GHraisoni COLLEGE
Engineering and
Management
Pune

G. H. Raisoni College of Engineering & Management, Wagholi, Pune – 412 207

Department of Master of Computer Applications

D-19

Lab Manual (2024-25)

Pattern-2024 Class: FYMCA

Term: I

Front-End Development-MCANL103

Faculty Name:

1. **Asst. Prof. Kishor Markad**
2. **Asst. Prof. Vrushasen Ware.**

G. H. Raisoni College of Engineering and Management,
Wagholi, Pune 412207

Department: MCA

Course Details

Course : MCANL103 Subject: HTML, CSS and JavaScript

Class: FY MCA	**Division: A & B**
Internal Marks: 25	**External marks: 25**
Credits: 2	**Pattern: 2024-25**

COURSE OUTCOME	
	After completing this course, students will be able to
CO1	1. Build Strong expertise to develop front end application using HTML5
CO2	2. Implement MVC and responsive design to scale well across PC, tablet and Mobile Phone
CO3	3. Understand core features of JavaScript and Apply OOP concepts by learning Java Script
CO4	4. Build interactive and user-friendly front end applications using JavaScript

Course Outcomes	Program Outcomes and Program Specific Outcomes														
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO1	1		3		3				2	2			3		2
CO2	1		3		3				2	2			3		2
CO3	1		3		3				2	2			3		2
CO4	1		3		3				2	2			3		2

<table>
<tr><th>Sr.No</th><th>List of Laboratory Assignments (15)</th></tr>
<tr><td>1</td><td>Write HTML code to design a web as per given specification. Divide the browser screen into two frames. The first frame will display the heading. Divide the second frame into two columns. The frame on the left should be name of cities consisting of hyperlinks. Clicking on any one of these hyperlinks will display related information in right hand side frame as shown below
<table>
<tr><td colspan="2"><b>IT Industries in INDIA</b></td></tr>
<tr><td><b>City</b>
1. Pune
2. Mumbai</td><td><b>Pune</b>
• Infosys
• Persistent</td></tr>
</table>
</td></tr>
<tr><td>2</td><td>Create HTML web page with following specifications
i) Title should be about your College.
ii) Put image in the background
iii) Place your college name at the top of page in large text followed by address in smaller size.
iv) Add names of courses offered, each in different color, style and font
v) Add scrolling text about college.
vi) Add any image at the bottom.</td></tr>
<tr><td>3</td><td>Write the HTML code which generates the following output.(use internal CSS to format the table
<table>
<tr><th>Country</th><th colspan="2">Population (in Crores)</th></tr>
<tr><td rowspan="3">INDIA</td><td>1998</td><td>85</td></tr>
<tr><td>1999</td><td>90</td></tr>
<tr><td>2000</td><td>100</td></tr>
<tr><td rowspan="3">USA</td><td>1998</td><td>30</td></tr>
<tr><td>1999</td><td>35</td></tr>
<tr><td>2000</td><td>40</td></tr>
<tr><td rowspan="3">UK</td><td>1998</td><td>25</td></tr>
<tr><td>1999</td><td>30</td></tr>
<tr><td>2000</td><td>35</td></tr>
</table>
</td></tr>
</table>

4	Write HTML code which generates the following output and display each element of list in different size, color & font. Use inline CSS to format the list. 1. DYP • Courses ▪ BCS ▪ BCA 2. Indira • Courses ▪ BCA ▪ MCs 3. ATSS • Courses ▪ BBA ▪ BCS
5	Design an HTML form to take the information of a customer for booking a travel plan consisting of fields such as name, address, contact no., gender, preferred season (Checkboxes), location type (to be selected from a list) etc. You should provide button to submit as well as reset the form contents. (All the fields should be properly aligned).
6	Design an ellipse using CSS.
7	Design a calendar using HTML and CSS.
8	Using HTML, CSS create an image slider which works with radio button.
9	Using HTML, CSS create display a text on top of an image using an overlay.
10	Using HTML, CSS create a custom hover and focus effect for navigation items, using CSS transformations.
11	Write a java script program to accept a number form user and calculate and display its sum of digits.
12	Write a java script program to accept a string from user and display the count of vowel characters from that string.
13	Write a JavaScript Program to read a number from user, store its factors into the array and display that array. (Handle on Click Event).
14	Write a JavaScript Program to design a digital clock & Calculator
15	Dashboard design using Power BI

