GUJARAT TECHNOLOGICAL UNIVERSITY, AHMEDABAD

On

"Web Designing"

Subject Code: 3170001

BECHELOR OF ENGINEERING

In

Computer Engineering

Vidhyadeep Institute of Engineering and Technology

Computer Engineering Department

August 2023

Web Designing

An INTERNSHIP REPORT

Submitted by

Rajput Manisha M

210940107537

Under subject of

SUMMER INTERNSHIP

B.E. IV, Semester– VII

in

Computer Engineering Department

Vidhyadeep Institute of Engineering & Technology, Anita, Kim

Gujarat Technological University, Ahmedabad

[August, 2023]

CERTIFICATE

This is to certify that the Summer Internship report submitted along with the project entitled **"Web Designing with React Js"** has been carried out by **Rajput Manisha** under my guidance in the subject of Summer Internship in Electrical Engineering, 7th Semester of Gujarat Technological University, Ahmedabad during the academic year 2023-24.

Internal Guide	Head of the Department
Dr. Sushma Ghode	Dr. Sushma Ghode
Sign	Sign

JOINING LETTER

Re : **Rajput Manisha Munnasingh** (210940107537)

Vidhyadeep Institute Of Engineering And Technology

We are very pleased to confirm your acceptance of an internship into **Web Design With React JS** with BrainyBeam Info-Tech Pvt. Ltd. Your duties and assignments for this position are as follows: Working with different Modules of respective technology

Your first day of work will be 27th Jul-2023. You will work for 6 days of a week. You will be reporting to your respective Developer.

Official location: You will be working in our branch office at **2nd Floor, Dhanlaxmi Chembers, Near Gujarat Vidhyapith, Ashram Road, Ahmedabad**

If you have any questions, please feel free to contact Miss Vrunda Patel. We are pleased you have decided to join BrainyBeam Info-Tech Pvt. Ltd.

BRAINYBEAM INFO-TECH PVT. LTD.
AHMEDABAD

With Best Regards,
Mr. Sagar Jasani
CEO.
BrainyBeam Info-Tech Pvt. Ltd
Ahmedabad (Gujarat-INDIA)

+91 90332 37336 sagar@brainybeaminfotech.com

COMPLETION CERTIFICATE

27th Jul-2023

TO WHOM IT MAY CONCERN

This is to certify that, Mr./Ms. **Rajput Manisha Munnasingh (210940107537)** Student of **Vidhyadeep Institute Of Engineering And Technology**, has successfully completed a 15 Days Internship in the field of **Web Design With React JS** during the period of **27th Jul-2023 to 10th Aug-2023**.

During the period of his internship program with us, he had been exposed to different processes and was found sincere and hardworking.

BRAINYBEAM INFO-TECH PVT. LTD.
AHMEDABAD

With Best Regards,
Mr. Sagar Jasani
CEO.
BrainyBeam Info-Tech Pvt. Ltd
Ahmedabad (Gujarat-INDIA)

Vidhyadeep Institute of Engineering & Technology

Anita, Kim

DECLARATION

I hereby declare that the Summer Internship report submitted along with the Summer Internship entitled **Web Development** submitted in partial fulfilment for the subject Summer Internship in Electrical Engineering to Gujarat Technological University, Ahmedabad, is a bonfire record of original project work carried out by me at **BrainyBeam Info-Tech Pvt. Ltd** under the supervision of **Dr.Sushma Ghode HOD** and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

Name of the Student _____

Sign of Student _____

ACKNOWELDGEMENT

I am greatly thankful to "**Vidhyadeep Institute of Engineering and Technology"** for providing us a platform that enhanced our skill and "**Gujarat Technological University"** to allow us to represent our skill and interest to the field of Computer Engineering world.

I would also like to thank **Mr. Sagar Jaini at BrainBeam PVT LTD,** for giving me the opportunity to work on the prestigious project. The internship opportunity I had with **BrainyBeam** Infotech was a great chance for learning and professional development.

I express my deepest thanks to place to my Guide **Dr.Sushma Ghode**, **Head of department**, Computer Engineering Department, VIEAT, Anita, Kim for taking part in useful decision & giving necessary advices and guidance, motivation, constant inspiration and above all for her ever-co-operating attitude that enabled us in bringing up this project in the present form.

I would like to thank **Mrs. Sushma Ghode Head of the Computer Engineering Department**, for his valuable advice and providing necessary facilities for our project work.

I perceive as this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way and I will continue to work on my improvement, in order to desired career objectives.

Name	**Enrollment Number**
Rajput Manisha	210940107537

EXECUTIVE SUMMARY

The 2-week Summer Internship Program at **BrainyBeam** Company for Multistory building is a vital part of the 1-year BE Computer Engineering Course. Since majority of the students come without any prior work experience, the Summer Internship adds worth to their CVs by giving each student immense learning. At this Company, a student can bag an internship through various means since the company gives us ample opportunities to interact with industry experts.

ABOUT THE COMPANY

At BrainyBeam Info-Tech, we see Innovation as a clear differentiator. Innovation, along with focus on deep, long-lasting client relationships and strong domain expertise, drives every facet of our day-to-day operations. BrainyBeam Info-Tech Info-Tech was founded with a vision to address growing businesses' needs of reducing the time to market and cost effectiveness required to develop and maintain unique and customized web and mobile solutions. We are uniquely and strategically positioned to partner with startups and leading brands to help them expand their business and offer the most effective and cost-efficient solutions that provide revenues and value to their business needs. All processes are well documented, that enable us to deliver more effective and innovative solutions with each project we work on. We provide services that conform to customers' requirements, at a satisfactory overall cost, having regard to functionality, reliability, durability, usability, appearance and safety.

TABLE OF CONTENTS

LIST OF FIGURES

CONTENTS

DAY-1

Introduction to web development:

Web development is the process of creating and building websites and web applications that are accessible via the internet. It encompasses a broad range of skills and technologies, combining both front-end and back-end development

Front-end Development:

Also known as client-side development, front-end development focuses on the visual and interactive aspects of a website. Front-end developers use following technologies such as HTML, CSS, PHP, Bootstrap, React, Django, Tailwind CSS, Flask etc... to create the structure, layout, and functionality that users interact with directly in their web browsers.

Back-end Development:

Back-end development, also referred to as server-side development, deals with the behind-the-scenes logic and operations of a website. Back-end developers work with server-side languages like PHP, Python, Ruby, or JavaScript (Node.js) to manage databases, handle user authentication, and process data, ensuring that the web application functions smoothly.

- backend: functionality ex. PHP, Python, Java, JavaScript, NodeJS
- database: store data ex. MySQL, Oracle, MariaDB, Mongo dB, Firebase

Introduction to HTML:

HTML (Hypertext Markup Language) is the fundamental building block of the World Wide Web. It is a standard markup language used to create and structure the content of web pages. Developed by Tim Berners-Lee in 1991, HTML allows web developers to define the elements and layout of a webpage, making it accessible and readable by web browsers.

Structure of HTML:

HTML is comprised of elements, represented by tags, which define the structure and content of a web page. Each HTML document starts with a <!DOCTYPE> declaration that specifies the version of HTML being used. The basic structure of an HTML document is as follows:

```
<!DOCTYPE html>
<html>
```

```
<head>
```

```
<!-- Metadata and title of the page -->
```

```
</head>

<body>

<! -- Content visible on the webpage -->

</body>
</html>
```

- The <!DOCTYPE html> declaration informs the browser that the document is written in HTML5, the latest version of HTML
- The <html> element is the root element of the HTML document and contains all the other elements
- The <head> element contains meta-information about the page, such as the title, links
- The <body> element contains the visible content of the webpage, including headings, paragraphs, images, links, and other elements

t

ernal stylesheets, or scripts.

HTML Elements and Tags:

HTML elements are represented by tags, which are enclosed in angle brackets < >. Tags come in pairs, with an opening tag to define the beginning of an element and a closing tag to define the end. The content to be displayed on the webpage is placed between the opening and closing tags.

For example, a basic paragraph element is defined using the <p> tag:

```
<p>This is a paragraph. </p>
```

Basic HTML tags:

- <p> tag

The <p> tag defines a paragraph.

Browsers automatically add a single blank line before and after each <p> element. - <pre> tag

The <pre> tag defines preformatted text.

Text in a <pre> element is displayed in a fixed-width font, and the text preserves both spaces and line breaks. The text will be displayed exactly as written in the HTML source code.

heading:

HTML headings are titles or subtitles that you want to display on a webpage.

list in HTML

HTML lists allow web developers to group a set of related items in lists.

