(PRo)

To create any specific blog of web site using bootstrap framework classes.

F. Expected Affective domain Outcome(ADos)

1. Follow Coding standards and practices.
2. Maintain tools and equipment.
3. Follow safety practices.
4. Follow ethical practices

G. Prerequisite Theory:

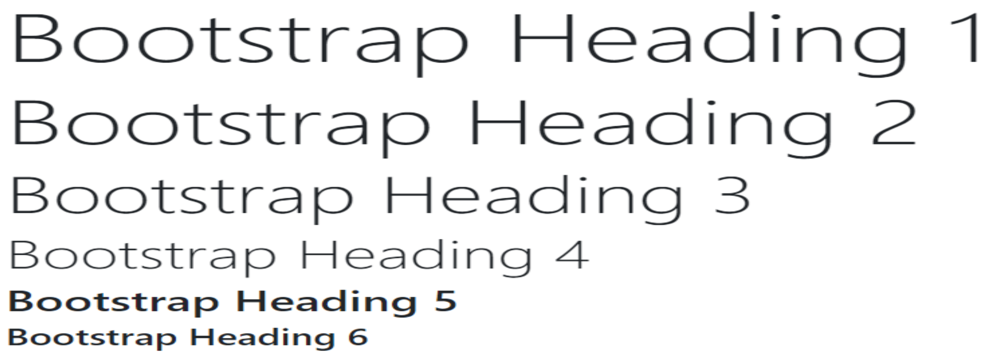
Typography: Bootstrap typography refers to the predefined styles and classes provided by the Bootstrap framework for creating visually appealing and consistent text layouts in web applications. Bootstrap is a popular front-end framework that offers a wide range of CSS and JavaScript components to streamline the development process.

The typography features of Bootstrap focus on ensuring readability, hierarchy, and aesthetic appeal in text elements. By utilizing Bootstrap's typography classes, developers can easily create well-designed and responsive typography without having to write custom CSS from scratch.Here are some key aspects of Bootstrap typography:

1. **Bootstrap headings:**

Bootstrap headings are a set of styles and classes provided by the Bootstrap framework that allow you to create visually appealing and properly structured headings for your web pages. They provide consistent typography and hierarchy to improve the readability and organization of your content.Here's an example of how you can use Bootstrap headings:

```
<html>
<head>
<!-- Include Bootstrap CSS -->
<link rel="stylesheet" href="bootstrap.min.css">
</head>
<body>
<h1 class="display-1">Bootstrap Heading 1</h1>
<h2 class="display-2">Bootstrap Heading 2</h2>
<h3 class="display-3">Bootstrap Heading 3</h3>
<h4 class="display-4">Bootstrap Heading 4</h4>
<h5 class="h1">Bootstrap Heading 5</h5>
<h6 class="h2">Bootstrap Heading 6</h6>
</body>
</html>
```



Bootstrap Heading 1

Bootstrap Heading 2

Bootstrap Heading 3

Bootstrap Heading 4

Bootstrap Heading 5

Bootstrap Heading 6

In the above example, we include the Bootstrap CSS file using a <link> tag in the <head> section of our HTML document. This allows us to access the Bootstrap styles and classes.

Then, we use various Bootstrap classes to create headings with different sizes and styles. Here's a breakdown of the classes used:display-1, display-2, display-3, display-4: These classes represent the larger headings, with display-1 being the largest and display-4 being the smallest.

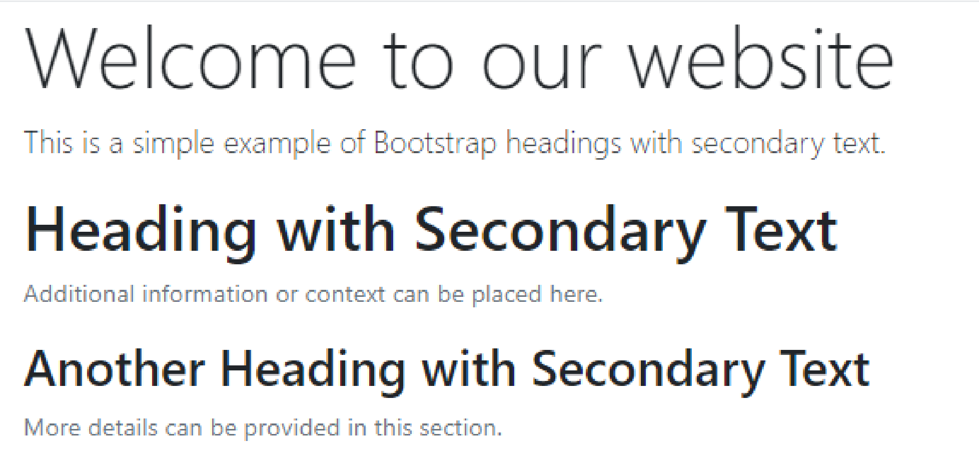
h1, h2: These classes are used to create headings that have the same size as the corresponding HTML <h1> and <h2> tags, respectively.

By applying these classes to your <h1> to <h6> tags, you can easily style and structure your headings according to the Bootstrap guidelines.

2. Bootstrap headings

Bootstrap headings with secondary text are a feature provided by the Bootstrap framework that allows you to add a small block of additional text below the main heading. This secondary text can be used to provide further context or additional information related to the heading. Here's an example of how Bootstrap headings with secondary text can be used:

```
<html>
<head>
<title>Bootstrap Headings with Secondary Text Example</title>
<!-- Include Bootstrap CSS -->
<link rel="stylesheet" href="bootstrap.min.css">
</head>
<body>
<div class="container">
<h1 class="display-4">Welcome to our website</h1>
<p class="lead">This is a simple example of Bootstrap headings with secondary text.</p>
<h2 class="h1">Heading with Secondary Text</h2>
<p class="text-muted">Additional information or context can be placed here.</p>
<h3 class="h2">Another Heading with Secondary Text</h3>
<p class="text-muted">More details can be provided in this section.</p>
</div>
</body>
</html>
```



Welcome to our website

This is a simple example of Bootstrap headings with secondary text.

Heading with Secondary Text

Additional information or context can be placed here.

Another Heading with Secondary Text

More details can be provided in this section.

In the example above, the Bootstrap classes display-4, lead, h1, h2, and h3 are used to style the headings with different sizes and styles. The text-muted class is applied to the <p> tags to give the secondary text a muted or faded appearance. You can customize these styles according to your needs.

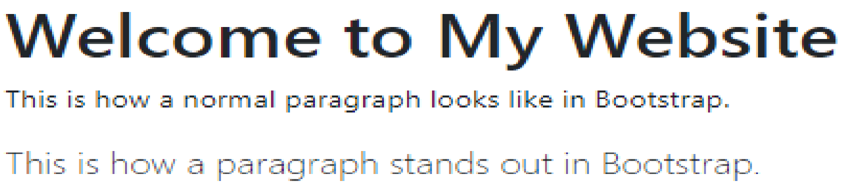
By using Bootstrap headings with secondary text, you can enhance the visual hierarchy of your page and provide additional information or context to your users.

3. **Bootstrap Normal And Lead Paragraph**

The lead class is a typography class provided by Bootstrap. It is used to emphasize or highlight a particular paragraph or section of text, typically used for introductory or summary content. The lead class adds some extra spacing, font size, and font weight to make the text stand out.

Here's an example of how the "lead" class can be used in HTML:

```
<html>
<head>
<link rel="stylesheet" href="bootstrap.min.css">
</head>
<body>
<div class="container">
<h1>Welcome to My Website</h1>
<p>This is how a normal paragraph looks like in Bootstrap.</p>
<p class="lead">This is how a paragraph stands out in Bootstrap.</p>
</div>
</body>
</html>
```



Welcome to My Website

This is how a normal paragraph looks like in Bootstrap.

This is how a paragraph stands out in Bootstrap.

One example of a Bootstrap class is the "lead" class. The "lead" class is used to apply a larger font size and increased line height to a paragraph of text, making it stand out and draw attention. It is often used to highlight important introductory or summary information on a webpage.

In the above example, the Bootstrap CSS file is included using the <link> tag to access the Bootstrap framework. The "container" class is used to create a container to hold the content. The <h1> element is used for the heading, and the <p> element with the "lead" class is used for the lead paragraph. As a result, the lead paragraph will be rendered with a larger font size and increased line height, making it more prominent compared to the regular paragraphs.

By utilizing Bootstrap classes like "lead," developers can quickly enhance the appearance and styling of their web pages, achieving a consistent and professional design without having to write custom CSS from scratch.

4. <u>Bootstrap Text alignment</u>

Text alignment in Bootstrap can be achieved using the "text" class along with alignment modifiers.

Here are some examples of Bootstrap text alignment classes:

- Left Alignment:
  To align text to the left, you can use the class "text-left". Here's an example:
  <p class="text-left">This text is aligned to the left.</p>
- Center Alignment:
  To center-align text, you can use the class "text-center". Here's an example:
  <p class="text-center">This text is centered.</p>
- Right Alignment:
  To align text to the right, you can use the class "text-right". Here's an example:
  <p class="text-right">This text is aligned to the right.</p>
- Justify Alignment:
  To justify text, you can use the class "text-justify". Here's an example:
  <p class="text-justify">This text is justified, meaning it is aligned on both the left and right sides.</p>

You can apply these text alignment classes to various HTML elements like paragraphs (p), headings (h1, h2, etc.), divs, or any other element that contains text content. By using these classes, you can easily control the alignment of your text within the Bootstrap framework.

5. Bootstrap text formatting

You can apply text formatting using predefined CSS classes. Bootstrap provides several classes that can be used to style text elements in a consistent and flexible mannerthat allow you to modify the appearance of text.These classes can be used to emphasize or highlight specific parts of your content.

Here are some examples of Bootstrap text formatting classes:

1. **text-primary**: This class sets the text color to a primary color defined by Bootstrap. It is useful for highlighting important information.

   Example: **<h1 class="text-primary">Welcome to our website!</h1>**

2. **text-secondary:** This class sets the text color to the secondary color defined by Bootstrap. It is used to display secondary information.

   Example:**<p class="text-secondary">This is some additional information.</p>**

3. **text-success**: This class sets the text color to indicate success or positive outcomes. It is commonly used for displaying success messages.

   Example: **<p class="text-success">Your transaction was successful.</p>**

4. **text-danger**: This class sets the text color to indicate danger or negative outcomes. It is often used for displaying error messages or warnings.

   Example: **<p class="text-danger">Error: Invalid input.</p>**

5. **text-info**: This class sets the text color to indicate informational or neutral content. It can be used for displaying informative messages.

   Example: **<p class="text-info">Please read the terms and conditions carefully.</p>**

6. **text-warning**: This class sets the text color to indicate warnings or cautionary information. It is useful for drawing attention to important details.

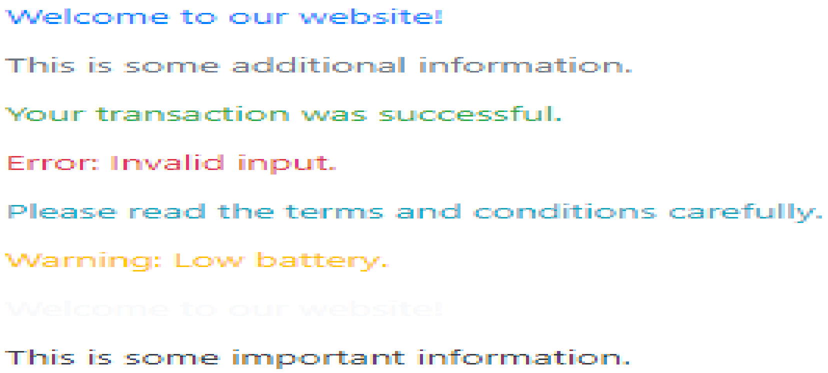
   Example: **<p class="text-warning">Warning: Low battery.</p>**

7. **.text-light**: This class sets the text color to a light shade. It is used when you want to display text on a dark background. Example:

   <p class="text-light">Welcome to our website!</p>

8. **.text-dark**: This class sets the text color to a dark shade. It is used when you want to display text on a light background. Example:

   <p class="text-dark">This is some important information.</p>



Welcome to our website!

This is some additional information.

Your transaction was successful.

Error: Invalid input.

Please read the terms and conditions carefully.

Warning: Low battery.

Welcome to our website!

This is some important information.

These are just a few examples of the text formatting classes provided by Bootstrap. By using these classes, you can easily apply consistent and visually appealing styles to your text elements without writing custom CSS.

6. Bootstrap text transformation :

The text transformation classes in Bootstrap are used to apply different styles to text, such as converting it to uppercase, lowercase, or capitalizing the first letter of each word. These classes are part of Bootstrap's utility classes, which can be easily applied to HTML elements.

Here are the three text transformation classes provided by Bootstrap:

text-lowercase: This class converts all the letters in the text to lowercase. For example, if you have the following HTML:

<p class="text-lowercase">HELLO, WORLD!</p>

text-uppercase: This class converts all the letters in the text to uppercase. For example:

<p class="text-uppercase">hello, world!</p>

text-capitalize: This class capitalizes the first letter of each word in the text. For example:

<p class="text-capitalize">hello, world!</p>

These text transformation classes can be used on various HTML elements, such as paragraphs (<p>), headings (<h1>, <h2>, etc.), and spans (<span>). By applying these classes, you can easily change the appearance of text in your web page without writing custom CSS styles.

7. Bootstrap blockquotes formatting :

Bootstrap provides several classes for formatting blockquotes, which are used to highlight and style quotations or excerpts of text. Here are some commonly used Bootstrap blockquote formatting classes along with examples:

.blockquote: This class is used to style the entire blockquote section.

```
<blockquote class="blockquote">
<p>The world is a dangerous place to live; not because of the people who are evil, but because of the people who don't do anything about it.</p>
<footer class="blockquote-footer">Albert Einstein</footer>
</blockquote>
```

.blockquote-footer: This class is used to style the footer of the blockquote, typically used to display the source or author of the quote.

```
<blockquote class="blockquote">
<p>The world is a dangerous place to live; not because of the people who are evil, but because of the people who don't do anything about it.</p>
<footer class="blockquote-footer">Albert Einstein</footer>
</blockquote>
```

.**blockquote-reverse:** This class is used to reverse the alignment of the blockquote, making the footer appear above the quote.

```
<blockquote class="blockquote blockquote-reverse">
<p>The world is a dangerous place to live; not because of the people who are evil, but because of the people who don't do anything about it.</p>
<footer class="blockquote-footer">Albert Einstein</footer>
</blockquote>
```

.blockquote-primary, .blockquote-secondary, .blockquote-success, .blockquote-info, .blockquote-warning, .blockquote-danger: These classes are used to apply different color variations to the blockquote.

```
<blockquote class="blockquote blockquote-primary">
<p>The world is a dangerous place to live; not because of the people who are evil, but because of the people who don't do anything about it.</p>
<footer class="blockquote-footer">Albert Einstein</footer>
</blockquote>
```

These are just a few examples of Bootstrap blockquote formatting classes. By combining these classes or customizing them with your own CSS, you can create various styles for blockquotes that suit your design needs.

8. Bootstrap truncating long text :

One useful feature of Bootstrap is its ability to truncate long text, which means cutting off the text and adding an ellipsis (...) to indicate that the content has been shortened.

To truncate long text in Bootstrap, you can use the CSS class "text-truncate" on the HTML element that contains the text you want to truncate. This class applies the necessary styles to limit the text's width and add the ellipsis when it overflows its container.