Assignment 1

Aim: Write HTML code to design a web as per given specification. Divide the browser screen into two frames. The first frame will display the heading. Divide the second frame into two columns. The frame on the left should be name of cities consisting of hyperlinks. Clicking on any one of these hyperlinks will display related information in right hand side frame as shown below

IT Industries in INDIA	
City 3. Pune 4. Mumbai	**Pune** • Infosys • Persistent

Theory : HTML frames allow authors to present documents in multiple views, which may be independent windows or subwindows. Multiple views offer designers a way to keep certain information visible, while other views are scrolled or replaced. For example, within the same window, one frame might display a static banner, a second a navigation menu, and a third the main document that can be scrolled through or replaced by navigating in the second frame.

Each frame would contain its HTML content, and together they would form an interactive webpage where users can navigate through different sections seamlessly.

Layout Of Frames HTML

A frameset HTML document, which defines layout of frames on a webpage, is structured differently than a regular HTML document. A regular HTML document includes a HEAD section and a BODY section. However, a frameset document includes a HEAD section and a FRAMESET section instead of the BODY section.

The FRAMESET section indicates the layout or we can say it indicates how different frames are organized and displayed in the main browser window. It can also include a NOFRAMES element, which provides alternative content for browsers that don’t support frames or are set not to display them.

Implementing frames in web development lets you split a webpage into multiple sections. Framesets divide the page into rows and columns, while iframes embed external content. You can customize these frames with attributes like columns, rows, width, and height for a more dynamic user experience.

Here’s a closer look at how to use them practically:

1. Creating a Frameset

To create a frameset, use the ‘<frameset>’ tag, and Inside this tag, you can specify rows and columns as required.

For example:

```
<frameset rows=”50%,50%”>
```

Using the above code creates a frameset with two rows of equal height.

You can also use the ‘<frame>’ tag to define individual frames and can set the size of each frame with attributes like ‘cols’ and ‘rows’, using percentages or pixel values.

2. Using iframes

Iframes allow you to embed external content within a webpage.

The syntax for an iframe is:

```
<iframe src=”external.html”></iframe>
```

By using the ‘src’ attribute, you can load another HTML document or web resource inside the iframe. You can customize iframes with attributes like ‘width’ and ‘height’ to set their size. Additional attributes include ‘frameborder’ to show or hide the border, and ‘scrolling’ to enable or disable scrolling within the iframe.

Source Code :

Main Frame

```
 <!DOCTYPE html>
<html lang="en">
<head>
    <title>Frames</title>
</head>
<body>
    <frameset row="100,*">
        <frame src="frame1.html" name="headerFrame" noresize>
        <frameset cols="200,*">
        <frame src="frame2.html" name="citiesFrame">
        <frame src="frame3.html" name="infoFrame">
    </frameset>
</body>
</html>
```

Frame1

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Frame 1</title>
</head>
<body>
        <h1>IT Industries in India</h1>
</body>
</html>
```

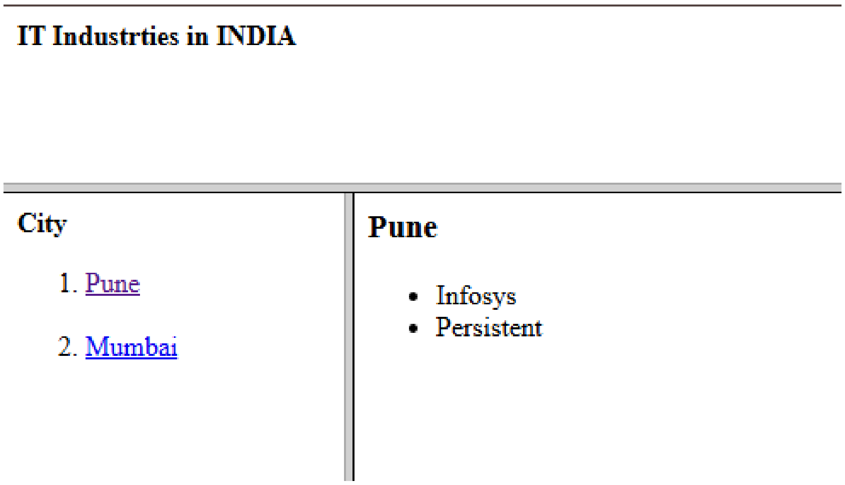
Frame2

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>frame2</title>
</head>
<body>
    <h1>City</h1>
    <ul>
        <li>Pune</li>
        <li>Mumbai</li>
    </ul>
</body>
</html>
```