Unordered HTML List: An unordered list starts with the <ul> tag. Each list item starts with the <li> tag. The list items will be marked with bullets (small black circles) by default.

Ordered HTML List: An ordered list starts with the <ol> tag. Each list item starts with the <li> tag. The list items will be marked with numbers by default.

HTML Description Lists: HTML also supports description lists. A description list is a list of terms, with a description of each term. The <dl> tag defines the
description list, the <dt> tag defines the term (name), and the <dd> tag describes each term:

```
<h2>An Unordered HTML List</h2>

<ul>

<li>Coffee</li><li>Tea</li>

<li>Milk</li>

</ul>

<h2>An Ordered HTML List</h2>

<ol>

<li>Coffee</li><li>Tea</li>

<li>Milk</li>

</ol>
```

```
<h2>A Description List</h2>
<dl>
<dt>Coffee</dt>
<dd>- black hot drink</dd>
<dt>Milk</dt>
<dd>- white cold drink</dd>
```

HTML Attributes:

HTML attributes provide additional information about HTML elements.

For example:

href in <a> tag, src in <img> tag,

style attribute for inline css, class,

id, name etc...

Block element vs inline element:

Block-level Elements

A block-level element always starts on a new line, and the browsers automatically add some space (a margin) before and after the element.

A block-level element always takes up the full width available (stretches out to the left and right as far as it can).

For example, <p>, <h>, are block elements.

Inline Elements

An inline element does not start on a new line.

An inline element only takes up as much width as necessary.

This is an inline element inside a paragraph.

DAY-2

Basic introduction of css:

CSS, which stands for Cascading Style Sheets, is a fundamental technology used in web development to control the presentation and layout of HTML documents. It is a stylesheet language that describes how HTML elements should be displayed on a web page or in other media.

We can implement css in 3 ways:

1 inline: within the opening tags of the element. If we use inline css and also use external or internal than first priority is given to inline css.

```
<h1 style="color: blue; text-align: center;">This is a heading</h1>
```

2 internal: style tag within the head tag. If we use internal and external css than first priority is given to internal css.

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
   <style>
      h1 {
         color: maroon;
         margin-left: 40px;
      }
   </style>

</head>

<body>
   <h1>This is a heading</h1>
</body>
</html>
```

3. external: make an external css file and link it.

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

   <link rel="stylesheet" href="my style.css">
</head>

<body>
   <h1>This is a heading</h1>
   <p> This is a paragraph. </p>
</body>
</html>
```

mystyle.css

```
body {
  background-color:lightblue;
}

h1 {
  color: navy;
  margin-left: 20px;
}
```

CSS Selectors:

1 **Id:**

The id selector uses the id attribute of an HTML element to select a specific element. The id of an element is unique within a page, so the id selector is used to select one unique element. To select an element with a specific id, write a hash (#) character, followed by the id of the element.

2 **Class:**

The class selector selects HTML elements with a specific class attribute. To select elements with a specific class, write a period (.) character, followed by the class name.

3 **Tag:**

The element selector selects HTML elements based on the element name.

4 **Universal:**

The universal selector (*) selects all HTML elements on the page.

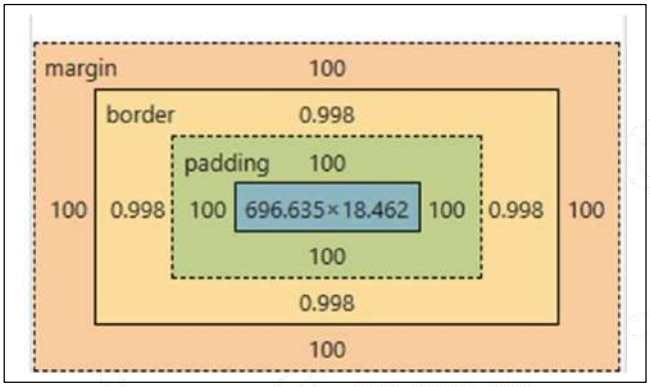
5 **Group:**

The grouping selector selects all the HTML elements with the same style definitions.

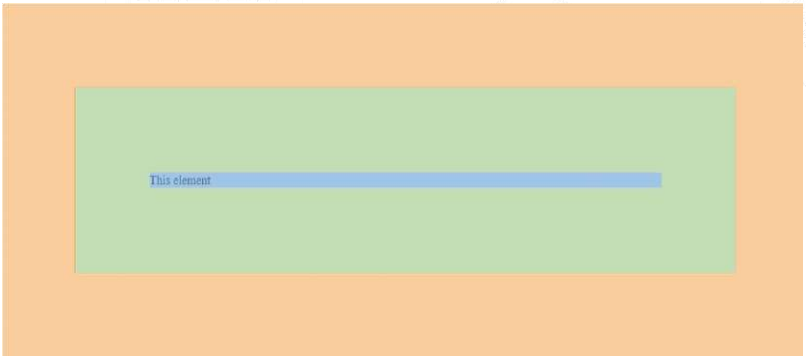
Padding: The CSS padding properties are used to generate space around an element's content, inside of any defined borders. There are properties for setting the padding for each side of an element (top, right, bottom, and left).

Margin: The CSS margin properties are used to create space around elements, outside of any defined borders. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

Example:




```
<head> <style> div { margin: 100px; padding:100px; border: 1px solid #4CAF50;
}</style></head>
<body>
<div>This element</div>
</body>
```



CSS box-sizing property:

The box-sizing property allows us to include the padding and border in an element's total width and height. If you set box-sizing: border-box; on an element, padding and border are included in the width and height.

DAY-3

CSS Properties:

Property	Use	Values
color	The color property specifies the color of text.	Color-name
background-color	The background-color property sets the background color of an element.	Color-name
font-size	The font-size property sets the size of a font.	Font-size in pixel
font-weight	The font-weight property sets how thick or thin characters in text should be displayed.	normal, bold, bolder, lighter, 100-900
font-family	The font-family property specifies the font for an element.	Name of font-family
text transform	The text-transform property controls the capitalization of text.	none, capitalize, uppercase , lowercase
text decoration	The text-decoration property specifies the decoration added to text, and is a shorthand property for:	• Text-decoration-line • text-decoration-color • text-decoration-style • text-decoration thickness

text-align	The text-align property specifies the horizontal	left, right,
	alignment of text in an element.	center, justify

Task1:

Aim: Create a card with image, title text and description using css.

Code: task1.html

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="./css/t
    <title>Document</title>
</head>
<body>
    <center>
        <div class="card">
            <img src="./image/img1.jp
            <h4>Title</h4>
            <p>This is description abo
chfaehhgeur23y23ugqhqh you can see
            <button>Click</button>
        </div>
    </center>
</body>
</html>
```

DAY-4

Background-image and it's properties:

The background-image property sets one or more background images for an element.

By default, a background-image is placed at the top-left corner of an element, and repeated both vertically and horizontally.

- background-size property:

  The background-size property specifies the size of the background images.

  Values:

  cover: Resize the background image to cover the entire container, even if it has to stretch the image or cut a little bit off one of the edges.

  contain: Resize the background image to make sure the image is fully visible.

  auto: Default value. The background image is displayed in its original size.

  length: Sets the width and height of the background image.

- background-repeat property:

  The background-repeat property sets if/how a background image will be repeated.

  Values:

  repeat: The background image is repeated both vertically and horizontally. repeat-x: The background image is repeated only horizontally. repeat-y: The background image is repeated only vertically.

  no-repeat: The background-image is not repeated.

- background-position property:

  The background-position property sets the starting position of a background image.

  Values: left top, left center, left bottom, right top, right center, center top, center center, center bottom

Float property:

The float property specifies whether an element should float to the left, right, or not at all.

Values: none, left, right

Clear property:

The clear property controls the flow next to floated elements.

The clear property specifies what should happen with the element that is next to a floating element.

Values: left, right, both.

Task1:

Aim: Create two rectangular cards with image, title text and description using css float property. Code: task2.html