Here's an example of how you can use the "text-truncate" class in Bootstrap:

```
<div class="container">
<p class="text-truncate">
The world is a dangerous place to live; not because of the people who are evil, but because of the people who don't do anything about it.
</p>
</div>
```

In the example above, the <p> element contains a long text that may exceed the container's width. By applying the "text-truncate" class to the <p> element, Bootstrap will automatically truncate the text and add the ellipsis if necessary.

The text is cut off after a certain number of characters, and the ellipsis indicates that the content has been truncated. This approach is particularly useful when you have limited space and want to prevent long text from breaking the layout or overflowing its container.

Note that the "text-truncate" class works best on block-level elements like <p>, <div>, or <span>. It may not have the desired effect on inline elements such as <span> within a paragraph. In such cases, you may need to wrap the inline element with a block-level element and apply the class to the outer container.

9. Bootstrap handling Text Overflow:

Two classes commonly used for text wrapping and handling overflow in Bootstrap are "text-wrap" and "text-overflow".

Text Wrap (class: text-wrap):

The "text-wrap" class in Bootstrap is used to enable word wrapping for long lines of text within a container. By default, text in Bootstrap will not wrap and will extend beyond the container's boundaries, potentially causing layout issues. The "text-wrap" class ensures that text automatically wraps within the available space.

Example:

```
<div class="container">
<p class="text-wrap">
```

Computer engineering is a discipline within the field of engineering that focuses on the design, development, and implementation of computer systems and hardware. It combines principles of electrical engineering and computer science to create and improve computer technology.
</p>
</div>

In this example, the "text-wrap" class is applied to the <p> element. This ensures that the long paragraph of text will wrap within the container, preventing it from overflowing and breaking the layout.

Text Overflow (class: text-overflow):

The "text-overflow" class in Bootstrap is used to handle overflow when text content exceeds the width of its container. It provides options to truncate the overflowing text and display an ellipsis (...) or to make it scrollable within the container.

Example 1: Text Truncation with Ellipsis

<div class="container">
<p class="text-overflow">
Computer engineering is a discipline within the field of engineering that focuses on the design, development, and implementation of computer systems and hardware. </p>
</div>

In this example, the "text-overflow" class is applied to the <p> element. If the text overflows the container's width, it will be truncated with an ellipsis (...) at the end.

Example 2: Text Scrollable within Container

<div class="container">
<p class="text-overflow" style="overflow: auto;">
Computer engineering is a discipline within the field of engineering that focuses on the design, development, and implementation of computer systems and hardware. </p>
</div>

In this example, the "text-overflow" class is applied to the <p> element, and the inline style "overflow: auto;" is added. This makes the text content scrollable horizontally within the container when it exceeds the available width.

Note: The "text-overflow" class itself does not enable scrolling; it is achieved by using CSS overflow properties.

H. Program Code & Output Source code:

SOURCE CODE

OUTPUT

Signature

Practical No.7: Design responsive web page for student registration form using bootstrap form layout, form control, bootstrap buttons.

A. Objective:

- Utilize various form layout classes, form controls classes and button classes to make responsive web page.
- Create well-designed and mobile-friendly forms.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the *engineering* problems.
2. **Problem analysis**: Identify and analyse well-defined *engineering* problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
5. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes *in field of engineering.*

C. Expected Skills to be developed based on competency:

Create flexible and visually appealing forms that adapt to different screen sizes and provide a seamless user experience by using various form classes..

D. Expected Course Outcomes(Cos)

Build different webpages layouts adhering to all platforms and sizes.

E. Practical Outcome(PRo)

Design responsive web page using various form layout classes, form controls classes and button classes.

F. Expected Affective domain Outcome(ADos)

1. Follow Coding standards and practices.
2. Maintain tools and equipment.
3. Follow safety practices.
4. Follow ethical practices

G. Prerequisite Theory:

Bootstrap Form control: The Bootstrap framework provides a set of form control classes that can be used to style and enhance HTML form elements. These classes make it easy to create consistent and visually appealing forms in web applications. Here are some commonly used form control classes in Bootstrap, along with examples:

1. **.form-control:** This class is applied to input elements, such as text fields, select dropdowns, and text areas, to create a standardized look and feel.

   Example:

   ```
   <input type="text" class="form-control" placeholder="Enter your name">
   ```

2. **.form-check:** This class is used to create custom-styled checkboxes and radio buttons.

   Example:

   ```
   <div class="form-check">
   <input class="form-check-input" type="checkbox" value=""
   id="exampleCheckbox">
   <label class="form-check-label" for="exampleCheckbox">
      Checkbox label
   </label>
   </div>
   ```

3. .**form-select**: This class is applied to <select> elements to style them as custom dropdowns.

   Example:

   ```
   <select class="form-select">
   <option selected>Select an option</option>
   <option value="1">Option 1</option>
   <option value="2">Option 2</option>
   <option value="3">Option 3</option>
   </select>
   ```

4. .**form-range**: This class is used to style range input sliders.

   Example:

   ```
   <input type="range" class="form-range" min="0" max="100">
   ```

5. .**form-floating**: This class is applied to create floating label inputs where the label moves above the input when it is focused or has a value.

   Example:

   ```
   <div class="form-floating">
   <input type="email" class="form-control" id="floatingInput"
   placeholder="name@example.com">
   <label for="floatingInput">Email address</label>
   ```

```
</div>
```

6. .**form-check-input**: This class is used to horizontally align multiple checkboxes or radio buttons.

   Example:

```
<div class="form-check-inline">
<input class="form-check-input" type="checkbox" value=""
id="inlineCheckbox1">
<label class="form-check-label" for="inlineCheckbox1">Option
1</label></div>
<div class="form-check-inline">
<input class="form-check-input" type="checkbox" value=""
id="inlineCheckbox2">
<label class="form-check-label" for="inlineCheckbox2">Option 2</label>
</div>
```

7. .**form-group**: This class is used to group form elements together for better organization and styling. It provides a container for form controls such as input fields, checkboxes, radio buttons, and select dropdowns. The "form-group" class helps in aligning and spacing form controls properly.

   Example:

```
<div class="form-group">
<label for="country">Country</label>
<select class="form-control" id="country">
<option>USA</option>
<option>Canada</option>
<option>UK</option>
</select>
</div>
```

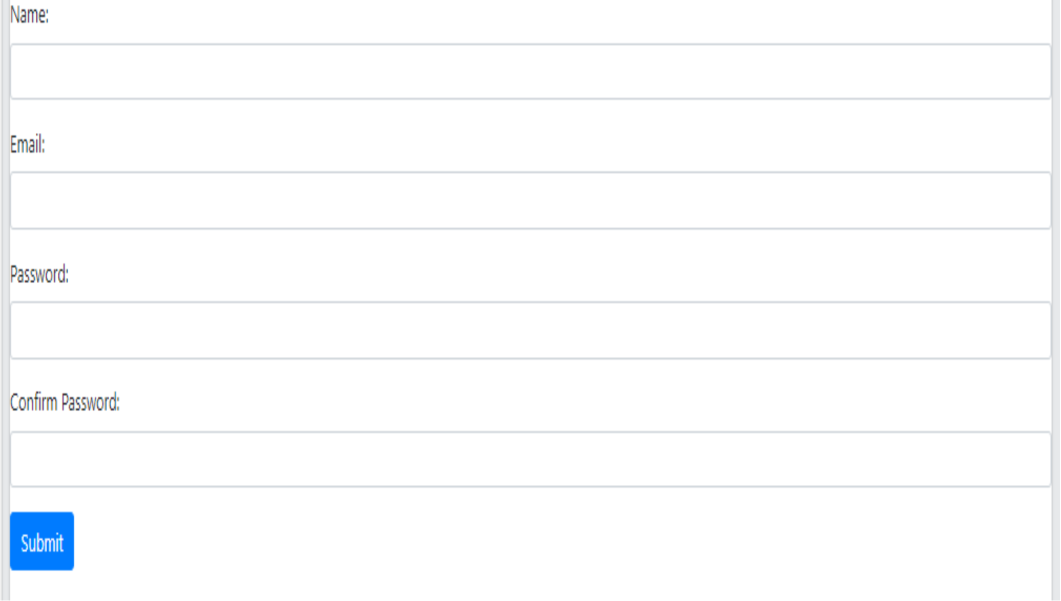
By utilizing these classes, you can easily style and customize form elements to fit your design requirements while maintaining consistency throughout your web application.

Bootstrap Form Layout : The Bootstrap framework provides several form layout options that developers can use to create well-designed and responsive forms on their websites. Here are some of the form layout options available in Bootstrap:

Vertical Form Layout: The Vertically Form Layout in Bootstrap is a way to arrange form elements vertically, stacking them on top of each other. This layout is commonly used when designing forms to ensure a clean and organized appearance. In this layout, form labels are placed above the corresponding input fields or form controls.

To create a Vertically Form Layout in Bootstrap, you can use the .form-group class to wrap each form group, which consists of a label and its associated form control. Here's an example:

```
<form>
<div class="form-group">
<label for="name">Name:</label>
<input type="text" class="form-control" id="name"></div>
<div class="form-group">
<label for="email">Email:</label>
<input type="email" class="form-control" id="email"></div>
<div class="form-group">
<label for="password">Password:</label>
<input type="password" class="form-control" id="password"></div>
<div class="form-group">
<label for="confirm-password">Confirm Password:</label>
<input type="password" class="form-control" id="confirm-password"></div>
<button type="submit" class="btn btn-primary">Submit</button>
</form>
```

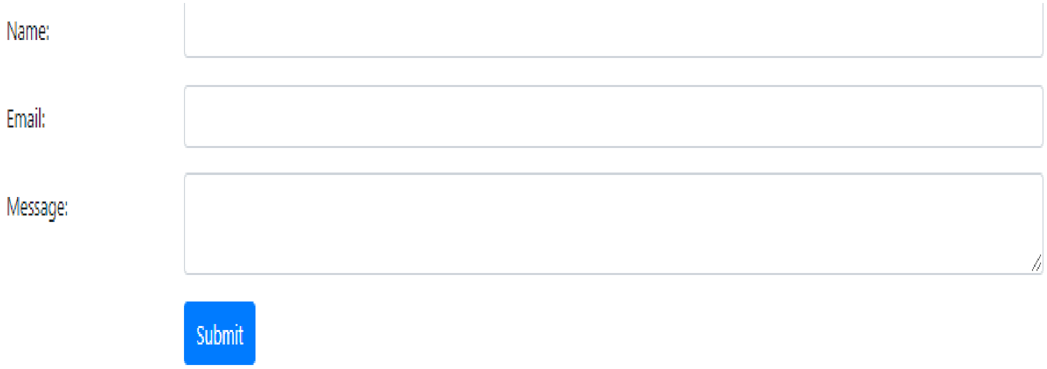


In the above example, each form field is wrapped inside a <div> element with the .form-group class. The <label> tags are used to describe the purpose of each input field, and the <input> tags represent the form controls. The .form-control class is applied to each input field to style it as a Bootstrap form control.By default, Bootstrap's CSS will stack the form groups vertically, with the labels positioned above the input fields. This layout provides a clean and organized appearance for your forms, making it easier for users to fill them out.

Horizontal Form layout

In the Bootstrap framework, the horizontal form layout is a way to structure form elements in a horizontal manner, where labels are placed alongside their corresponding input fields. This layout is useful when you have multiple form fields and want to display them in a more compact and aligned manner. Here's an example of how to implement a horizontal form layout in Bootstrap:

```
<form>
<div class="form-group row">
<label for="name" class="col-sm-2 col-form-label">Name:</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="name"></div></div>
<div class="form-group row">
<label for="email" class="col-sm-2 col-form-label">Email:</label>
<div class="col-sm-10">
<input type="email" class="form-control" id="email"></div></div>
<div class="form-group row">
<label for="message" class="col-sm-2 col-form-label">Message:</label>
<div class="col-sm-10">
<textarea class="form-control" id="message"></textarea></div></div>
<div class="form-group row">
<div class="col-sm-10 offset-sm-2">
<button type="submit" class="btn btn-primary">Submit</button></div></div>
</form>
```



In the example above, we have a **<form>** element containing multiple form fields structured using the horizontal form layout. Here's a breakdown of the important Bootstrap classes used:

- The outer **<div>** elements with the class **form-group row** define each form field group and ensure proper alignment.
- The **<label>** elements use the class **col-form-label** to style the labels.

- The **<input>**, **<textarea>**, or other form elements are wrapped inside a **<div>** with the class **col-sm-10** to create the input field container. The **col-sm-10** class ensures that the input fields occupy the remaining horizontal space in the form.
- The **<label>** and **<input>** (or other form elements) are placed inside a **<div>** with the class **col-sm-2** to allocate a fixed width for the labels.
- The **offset-sm-2** class on the last **<div>** element offsets the submit button by two columns to align it with the form fields.

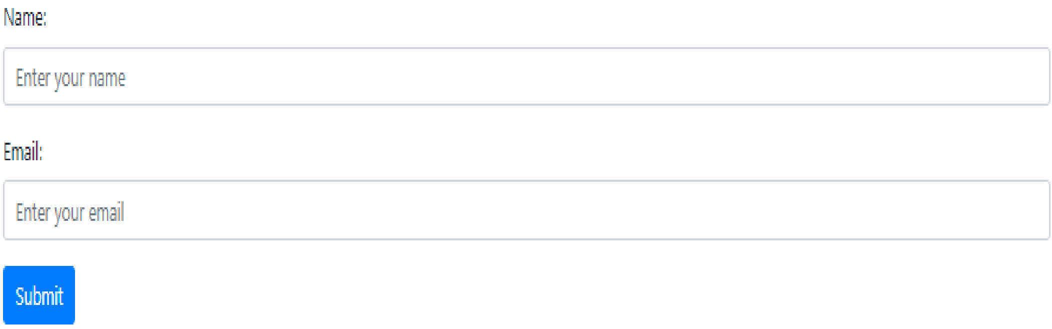
By using these Bootstrap classes, the form fields and labels will be displayed horizontally, making the form more compact and visually aligned.

Inline Form Layout

The inline form layout is a feature of the Bootstrap framework that allows you to create forms where the form controls and labels are placed horizontally in a single line. This layout is useful when you have a limited amount of form controls and want to display them in a compact and aligned manner.

To create an inline form layout in Bootstrap, you can use the .form-inline class along with the appropriate form control and label classes. Here's an example:

```
<form class="form-inline">
<div class="form-group">
<label for="name">Name:</label>
<input type="text" class="form-control" id="name" placeholder="Enter your name"></div>
<div class="form-group">
<label for="email">Email:</label>
<input type="email" class="form-control" id="email" placeholder="Enter your email"></div>
<button type="submit" class="btn btn-primary">Submit</button>
</form>
```



In the above example, the form element has the form-inline class to indicate that it should be rendered as an inline form. The form controls and labels are wrapped in div elements with the form-group class to provide spacing and alignment.

The label elements are associated with their corresponding form controls using the for attribute. The form controls themselves use the form-control class to style them consistently.The final button is a regular Bootstrap button styled with the btn and btn-primary classes.