Frame3

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>frame2</title>
</head>
<body>
    <h1>Pune</h1>
    <ul>
        <li>Infosys</li>
        <li>Persistent</li>
    </ul>
  </body>
  </html>
```

Output :



IT Industrties in INDIA

City

1. Pune
2. Mumbai

Pune

- Infosys
- Persistent

Conclusion : This assignment demonstrate the use of HTML iframe & frameset.

Assignment 2

Aim :

Create HTML web page with following specifications

i) Title should be about your college.

ii) Put image in the background

iii) Place your college name at the top of page in large text followed by address in smaller size.

iv) Add names of courses offered, each in different color, style and font

v) Add scrolling text about college.

vi) Add any image at the bottom.

Theory :

Basic HTML Document Structure

Let's begin by analyzing the primary structure of an HTML document. A minimal HTML record consists of the following factors:

```
<!DOCTYPE html>
<html>
<head>
   <title>Document Title</title>
</head>
<body>
   <!-- Content goes here -->
</body>
</html>
```

<!DOCTYPE html>: This declaration specifies the HTML version getting used. In this example, it's HTML5.

<html>: The root element that wraps the complete HTML record.

<head>: Contains meta-information about the document, the name, man or woman set, linked stylesheets, and extra.

<title>: Sets the HTML file's identity displayed in the browser's title bar or tab.

<body>: Houses the content material of the HTML report, including textual content, pictures, links, and different factors.

Headings (<h1> to <h6>): Headings are important for structuring the content within the <body> section. They define the hierarchy of information, where <h1> is the highest-level heading.

Paragraphs (<p>): The <p> tag is used for creating paragraphs, which help to group text into readable sections. This tag automatically adds some spacing between paragraphs to improve readability.

Images (<img>): The <img> tag is used to add images to a webpage. It requires the src attribute to specify the image path and the alt attribute for accessibility.

Links (<a>): Links are created with the <a> tag. The href attribute defines the destination URL, and the text within the <a> tag serves as the clickable link text.

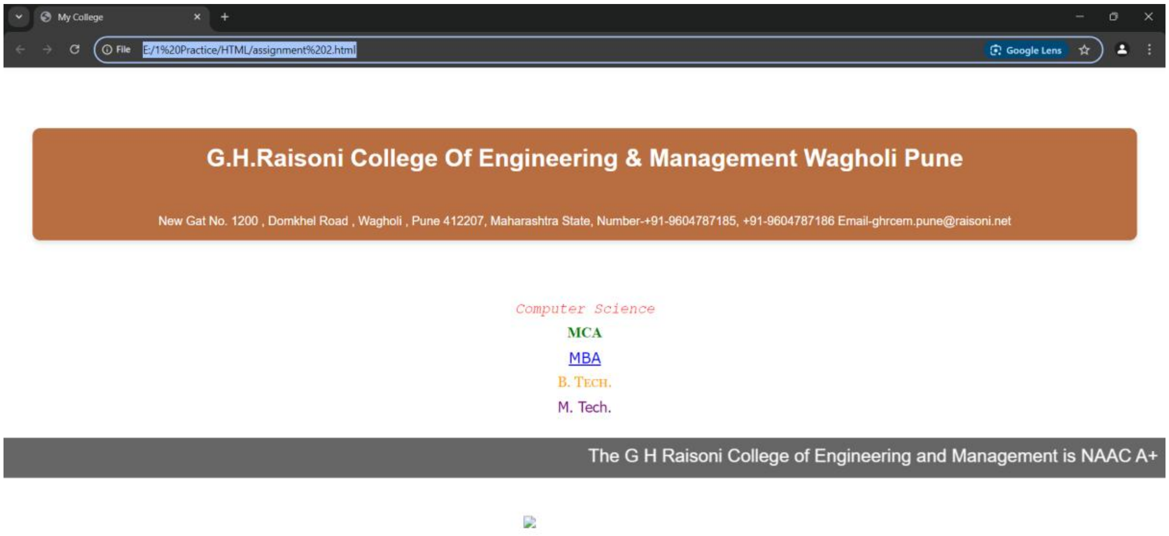
Divisions (<div>): The <div> tag is a container used to group other elements together, often for layout. It does not add any style or structure on its own but is useful for applying CSS styles to sections of content.

Source Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My College</title>
    <style>
        .container{
            margin: 5rem auto;
            width: 95%;
            background-color: #ae3535;
            border-radius: 10px;
            box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
            overflow: hidden;
            margin-bottom: 5rem;
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            text-align: center;
            transition: background-color 0.3s;
                &:hover{
                    background-color: #b86e40;
            }    }

        body {
            background-image: url('background.jpg');
            background-size: cover;
            background-repeat: no-repeat;
            color: white;
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            text-align: center;
        }
        .header {
            margin-top: 50px;
        }
        .header h1 {
            font-size: 3em;
            margin: 0;
        }
        .header p {
            font-size: 1.5em;
            margin: 10px 0 20px 0;
        }
        .courses {
            margin: 30px 0;
        }
        .courses div {
            margin: 10px;
            font-size: 1.2em;
        }
        .scroll-text {
            margin: 30px 0;
            font-size: 1.5em;
            background-color: rgba(0, 0, 0, 0.6);
            padding: 10px;
        }
        .footer-image {
            margin: 50px 0;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>G.H.Raisoni College Of Engineering & Management Wagholi
Pune</h1><br>
        <p>New Gat No. 1200 , Domkhel Road , Wagholi , Pune 412207, Maharashtra
State,
            Number-+91-9604787185, +91-9604787186
            Email-ghrcem.pune@raisoni.net</p>
        </div>



    <div class="courses">
```

```
        <div style="color: red; font-family: 'Courier New', Courier, monospace; font-style: italic;">Computer Science</div>
        <div style="color: green; font-family: 'Times New Roman', Times, serif; font-weight: bold;">MCA</div>
        <div style="color: blue; font-family: 'Verdana', sans-serif; text-decoration: underline;">MBA</div>
        <div style="color: orange; font-family: 'Georgia', serif; font-variant: small-caps;">B. Tech.</div>
        <div style="color: purple; font-family: 'Tahoma', sans-serif; font-size: larger;">M. Tech.</div>
    </div>

    <div class="scroll-text">
        <marquee behavior="scroll" direction="left">The G H Raisoni College of Engineering and Management is NAAC A+ accredited and affiliated to the Savitribai Phule University. </marquee>
    </div>
    <div class="footer-image">
        <img src="college_logo.png" alt="G H Raisoni College" style="width: 200px;">
    </div>
</body>
</html>
```