```
<html lang="en">
<head>
<meta charset="UTF-8">
<link rel="stylesheet” href="./css/task2.css"/>
<title>Document</title>
</head>
<body>
<center>
<div class="container">
<div class="card">
<img src="./image/img1.jpg" alt="img">
<h4>Title</h4>
<p>This is description about above thing chfaehhgeur23y23ugqhqh you can see it in image above. </p>
<button>Click</button>
</div>
<div class="card">
<img src="./image/img1.jpg" alt="img">
<h4>Title</h4>
<p>This is description about above thing chfaehhgeur23y23ugqhqh you can see it in image above. </p>
<button>Click</button>
</div>
</div>
</center>
</body>
```

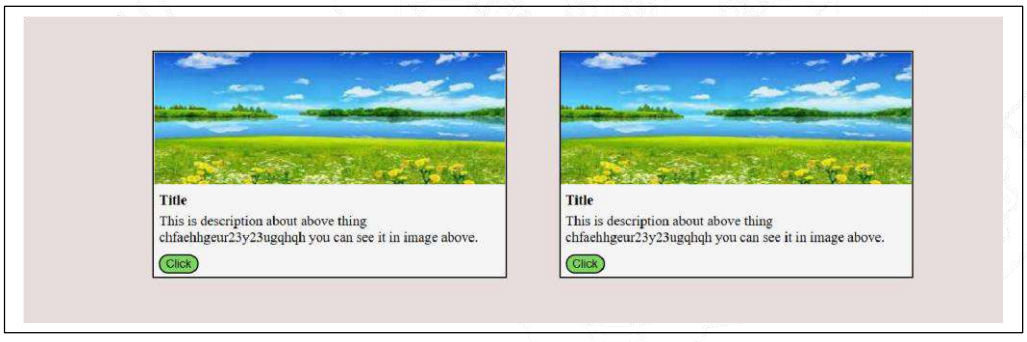
</html>

Code: task2.css

```
. card{  border:2px solid black;      float: left;
width:400px;
 height:255px  background-color:  WhiteSmoke;
margin:30px;    text-align: left;
}  img  {      height:150px;
width:400px;
}

button{          background-color:  rgb(126,  214,  97);
border-radius:15px
}
p,button,h4{     margin:5px;
}
.container{     margin:30px;     display:
inline-block;
} center{     background-color: rgb(231, 221, 221);
}
```

Output:



Task 2: create navbar with logo image and items using css float property.

Code: task3.html

```
<html lang="en">
<head>
<meta charset="UTF-8">
```

<title>Document</title>

<link rel="stylesheet" href=".\css\task3.css"></head>

<body>

<img src=".\image\img2tf.png" class="logo">

<li class="l-item">Python

<li class="l-item">SkLearn

<li class="l-item">Pandas

<li class="l-item">Keras

</body>

</html> Code: task3.css ul{ list-

style-type: none;}

li{ float: left;}

img. Logo{ height:55px;} .nav{ height: 55px;

background-color: rgb(0, 200, 255); color: white;

} .l-item{ margin: 15px;

font-size: 20px;} output:

Task 3: create card with left side image and right-side text using css float property.

Code: task4.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

```
<link rel="stylesheet"  href="./css/task4.css"/>
</head>
<body>
<div class="container">
<div id="p1">
<p>
Lorem ipsum dolor sit met connecter adipescent elite. Nula consequent dolorous rescinds labore, olio libero ad, suscept id quiz, Quique at qualm? Ex hic distinction nescient Omnis, fugit teeter beatae.
</p>
</div>
<div id="i1">
<img src="./image/img1.jpg" alt="img">
</div>
</div>
<div class="container">

<div id="i1">
<img src="./image/img1.jpg" alt="img">
</div>
<div id="p1">
<p>
Lorem ipsum dolor sit abet connecter adipescent elite. Nula consequent dolorous rescinds labore, olio libero ad, suscept id quiz, Quique at qualm? Ex hic distinction nescient omnis, fugit teeter beatae.
</p>
</div>
</div>
</body>
</html>
```

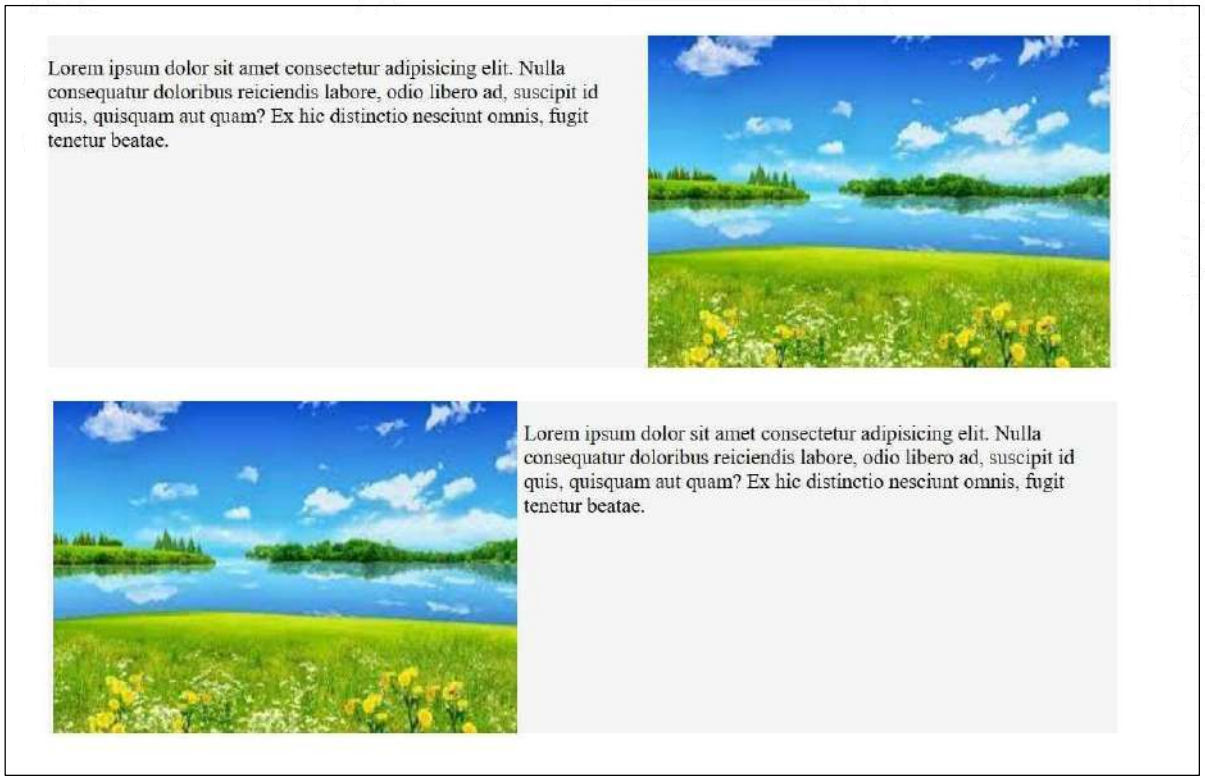
Code: task4.css

```
#p1{
float:  left;          height:  255px;
width: 455px;
```

```
}
#i1{
float: left;  height: 255px;
width:355px;  margin-left:
5px;  margin-right:5px;
}

img{  height: 255px;
width:355px;
}
.container{  display: block;  height: 255px;
width:820px;  margin:25px;  background-
color: WhiteSmoke;
}
```

Output:



Task 4: create nav-bar and container at left-side and another container on right side using float property.

Code: task5.html:

```
<html lang="en">
<head>
```

```
<meta charset="UTF-8">
<link href=".\css\task5.css" rel="stylesheet">
<title>Document</title>
</head> <body>
<div>
<ul class="nav">
<div class="nav-item"> <li > <img class="logo" src="https://picsum.photos/200"
alt="img"></li></div>
<div class="nav-item"> <li>Python</li></div>
<div class="nav-item"><li >JavaScript</li></div>
<div class="nav-item"><li >Java</li></div>
<div class="nav-item"><li >HTML</li></div>
<div class="nav-item"><li >C++</li></div>
</ul>
</div>
<div class="container1">
<ul class="ul2">
<div class="col"> <li>column1</li></div>
<div class="col"> <li>column2</li></div>
<div class="col"> <li>column3</li></div>
<div class="col"> <li>column4</li></div>
<div class="col"> <li>column5</li></div>
<div class="col"> <li>column6</li></div>
<div class="col"> <li>column7</li></div>
<div class="col"> <li>column8</li></div>
<div class="col"> <li>column9</li></div>
</ul>
</div>
<div class="container2">
<div class="p1">
<p>
```

Lorem ipsum dolor sit amet connecter adipescent elite. Nula consequent dolorous rescinds labore, olio libero ad, suscept id quiz, Quique at qualm? Ex hic distinction nescient Omni's, fugit teeter beatae.

Lorem ipsum dolor sit abet connecter adipescent elite. Nula consequent dolorous rescinds labore, olio libero ad, suscept id quiz, Quique at qualm? Ex hic distinction nescient Omni's, fugit teeter beatae.

Lorem ipsum dolor sit amet connecter adipescent elite. Nula consequent dolorous rescinds labore, olio libero ad, suscept id quiz, Quique at qualm? Ex hic distinction nescient Omni's, fugit teeter beatae.

Lorem ipsum dolor sit amet connecter adipescent elite. Nula consequent dolorous rescinds labore, olio libero ad, suscept id quiz, Quique at qualm? Ex hic distinction nescient Omni's, fugit teeter beatae.

</body>

</html>

Code: task5.css

.nav-item {