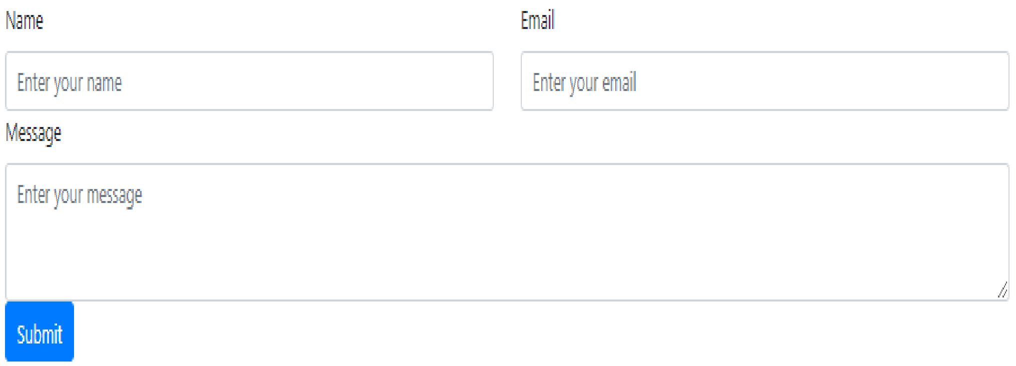
When you use this code, the form controls and labels will appear side by side in a single line, and the button will be aligned to the right. You can further customize the layout and styling using additional Bootstrap classes and CSS.

Column Sizing Form Layout

The column sizing form layout in the Bootstrap framework allows developers to create responsive and visually appealing forms by utilizing the grid system. This layout provides a flexible way to arrange form elements into multiple columns, adjusting their sizes based on different screen sizes.

To create a column sizing form layout in Bootstrap, you need to use the grid classes (col-*) along with form components such as labels, inputs, and buttons. Here's an example:

```
<form>
<div class="row">
<div class="col-md-6">
<label for="inputName">Name</label>
<input type="text" class="form-control" id="inputName"
placeholder="Enter your name"></div>
<div class="col-md-6">
<label for="inputEmail">Email</label>
<input type="email" class="form-control" id="inputEmail"
placeholder="Enter your email"></div></div>
<div class="row">
<div class="col-md-12">
<label for="inputMessage">Message</label>
<textarea class="form-control" id="inputMessage" placeholder="Enter
your message" rows="3"></textarea></div></div>
<div class="row">
<div class="col-md-6">
<button type="submit" class="btn btn-
primary">Submit</button></div></div>
</form>
```



In this example, the form is divided into rows (<div class="row">) to contain the form elements. The col-md-6 class is used to create two equal-width columns for the "Name" and "Email" fields on medium-sized screens and larger. On smaller screens, the columns will stack vertically.

The "Message" field occupies the full width of the row using col-md-12, ensuring it stretches across the entire row on medium-sized screens and larger.

Finally, the "Submit" button is placed in a single column using col-md-6, which creates a half-width column on medium-sized screens and larger.

Responsive Form Layout: Bootstrap provides a responsive form layout system that allows developers to easily create well-designed and mobile-friendly forms. The form layout in Bootstrap is based on a grid system, making it flexible and adaptable to different screen sizes.

To create a responsive form layout in Bootstrap, you typically structure your form using the following components:

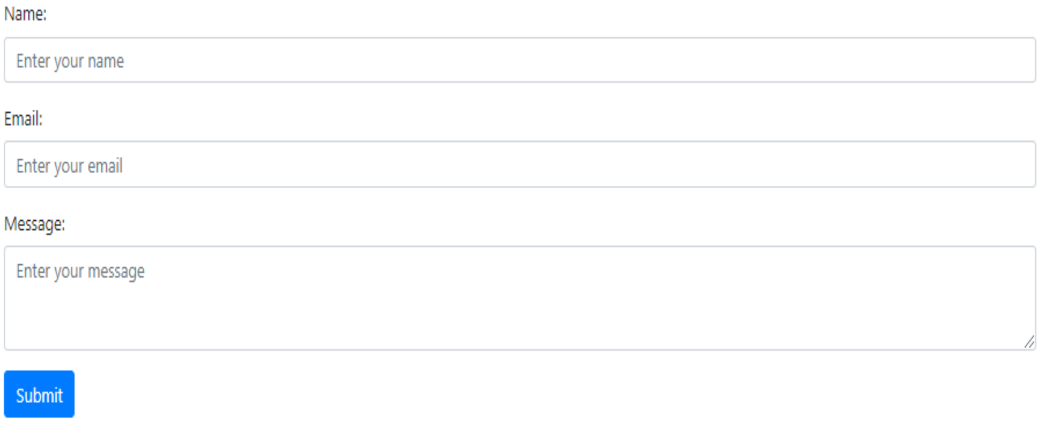
Form Container: Wrap your form elements inside a <form> tag to create a container for your form.

- Grid System: Use Bootstrap's grid classes (container, row, and col-*) to create a responsive grid structure for your form layout. The grid system allows you to divide the form into multiple columns and adjust their widths based on the screen size.
- Form Groups: Wrap related form elements, such as labels and input fields, inside a <div> element with the class .form-group. This class adds styling and spacing to group the elements together.
- Labels: Use the <label> element to provide a text description or name for each input field. Associate labels with their corresponding input fields using the for attribute and the id attribute of the input field.
- Input Fields: Use various input field types, such as <input>, <select>, or <textarea>, to collect user input. Apply Bootstrap's form control classes (e.g., .form-control) to style and standardize the appearance of the input fields.

- Form Validation: Bootstrap also provides built-in classes for form validation states. You can add the classes .is-valid and .is-invalid to input fields or form groups to indicate valid or invalid input, respectively.

Here's an example of a responsive form layout using Bootstrap:

```
<form>
<div class="form-group">
<label for="nameInput">Name:</label>
<input type="text" class="form-control" id="nameInput"
placeholder="Enter your name"></div>
<div class="form-group">
<label for="emailInput">Email:</label>
<input type="email" class="form-control" id="emailInput"
placeholder="Enter your email"></div>
<div class="form-group">
<label for="messageInput">Message:</label>
<textarea class="form-control" id="messageInput" rows="3"
placeholder="Enter your message"></textarea></div>
<button type="submit" class="btn btn-primary">Submit</button>
</form>
```

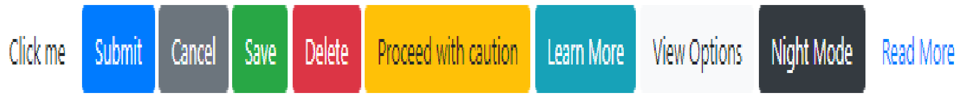


In the example above, the form elements are wrapped inside a <form> tag. Each form element is enclosed within a <div> element with the .form-group class to group them together. The labels are associated with their respective input fields using the for and id attributes.

By utilizing the grid system and form control classes provided by Bootstrap, this form layout will automatically adjust its appearance and responsiveness based on the screen size, providing an optimal user experience across devices.

Bootstrap Button classes: Bootstrap is a popular front-end framework that provides a wide range of pre-styled components to build responsive and modern websites. One of its notable features is the set of button classes that allow you to create visually appealing buttons with ease. Here are some common Bootstrap button classes along with examples:

1. **.btn**: This is the base class for creating a button. It provides a default button style. Example: **<button class="btn">Click me</button>**
2. **.btn-primary**: This class gives the button a primary color, indicating its importance or prominence.
   Example: **<button class="btn btn-primary">Submit</button>**
3. **.btn-secondary**: This class is used for creating secondary buttons, often used for less important actions.
   Example: **<button class="btn btn-secondary">Cancel</button>**
4. **.btn-success**: This class represents a success state for the button, typically used for actions that indicate successful completion.
   Example: **<button class="btn btn-success">Save</button>**
5. **.btn-danger**: This class signifies a dangerous or critical action, such as deleting data.
   Example: **<button class="btn btn-danger">Delete</button>**
6. **.btn-warning**: This class indicates a warning or cautionary action.
   Example: **<button class="btn btn-warning">Proceed with caution</button>**
7. **.btn-info**: This class is used for informational actions.
   Example: **<button class="btn btn-info">Learn More</button>**
8. **.btn-light**: This class creates a light-colored button, suitable for subtle or low-contrast actions.
   Example: **<button class="btn btn-light">View Options</button>**
9. **.btn-dark**: This class generates a dark-colored button, often used for actions that need more visual emphasis.
   Example: **<button class="btn btn-dark">Night Mode</button>**
10. **.btn-link**: This class creates a button that looks like a link, with no background or border.
    Example: **<button class="btn btn-link">Read More</button>**



You can further customize these buttons using additional classes and CSS to achieve the desired visual effects.

H. Program Code & Output Source code:

SOURCE CODE

OUTPUT

Signature

Practical No.8: Design responsive web page that shows odd (sem1, sem3, sem5) and even (sem2, sem4, sem6) semester consider as menu, courses of each semester as submenu using button groups and button toolbar component.

A. Objective:

Learn utilization of various bootstrap button and drop down classes for design responsive either nested or multilevel submenu.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the *engineering* problems.
2. **Problem analysis**: Identify and analyse well-defined *engineering* problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
5. **Engineering practices for society, sustainability and environment**: Apply appropriate technology in context of society, sustainability, environment and ethical practices.
6. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes *in field of engineering.*

C. Expected Skills to be developed based on competency:

Design a consistent and user-friendly navigation system with customization of navigation up to multilevel submenu.

D. Expected Course Outcomes(Cos)

Apply reusable bootstrap components to design effective user-friendly web pages.

E. Practical Outcome(PRo)

Design responsive multi-level or nested submenu using various drop down and button classes for branch specific odd-even semester with subjects .

F. Expected Affective domain Outcome(ADos)

1. Follow Coding standards and practices.
2. Maintain tools and equipment.
3. Follow safety practices.
4. Follow ethical practices

G. Prerequisite Theory:

In Bootstrap, the navbar is a responsive navigation component that provides a horizontal menu bar that collapses into a mobile-friendly "hamburger" menu on smaller screens. It's a fundamental part of creating a navigation menu in Bootstrap-based websites. Let's dive into the explanation of each term with an example:

Navbar class:

The navbar is a top-level container that holds the entire navigation menu. It usually contains the site's logo or brand name on the left and the navigation links on the right. The navbar can be customized with various CSS classes to control its appearance and behavior.

Navbar Toggle:

The navbar toggle is the button that appears on smaller screens to toggle the collapse of the navbar and show the navigation links as a dropdown menu. It is represented by the "hamburger" icon (three horizontal lines) by default. The navbar toggle is used in conjunction with the Bootstrap JavaScript component to enable responsive behavior. In the example above, the navbar-toggler class is added to the button element to create the toggle functionality.

Navbar-item:

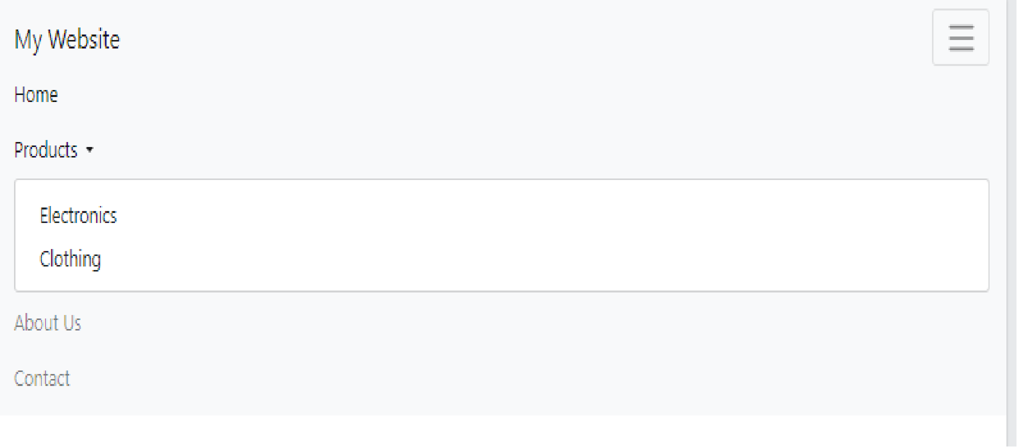
In the example provided earlier, navbar-item is not a specific Bootstrap class. Instead, it's a custom class defined by the user to style the individual navigation items. In Bootstrap, the individual navigation items are represented by the nav-item class. You can add your custom CSS classes to further style or customize the appearance of the navigation items according to your design preferences.

Nav-item dropend:

The dropend class is used in combination with nav-item to create dropdown menus that open towards the right. This is useful when you want to display submenus in a horizontal navbar, and the dropdown items should appear on the right side of the parent item. It is an alternative to the default drop-down behavior, which opens towards the bottom. To create a dropdown that opens to the right, you'll use the dropend class as shown in the example below.

```
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Bootstrap Navbar Example</title>
```

```
<link rel="stylesheet" href="bootstrap.min.css">
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-light bg-light">
<a class="navbar-brand" href="#">My Website</a>
<button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#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 dropdown dropend">
<a class="nav-link dropdown-toggle" href="#" id="navbarDropdown1"
role="button" data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false">Products</a>
<ul class="dropdown-menu" aria-labelledby="navbarDropdown1">
<li><a class="dropdown-item" href="#">Electronics</a></li>
<li><a class="dropdown-item" href="#">Clothing</a></li></ul>
</li><li class="nav-item">
<a class="nav-link" href="#">About Us</a>
</li><li class="nav-item">
<a class="nav-link" href="#">Contact</a></li></ul>
</div></nav>
<script src="jquery-3.5.1.slim.min.js"></script>
<script src="bootstrap.bundle.min.js"></script>
<script src="bootstrap.min.js"></script>
</body>
</html>
```



In this example, we have created a simple Bootstrap Navbar with a toggle button (navbar-toggler) for mobile responsiveness. We've also used the navbar-nav class to define the unordered list (<ul>) that holds all the navigation items.

One of the navigation items has the dropend class, which indicates that the dropdown menu for the "Products" link should be aligned to the right side of the link. The dropdown-menu class is used to create the actual dropdown menu that appears when you click on "Products."

When you run this HTML code, you'll see a responsive navigation bar with a dropdown menu that expands to the right side of the "Products" link.

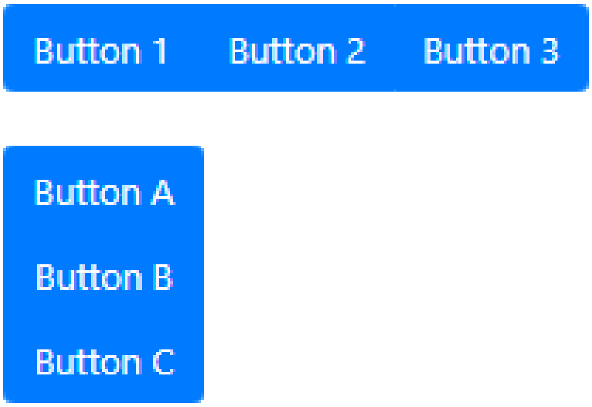
Button Group class :

Bootstrap provides a variety of pre-styled components, including buttons. Here are some of the commonly used button classes in Bootstrap:

1. **btn**: This is the base class for creating buttons. It is used to style a basic button with default appearance.
2. **btn-primary**: This class is used to create a prominent button with a primary color, indicating the main action on a page.
3. **btn-secondary**: This class creates a button with a secondary color, typically used for less important actions or as a subtle alternative to the primary button.
4. **btn-success**: Use this class to create a button with a color that represents a successful action, like saving a form or completing a task.
5. **btn-danger**: This class creates a button with a color that indicates a dangerous or critical action, such as deleting an item or canceling an important operation.
6. **btn-warning**: This class is used to create a button with a warning color, often used to signal potential issues or draw attention to a specific part of the page.
7. **btn-info**: Use this class to create a button with an informative color, suitable for providing additional information or details.
8. **btn-light**: This class creates a button with a light color, often used in cases where a less prominent, subtle appearance is desired.
9. **btn-dark**: This class creates a button with a dark color, often used for creating buttons with a higher contrast on lighter backgrounds.
10. **btn-link**: This class creates a button with a link-like appearance, styled to look like a hyperlink rather than a traditional button.
11. **btn-outline-primary**, **btn-outline-secondary**, etc.: These classes create buttons with a transparent background and colored borders, matching the respective primary, secondary, etc., color schemes.
12. **btn-lg**, **btn-sm**, **btn-block**: These classes are used to modify the size and layout of buttons. **btn-lg** creates a larger button, **btn-sm** creates a smaller button, and **btn-block** makes the button stretch the full width of its container.