Output :



G.H.Raisoni College Of Engineering & Management Wagholi Pune

New Gat No. 1200 , Domkhel Road , Wagholi , Pune 412207, Maharashtra State, Number-+91-9604787185, +91-9604787186 Email-ghrcem.pune@raisoni.net

Computer Science

MCA

MBA

B. Tech.

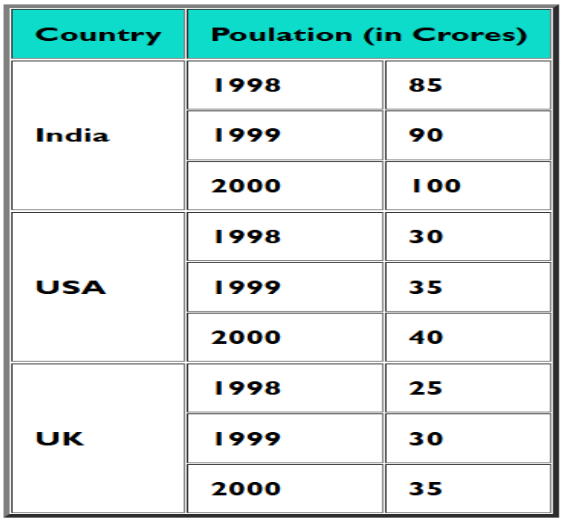
M. Tech.

The G H Raisoni College of Engineering and Management is NAAC A+

Conclusion : This assignment demonstrate the basic structure of a web page.

Assignment 3

Aim : Write the HTML code which generates the following output.(use internal CSS to format the table)



Country	Poulation (in Crores)	
India	1998	85
	1999	90
	2000	100
USA	1998	30
	1999	35
	2000	40
UK	1998	25
	1999	30
	2000	35

Theory :

What Is an HTML Table?