```
    float: left;
    margin: 15px;
}

.nav {
    list-style-type: none;
    display: block;
    height: 55px;
    background-color: black;
    color: white;
    padding-left: 2px;
    margin-left: 30%;
    margin-right: 30%;
    border: 2px solid white;
    border-radius: 15px;
}

.logo {
    height: 25px;
    width: 25px;
    border-radius: 50%;
}

.container1 {
    height: 100%;
    float: left;
    display: inline;
    background-color: WhiteSmoke;
}

.container2 {
    margin: 20px;
    height: 100%;
    width: 60%;
    border: 2px solid green;
    float: left;
    display: inline;
}

ul {
    list-style-type: none;
}

.ul2 {
    list-style-type: none;
    margin: 0px;
    padding: 0px;
}

.col {
    margin: 30px;
}

.p1 {
    display: inline;
```

```
}

p {
   margin: 25px;
```

output:


```
}
```

DAY-5

Position property:

The position property specifies the type of positioning method used for an element (static, relative, absolute, fixed).

static	Default value. Elements render in order, as they appear in the document flow.
absolute	The element is positioned relative to its first positioned (not static) ancestor element.
relative	The element is positioned relative to its normal position, so "left:20px" adds 20 pixels to the element's LEFT position.
fixed	The element is positioned relative to the browser window.

After position property we can apply top, left, bottom, right values negative or positive both.

Overflow property:

The overflow property specifies what should happen if content overflows an element's box.

Values:

hidden: The overflow is clipped, and the rest of the content will be invisible.

scroll: The overflow is clipped, but a scroll-bar is added to see the rest of the content.

Hover:

The: hover selector is used to select elements when you mouse over them.

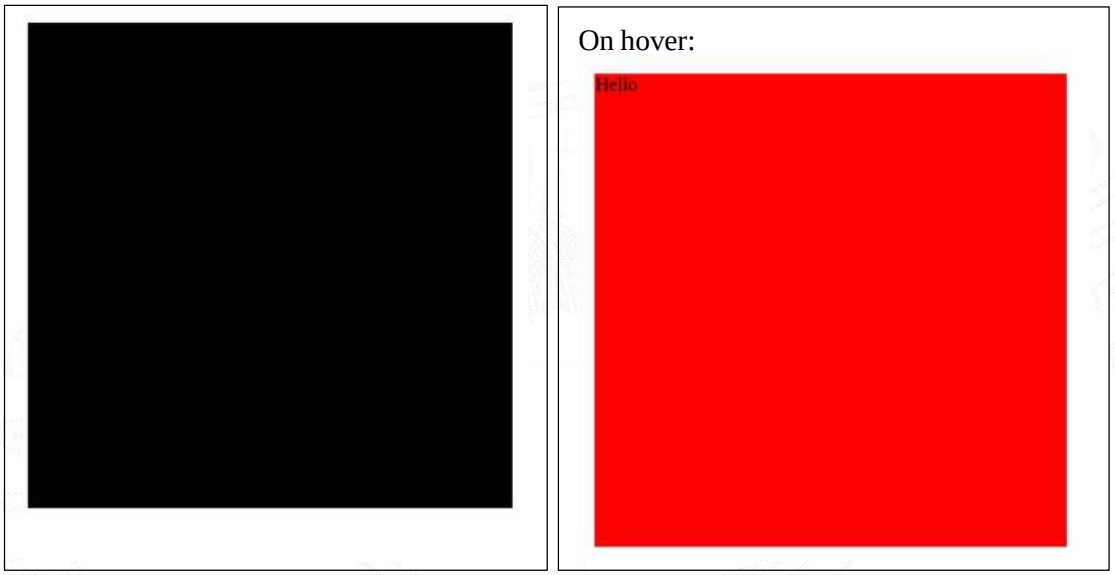
If we want to apply change on another element when hover on element than we can perform it by.

classmate: hover > .classmate {}

Example:

```
<!DOCTYPE html>
<html>
<head>
<style> .container{     margin:30px;
```

```
    background-color: black;
    display:inline-block;
    height:400px;
    width:400px;
}
.container:hover{
    background-color: red;}
</style>
</head>
<body>
<div class="container">Hello</div>
</body>
</html>
```



z-index:

When two or more containers are at same position or 1 container covers another container in that case we can assign priority that which element show above by using z-index. We can assign 1 or -1 to z-index.

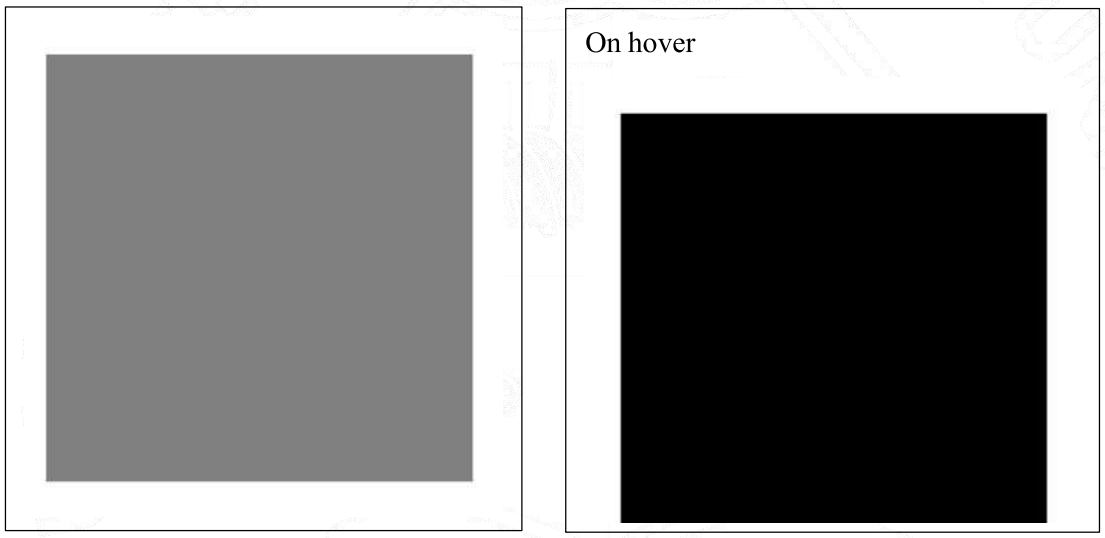
Opacity: we can give value between range 0-1.

```
<!DOCTYPE html>
<html>
<head>
<style> .container{ margin:30%; background-color: black; display:inline-block; height:400px; width:400px; opacity:0.5;
```

```
    }
    .container:hover{
        opacity:1;
    }
</style>
</head>
<body>
<div  class="container">

</div>
</body>
</html>
```

Output:



Task 1: create 3 cards with image and text on image using position property.

Code: task6.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Document</title>
<link rel="stylesheet" href=".\css\task6.css">
</head>
<body>
<div class="container">
<div class="card">
<div class="img1">
```

```
<img src="https://picsum.photos/200">
<div class="p1">
<h4>Title</h4>
<p>Lorem ipsum, dolor sits amet connecter adipescent elite.
Provident, facer. </p>
</div>
</div>
</div>
<div class="card">
<div class="img1">
<img src="https://picsum.photos/200">
<div class="p1">
<h4>Title</h4>
<p>Lorem ipsum, dolor sits amet connecter adipescent elite.
Provident, facers. </p>
</div>
</div>
</div>
<div class="card">
<div class="img1">
<img src="https://picsum.photos/200">
<div class="p1">
<h4>Title</h4>
<p>Lorem ipsum, dolor sits amet connecter adipescent elite.
Provident, facete. </p>
</div>
</div>
</div>
</div>
</body>
</html>
```

Code: task6.css.

```
card {
    height: 300px;
```

```
    width: 240px;
    float: left;
    margin: 11px;
}

.img1 {
    position: relative;
    width: 240px;
    height: 300px;
}

img {
    height: 300px;
    width: 240px;
}

.p1 {
    position: absolute;
    left: 0px;
    bottom: 0px;
    margin: 15px;
    font-weight: bold;
    color: WhiteSmoke;
}

.container {
    background-color: whitesmoke;
    height: 330px;
    padding: 25px;
}
```

output:

Task 2: create 3 cards with center image and text using position property.