Bootstrap allows you to group a series of buttons together (on a single line) in a button group. "btn-group" class in Bootstrap is used to group a set of buttons together, creating a cohesive button group. It allows you to style and align related buttons horizontally or vertically.Here's an example of how to use the "btn-group" class in Bootstrap:

```
<html>
<head>
<title>Button Group Example</title>
<link rel="stylesheet" href="bootstrap.min.css">
</head>
<body>
<div class="container mt-4">
<!-- Horizontal button group -->
<div class="btn-group">
<button type="button" class="btn btn-primary">Button 1</button>
<button type="button" class="btn btn-primary">Button 2</button>
<button type="button" class="btn btn-primary">Button 3</button>
</div><br><br>
<!-- Vertical button group -->
<div class="btn-group-vertical">
<button type="button" class="btn btn-primary">Button A</button>
<button type="button" class="btn btn-primary">Button B</button>
<button type="button" class="btn btn-primary">Button C</button></div>
</div>
<!-- Include Bootstrap JS -->
<script src="jquery.min.js"></script>
<script src="bootstrap.min.js"></script>
</body>
</html>
```



In this example, we have two button groups:

Horizontal button group: Three buttons ("Button 1," "Button 2," and "Button 3") are grouped together using the "btn-group" class. This creates a simple horizontal button group.

Vertical button group: Three buttons ("Button A," "Button B," and "Button C") are grouped together using the "btn-group-vertical" class. This creates a vertical button group.

By using the "btn-group" class, you can easily group buttons together and apply Bootstrap's button styles. The "btn-group-vertical" class allows you to create vertical button groups, which are useful for certain layouts and design requirements.

In Bootstrap, the "role" attribute of the button group class is used to define the purpose or function of the button group for accessibility purposes. The "role" attribute is an important part of making web content more accessible to users with disabilities, particularly those who use screen readers or other assistive technologies. It helps to convey the intended purpose of the button group to these users, making their browsing experience more inclusive.

The "role" attribute for the button group class can take one of the following values:

1. **"group":** This is the most common and appropriate value for the button group class. It indicates that the buttons within the group are related and form a cohesive unit.

   Example:

   ```
   <div class="btn-group" role="group" aria-label="Navigation Buttons">
   <!-- Buttons here -->
   </div>
   ```

2. **"toolbar":** Use this value when the button group represents a toolbar, such as a set of buttons for text formatting or other actions typically found in a toolbar.

   Example:

   ```
   <div class="btn-group" role="toolbar" aria-label="Text Formatting Toolbar">
   <!-- Buttons here -->
   </div>
   ```

3. **"presentation":** In some rare cases, you might need to use this value to indicate that the button group is being used for presentation purposes only and does not have any interactive functionality.

   Example:

   ```
   <div class="btn-group" role="presentation" aria-label="Decoration">
   <!-- Buttons used for decoration only -->
   </div>
   ```

It's important to select the appropriate value for the "role" attribute based on the intended purpose of the button group. By doing so, you help assistive technologies accurately convey the meaning and functionality of the button group to users with disabilities.The role attribute is not exclusive to Bootstrap but is a standard part of HTML, and it helps make web content more accessible to users with disabilities. It is used to describe the purpose or type of an element when the default role implied by the element's tag is not sufficient.

Here's an example of how you can use the role attribute with a Bootstrap button:

```
<!-- Standard Bootstrap button without the role attribute -->
<button class="btn btn-primary">Click me!</button>
<!-- Bootstrap button with the role attribute added -->
<button class="btn btn-primary" role="button">Click me!</button>
```

In the first example, the button does not have the role attribute explicitly set. However, since it is a <button> element, its default role is already "button," and it will be correctly interpreted as such by assistive technologies.

In the second example, the role attribute is added with the value "button." This is redundant for a standard <button> element but can be helpful in situations where you use a different element (like a <span> or <div>) styled to look like a button. By explicitly setting the role attribute to "button," you ensure that assistive technologies still recognize it as a button and provide the appropriate feedback to users with disabilities.

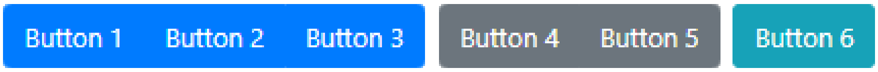
Remember that assistive technologies can infer the role of many elements automatically, so you should use the role attribute sparingly and only when necessary. In the case of a standard Bootstrap button, the role attribute is not required, as the element's default role is already "button."

<u>btn-toolbar class</u>

To create a Button Toolbar in Bootstrap, you can use the btn-toolbar class along with appropriate button classes to style the individual buttons.Here's an example of how to create a Button Toolbar using Bootstrap:

```
<div class="btn-toolbar" role="toolbar" aria-label="Button Toolbar Example">
<div class="btn-group mr-2" role="group" aria-label="First group">
<button type="button" class="btn btn-primary">Button 1</button>
<button type="button" class="btn btn-primary">Button 2</button>
<button type="button" class="btn btn-primary">Button 3</button>
</div>
<div class="btn-group mr-2" role="group" aria-label="Second group">
<button type="button" class="btn btn-secondary">Button 4</button>
<button type="button" class="btn btn-secondary">Button 5</button>
```

```
</div>
<div class="btn-group" role="group" aria-label="Third group">
<button type="button" class="btn btn-info">Button 6</button>
</div>
</div>
```



In this example, we use the btn-toolbar class to create the button toolbar container. Inside the container, we use btn-group classes to group related buttons together. Each btn-group element can contain one or more buttons. The mr-2 class is used to add some margin between button groups for spacing.

For accessibility, we include the role attribute with the value "toolbar" to indicate the container's role as a toolbar. The aria-label attribute is also added to provide a descriptive label for the toolbar, which is useful for assistive technologies.

Each individual button within the toolbar can have its own Bootstrap button classes, such as btn-primary, btn-secondary, or any other button variant you want to use.

H. Program Code & Output Source code:

SOURCE CODE

OUTPUT

Signature

Practical No.9: Use different bootstrap input groups components to create responsive webpage for job application or any other kind of application.

A. Objective:

Extending functionality of input element of form control to understand purpose of various bootstrap input group component classes and design a powerful and flexible user interface elementto enhance form controls by adding additional elements or buttons before, after, or on both sides of an input field.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the *engineering* problems.
2. **Problem analysis**: Identify and analyse well-defined *engineering* problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
5. **Engineering practices for society, sustainability and environment**: Apply appropriate technology in context of society, sustainability, environment and ethical practices.
6. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes *in field of engineering.*

C. Expected Skills to be developed based on competency:

Design highly customizable form usingBootstrap input groupsto enhance the user experience by providing a better and more interactive input interface

D. Expected Course Outcomes(Cos)

Apply reusable bootstrap components to design effective user-friendly web pages.

E. Practical Outcome(PRo)

To create responsive webpage for job application or any other kind of application using bootstrap input group components.

F. Expected Affective domain Outcome(ADos)

1. Follow Coding standards and practices.
2. Maintain tools and equipment.
3. Follow safety practices.
4. Follow ethical practices

G. Prerequisite Theory:

Bootstrap's Input Group component: Bootstrap's Input Group component is a powerful feature that allows you to extend and enhance form controls by adding additional elements before, after, or on both sides of an input field. This can be useful for creating stylish and functional form layouts. Input groups can contain various elements such as buttons, dropdowns, text, and icons, and they are designed to seamlessly integrate with Bootstrap's grid system.

Input Group Container:

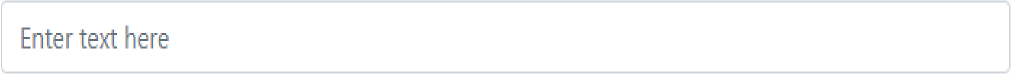
The Input Group container is the main wrapper that holds all the elements of the input group. It is created using the div element with the class .input-group. Inside this container, you'll place the input field and any additional elements that you want to include.

```
<div class="input-group">
<!-- Input field and additional elements go here -->
</div>
```

Input Field:

The input field is the core form control element where users can input their data. It can be any HTML input type, such as text, number, email, etc. You can add the input field by placing it inside the input group container.

```
<div class="input-group">
<input type="text" class="form-control" placeholder="Enter text here">
</div>
```



Enter text here

Add-ons (Prepend and Append Elements):

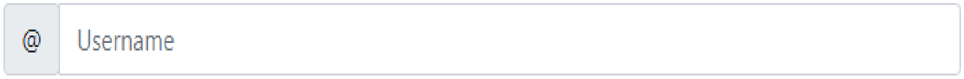
In Bootstrap, "prepend" and "append" are classes used to add content or elements before or after an input field, respectively. These classes are helpful when you want to provide additional context, icons, or other elements directly attached to an input field. This can be commonly seen in forms where you want to label or signify the purpose of the input field in a more visually appealing way.

Let's go through a step-by-step example of using "prepend" and "append" classes with Bootstrap to enhance the appearance of input fields.

To create add-ons, use the .input-group-prepend or .input-group-append class with a <span> or <div> element.

Let's take the example of a Bootstrap input group to demonstrate how to use prepend and append inputs.

```
<div class="input-group mb-3">
<div class="input-group-prepend">
<span class="input-group-text">@</span>
</div>
<input type="text" class="form-control" placeholder="Username" aria-label="Username" aria-describedby="basic-addon1">
</div>
```

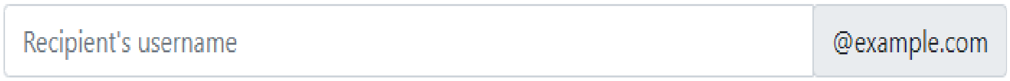


In this example, we have an input field for a username. The "@" symbol should appear before the input field. To achieve this, we use the input-group-prepend class. It creates a container element (a div in this case) that will hold the content we want to appear before the input field.

Within the input-group-prepend container, we place a span element with the class input-group-text. This class applies the necessary styles to make the "@" symbol appear properly and align it with the input field.

Similarly, let's add an example for appending content:

```
<div class="input-group mb-3">
<input type="text" class="form-control" placeholder="Recipient's username" aria-label="Recipient's username" aria-describedby="basic-addon2">
<div class="input-group-append">
<span class="input-group-text">@example.com</span>
</div>
</div>
```



In this case, we want to append "@example.com" after the input field. To achieve this, we use the input-group-append class. This class creates a container element that will hold the content to appear after the input field.

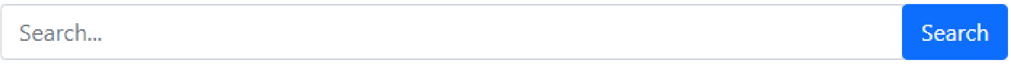
Inside the input-group-append container, we place a span element with the class input-group-text. The content of this span element will be "@example.com," which will be displayed after the input field.

By using these "prepend" and "append" classes provided by Bootstrap, you can easily enhance the appearance and functionality of your input fields and add relevant content before or after them as needed.

Button Add-ons:

Input groups can also include buttons as add-ons, which can be used for various purposes like search buttons or submit buttons.

```
<div class="input-group">
<input type="text" class="form-control" placeholder="Search...">
<div class="input-group-append">
<button class="btn btn-primary" type="button">Search</button>
</div>
</div>
```

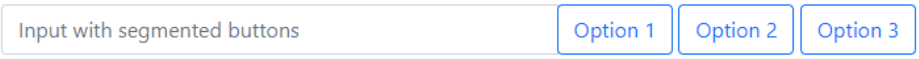


In this example, we have a search input with a search button as the add-on on the right side.

Segmented Buttons:

You can also create segmented button groups as input add-ons.

```
<div class="input-group">
<input type="text" class="form-control" placeholder="Input with segmented buttons">
<div class="input-group-append">
<button class="btn btn-outline-primary" type="button">Option 1</button>
<button class="btn btn-outline-primary" type="button">Option 2</button>
<button class="btn btn-outline-primary" type="button">Option 3</button>
</div>
</div>
```



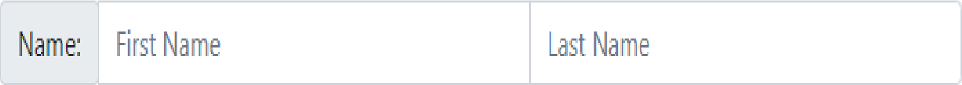
In this example, we have an input element with three segmented buttons as the add-on on the right side.

Multiple inputs within an input group

you can create a group of multiple inputs using the "prepend" and "append" classes, typically in conjunction with Bootstrap or other CSS frameworks. These classes allow you to visually group input elements together, such as combining input fields with additional content like buttons or text.

Here's an example of how you can use the "prepend" classe with Bootstrap to create a group of inputs:

```
<div class="container mt-5">
<div class="input-group">
<div class="input-group-prepend">
<span class="input-group-text">Name:</span></div>
<input type="text" class="form-control" placeholder="First Name">
<input type="text" class="form-control" placeholder="Last Name">
</div>
</div>
```



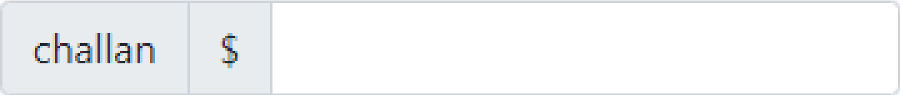
In this example, we create an input group with two text input fields for the first name and last name. We use the "input-group" class to define the container for the inputs, and within this container, we use "input-group-prepend" to specify the content that should appear before the inputs.

The content in the "input-group-prepend" div is specified by the "input-group-text" class, which gives it a visually distinct appearance from regular text. In this case, we use it to add the label "Name:" before the input fields.

Multiple Addons within an input group

Multiple addons could be stacked or mixed together with other types. To achieve this, you can add different elements such as text, buttons, or icons before the input field to enhance its functionality or provide context.To create multiple addons within an input group using the "prepend" class, you can follow this structure:

```
<div class="input-group">
<div class="input-group-prepend">
<span class="input-group-text">challan</span>
 <span class="input-group-text">$</span>
</div>
<input type="text" class="form-control" placeholder="Input field">
<!-- Additional addons can be added here if needed -->
</div>
```



In this example,outermost element with the class "input-group" defines the container for the input group.Each addon is placed within a div with the class "input-group-prepend". This class is used to position the addon elements before the input field.

Inside each "input-group-prepend" div, there is a <span> element with the class "input-group-text". This class adds styles to the text elements and provides a consistent look for the addons.The input field is placed after the addon elements and is styled using the "form-control" class, making it responsive and visually consistent with Bootstrap styles.