An HTML table is a structured data grid organized into rows and columns. It visually represents tabular data, such as financial reports, product listings, schedules, etc. Tables present information in an organized and easily readable format, which is especially helpful when dealing with large data sets.

HTML tables consist of rows (represented by `<tr>` tags) and columns (represented by `<td>` or `<th>` tags).

Each row contains data cells, either standard data cells `<td>` or header cells `<th>`.

Header cells typically appear at the top of columns and briefly describe the content in that column.

How Does a Table Work?

HTML tables use a combination of tags to structure and display data in a tabular format.

1. Table Element `<table>`: The `<table>` tag serves as the container for the entire table. It encapsulates all other table-related tags.
2. Table Row Element `<tr>`: Rows are created using the `<tr>` tag, and each row represents a horizontal line in the table. Rows contain one or more data cells.
3. Table Header Cell `<th>`: Header cells label columns or provide additional information about the data in the column. They are contained within `<tr>` elements.
4. Table Data Cell `<td>`: Data cells contain the actual data and are placed within `<tr>` elements. These cells make up the majority of the table's content.
5. Table Caption `<caption>`: The `<caption>` tag is optional and provides a title or description for the entire table. It appears above or below the table, typically

centered.

6. Column Grouping `<colgroup>` and `<col>`: These tags are used to group and style columns. `<colgroup>` is a container for one or more `<col>` elements, which define column-specific properties.
7. Table Body `<tbody>`, Table Head `<thead>`, and Table Footer `<tfoot>`: These elements group rows within the table. `<tbody>` contains the main data rows, `<thead>` holds header rows, and `<tfoot>` is used for footer rows.

Internal CSS:-

Internal CSS is a method for defining CSS styles directly within an HTML document. It’s particularly useful for applying unique styles to a single web page, and it’s embedded within the <style> element located in the <head> section of the HTML file.

How to Use Internal CSS ?

To use internal CSS, you need to include CSS rules within a <style> tag inside the HTML document’s <head>. This allows you to define styles by selecting HTML elements or classes and applying styling rules within the tag. The styles defined by internal CSS apply only to the specific web page where they are included.

8.

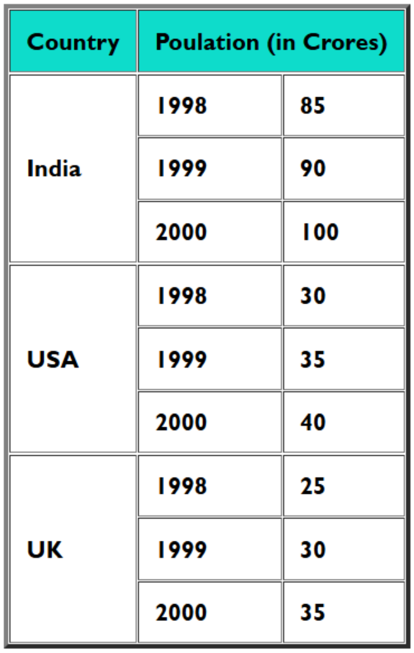
HTML Table Tags

1. `<table>` : Container for the entire table.
2. `<tr>` : Represents a row within the table.
3. `<th>` : Defines header cells within rows.
4. `<td>` : Represents standard data cells within rows.
5. `<caption>`: Provides a title or description for the table.
6. `<colgroup>`: Groups columns for styling purposes.
7. `<col>` : Defines column-specific properties.
8. `<tbody>` : Groups main data rows within the table.
9. `<thead>` : Groups header rows within the table.
10. `<tfoot>` : Groups footer rows within the table.