Code: task7.html:

```
<html lang="en">

<head>

<meta charset="UTF-8">
```

```
<link rel="stylesheet" href=".\css\task7.css">
<title>Document</title>
</head>
<body>
<div class="container">
<div class="card">
<div class="img1">
<img src="https://picsum.photos/200">
</div>
<div class="p1">
<h4>Lorem, ipsum dolor. </h4>
<h4>Lorem. </h4>
<p>Lorem ipsum, dolor sits amet connecter adipescent elite.
Provident, faker0. </p>
</div>
</div>
<div class="card">
<div class="img1">
<img src="https://picsum.photos/200">
</div>
<div class="p1">
<h4>Lorem, ipsum dolor. </h4>
<h4>Lorem. </h4>
<p>Lorem ipsum, dolor sits amet connecter adipescent elite.
Provident, faker. </p>
</div>
</div>
<div class="card">
<div class="img1">
<img src="https://picsum.photos/200">
</div>
<div class="p1">
<h4>Lorem, ipsum dolor. </h4>
```

```
<h4>Lorem. </h4>

<p>Lorem ipsum, dolor sits amet connecter adipescent elite.
Provident, faker. </p>

</div>

</div>

</div>

</body>

</html>
```

Code: task7.css.

```
card {
   float: left;
   height: 250px;
   width: 300px;
   position: relative;
   border-radius: 15px;
   margin: 50px;
   border: 2px solid green yellow;
   background-color: whitesmoke;
}

.img1 {
   height: 70px;
   width: 70px;
   border: 2px solid green yellow;
   border-radius: 50%;
   position: absolute;
   top: -35px;
   left: 115px;
}

img {
   height: 70px;
   width: 70px;
   border-radius: 50%;
}

.p1 {
   position: absolute;
   top: 40px;
   left: 0px;
}

h4 {
   margin-left: 70px;
   margin-right: 70px;
}

p {
```

```
    margin: 25px;
}

.container {
    height: 400px;
    width: 100%;
    background-color: Alice blue;
    position: absolute;
    top: 150px;
    left: 0px;
    padding: 50px;
```

output:


```
}
```

Task 2:

Code: task8.html

```
<html lang="en">

<head>

<meta charset="UTF-8">

<title>Document</title>

<link href=".\css\task8.css" rel="stylesheet">

</head>

<body>

<div class="container2">

<div class="i2"><img src="https://picsum.photos/200" style=" height:700px;width:450px;border-radius:15px">

<div class="p1">Lorem ipsum, dolor sits amet connecter adipescent elite.
Provident, faker.

</div>

</div>

</div>
```

```
<div class="container">

<div class="container1">

<div class="i1"><img src="https://picsum.photos/200" style=" height:300px; width:800px; border-radius:15px">

<div class="p1">Lorem ipsum, dolor sits amet connecter adipescent elite.
Provident, faker.

</div>

</div>

</div>

<div class="container3">

<div class="i3"><img src="https://picsum.photos/200" style=" height:350px; width:350px; border-radius:15px">

<div class="p1">Lorem ipsum, dolor sits amet connecter adipescent elite.
Provident, faker.

</div>

</div>

</div>

<div class="container4">

<div class="i3"><img src="https://picsum.photos/200" style=" height:350px; width:350px;border-radius:15px">

<div class="p1">Lorem ipsum dolor sit amet, connecter adipescent elite. Saipem, enemy? </div>

</div>

</div>

</div>

</body>

</html>
```

Code:task8.css

```
.p1{

margin:20px;     color: WhiteSmoke;

position: absolute;     bottom:0px;

left:0px;} .i1{     height:300px;

width:800px;     position: relative;

border-radius: 15px;} .i2{

height:700px;     width:450px;

position: relative;     border-radius:
```

15px;} .i3{ height:350px;

width:350px; position: relative;

border-radius: 15px;} .container1{

position: absolute; top:0px;

left:0px;} .container2{ position:

absolute; top:10px;

right:100px;} .container3{

position: absolute; bottom:0px;

left:0px;} .container4{ position:

absolute; bottom:0px;

right:0px;} .container{ position:

relative; height: 700px;

width:800px;}

CSS Flexbox is a powerful layout model that allows you to create flexible and responsive designs for web pages. It provides an efficient way to distribute space, alignment, and positioning of elements within a container. The flex layout consists of a parent container (flex container) and its child elements (flex items). The flex container can be displayed horizontally or vertically, and it controls the layout of its flex items.

To use flexbox, you need to set the display property of the container to flex, and then use various flex-related properties to control the layout of the child elements.

- flex-direction:

The flex-direction property specifies the direction of the flexible items.

Values: row, column.

- justify-content:

  The justify-content property aligns the flexible container's items when the items do not use all available space on the main-axis (horizontally).

  Values:

  center: Items are positioned in the center of the container.

  space-between: Items will have space between them.

  space-around: Items will have space before, between, and after them.

  space-evenly: Items will have equal space around them.

- align-content:

  The align-content property specifies how flex lines are distributed along the cross axis in a flexbox container.

CSS Gradients:

- CSS Linear Gradients:

To create a linear gradient, you must define at least two-color stops. Color stops are the colors you want to render smooth transitions among. You can also set a starting point and a direction (or an angle) along with the gradient effect. Syntax:

background-image: linear- gradient (direction, color-stop1, color-stop2, ...);

- CSS Radial Gradients:

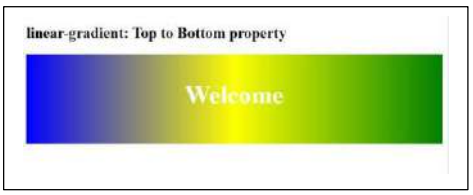
A radial gradient is defined by its center.

To create a radial gradient, you must also define at least two color stops. Syntax

background-image: radial-gradient (shape size at position, start-color, ..., last-color);

Example:

```
<!DOCTYPE html>
<html>
<head>
<title>linear-gradient function</title>
<style>
.gradient {            height: 100px;
background:              linear-gradient(to left
,green, yellow, blue);
Text-align: center;          padding-top: 40px;
font-size: 40px;          color: white;
font-weight: bold;
}
</style>
</head>
<body>
<h2>linear-gradient: Top to Bottom property</h2>
<div class="gradient">Welcome</div>
</body>
</html>
```



Task1: create a navbar using flex property that have logo and items and when hover on item sub-items dropdown display.

Code: task9.html

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link href=".\css\task9.css" rel="stylesheet">
    <title>Document</title>
</head>
<body>
    <div class="nav">
        <div class="logo">
        <img src="./image/img1.jpg" alt="img">
        </div>
        <div class="sub-nav">
            <div class="nav-items">Home</div>
```

```
        <div class="nav-items con">contact
            <div class="dd1">
                <ul>
                    <li class="di">contact1</li>
                    <li class="di sub2">contact2
                        <ul class="dd2">
                            <li class="di2">mobile</li>
                            <li
class="di2">WhatsApp</li>
                            <li class="di2">email</li>
                        </ul>
                    </li>
                    <li class="di">contact3</li>
                    <li class="di">contact4</li>
                </ul>
            </div>
        </div>
        <div class="nav-items">xyzzy</div>
        <div class="nav-items">ABC</div>
    </div>
</body>
</html>
```

Code:task9.css

```
* {
    margin: 0px;
    padding: 0px;
    list-style-type: none;
}

.nav {
```

```
    display: flex;
    justify-content: space-around;
    background-color: black;
    color: white;
    border: 2px solid white;
    width: 70%;
    border-radius: 20px;
    justify-content: space-between;
    align-items: center;
    margin-left: 100px;
}

.logo {
    margin-left: 20px;
    height: 55px;
    width: 55px;
}

img {
    height: 55px;
    width: 55px;
    border-radius: 50%;
}

.sub-nav {
    display: flex;
    align-items: center;
    justify-content: center;
    align-items: center;
}

.nav-items {
    margin-left: 10px;
```

```
    height: 40px;
    width: 60px;
}

. con {
    position: relative;
}

.dd1 {
    position: absolute;
    background-color: red;
    display: none;
    width: 100%;
}

.di {
    padding: 5px;
    opacity: 0.7;
    border: 1px solid black;
    box-sizing: border-box;
}

.Di: hover {
    padding: 0px;
    opacity: 1;
    box-sizing: border-box;
}

. Conover>.dd1 {
    display: block;
}

. sub2 {
```

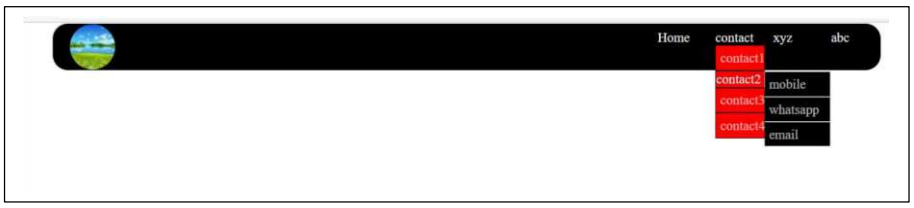
```
    position: relative;
}