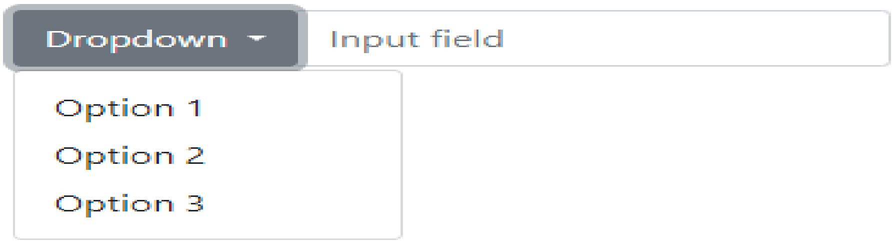
You can repeat the structure to add more addons before the input field if needed.

Button dropdowns within an input group

In Bootstrap, button dropdowns within an input group combine the features of input groups and dropdowns to create a powerful and flexible user interface element. Input groups in Bootstrap are used to enhance form controls by adding additional elements or buttons before, after, or on both sides of an input field. Button dropdowns, on the other hand, allow you to create a button that reveals a dropdown menu when clicked. When these two components are combined, you get button dropdowns within an input group, providing users with a wide range of options to choose from within a form field.

To create the HTML structure for the input group with the button dropdown. Here's an example of how it can be done:

```
<div class="input-group">
<div class="input-group-prepend">
<!-- Add a regular button or an icon (e.g., a search icon) here -->
<button class="btn btn-outline-secondary dropdown-toggle" type="button"
data-bs-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
    Dropdown
</button>
<!-- Add a dropdown menu here -->
<div class="dropdown-menu">
<!-- Dropdown items -->
<a class="dropdown-item" href="#">Option 1</a>
<a class="dropdown-item" href="#">Option 2</a>
<a class="dropdown-item" href="#">Option 3</a>
</div></div>
<!-- Add an input field here (if required) -->
<input type="text" class="form-control" placeholder="Input field">
</div>
```



In the example above, the input-group class creates the input group container. Inside the container, we have input-group-prepend, which indicates the elements that should appear before the input field.

Within input-group-prepend, we place the button element (<button>) with the class btn, along with any other necessary classes such as btn-outline-secondary to style it. The dropdown-toggle class is crucial as it tells Bootstrap to activate the dropdown behavior for this button. The data-bs-toggle="dropdown" attribute and the aria attributes are used to manage the dropdown behavior and accessibility.

The dropdown menu itself is defined using the dropdown-menu class. Inside this element, you can add various options (<a> elements with the class dropdown-item) that will be displayed as items in the dropdown menu.

H. Program Code & Output Source code:

SOURCE CODE

OUTPUT

Signature

Practical No.10: UseNavigation tabs/Pills to create responsive web page for summarize of all individual units of any one course.

A. Objective:

Understand various tabs and pills components of bootstrap to manage large content information into small specific sections by giving specific name or title of various sections and navigates to each other's without page refreshing.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the *engineering* problems.
2. **Problem analysis**: Identify and analyse well-defined *engineering* problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
5. **Engineering practices for society, sustainability and environment**: Apply appropriate technology in context of society, sustainability, environment and ethical practices.
6. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes *in field of engineering.*

C. Expected Skills to be developed based on competency:

Familiar with tabs and pills classes of bootstrap framework to customize/manage information as per needs and navigates between them using tab or pill components to provide more interactive interface.

D. Expected Course Outcomes(Cos)

Apply reusable bootstrap components to design effective user-friendly web pages.

E. Practical Outcome(PRo)

To create navigation bar for summarize unit wise details of any one course of computer engineering branch using either tabs or pills classes of bootstrap.

F. Expected Affective domain Outcome(ADos)

1. Follow Coding standards and practices.
2. Maintain tools and equipment.
3. Follow safety practices.
4. Follow ethical practices

G. Prerequisite Theory:

Bootstrap Tabs/Pills : Bootstrap tabs provide a clean and organized way to present content in a webpage, allowing users to switch between different sections without loading separate pages. Tabs are great for grouping related information, such as different categories, sections, or steps in a process.

Bootstrap "pill" refers to a specific style of navigation for creating tabbed interfaces. The Bootstrap "pill" style provides a clean and visually appealing way to organize content into separate sections using tabs. Each tab represents a different section of content, and users can switch between tabs to view the content of each section without navigating to a new page.

Tabbed navigation is a common UI pattern that allows users to switch between different content sections within the same page, typically organized as tabs. When a tab is activated, its corresponding content panel becomes visible while others are hidden. In Bootstrap, tabs are implemented using HTML, CSS, and JavaScript, and they are easy to set up and customize.

Following components are used to design responsive navigation using tabs or pills.

tab-content class: The tab-content class is used to create a container for all the content panels (tab panes) that are associated with the tab navigation. It acts as a parent element to hold the individual tab content sections.

tab-pane class: The tab-pane class is used to represent individual content sections associated with each tab. It is essential to add this class to create the content area for each tab.

data-toggle attribute: data-toggle attribute is used to control the behavior of various interactive components, such as dropdowns, modals, and tabs, without writing any JavaScript code explicitly. It is an essential part of Bootstrap's JavaScript plugins, providing an easy way to trigger specific actions or display hidden content.

data-toggle="tab" attribute is used in conjunction with the Tab JavaScript plugin to create tabbed navigation **.nav-tabs** class is used while use data-toggle=tab data attribute in design as shown below example.

data-toggle="pill" is a data attribute used to create tabbed navigation that switches between content panes, also known as **"pill" navigation**. This feature allows you to organize related content into separate tabs, making it easy for users to switch between different sections of content without navigating to a new page..**nav-pills** class instead of .**nav-tabs** class is used while use **'data-toggle=pill' instead of** data-toggle=tab data attribute in design for pill navigation

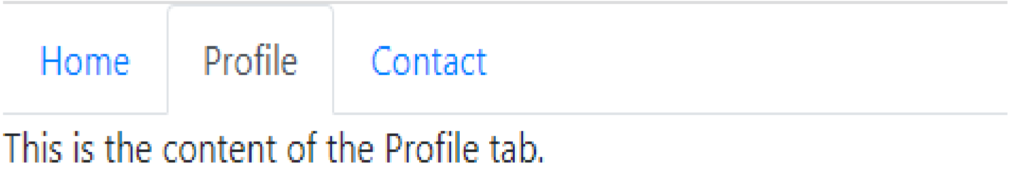
First you have to include the Bootstrap CSS and JavaScript files in your HTML file. You can do this by adding the following lines to your <head> section:

```
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css">

<script
src="https://cdn.jsdelivr.net/npm/jquery@3.6.4/dist/jquery.slim.min.js"></script>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js"></script>
```

Next, you need to create the HTML structure for the tabs and their associated content.

Here's an example of how it can be done:

```
<ul class="nav nav-tabs">
<li class="nav-item" >
<a class="nav-link active"  data-toggle="tab" href="#home" >Home</a>
</li><li class="nav-item">
<a class="nav-link"  data-toggle="tab" href="#profile" >Profile</a>
</li><li class="nav-item">
<a class="nav-link" data-toggle="tab" href="#contact"
>Contact</a></li></ul>
<div class="tab-content" >
<div class="tab-pane fade show active" id="home" >
<p>This is the content of the Home tab.</p></div>
<div class="tab-pane fade" id="profile" >
<p>This is the content of the Profile tab.</p></div>
<div class="tab-pane fade" id="contact" >
<p>This is the content of the Contact tab.</p></div></div>
```



This is the content of the Profile tab.

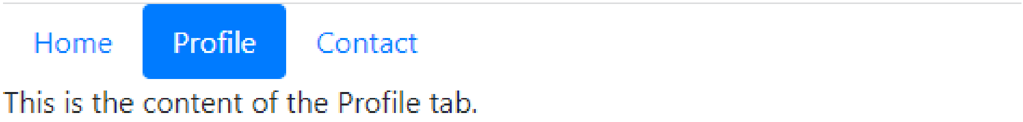
Let's break down the code above:

- The **ul** element with the class **nav** and **nav-tabs** represents the tab navigation itself. Each tab is represented by an**li** element with the class **nav-item**.
- Inside each **nav-item**, there's an **a** element with the class **nav-link**. The **data-toggle="tab"** attribute is added to each **a** element to enable tab switching functionality. The **href** attribute of each **a** element points to the corresponding

content section (i.e., the **tab-pane**) that should be displayed when the tab is activated.

- The content sections for each tab are represented by **div** elements with the class **tab-pane**. The **id** attribute of each **div** element should match the **href** value of the corresponding **a** element to establish the association between tabs and their content.
- The **tab-content** class is applied to the container **div** element that holds all the content sections (**tab-pane**). This class is essential for Bootstrap's Tab plugin to work correctly.
- By default, the first tab has the **show active** classes applied to its corresponding **tab-pane** to indicate that it should be displayed initially. When users click on other tabs, the **show active** classes will be toggled to show the appropriate content section while hiding the others.

If you replace **nav-tabs**class with **nav-pills** and**data-toggle="tab"**with**data-toggle="pill"**in above code then you will get following pill navigation output:



This is the content of the Profile tab.

H. Program Code & Output Source code:

SOURCE CODE

OUTPUT

Signature

Practical No.11: UseNav bar component to create responsive fixed to top menu design includes logo, menu, drop down menu, form input elements such as sign-up button, search mechanism etc. And fixed to bottom menu design contains footer information.

A. Objective:

Understand usage of various classes of nav bar component to design responsive navigation bar including various option such as logo, drop down menu, form control input fields at specific position of viewport.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the *engineering* problems.
2. **Problem analysis**: Identify and analyse well-defined *engineering* problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
5. **Engineering practices for society, sustainability and environment**: Apply appropriate technology in context of society, sustainability, environment and ethical practices.
6. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes *in field of engineering.*

C. Expected Skills to be developed based on competency:

Design customize navigation bar as per needs of website to provide more interactive interface to user.

D. Expected Course Outcomes(Cos)

Apply reusable bootstrap components to design effective user-friendly web pages.

E. Practical Outcome(PRo)

To create responsive header and footer navigation bar at specific position of webpage includes various option such as logo, menu, drop down menu, form input elements.

F. Expected Affective domain Outcome(ADos)

1. Follow Coding standards and practices.
2. Maintain tools and equipment.

3. Follow safety practices.
4. Follow ethical practices

G. Prerequisite Theory:

Bootstrap framework provides a versatile and customizable Navbar (navigation bar) component that allows you to create responsive and user-friendly navigation menus for your website or web application. The Bootstrap Navbar is designed to work on various screen sizes and devices, making it a powerful tool for building modern and mobile-friendly interfaces.

navbar-brand class

The navbar-brand class is a specific class used within Bootstrap's Navbar component. The Navbar is a horizontal navigation bar that typically appears at the top of a website or web application. It is commonly used to house links to various pages or sections of the site.

The purpose of the navbar-brand class is to create a stylized logo or brand name within the Navbar. This class is usually applied to an anchor <a> element, which serves as a link back to the homepage or primary landing page of the website. It is typically placed on the left side of the Navbar.

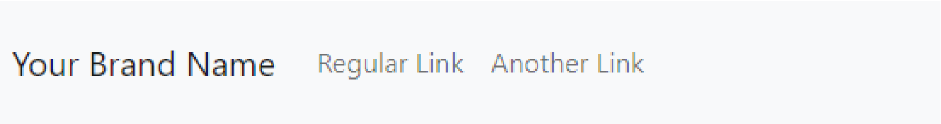
The navbar-brand can contain either a logo (usually an image) or plain text (the brand name). Developers can use an <img> tag within the anchor element to display a logo, or simply use text.

Using a logo (image) example:

```
<a class="navbar-brand" href="#home">
<img src="path/to/your-logo.png" alt="Your Logo">
</a>
```

Using plain text (brand name) example:

```
<a class="navbar-brand" href="#home">Your Brand Name</a>
```



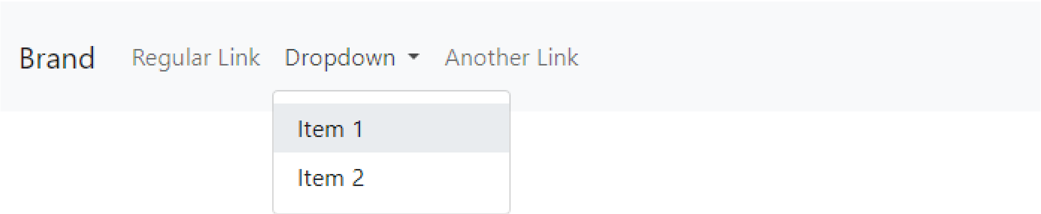
Dropdown class:

The dropdown class is used to create a dropdown menu within the navigation bar. It allows you to group related links under a single parent item. When users click on the parent item, the dropdown menu reveals additional links or content associated with that item.

Example of a basic dropdown structure:

```
<nav class="navbar">
```

```
<ul class="navbar-nav">
<li class="nav-item">
<a class="nav-link" href="#">Regular Link</a></li>
<li class="nav-item dropdown">
<a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
    Dropdown</a>
<div class="dropdown-menu" aria-labelledby="navbarDropdown">
<!-- Dropdown menu items -->
<a class="dropdown-item" href="#">Item 1</a>
<a class="dropdown-item" href="#">Item 2</a>
<!-- Additional dropdown items go here -->
</div></li>
<li class="nav-item">
<a class="nav-link" href="#">Another Link</a></li></ul>
</nav>
```



The nav-item class represents a single item in the navigation bar, such as a link or a dropdown.The dropdown class is added to the nav-item that contains the dropdown menu.The nav-link class represents a navigation link, which is a clickable element that will either take the user to a new page or open the dropdown menu when clicked.

The dropdown-toggle class is used on the link that should trigger the dropdown to be shown. It adds the dropdown arrow indicator and enables the dropdown functionality.The dropdown-menu class is applied to the container that holds the actual dropdown items.

The aria-labelledby attribute is important for accessibility. It associates the dropdown menu with its triggering link, improving screen reader support and navigation.The content of the dropdown menu is defined within the <div class="dropdown-menu"> container. Each individual item within the dropdown is represented by an anchor (<a>) element with the class dropdown-item. You can add as many items as needed.

When the user clicks on the link with the dropdown-toggle class, the dropdown menu will appear below the link. It can be customized to slide down, fade in, or use other animations based on your CSS settings and the version of Bootstrap you are using.

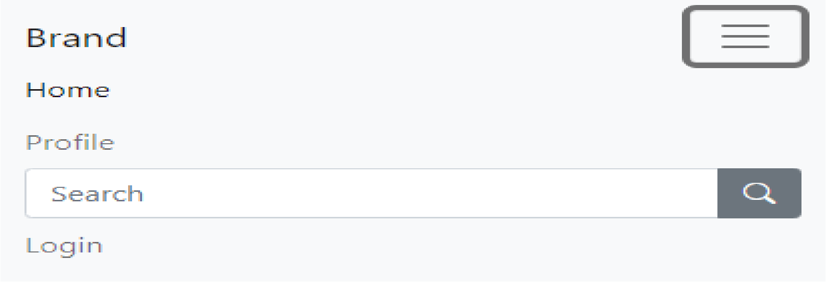
form-control class

the form-control class in Bootstrap is primarily used for styling form elements like text inputs, textareas, and select dropdowns. It is not a class typically used directly on a navigation bar (navbar).