Source Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>create table using internal CSS </title>
    <style>
        body{
            text-align: -webkit-center;
            position: relative;
            font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;
            font-size: 20px;
            padding-top: 5rem;
        }
        .table-frame tr td {
            font-size: 30px;
            font-weight: 900;
            padding: 20px;
        }
        .tr-frame1{
            background-color: #0edccb;
            color: #000;
        }
    </style>
</head>
<body>
        <table border="6" class="table-frame">
        <tr class="tr-frame1">
            <td>Country</td>
            <td colspan="2">Poulation (in Crores)</td>
        </tr>
   <tr>
            <tr>
                <td rowspan="3">India</td>
                <td>1998</td>
                <td>85</td>
 </tr>
            <tr>
                <td>1999</td>
                <td>90</td>
            </tr>
            <tr>
                    <td>2000</td>
                    <td>100</td>
            </tr>
     </tr>
        <tr>
      <tr>
                <td rowspan="3">USA</td>
                <td>1998</td>
                <td>30</td>
            </tr>
            <tr>
                <td>1999</td>
                <td>35</td>
   </tr>
            <tr>
                <td>2000</td>
                <td>40</td>
            </tr>
</tr>
        <tr>
    <tr>
                <td rowspan="3">UK</td>
                <td>1998</td>
                <td>25</td>
   </tr>
                <tr>
                    <td>1999</td>
                    <td>30</td>
       </tr>
                <tr>
                    <td>2000</td>
                    <td>35</td>
   </tr>
        </tr>
        </table>
</body>
</html>
```

Output :



Country	Poulation (in Crores)	
India	1998	85
	1999	90
	2000	100
USA	1998	30
	1999	35
	2000	40
UK	1998	25
	1999	30
	2000	35

Conclusion : with the help of this assignment you understand how to use HTML table element.

Assignment 4

Aim : Write HTML code which generates the following output and display each element of list in different size, color & font. Use inline CSS to format the list.

1. DYP
    - Courses
        - BCS
        - BCA
2. Indira
    - Courses
        - BCA
        - MCS
3. ATSS
    - Courses
        - BBA
        - BCS

Theory :

1. HTML Lists are used to create a list of information in an HTML page.
2. HTML Lists may contain (i) Unordered Information (ii) Ordered Information (iii) Definitions.
3. Unordered Information can be presented using ul and li elements.
4. Ordered Information can be presented using ol and li elements.
5. Definitions can be presented using dl, dt and dd elements.
6. There are two more elements dir and menu may also be used to create multi column directory list and single column list. But these elements are deprecated.
7. HTML Lists should not be used for indenting text. It should be used for creating a list of information only. In modern web design trends, navigation is a good example of using lists.

HTML ul tag and element

1. HTML ul (unordered list) element is used to create an unordered list of items (information) in an HTML page.
2. Items included in an unordered list do not have any numbering.
3. The order of appearance of items in an unordered list does not have any significance.
4. By default, items in an unordered list are preceded by circle bullet. Styles can be used to modify the form of bullet.

Inline CSS: - Inline CSS is a method that applies CSS styling directly to HTML elements using the ‘style’ attribute. This approach allows developers to define styles for individual elements, making it an effective tool for applying unique styles to specific HTML elements.

When to Use Inline CSS ?

When Should You Use Inline CSS? Inline styles are particularly useful in web development when you need to apply specific, one-off styles to an element. This approach ensures that global styles remain unaffected. Additionally, inline styles are ideal when you need to generate dynamic styles programmatically, based on data or user interactions.

How to use inline CSS ?

To use inline CSS, add a “style” attribute to an HTML element and define CSS properties and values within double quotes, Inline CSS has the highest priority out of external, internal CSS, and inline CSS.

Syntax

```
 <ul>
<li>List item one</li>
<li>List item two </li>
</ul>
```

HTML ol tag and element

1. HTML ol (ordered list) element is used to create an ordered list of items (information) in an HTML page.
2. Items included in an ordered list have a numbering.
3. An item appears in order of significance in an ordered list.
4. By default, items in an ordered list are preceded by numbers (1,2,3.....).

Syntax

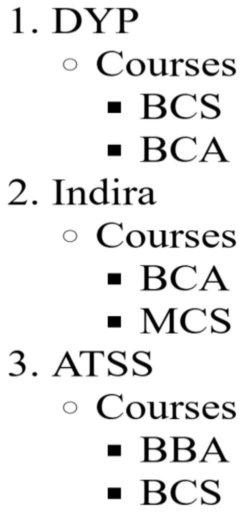
```
<ol>
<li>List item one</li>
<li>List item two </li>
</ol>
```

Source Code:

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

```
<head>    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Element of List</title>
</head>
<body >
    <ol >
        <li>DYP</li>
        <ul>           <li>Courses</li>
            <ul>
                <li>BCS</li>                <li>BCA</li>
            </ul>      </ul>
        <li>India</li>
        <ul>
            <li>Courses</li>
            <ul>
                <li>BCA</li>
                <li>MCS</li>
            </ul>       </ul>
        <li>ATSS</li>
        <ul>
            <li>Courses</li>
            <ul>
            <li>BBA</li>
            <li>BCS</li>
        </ul>       </ul>
    </ol>
</body>
</html>
```

Output:



```
1. DYP
     ○ Courses
          ▪ BCS
          ▪ BCA
2. Indira
     ○ Courses
          ▪ BCA
          ▪ MCS
3. ATSS
     ○ Courses
          ▪ BBA
          ▪ BCS
```

Conclusion: After completing this assignment we can use HTML list element

Assignment 5

Aim : Design an HTML form to take the information of a customer for booking a travel plan consisting of fields such as name, address, contact no., gender, preferred season (Checkboxes), location type (to be selected from a list) etc. You should provide button to submit as well as reset the form contents. (All the fields should be properly aligned).

Theory :

HTML Form

An HTML form is a section of a document which contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, menus etc.

An HTML form facilitates the user to enter data that is to be sent to the server for processing such as name, email address, password, phone number, etc. .

HTML Form Syntax

```
        <form action="server url" method="get|post">
          //input controls e.g. textfield, textarea, radiobutton, button
        </form>
```

The <input> Element

The HTML <input> element is the most used form element.
An <input> element can be displayed in many ways, depending on the type attribute.

Here are some examples of <input> elements:

Type	**Description**
<input type="text">	Displays a single-line text input field
<input type="radio">	Displays a radio button (for selecting one of many choices)
<input type="checkbox">	Displays a checkbox (for selecting zero or more of many choices)
<input type="submit">	Displays a submit button (for submitting the form)
<input type="button">	Displays a clickable button

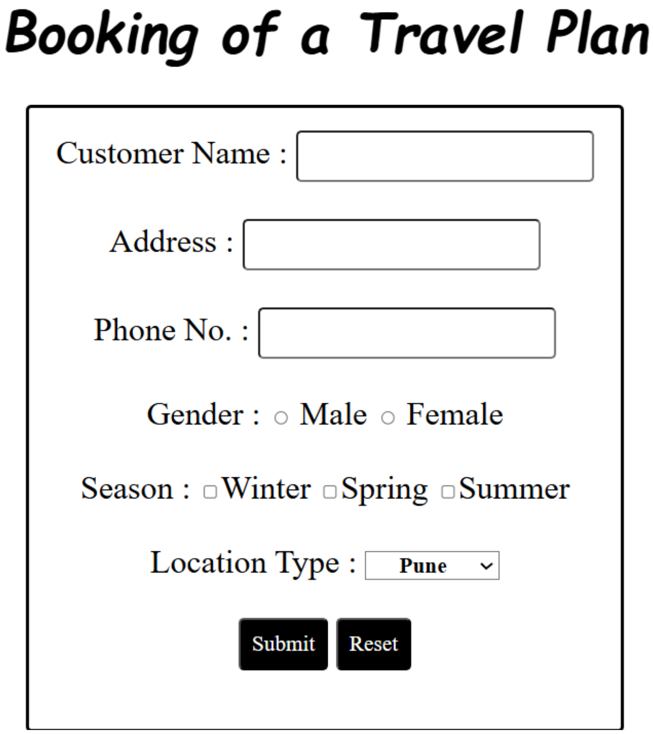
Source Code

```
<!DOCTYPE html>
<html>
        <head>
                <title>Information of a Customer </title>
          <style>
                body{
```

```
 text-align: center;
 font-size: 30px;
}
form{
    margin-top: 20px;
    padding: 20px;
    background-color: lightblue;
    border: 3px black solid;
    border-radius: 5px;
    margin:0 auto ;
    width: 500px;
}
input{
   font-family: fangsong;
    font-size: 20px;
    padding: 10px;
    border-radius: 5px;
}
h1{
text-align: center;
font-size: 50px;
font-family: cursive;
 font-style: oblique;
 }
 .loc_opt{
    font-size: 20px;
    text-align: center;
    font-family: fangsong;
    font-weight: bold;
    color: #000000;
    cursor: pointer;
    background-color: #ffffff;
 }
 .sub{
    text-align: center;
    font-size: 20px;
    font-family: fangsong;
    color: #ffffff;
    cursor: pointer;
    background-color: #000000;
}
 .sub:hover{
```