.dd2 {
    display: none;
    height: 90px;
    width: 80px;
    background-color: black;
    color: white;
    position: absolute;
    right: -80px;
    top: 0px;
}

.sub2: hover>.dd2 {
    display: block;
}

.di2 {
    padding: 5px;
    opacity: 0.7;
    border: 1px solid white;
    box-sizing: border-box;
}

.di2: hover {
    padding: 0px;
    opacity: 1;
    box-sizing: border-box;
}Output:
```



DAY-7

Media-query: for making responsive website

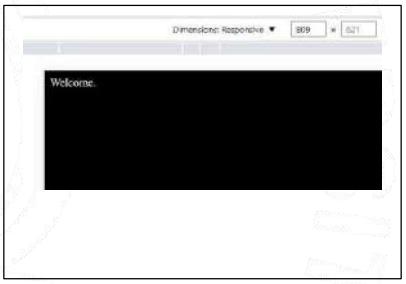
Syntax:

```
@Media screen and (min-width/max-width: value px) {
   selector {
      css
   }
}
```

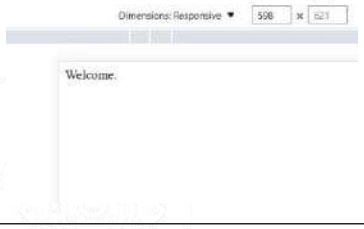
- ✓ min-width: for screen minimum width and above it.
- ✓ max-width: for screen maximum width and below it.

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
<style>        body{
background-color: black;
color: white;
}
@media only screen and (max-width:600px){
body{background-color: white; color: black;}
```



Width less than 600px:



For apply media query we have to apply meta tag:

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

DAY-8

Bootstrap:

Bootstrap is a html, css framework using which we can easily create responsive UI using classes which are already defined.

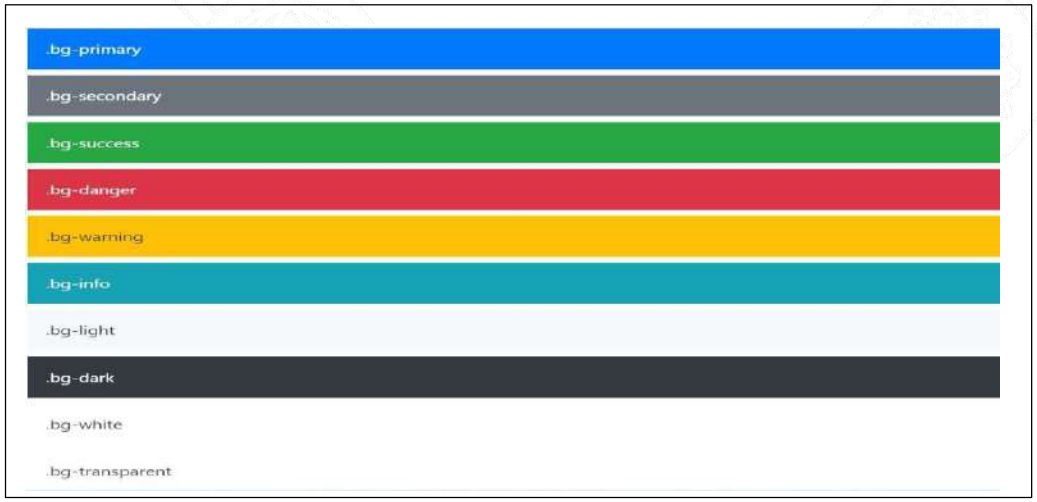
We can download bootstrap folder or we can link the bootstrap using <link> tag.

Color property using bootstrap:

Output:

Background color:

Similar to the contextual text color classes, easily set the background of an element to any contextual class. Anchor components will darken on hover, just like the text classes. Background utilities do not set color, so in some cases you'll want to use utilities.



```
<!DOCTYPE html>
<html lang="en">

<head>


<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css"

rel="stylesheet" integrity="sha384-
4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+PtmoHDEXuppvnDJzQIu9" cross
origin="anonymous">    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
HwwvtgBNo3bZJJLYd8oVXjrBZt8cqVSpeBNS5n7C8IVInixGAoxmnlMuBnhbgrkm" cross
origin="anonymous"></script>


<title>Document</title>
</head>

<body>


<div class="p-3 mb-2 bg-primary text-white">.bg-primary</div>
<div class="p-3 mb-2 bg-secondary text-white">.bg-secondary</div>


<div class="p-3 mb-2 bg-success text-white">.bg-success</div>


<div class="p-3 mb-2 bg-danger text-white">.bg-danger</div>


<div class="p-3 mb-2 bg-warning text-dark">.bg-warning</div>
<div class="p-3 mb-2 bg-info text-white">.bg-info</div>
```

DAY-9

Bootstrap:

Exploring bootstrap website: https://getbootstrap.com/docs/4.6/gettingstarted/introduction/

Exploring various concepts:

element	Class-name	description
container	.container	
Container fluid	.container-fluid	
alerts	. alert, . alert-primary, . alert-success, . alert-secondary, . alert-danger, . alert-warning, . alert-info, . alert-dark, . alert-light	Provide contextual feedback messages for typical user actions with the handful of available and flexible alert messages.
badges	.badge , .badge-light, .badge-primary, .badge-success, .badge-secondary, .badge-danger, .badge-warning, .badge-info, .badge-dark,	Used as our small count and labeling component.
Buttons	.btn , .btn-light, .btn-primary, .btn-success, .btn-secondary, .btn-danger, .btn-warning, .btn-info, .btn-dark, .btn-link	Use Bootstrap’s custom button styles for actions in forms, dialogs, and more with support for multiple sizes, states, and more.
Cards	.card, .card-img-top, .card-title, .card-body, .card-text, .card-link,	Bootstrap’s cards provide a flexible and extensible content container with multiple variants and options.

Carousel	carousel, carousel-item, active, carousel-control-prev, carousel-control-next, carousel-control-next-icon, carousel-control-prev-icon	A slideshow component for cycling through elements—images or slides of text— like a carousel.
Forms	.form-group, .form-control, form-control-file	Examples and usage guidelines for form control styles, layout options, and custom components for creating a wide variety of forms.
Modal	.modal, .modal-body, .modal-header, .modal-footer, .modal-content, .modal-dialog	Use Bootstrap's JavaScript modal plugin to add dialogs to your site for lightboxes, user notifications, or completely custom content.

Task :create carousel with left side image and right side text using bootstrap.

Code:

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.m in.css" rel="stylesheet" integrity="sha384-4bw+/ape/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+PtmoHDEXuppv nDJzQIu9" cross origin="anonymous">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bun dle.min.js" integrity="sha384-HwwvtgBNo3bZJJLYd8oVXjrBZt8cqVSpeBNS5n7C8IVInixGAoxmnlMuBnhb grim" cross origin="anonymous"></script>

<title>Document</title>

</head>
```

```
<body>

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


<div class="carousel-inner bg-primary " style="height:300px; width:
100%; ">

<div class="carousel-item active">


<div class="container row d-flex">

<div class="col-6 d-flex justify-content-end">

<img src=".\image\i1.jpg" class="border border-2 border white" alt=".." style="height:300px;
width:300px; border-radius: 50%;">

</div>

<div class="col-6 d-flex justify-content-center align-items center">

<p class=""><p>Lorem ipsum, dolor sits amet connecter adipescent elite.
Provident, faker. </p>

?</p>

</div>

</div>


</div>

<div class="carousel-item">


<div class="container row d-flex">

<div class="col-6 d-flex justify-content-end">

<img src=".\image\img1.jpg" class="border border-2 border white" alt=".." style="height:300px;
width:300px; border-radius: 50%;">

</div>

<div class="col-6 d-flex justify-content-center align-items center">

<p class=""><p>Lorem ipsum, dolor sits amet connecter adipescent elite.
Provident, faker. </p>

? </p>

</div>

</div>
```

```
</div>
<div class="carousel-item ">


<div class="container row d-flex justify-content-center">
<div class="col-6 d-flex justify-content-end">
<img src=".\image\img2tf.png" class="border border-2 border-white" alt=".."
style="height:300px; width:300px; border-radius: 50%;">
</div>
<div class="col-6 d-flex justify-content-center align-items center">
<p class=""><p>Lorem ipsum, dolor sits amet connecter adipescent elite.
Provident, faker. </p>
 ? </p>
</div>
</div>