If you want to include form elements like text inputs or search fields in a Bootstrap navigation bar, you can do so by using the appropriate components, such as the form-inline class or input-group class, in combination with the form control classes.

Here's an example of how you can add a search field to a Bootstrap navigation bar using the form-inline class and the form-control class:

```
<div class="m-4">
<nav class="navbar navbar-expand-lg navbar-light bg-light">
<div class="container-fluid">
<a href="#" class="navbar-brand">Brand</a>
<button type="button" class="navbar-toggler" data-bs-toggle="collapse"
data-bs-target="#navbarCollapse">
<span class="navbar-toggler-icon"></span></button>
<div class="collapse navbar-collapse justify-content-between"
id="navbarCollapse">
<div class="navbar-nav">
<a href="#" class="nav-item nav-link active">Home</a>
<a href="#" class="nav-item nav-link">Profile</a>
<form class="form-inline">
<div class="input-group">
<input type="text" class="form-control" placeholder="Search">
<button type="button" class="btn btn-secondary"><i class="bi-
search"></i></button></div>
</form></div>
<div class="navbar-nav">
<a href="#" class="nav-item nav-link">Login</a></div></div></div>
</nav></div>
```



In this example,The form-inline class is applied to the form element to create an inline form within the navigation bar.The form-control class is applied to the search input field to style it as a Bootstrap form control.The "bi-search" class is used to display a search icon from the Bootstrap Icons library.

This creates a responsive navigation bar with a search field on the right side of the Navbar. As the screen size reduces, the navigation links collapse into a hamburger menu, and the search field adjusts accordingly.

navbar fixed-top class

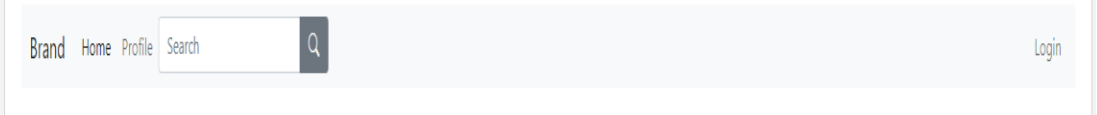
In Bootstrap, the navbar fixed-top class is used to create a navigation bar (Navbar) that sticks to the top of the viewport even when the user scrolls down the page. The fixed-top navbar bar is fixed at the top of the viewport and does not scroll with the rest of the page.It is a popular technique for keeping the navigation always visible and accessible to users, especially on long pages.

Here's an example of how to create a fixed-top Navbar using Bootstrap:

In above example simply add **fixed-top class** as given below.

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

The fixed-top class is added to the navbar element. This class makes the Navbar stick to the top of the viewport while scrolling down the page.



navbar fixed-bottom class

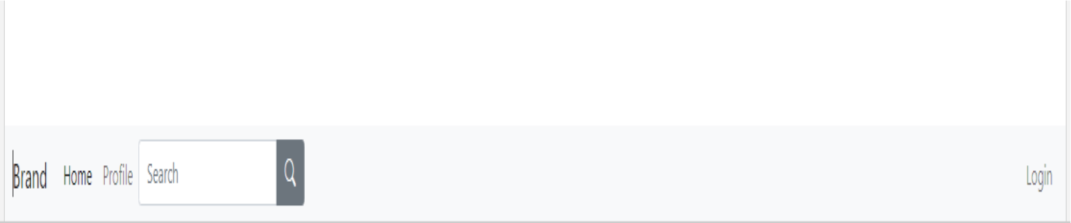
navbar fixed-bottom class is used to create a navigation bar (Navbar) that sticks to the bottom of the viewport even when the user scrolls down the page. It's similar to the fixed-top Navbar, but it remains fixed at the bottom of the screen.

Here's an example of how to create a fixed-bottom Navbar using Bootstrap:

In above example simply add **fixed-bottom class** as given below.

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

The fixed-bottom class is added to the navbar element. This class makes the Navbar stick to the bottom of the viewport while scrolling down the page.



Bootstrap Card classes

In Bootstrap, the "card" component is used to create a flexible and stylish container to display content, such as images, text, and other elements. It provides a clean and consistent way to present information within a website or web application. The card component is used for things like blog posts, product listings, and other content blocks.Let's describe each of the card-related classes in detail:

Card (card class):

The card class is the fundamental class that defines the Bootstrap card component. A card is a flexible and extensible content container that can be used to display various types of content in a consistent and visually appealing way. It is commonly used for presenting information, images, and other types of media.

The basic structure of a card looks like this:

```
<div class="card">
<!-- Card content goes here -->
</div>
```

You can customize the appearance of the card using various other card-related classes and components.

Card Body (card-body class):

The card-body class is used to create the main content area within a card. It is where most of the textual or media content of the card is placed.

The card body class is often used like this:

```
<div class="card">
<div class="card-body">
<!-- Card content goes here -->
</div>
</div>
```

The content within the card-body will typically include card title, card text, images, buttons, or any other relevant information.

Card Title (card-title class):

The card-title class is used to create the title or heading section within a card. It is commonly used to display a brief title that describes the card's content.

The card title class is often used like this:

```
<div class="card">
<div class="card-body">
<h5 class="card-title">Card Title</h5>
```

```
<!-- Other content goes here -->
</div>
</div>
```

card-subtitle:

The card-subtitle class is used to display a subtitle or secondary information below the main title. It is similar to the card-title class but typically used for additional information.

```
<div class="card">
<div class="card-body">
<h5 class="card-title">Card Title</h5>
<h6 class="card-subtitle mb-2 text-muted">Card Subtitle</h6>
<!-- Additional content of the card goes here -->
</div></div>
```

card-img class:

The card-img class is used to display an image within the card. It is commonly placed at the top of the card and can be used to showcase a thumbnail or an illustration related to the card content.

```
<div class="card">
<img src="image.jpg" class="card-img-top" alt="Card Image">
<div class="card-body">
<!-- Card content with image -->
</div></div>
```

card-text class:

The card-text class is used to add regular text content inside the card. It provides default font styling for paragraphs or text content.

```
<div class="card">
<div class="card-body">
<h5 class="card-title">Card Title</h5>
<p class="card-text">This is some sample text within the card.</p>
<!-- Other card content -->
</div></div>
```

card-header class:

The card-header class is used to create a header section for the card. It typically contains elements such as additional titles, buttons, or icons that provide context or actions related to the card.

```
<div class="card">
<div class="card-header">
   Card Header
</div><div class="card-body">
<!-- Card content -->
</div></div>
```

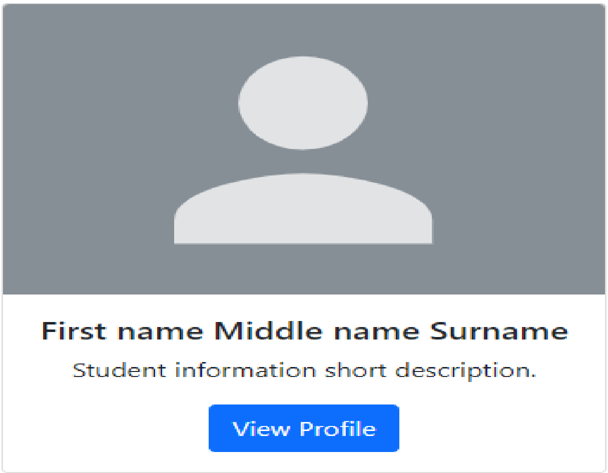
card-footer class:

The card-footer class is used to create a footer section for the card. It can contain elements like buttons, links, or text that provide additional information or actions related to the card content.

```
<div class="card">
<div class="card-body">
<!-- Card content -->
</div><div class="card-footer">
   Card Footer
</div></div>
```

Below is an example of a simple Bootstrap shopping cart using cards to display the student information:

```
<div class="container mt-4">
<div class="row"><div class="col-md-4">
<div class="card">
<img src="r1.jpg" class="card-img-top" alt="Student Name">
<div class="card-body">
<h5 class="card-title">First name  Middle name Surname</h5>
<p class="card-text">Student information short description.</p>
<button class="btn btn-primary">View more</button></div></div>
</div></div>
</div>
```



First name Middle name Surname

Student information short description.

H. Program Code & Output Source code:

SOURCE CODE

OUTPUT

Signature

Practical No.12: Use media, rounded media, Nested media object to create responsive web page for all family members in hierarchical order.

A. Objective:

Understand the usage of media objects, rounded media and nested media objects while designing responsive web page.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the *engineering* problems.
2. **Problem analysis**: Identify and analyse well-defined *engineering* problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
5. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes *in field of engineering.*

C. Expected Skills to be developed based on competency:

Studentsrepresentation skill will be developed by using different media objects to display organisation chart, family chart.

D. Expected Course Outcomes(Cos)

Apply reusable bootstrap components to design effective user-friendly web pages.

E. Practical Outcome(PRo)

Apply different media objects to prepare different charts.

F. Expected Affective domain Outcome(ADos)

1. Follow Coding standards and practices.
2. Maintain tools and equipment.
3. Follow safety practices.
4. Follow ethical practices

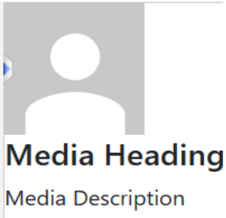
G. Prerequisite Theory:

Prerequisite Theory: Here's a detailed theory for the Bootstrap media, rounded media, and nested media components, along with an example:

Bootstrap Media Component: The Bootstrap Media component provides a flexible way to display images, videos, and other media content along with accompanying text. It consists of a parent div with the class media, which contains two main components: media-left and media-body. The media-left component is used to display the media content, such as an image or video thumbnail, while the media-body component holds the accompanying text or additional information. Here's an example:

```
<div class="media">
<div class="media-left">
<img src="image.jpg" alt="Media Image"></div>
<div class="media-body">
<h4 class="media-heading">Media Heading</h4>
<p>Media Description</p>
</div></div>
```

Output:



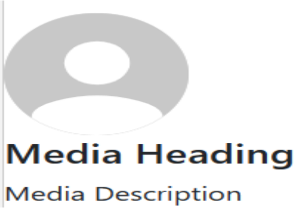
Media Heading

Media Description

Bootstrap Rounded Media Component: The Bootstrap Rounded Media component adds a rounded shape to the media content, such as images or video thumbnails, by applying the rounded-circle class to the media content element. This class gives a circular shape to the media content, creating a visually appealing rounded media object. Here's an example:

```
<div class="media">
<div class="media-left">
<img class="rounded-circle" src="image.jpg" alt="Media
Image"></div>
<div class="media-body">
<h4 class="media-heading">Media Heading</h4>
<p>Media Description</p>
</div>
</div>
```

Output:



Media Heading

Media Description

Bootstrap Nested Media Component: The Bootstrap Nested Media component allows you to create a hierarchical structure by nesting media components inside each other. This is useful when you want to display nested content, such as comments, replies, or subcategories. Simply include a media component inside the media-body section of another media component to create the nested structure. Here's an example:

```
<div class="media">
<div class="media-left">
<img class="rounded-circle" src="image.jpg" alt="Media Image">
</div>
<div class="media-body">
<h4 class="media-heading">Media Heading</h4>
<p>Media Description</p>
<!-- Nested Media Component -->
<div class="media">
<div class="media-left">
<img class="rounded-circle" src="nested_image.jpg" alt="Nested Media Image">
</div>
<div class="media-body">
<h4 class="media-heading">Nested Media Heading</h4>
<p>Nested Media Description</p>
</div></div></div>
</div>
```

In this example, we have a parent media component with a rounded media object. Inside the media-body section of the parent media component, there's another nested media component with its own rounded media object. This structure can be repeated to create multiple levels of nesting as needed.The Bootstrap media, rounded media, and nested media components provide powerful tools for displaying media content and hierarchical structures. Use these components to enhance your website or application's visual design and improve the presentation of media and nested content. Customize the content and styles based on your specific needs to create engaging and informative layouts.

```
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<title>Family Members</title></head>
<body>
```

```
<div class="container">
<h1>Family Members</h1>
<div class="media">
<img src="person1.jpg" class="mr-3 rounded" alt="Person 1">
<div class="media-body">
<h5 class="mt-0">John Doe</h5>
<p>Father</p>
<div class="media">
<img src="person2.jpg" class="mr-3 rounded" alt="Person 2">
<div class="media-body">
<h5 class="mt-0">Jane Doe</h5>
<p>Mother</p>
<div class="media">
<img src="person3.jpg" class="mr-3 rounded" alt="Person 3">
<div class="media-body">
<h5 class="mt-0">Sarah Doe</h5>
<p>Daughter</p></div></div>
<div class="media">
<img src="person4.jpg" class="mr-3 rounded" alt="Person 4">
<div class="media-body">
<h5 class="mt-0">David Doe</h5>
<p>Son</p>
</div></div></div></div>
</div></div>
</div>
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.mi
n.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"><
/script>
</body>
</html>
```

I. Program Code & Output Source code:

SOURCE CODE

OUTPUT

Signature

Practical No.13: Design a smooth page transition between homepage, about and contact us page using bootstrap transition plugin.

A. Objective:

Use smooth page transitions among pages in responsive website using transition plugin.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the *engineering* problems.
2. **Problem analysis**: Identify and analyse well-defined *engineering* problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
5. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes *in field of engineering.*

C. Expected Skills to be developed based on competency:

Students navigate between pages with smooth transition using transition plug-in.

D. Expected Course Outcomes(Cos)

Develop interactive features rich web pages using Bootstrap jQuery plug-ins..

E. Practical Outcome(PRo)

Apply transition plug-in to make smooth transition between pages of responsive website.

F. Expected Affective domain Outcome(ADos)

1. Follow Coding standards and practices.
2. Maintain tools and equipment.
3. Follow safety practices.
4. Follow ethical practices

G. **Prerequisite Theory:** The concept of a "transition plugin" in Bootstrap refers to the functionality provided by Bootstrap's JavaScript library to smoothly animate the transition between different states of an element on a webpage. The transitions can be applied to various elements, such as modals, dropdowns, tabs, carousels, and more. This plugin utilizes CSS transitions and JavaScript to achieve the desired effects. Here's a detailed theory of how the transition plugin works in Bootstrap, along with an example:

CSS Transitions: Bootstrap leverages CSS transitions to animate the properties of elements. CSS transitions allow you to define the duration, timing function, delay, and other properties related to the animation. Bootstrap uses these transitions to create smooth and visually appealing effects during state changes.

JavaScript Event Handlers: Bootstrap's transition plugin utilizes JavaScript event handlers to trigger the transitions. These event handlers listen for specific events, such as the opening or closing of an element, and apply the necessary CSS classes to initiate the transitions.**Transition.js** is a simple helper for transition End events as well as a CSS changeover emulator. It is used by the other plugins to check for CSS transition support and to catch hanging transitions.