</div>
</div>
<button class="carousel-control-prev" type="button" data-target="#carousel Example" data-bs-
slide="prev">
<span class="carousel-control-prev-icon" aria hidden="true"></span>
<span class="visually-hidden">Previous</span>
</button>
<button class="carousel-control-next" type="button" data-target="#carousel Example" data-bs-
slide="next">
<span class="carousel-control-next-icon" aria hidden="true"></span>
<span class="visually-hidden">Next</span>
</button>


</div>
</body>
</html>
```

Output:

DAY10

- **JavaScript**

JavaScript is a widely used programming language primarily known for its role in web development. It enables interactive and dynamic behavior on websites, making them more engaging and user-friendly. JavaScript allows you to create, manipulate, and modify website content in real-time without requiring a page refresh.

Key features of JavaScript include:

Client-Side Scripting: JavaScript runs in web browsers, allowing it to manipulate the Document Object Model (DOM) of a webpage. This means you can change elements on a page, handle user interactions (such as clicks and form submissions), and update content without needing to communicate with a server.

Event-Driven Programming: JavaScript is event-driven, meaning it responds to events like user actions (clicks, key presses, etc.) or changes in the webpage state. You can define functions to execute when specific events occur.

Dynamic Content: JavaScript allows you to create dynamic and interactive content, such as animations, form validations, real-time updates, and more. This enhances user experience by making websites more responsive and engaging.

Libraries and Frameworks: There are numerous libraries and frameworks built on top of JavaScript, such as jQuery, React, Angular, and Vue.js, which simplify complex tasks and provide reusable components

for building web applications.

Asynchronous Programming: JavaScript supports asynchronous programming, allowing you to execute tasks without blocking the main thread. This is crucial for handling tasks like fetching data from servers or performing time-consuming operations without freezing the user interface.

Applications:

Web Development: JavaScript is primarily known for its role in web development. It is used to create interactive and dynamic websites, handle user interactions, update content in real-time, and enhance the user experience.
Front-End Frameworks and Libraries: JavaScript frameworks and libraries like React, Angular, and Vue.js are used to build complex user interfaces and single-page applications (SPAs). These tools provide efficient ways to manage state, handle routing, and create reusable UI components.

Mobile App Development: Frameworks like React Native and frameworks powered by technologies like Apache Cordova allow developers to build mobile apps using JavaScript. This enables code reuse between web and mobile platforms.

Game Development: JavaScript can be used to create browser-based games and interactive multimedia content. HTML5 game engines like Phaser and Three.js utilize JavaScript to create engaging gaming experiences.

Datatypes in JS:

Primitive Data Types:

Number: Represents both integer and floating-point numbers.

String: Represents a sequence of characters, such as text.

Boolean: Represents a logical value, either true or false.

Null: Represents the intentional absence of any value.

Undefined: Represents a variable that has been declared but hasn't been assigned a value.

Reference Data Types:

Object: Represents a collection of key-value pairs, where values can be of any data type, including other objects. Objects can be created using object literals or through constructors.

Array: A special type of object that holds an ordered list of values, typically of the same type. Arrays are used for storing collections of data.

RegExp (Regular Expression): Represents a pattern used for matching strings.

We can write JS anywhere in html document within

<script> </script>

Or we can also write JS in external file with extension .js and it can be import using <script src=" PATH" ></script>.

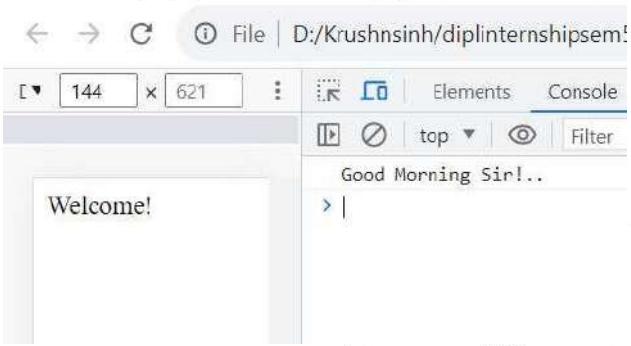
Basic JavaScript program:

```
<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="UTF-8">
   <title>Document</title>
</head>

<body>
   Welcome!
   <script>  console.log ("Good Morning Sir!");
   </script>
</body>

</html>
```



Here console.log() used to print or show output in console panel. alert() method:

The alert() method displays an alert box with a message and an OK button.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Document</title>
</head> <body> <script>   alert
("Good Morning Sir! ");
</script>
</body>
</html>
```

Document Object Model (DOM)

The document object represents the whole html document.

When html document is loaded in the browser, it becomes a document object. It is the root element that represents the html document. It has properties and methods. By the help of document object, we can add dynamic content to our web page.

Methods of document object
We can access and change the contents of document by its methods.
The important methods of document object are as follows:

Method	Description
write("string")	writes the given string on the document.
getElementById ()	returns the element having the given id value.
getElementsByName ()	returns all the elements having the given name value.
getElementsByTagName ()	returns all the elements having the given tag name.
getElementsByClassName ()	returns all the elements having the given class name.

Operators in JS:

Arithmetic operators:

Operator	Description	Example
+	Addition	10+20 = 30
-	Subtraction	20-10 = 10
*	Multiplication	10*20 = 200
/	Division	20/10 = 2

%	Modulus (Remainder)	20%10 = 0
++	Increment	var a=10; a++; Now a = 11
--	Decrement	var a=10; a--; Now a = 9

JavaScript Comparison Operators

The JavaScript comparison operator compares the two operands. The comparison operators are as follows:

Operator	Description	Example
==	Is equal to	10==20 = false
===	Identical (equal and of same type)	10==20 = false
!=	Not equal to	10! =20 = true
!==	Not Identical	20! ==20 = false
>	Greater than	20>10 = true
>=	Greater than or equal to	20>=10 = true
<	Less than	20<10 = false
<=	Less than or equal to	20<=10 = false

JavaScript Logical Operators:

Operator	Description	Example
&&	Logical AND	(10==20 && 20==33) = false

||	Logical OR	(10==20 || 20==33) = false
!	Logical Not	! (10==20) = true

JavaScript Assignment Operators

Operator	Description	Example
=	Assign	10+10 = 20
+=	Add and assign	var a=10; a+=20; Now a = 30
-=	Subtract and assign	var a=20; a-=10; Now a = 10
*=	Multiply and assign	var a=10; a*=20; Now a = 200
/=	Divide and assign	var a=10; a/=2; Now a = 5
%=	Modulus and assign	var a=10; a%=2; Now a = 0

DAY-11

JavaScript If-else

The JavaScript if-else statement is used to execute the code whether condition is true or false. There are three forms of if statement in JavaScript.

1. If Statement
2. If else statement
3. if else if statement

JavaScript If statement

It evaluates the content only if expression is true. The signature of JavaScript if statement is given below.

```
if(expression) {
        //content to be evaluated
}
```

JavaScript If...else Statement

It evaluates the content whether condition is true of false. The syntax of JavaScript if-else statement is given below.

```
if(expression) {
        //content to be evaluated if condition is true
        } else {
        //content to be evaluated if condition is false}
```

JavaScript If...else if statement
It evaluates the content only if expression is true from several expressions.
The signature of JavaScript if else if statement is given below.

```
if(expression1) {
//content to be evaluated if expression1 is true
} else if(expression2) {
//content to be evaluated if
expression2 is true
} else if(expression3) {
        //content to be evaluated if expression3 is true
    }
else {
        //content to be evaluated if no expression is true
}
```

JavaScript Loops

The JavaScript loops are used to iterate the piece of code using for, while, do while or for-in loops. It makes the code compact. It is mostly used in array.

There are four types of loops in JavaScript.

1. for loop
2. while loop
3. do-while loop
4. for-in loop

1) JavaScript For loop

The JavaScript for loop iterates the elements for the fixed number of times. It should be used if number of iterations is known. The syntax of for loop is given below.

```
for (initialization; condition; increment)
{
code to be executed
}
```

2) JavaScript while loop

The JavaScript while loop iterates the elements for the infinite number of times. It should be used if number of iterations is not known. The syntax of while loop is given below.

```
while (condition)
{
code to be executed
}
```

3) JavaScript do while loop

The JavaScript do while loop iterates the elements for the infinite number of times like while loop. But code is executed at least once whether condition is true or false. The syntax of do while loop is given below.

```
do {code to be executed
} while (condition);
```

JavaScript Array

Array is an object that represents a collection of similar type of elements.

But, in JavaScript array we can store different types of elements.