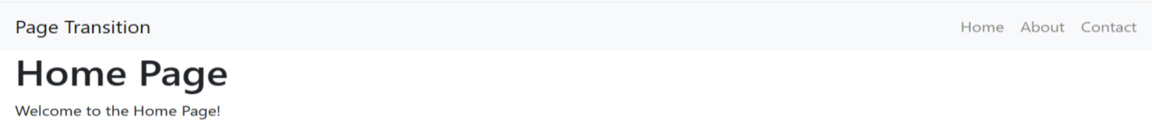
CSS Classes: Bootstrap adds and removes CSS classes to elements to control the transitions. These classes define the starting and ending states of the elements and include properties that define the animation. The commonly used classes are:

.fade: Adds a fading effect to an element.
.collapse: Hides or shows an element with a sliding animation.
.show: Displays a hidden element.
.collapsing: Applied during the animation phase of the collapse effect.
.in: Applied to indicate the element is in the target state.
Example: Transition Plugin in Bootstrap (Collapsible element),Let's consider an example of a collapsible element, such as an accordion, where clicking a header reveals or hides the content below it.
HTML structure:

```
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Bootstrap Page Transition Example</title>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<link rel="stylesheet" href="styles.css">
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-light bg-light">
<div class="container">
```

```
<a class="navbar-brand" href="#">Page Transition</a>
<ul class="navbar-nav ml-auto">
<li class="nav-item">
<a class="nav-link" href="#home">Home</a></li>
<li class="nav-item">
<a class="nav-link" href="#about">About</a </li>
<li class="nav-item">
<a class="nav-link" href="#contact">Contact</a></li>
</ul>
</div>
</nav>
<div class="container page-content">
<div id="home" class="page active">
<h1>Home Page</h1>
<p>Welcome to the Home Page!</p></div>
<div id="about" class="page">
<h1>About Page</h1>
<p>This is the About Page.</p></div>
<div id="contact" class="page">
<h1>Contact Page</h1>
<p>Contact us here.</p></div>
</div>
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script>
  $(document).ready(function () {
    $(".nav-link").on("click", function (event) {
event.preventDefault();
var targetPage = $(this).attr("href");
      // Fade out the current active page
      $(".page.active").addClass("fade-out");
setTimeout(function () {
        $(".page.active").removeClass("active fade-out");
        // Show the target page after the fade-out transition
        $(targetPage).addClass("active fade-in");
      }, 500); // Adjust the delay based on the CSS transition duration
      // Update the active nav-link
      $(".nav-link").removeClass("active");
      $(this).addClass("active");
    });
  });
</script>
</body>
</html>
```

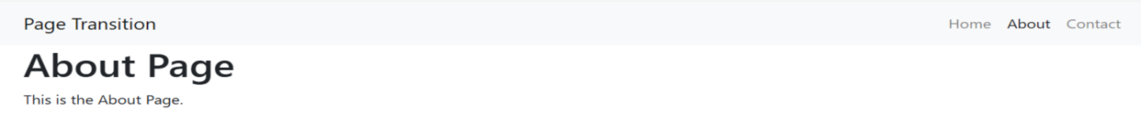
Output:



Page Transition

Home Page

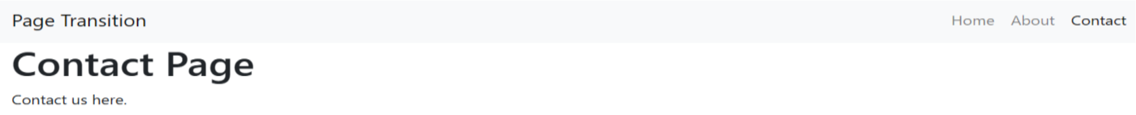
Welcome to the Home Page!



Page Transition

About Page

This is the About Page.



Page Transition

Contact Page

Contact us here.

Explanation: In this example, we've created a basic Bootstrap layout with a navigation bar and three content sections representing different pages. When you click on a navigation link, we use jQuery to handle the click event. We apply CSS classes to the current and target pages to initiate the fade-out/fade-in transitions using CSS. The active class is used to show the current page, and the fade-out and fade-in classes control the opacity transition.Keep in mind that this is just a basic example. For more complex page transitions, you might need to use CSS animations or leverage third-party libraries that offer more advanced transition effects.

H. Program Code & Output Source code:

SOURCE CODE

OUTPUT

Signature

Practical No.14: Design a webpage with different modal dialog for “Save record confirmation”, “Delete record confirmation” using model dialog plugin of bootstrap.

A. Objective:

Use of model dialog plug-into give alerts and confirmation messages to use in web pages.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the *engineering* problems.
2. **Problem analysis**: Identify and analyse well-defined *engineering* problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
5. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes *in field of engineering.*

C. Expected Skills to be developed based on competency:

Students design page with different modal dialogs.

D. Expected Course Outcomes(Cos)

Develop interactive features rich web pages using Bootstrap jQuery plug-ins..

E. Practical Outcome(PRo)

Design responsive web page using modal dialog plug-in.

F. Expected Affective domain Outcome(ADos)

1. Follow Coding standards and practices.
2. Maintain tools and equipment.
3. Follow safety practices.
4. Follow ethical practices

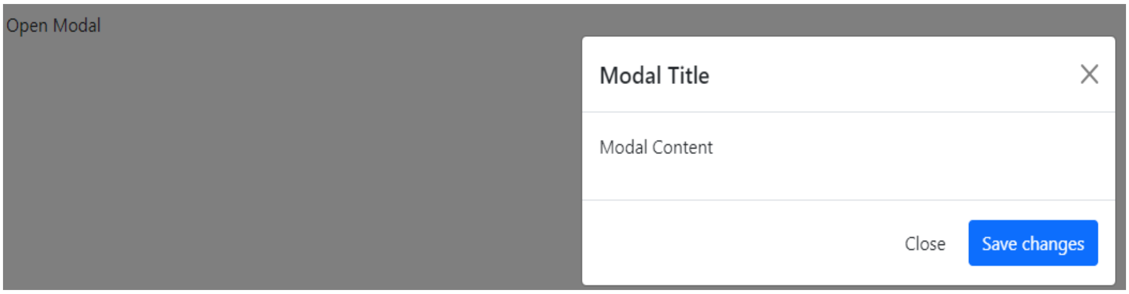
G.

Prerequisite Theory: The Modal component in Bootstrap allows you to create a modal dialog, a popup window that appears on top of the current page. Modals are commonly used for displaying additional content, forms, notifications, or interactive elements. They provide a way to focus the user's attention and temporarily interrupt their current workflow. To use the Modal component in Bootstrap, you need to include the Bootstrap CSS and JavaScript files, as well as the jQuery library, in your HTML file. Here's how you can create a modal dialog using the Modal component.

Example:

```
<!-- Button to trigger the modal -->
<button type="button" class="btn" data-bs-toggle="modal" data-bs-target="#myModal">
  Open Modal
</button>
<!-- The modal dialog -->
<div class="modal" id="myModal">
<div class="modal-dialog">
<div class="modal-content">
<!-- Modal header -->
<div class="modal-header">
<h5 class="modal-title">Modal Title</h5>
<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button></div>
<!-- Modal body -->
<div class="modal-body">
<p>Modal Content</p></div>
<!-- Modal footer -->
<div class="modal-footer">
<button type="button" class="btn" data-bs-dismiss="modal">Close</button>
<button type="button" class="btn btn-primary">Save changes</button>
</div></div></div>
</div>
```

Output:



In this example, we have a button element that triggers the modal dialog. It has the **data-bs-toggle="modal"** attribute to indicate that it should open a modal. The **data-bs-target="#myModal"** attribute specifies the target modal using its ID.

Inside the modal dialog, we have the necessary HTML structure. The modal's content is contained within the **<div class="modal">** element. It consists of a modal dialog (**'<div class="modal-dialog">'**) and modal content (**'<div class="modal-content">'**). The content is divided into three sections: modal header, modal body, and modal footer.

JavaScript Initialization: In the JavaScript code below, we use the **bootstrap.Modal** constructor to initialize the modal dialog. We pass the element reference of the modal using **document.getElementById('myModal').**

Example:

```
$(document).ready(function() {
var myModal = new
bootstrap.Modal(document.getElementById('myModal'));
});
```

Opening and Closing the Modal: To open or close the modal dialog rogrammatically, you can use JavaScript/jQuery methods.

To open the modal:

```
myModal.show();
```

To close the modal:

```
myModal.hide();
```

Options and Configurations: The Modal component offers various options and configurations to customize its behavior. You can pass an object with options when initializing the modal. For example:

```
$(document).ready(function() {
var myModal = new
bootstrap.Modal(document.getElementById('myModal'), {
backdrop: 'static',
keyboard: false
  });
});
```

In the code above, we set the **backdrop** option to 'static' to prevent the modal from closing when the user clicks outside of it. The **keyboard** option is set to **false** to disable closing the modal by pressing the Escape key. You can explore more options and configurations in the Bootstrap documentation for modals. Remember to include the necessary Bootstrap and jQuery files, and ensure that the modals are visually accessible and responsive on different devices.

H. Program Code & Output Source code:

SOURCE CODE

OUTPUT

Signature

Practical No.15: Design news story page to demonstrate usage of Scrollspy for multiple section, Tooltip for different photos, Collapsible and popover plugins of bootstrap.

A. Objective:

Use of plug-inslike Scrollspy, Tooltip, Collapsible and Popover to design attractive web page for different websites.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the *engineering* problems.
2. **Problem analysis**: Identify and analyse well-defined *engineering* problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
5. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes *in field of engineering.*

C. Expected Skills to be developed based on competency:

Students will design web pages using Scrollspy, Tooltip, Collapsible and Popover plug-in for modern websites.

D. Expected Course Outcomes(Cos)

Develop interactive features rich web pages using Bootstrap jQuery plug-ins.

E. Practical Outcome(PRo)

Design responsive web page using Scrollspy, Tooltip, Collapsible and Popover plug-ins.

F. Expected Affective domain Outcome(ADos)

1. Follow Coding standards and practices.
2. Maintain tools and equipment.
3. Follow safety practices.
4. Follow ethical practices

G. **Prerequisite Theory:** Scrollspy is a Bootstrap plugin that allows you to automatically update navigation links based on the scroll position of the page. It enhances the user experience by indicating the current active section or item in the navigation menu as the user scrolls through different sections of a webpage.

Scrollspy works by assigning an active class to the navigation item corresponding to the currently visible section. Here's a detailed breakdown of how Scrollspy works:

Enabling Scrollspy: To use Scrollspy, you need to enable it on the parent element that contains your navigation links. This is typically the <body> element or a container element. You can enable Scrollspy by adding the data-spy="scroll" attribute to the element.

Example:

```
        <body data-bs-spy="scroll">
```

Setting the Target: Scrollspy needs to know which navigation links to track and update. You specify the target navigation by adding the data-target attribute to the Scrollspy-enabled element. The target is typically the ID or class of the navigation menu element.

Example:

```
        <<body data-bs-spy="scroll" data-bs-offset="15" data-bs-
target="#myScrollspy">
```

Creating the Navigation: Next, you need to create a navigation menu with appropriate links that correspond to the sections on your page. Each navigation item should have an href attribute pointing to the corresponding section's ID.

Example:

```
<div class="container-lg my-2">
<h1>Bootstrap Scrollspy</h1>
        <div class="row">
        <div class="col-sm-3" id="myScrollspy">
        <div class="list-group">
        <a class="list-group-item list-group-item-action active"
        href="#section1">Section One</a>
        <a class="list-group-item list-group-item-action"
        href="#section2">Section Two</a>
        <a class="list-group-item list-group-item-action"
        href="#section3">Section Three</a>
        </div></div>
```

Assigning IDs to Sections: Each section on your page that you want to be tracked by Scrollspy should have a unique ID. These IDs should match the href values in the navigation links.

Example:

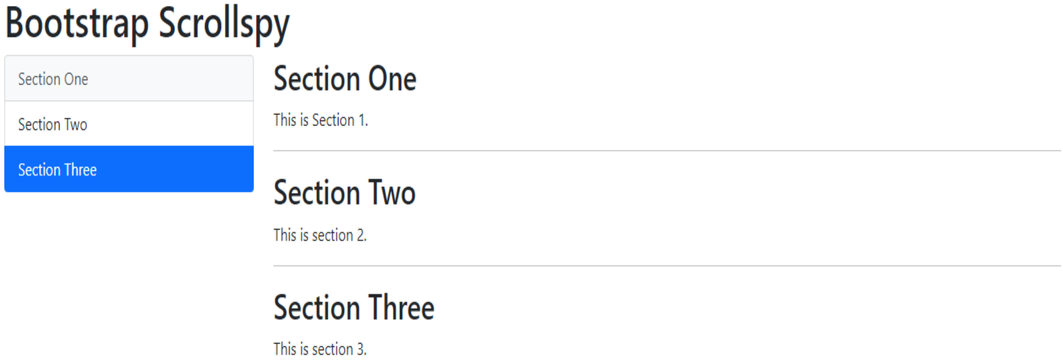
```
<div class="col-sm-9">
        <div id="section1">
                <h2>Section One</h2>
                <p>This is Section 1.</p></div>
        <hr><div id="section2">
                <h2>Section Two</h2>
                <p>This is section 2.</p></div>
        <hr><div id="section3">
                <h2>Section Three</h2>
                <p>This is section 3.</p></div>
 </div>
```

Finally, you need to include the Bootstrap JavaScript file that contains the Scrollspy plugin. You also need to include the jQuery library, as Scrollspy depends on it.

Example:

```
<script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.1.0/js/bootstrap.bundle.min.js"></script>
```

Output:



Bootstrap Scrollspy

Section One

This is Section 1.

Section Two

This is section 2.

Section Three

This is section 3.

With these steps in place, Scrollspy will automatically track the scroll position of the page and update the active navigation item as you scroll. The active class will be applied to the corresponding navigation item, indicating the currently visible section.

Note: Ensure that you have properly included the Bootstrap CSS file for the desired styling of the navigation menu and sections.

Tooltip: The Tooltip plugin in Bootstrap allows you to add informative tooltips to elements on your web page. Tooltips are small pop-up boxes that provide additional context or descriptions when the user hovers over or clicks on an element. They are commonly used to enhance the user experience by providing extra information about buttons, icons, or other interactive elements.

To use the Tooltip plugin in Bootstrap, you need to include the Bootstrap CSS and JavaScript files, as well as the jQuery library, in your HTML document. Here's an overview of how to use tooltips:

HTML Markup: To create a tooltip, you need to add specific data attributes to the element you want to apply the tooltip to. Typically, you use the data-toggle and title attributes.

Example:

```
<button type="button" class="btn" data-toggle="tooltip" title="This is a tooltip">Hover me</button>
```

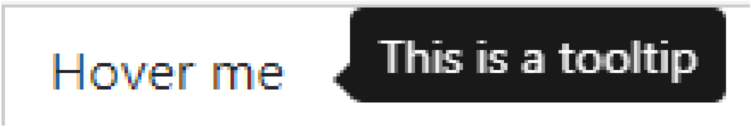
In this example, we have a <button> element with the class "btn". The data-toggle="tooltip" attribute indicates that this element will trigger a tooltip. The title attribute specifies the text that will appear in the tooltip.

Initializing Tooltips: To activate the Tooltip plugin, you need to initialize it using JavaScript. You can achieve this with the following code:

Example:

```
$(document).ready(function() {
  $('[data-bs-toggle="tooltip"]').tooltip();
});
```

Output:



In this code, we use the **$(document).ready()** function to ensure that the initialization code runs once the document has finished loading. The **$('[data-toggle="tooltip"]').tooltip();** line selects all elements with the **data-toggle="tooltip"** attribute and initializes the tooltip plugin for them.

Options and Customization: The Tooltip plugin offers various options to customize the appearance and behavior of tooltips. You can pass an options object to the tooltip() function to modify settings such as placement, delay, and triggering events.