The syntax of creating array using array literal is given below:

var array name= [value 1, value2, valueN];

```
<html>
<body>
<script>   var emp=["Sonoo", "Vimal”,
Ratan"];   console.log(emp);
</script>
</body>
</html>
```

JavaScript Objects

JavaScript Objects is used to store key-value pair.

Syntax:

```
object= {
key: value, key: value, key: value,
        …
}
```

Example:

```
<html>
<body>
<script>   emp= {id:102, name:” xyzzy”,
salary:40000}
document. Write (emp.id+" "+emp.name+" "+emp.
salary);
</script>
</body>
</html>
```

102 xyz 40000

```
//task1 odd-even console.log ("Task1:
odd-even");
var n=prompt ("Enter a
number:"); if(n%2==0)
{console.log (n+" is even.");
} if(n%2==1)
{console.log (n+" is odd.")
}
```

```
//task3 print 20 1
console.log("sub-task3 20-1 using
for loop"); for(i=20; i>0; i--){
console.log(i);
}
console.log ("sub-task3 20-1 using
while loop"); var i=20 while(i>0){
console.log(i);     i--;
}
```

```
// task2 percentage grade
console.log ("Task2: percentage
grade."); var percentage=prompt
("Enter percentage: ");
if(percentage>=0 && percentage<=100)
{if (percentage >= 80) {
console.log ("A Grade...");
}

else if(percentage >= 70){

console.log("B Grade...");

}
else if(percentage >= 60){
console.log("C Grade...");
}     else if(percentage >= 50){
console.log("D Grade...");
}  else{
console.log("require more headwork...");

}
} else {
console.log ("percentage value must be
range 0-100. ")

}
```

```
// task4 print all the even numbers between  1-20
console.log("sub-task4 print all the even numbers between  1-
20");


var num=2;
while(num<=20){

console.log(num);
num+=2;

} console.log("sub-task4 print all the odd numbers    between
1-20");

num=1;
while(num<=20) {  console.log(num);


num+=2;
}
```

```
// sub-task5 print table
console.log ("sub-task5 print table..."); var t=prompt ("Enter number of

which you want table.");


for (i=1; i<=10; i++) {
console.log (t+" X "+i+" = "+ (t*i));

}
```

DAY-12

String methods

char at ()	it provides the char value present at the specified index.
Concat ()	It provides a combination of two or more strings.
replace ()	it replaces a given string with the specified replacement.
Substr ()	It is used to fetch the part of the given string on the basis of the specified starting position and length.
Substring ()	It is used to fetch the part of the given string on the basis of the specified index.
Slice ()	It is used to fetch the part of the given string. It allows us to assign positive as well negative index.
to Lowercase ()	It converts the given string into lowercase letter.
to Uppercase ()	It converts the given string into uppercase letter.
Split ()	It splits a string into substring array, then returns that newly created array.
Trim ()	It trims the white space from the left and right side of the string.
length	It returns the length of a string.

Array methods:

length	It returns the length of an array.
Push ()	It adds new element at the end of array.
Pop ()	It removes the last element of an array.
Shift ()	It adds the new element at starting of an array.
Unshift ()	It removes the first element of an array.
to String ()	The JavaScript method to String () converts an array
	to a string of (comma separated) array values.

```
//task1 print and count vowels present in string var str= prompt
("Enter String:"); var count=0; for (var i=0; i<str. length; i++) {
if((str[i]=="a") || (str[i]=="e") || (str[i]=="i") || (str[i]=="o") ||
(str[i]=="u")) {         console.log(str[i]+" at index "+
i);       count++;
}} console.log ("Total vowels: "+ count);


//task2 convert characters at //even places in string into *.
var str=prompt ("Enter String:"); var str2=""; for (var i=0; i<str. length; i++) {     if(i%2==0) {
str2=str2.concat(str[i]);
} else {str2=str2.concat("*");
    }} console.log(str2);
    //task5 print largest word from given array of words.
    arr=["he","welcome","orange","meaning"];
    var largestWordIndex=0;
    for (var i=0; i<arr. length; i++) { if(arr[i]. length>
       arr[largestWordIndex]. length) { largestWordIndex=i;
       }
       }
       console.log ("Largest word in array: "+arr[largestWordIndex]);


//task3 print sum of all elements
//present in array
arr= [1,2,5,9,11]; var sum=0; for (var i=0; i<arr. length; i++) {     sum+=arr[i];
    } console.log(sum);
```

```
//task4 evaluate x per given values in array
//one by one.
arr=["x++","++x","x++","x--","--x"];
var x=prompt ("Enter value:");
for (var i=0; i<arr. length; i++) {if (arr[i]=="++x" ||
arr[i]=="x++")
{ x++;
} else { x--;
}} console.log("x="+x);
```

```
//task-1 write program that print both string //are equal after combining each element of //array.

var arr1=["ab","cd"]; var arr2= ["a”,

“bide"]; var str1=""; var str2=""; for

(var i=0; i<arr1.length; i++) {

str1+=arr1[i];

} for (var i=0; i<arr2.length; i++) {

str2+=arr2[i];

} console.log(str1); console.log(str2); if(str1==str2)

{console.log ("both strings are equal.");

} else {console.log ("both strings are not equal.");

}
```

```
//task2 print the string in reverse order word
//wise. var str="hello world how are
you"; var arr=St. Split (" "); var str2="";
for (i=arr. length-1; i>=0; i--) {
str2+=arr[i]+" ";
} console.log(str);
console.log(arr);
console.log(str2);
```

```
//task3 print string having maximum words from //array of
strings.
var arr= ["Good Morning!", "how are you","xyz","vcjhs
ghsdvhj yugh hhib"]; misworded=0;
for (var i=0; i<arr. length; i++) {
if((arr[maxWordIndex]. split (" ")). length <
(Arr[i]. split (" ")). length) {maxWordIndex=i;
}} console.log(arr[maxWordIndex]);
```

DAY-14

React Js

React is a JavaScript library for building user interfaces.

React is used to build single-page applications.

React allows us to create reusable UI components.

Example:

```
import React from "react";
import ReactDOM from "react-dom/client";

function Hello(props) {
return <h1>Hello World! < /h1>;
}

const container = document. getElementById('root');
const root = ReactDOM.createRoot(container);
root. render (<Hello />);

/*
The examples in this tutorial are created using the
create-react-app.

Install the create-react-app,
and you will be able to run the same examples on your computer.

In this example we create a component called 'Hello'.
The component is rendered in a container called 'root'.
*/
```

React Components

Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML via a render () function.
Components come in two types, Class components and Function components, in this chapter you will learn about Class components.

Create a Class Component

When creating a React component, the component's name must start with an upper-case letter.

The component has to include the extends React.Component statement, this statement creates an inheritance to React. Component, and gives your component access to React. Component's functions.

The component also requires a render () method, this method returns HTML.

Example: -

```
import React from 'react';
import ReactDOM from 'react-dom/client';

class Car extends React.Component {
render () {
return <h2>Hi, I am a Car! </h2>;
}
}

const container = document. getElementById('root');
const root = ReactDOM.createRoot(container);
root. Render (<Car />);
```

Hi, I am a Car!

DAY: -15

Styling React Using CSS

There are many ways to style React with CSS, this tutorial will take a closer look at three common ways:

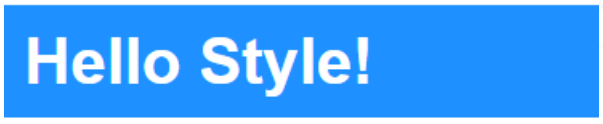
- Inline styling
- CSS stylesheets
- CSS Modules

Example: -

```
import React from 'react';
import React DOM from 'react-dom/client';

const Header = () => {
const myStyle = {
color: "white",
backgroundColor: "DodgerBlue",
padding: "10px",
font Family: "Sans-Serif"
};
return (
<>
<h1 style={myStyle}>Hello Style! </h1>
<p>Add a little style! </p>
</>
);
}

const root = ReactDOM.createRoot(document. getElementById('root'));
root. Render (<Header />);
```



Add a little style!

- This Internship is very good and very important for me because it's providing an extraand more understandable knowledge in web designing
- It was a great experience for me in which I learning something new which I don't know before the Internship.
- The Things that I Have learnt during internship the importance of time-management. Punctuality,

Limitation and Future Enhancement

- Rental Hub System is a Web designing Template Which is currently available in only local storage And that is the limitation of this project.
- Another limitation is that user cannot book Car because its static template

References

https://www.w3schools.com/css/css3_gradients.asp

https://stackoverflow.com/questions/54523630/multiplication-table-using-appendchild-and- html-table

apishttps://getbootstrap.com/

https://www.w3schools.com/bootstrap4/bootstrap_icons