Example:

```
$(document).ready(function() {
  $('[data-toggle="tooltip"]').tooltip({
placement: 'right',
delay: { show: 500, hide: 100 },
trigger: 'hover'
  });
});
```

In this example, we customize the tooltip's placement by setting placement: 'right', which displays the tooltip on the right side of the element. The delay option specifies the delay time (in milliseconds) for showing and hiding the tooltip. Finally, the trigger option is set to 'hover', meaning the tooltip appears when the user hovers over the element.These are just a few examples of how you can customize tooltips. You can explore more options in the Bootstrap documentation.Remember to include the necessary Bootstrap CSS and JavaScript files, as well as the jQuery library, before using the Tooltip plugin.

Collapsible plugin: The Collapsible plugin in Bootstrap allows you to create collapsible content sections that expand or collapse when triggered. It's commonly used for accordions, collapsible panels, and toggleable content.To use the Collapsible plugin in Bootstrap, you need to include the Bootstrap CSS and JavaScript files, as well as the jQuery library, in your HTML file.Here's how you can create a collapsible section using the Collapsible plugin.HTML Structure:The basic structure of a Bootstrap Collapse consists of two main components.

Trigger element (usually a clickable button or link): This element is used to initiate the collapse action.

Content element: This is the container that holds the collapsible content.

Data Attributes: Bootstrap Collapse uses data attributes to define the relationship between the trigger and content elements. The critical data attributes are:

data-toggle: This attribute defines the type of Bootstrap plugin to activate. For Collapse, it should be set to "collapse".

data-target: This attribute specifies the CSS selector of the content element that needs to be collapsed or expanded.

aria-expanded: This attribute is used for accessibility purposes to indicate whether the content is currently expanded or collapsed. It should have a value of "true" when expanded and "false" when collapsed.

aria-controls: This attribute links the trigger element to the content element by referencing the content element's ID.

Event Handling: When you click on the trigger element, Bootstrap's JavaScript code comes into play. It captures the click event and then checks the data-target attribute to find the corresponding content element that needs to be collapsed or expanded.

Collapsing and Expanding: Once the content element is identified, Bootstrap applies the necessary CSS classes and styles to perform the collapse or expand action. The primary CSS classes used are:

.collapse: This class is added to the content element that you want to make collapsible.

.collapse.show: This class is added to the content element when it is expanded, making it visible.

.collapsing: This class is temporarily added during the animation of the collapse or expand action. It helps to apply smooth transitions.

.collapsed: This class is added to the trigger element to indicate that the associated content is collapsed.

CSS Transition and Animation: Bootstrap utilizes CSS transitions to create a smooth animation effect when collapsing or expanding content. The transition properties are applied to the .collapse class, defining the duration and easing function for the animation.

Accessibility: Bootstrap also considers accessibility in the Collapse plugin implementation. The aria-expanded attribute is updated dynamically to ensure that screen readers can convey the correct state of the collapsible content.

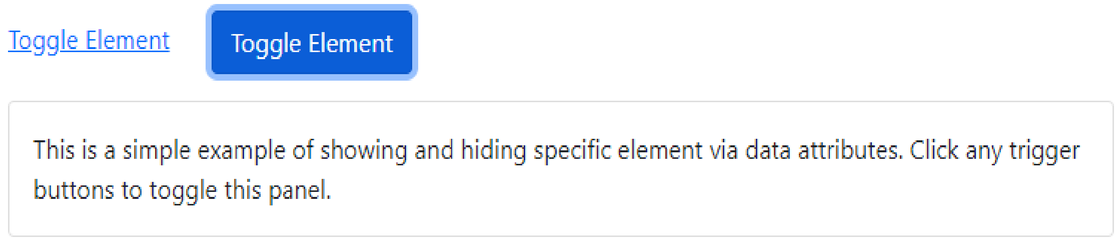
By following this theory, Bootstrap creates a functional and interactive Collapse component that helps improve the user experience by organizing content in a more structured and manageable manner.

Example:

```
<div class="m-4">
<p>
<!-- Trigger Buttons HTML -->
<a href="#myCollapse" data-bs-toggle="collapse">Toggle Element</a>
<button type="button" class="btn btn-primary ms-4" data-bs-toggle="collapse" data-bs-target="#myCollapse">Toggle Element</button>
</p>
```

```
<!-- Collapsible Element HTML -->
<div class="collapse show" id="myCollapse">
<div class="card card-body">This is a simple example of showing and hiding specific element via data attributes. Click any trigger buttons to toggle this panel.</div>
</div>
</div>
```

Output:



This is a simple example of showing and hiding specific element via data attributes. Click any trigger buttons to toggle this panel.

Collapsible sections can also be triggered programmatically using JavaScript/jQuery methods like **.collapse('show'), .collapse('hide'),** and **.collapse('toggle').**Remember to include the necessary Bootstrap and jQuery files, and ensure the collapsible sections are visually accessible and responsive on different devices.

Sure, here is a detailed theory for the Popover plugin in Bootstrap:

Popover: A popover is a small, informative window that appears when a user hovers over an element. It is similar to a tooltip, but it can contain more content. Popovers are often used to provide additional information about an element, such as a definition, a link, or a list of options.

The Popover plugin is a JavaScript library that is included with Bootstrap. It uses the Popper.js library to position the popover relative to the element that it is attached to. The plugin also provides a number of options that can be used to customize the appearance and behavior of the popover.

To create a popover, you need to add the **`data-toggle="popover"`** attribute to the element that you want the popover to be attached to. You can also specify the title and content of the popover using the `title` and `content` attributes, respectively.

For example, the following code would create a popover with the title "Popover title" and the content "This is the content of the popover":

html

```
<button data-toggle="popover" title="Popover title" content="This is the content of the popover">
  Click me to see the popover
</button>
```

The Popover plugin provides a number of options that can be used to customize the appearance and behavior of the popover. These options include:

* **placement:** The position of the popover relative to the element that it is attached to.

* **trigger:** The event that triggers the popover. The default trigger is hover, but you can also use click or focus.

* **content:** The content of the popover. This can be a string, a jQuery selector, or a function.

* **title:** The title of the popover.

* **delay:** The delay in milliseconds before the popover is shown.

* **container:** The container that the popover is displayed in. The default container is the body element, but you can also specify a different container.

Examples

Here are some examples of how to use the Popover plugin:

* To create a popover that appears when the user hovers over an element, you would use the hover trigger.

* To create a popover that appears when the user clicks on an element, you would use the click trigger.

* To create a popover that appears when the user focuses on an element, you would use the focus trigger.

* To create a popover that appears in the top right corner of an element, you would use the top right placement option.

* To create a popover that has a delay of 500 milliseconds before it is shown, you would use the delay option with a value of 500.

* To create a popover that is displayed in a specific container, you would use the container option with the selector for the container element.

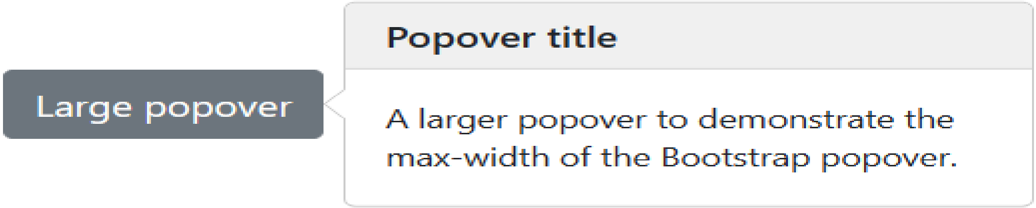
The Popover plugin is a powerful tool that can be used to add informative windows to your Bootstrap applications. It is easy to use and customize, and it provides a number of options that can be used to control the appearance and behavior of the popover.

Example:

```
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Popovers</title>
<link href="css/bootstrap.min.css" rel="stylesheet">
<script src="jquery.min.js"></script>
<script src="js/bootstrap.bundle.min.js"></script>
<style>
        .bs-example{
        margin: 100px;
```

```
   }
</style>
<script>
$(document).ready(function(){
        $('[data-bs-toggle="popover"]').popover();
});
</script></head>
<body>
<div class="bs-example">
<p><button type="button" class="btn btn-secondary ms-2" data-bs-toggle="popover" title="Popover title" data-bs-content="A larger popover to demonstrate the max-width of the Bootstrap popover.">Large popover</button></p>
</div></body>
</html>
```

Output:



Large popover

Popover title

A larger popover to demonstrate the max-width of the Bootstrap popover.

H. Program Code & Output Source code:

SOURCE CODE

OUTPUT

Signature

Practical No.16: Design animated photo gallery page using Carousel bootstrap plug-in with minimum seven photos.

A. Objective:

Use carousel plug-in to design attractive home page for different websites.

B. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the *engineering* problems.
2. **Problem analysis**: Identify and analyse well-defined *engineering* problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
5. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes *in field of engineering.*

C. Expected Skills to be developed based on competency:

Students will enhance their creativity by designing animated pages using carousel plug-in for modern websites.

D. Expected Course Outcomes(Cos)

Develop interactive features rich web pages using Bootstrap jQuery plug-ins..

E. Practical Outcome(PRo)

Design animated slider using carousel plug-in from bootstrap responsive framework.

F. Expected Affective domain Outcome(ADos)

5. Follow Coding standards and practices.
6. Maintain tools and equipment.
7. Follow safety practices.
8. Follow ethical practices

G. Prerequisite Theory: The Carousel Bootstrap plug-in is a popular feature in the Bootstrap framework that allows you to create image sliders or carousels on your website. It provides a user-friendly way to showcase multiple images or content in an interactive and visually appealing manner. In this response, I'll provide a detailed theory of how the Carousel plug-in works and provide an example to illustrate its usage.

The Carousel plug-in utilizes HTML, CSS, and JavaScript to create the carousel functionality. The core concept involves a container element that holds a set of slides, with one slide being displayed at a time. The user can navigate through the slides using navigation controls such as next and previous buttons or by utilizing indicators.

Here are the key components and their roles within the Carousel plug-in:

Carousel Container: This is an HTML element, usually a <div>, that wraps the entire carousel. It acts as a parent container for the carousel and provides a reference point for styling and JavaScript initialization.

Carousel Slides and captions: The slides are specified in a <div> with class **.carousel-inner**. The content of each slide is defined in a <div> with class **.item**. This can be text or images. The .active class needs to be added to one of the slides. Otherwise, the carousel will not be visible.

These are individual elements, typically <div> or <img>tags, that represent the content you want to display. Each slide can contain images, text, or any other HTML content. The Carousel plug-in automatically transitions between these slides. Below is example of carousel slide.

```
<div class="carousel-inner">
<div class="carousel-item active">
<img src="slide1.jpg"  alt="Slide 1">
<div class="carousel-caption">
<h3>Slide 1</h3>
<p>First slide description.</p>
</div>
</div>
```

Navigation Controls: Bootstrap provides built-in navigation controls, such as next and previous buttons, to navigate through the carousel slides. These buttons trigger the transitions to move between the slides. Below is example of carousel navigation controls.

```
<!-- Navigation Controls -->
<a class="carousel-control-prev" href="#myCarousel"
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="#myCarousel"
role="button" data-slide="next">
<span class="carousel-control-next-icon" aria-
hidden="true"></span>
<span class="sr-only">Next</span>
</a>
```

Indicators: The indicators are specified in an ordered list with class **.carousel-indicators**. The **data-target** attributes points to the id of the carousel. The **data-slide-to** attribute specifies which slide to go to, when clicking on the specific dot.

Indicators are small dots or icons that represent each slide within the carousel. They provide visual feedback to the user about the current slide and allow them to jump directly to a specific slide. Bootstrap's Carousel plug-in generates these indicators automatically based on the number of slides. Below is examples of indicators.

```
<!-- Indicators -->
<ol class="carousel-indicators">
<li data-target="#myCarousel" data-slide-to="0"
class="active"></li>
<li data-target="#myCarousel" data-slide-to="1"></li>
<li data-target="#myCarousel" data-slide-to="2"></li>
</ol>
```

JavaScript Initialization: To enable the Carousel plug-in, you need to initialize it using JavaScript. This involves selecting the carousel container element and calling the appropriate Bootstrap functions with the desired options.

Example:

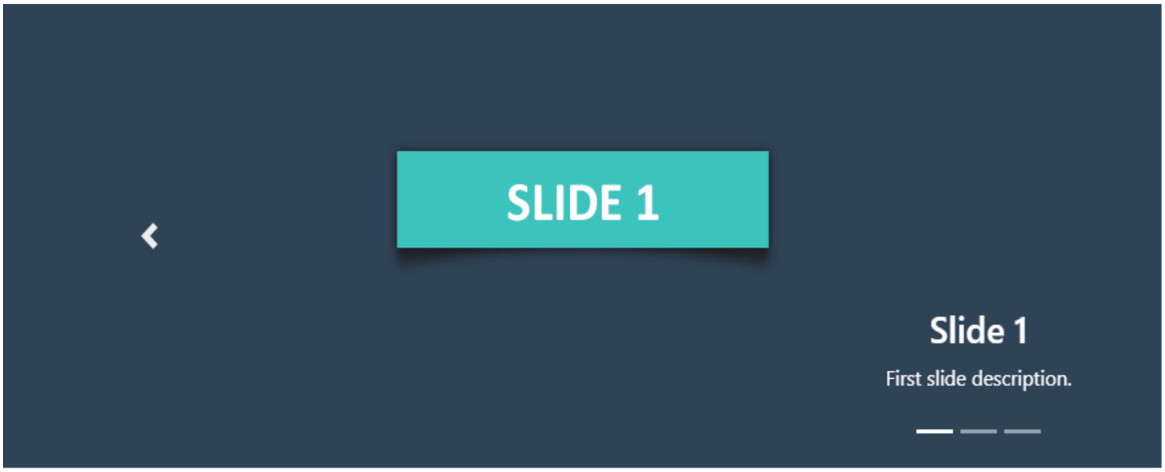
Let's consider a simple example to illustrate the Carousel plug-in:

```
<html>
<head>
<title>Carousel Example</title>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.cs
s">
</head>
<body>
```

```
<div id="myCarousel" class="carousel slide" data-ride="carousel">
<!-- Indicators -->
<ol class="carousel-indicators">
<li data-target="#myCarousel" data-slide-to="0" class="active"></li>
<li data-target="#myCarousel" data-slide-to="1"></li>
<li data-target="#myCarousel" data-slide-to="2"></li>
</ol>

<!-- Slides -->
<div class="carousel-inner">
<div class="carousel-item active">
<img src="slide1.jpg"  alt="Slide 1">
<div class="carousel-caption">
<h3>Slide 1</h3>
<p>First slide description.</p>
</div></div>
<div class="carousel-item">
<img src="slide2.jpg"  alt="Slide 2">
<div class="carousel-caption">
<h3>Slide 2</h3>
<p>Second slide description.</p>
</div></div>
<div class="carousel-item">
<img src="slide3.jpg"   alt="Slide 3">
<div class="carousel-caption">
<h3>Slide 3</h3>
<p>Third slide description.</p>
</div></div></div>
<!-- Navigation Controls -->
<a class="carousel-control-prev" href="#myCarousel" 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="#myCarousel" role="button" data-slide="next">
<span class="carousel-control-next-icon" aria-hidden="true"></span>
<span class="sr-only">Next</span>
</a>
</div>
```

```
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.3/dist/umd/popper.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js">
</script>
</body>
</html>
```

Output:



SLIDE 1

Slide 1

First slide description.

H. Program Code & Output Source code:

SOURCE CODE

OUTPUT

Signature