```
                    background-color: lightgreen;
                    color: #0015ff;
                    cursor: pointer;
                 }
                 .res{
                    text-align: center;
                    font-size: 20px;
                    font-family: fangsong;
                    color: #ffffff;
                    cursor: pointer;
                    background-color: #000000;
                 }
                 .res:hover{
                    background-color: lightgreen;
                    color: #0015ff;
                    cursor: pointer;
                 }
       </style>
</head>
<body>
        <h1>Booking of a Travel Plan</h1>
        <form class="f-type">
                <label for="cname">Customer Name :</label>
                <input type="text" id="cname" name="cname" ><br><br>
                 <label for="address">Address :</label>
                <input type="address" id="address" name="address" ><br><br>
                 <label for="number">Phone No. :</label>
                 <input type="number" name="phone" id="phone"><br><br>
                 <label for="gender">Gender :</label>
                 <input type="radio" name="gender" id="gender"> Male
                 <input type="radio" name="gender" id="gender"> Female <br><br>
                 <label for="season">Season :</label>
                 <input type="checkbox" name=" season" id="season" value="Winter">
                Winter
                 <input type="checkbox" name=" season" id="season" value="Spring">
                Spring
                 <input type="checkbox" name=" season" id="season" value="Summer">
                Summer <br><br>
                 <label for="location">Location Type :</label>
                 <select class="loc_opt" name="Location" size="1">
                          <option>Pune</option> <option>Ujjain</option>
                         <option>Kashmir</option>  <option>Delhi</option>
                    <option>Hyderabad</option></select><br><br>
                <input class="sub" type="Submit" value="Submit">
                <input class="res" type="Reset"><br><br>
               </form>
        </body>
</html>
```

Output:



Booking of a Travel Plan

Customer Name :

Address :

Phone No. :

Gender : ○ Male ○ Female

Season : □Winter □Spring □Summer

Location Type : Pune

Conclusion : After completing this assignment students will be able to use html form elements

Assignment 6

Aim : Design an ellipse using CSS.

Theory : Creating an ellipse using CSS involves styling an element (typically a <div>) and applying appropriate width, height, and border-radius values. The border-radius property is used to create rounded shapes, and when its value is set to 50% or more, it can turn a rectangular element into an ellipse.

1. **Proportion Control**:
   - The aspect ratio (width:height) determines how elongated or circular the ellipse appears. For a perfect circle, set width and height to equal values.
2. **Centering the Ellipse**:
   - Use margin: auto; or flexbox to center the ellipse within its container.
3. **Customization**:
   - Add a gradient background for a more visually appealing ellipse:

     ```
     background: radial-gradient(circle, lightblue, darkblue);
     ```
4. **Responsive Design**:
   - Use percentages for width and height to make the ellipse responsive:

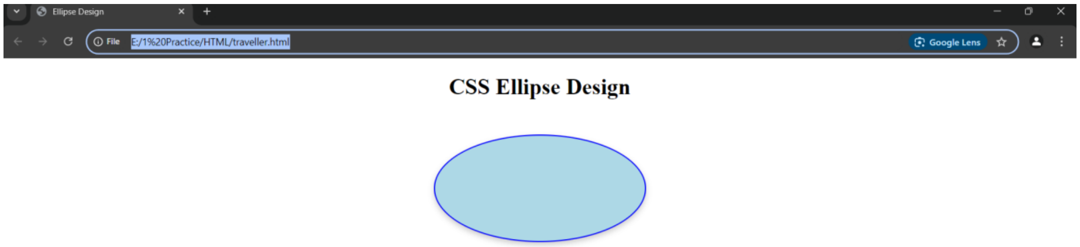
     ```
     width: 50%;
     height: 25%;
     ```

Source Code :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ellipse Design</title>
  <style>
    .ellipse {
      width: 300px;
      height: 150px;
      background-color: lightblue;
      border: 2px solid blue;
      border-radius: 50%;
      margin: 50px auto;
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
```

```
        }
      </style>
    </head>
    <body>
      <h1 style="text-align: center;">CSS Ellipse Design</h1>
      <div class="ellipse"></div>
    </body>
    </html>
```

Output :



CSS Ellipse Design

Conclusion : After completing this assignment students can understand how to create shapes using CSS and HTML.

Assignment 7

Aim : Design a calendar using HTML and CSS

Theory :

we will create a calendar using HTML, and CSS. HTML is a standard for creating the structure of web pages and defining elements like headings, paragraphs, and links whereas CSS (Cascading Style Sheets) complements HTML by controlling the visual presentation, enabling styling and layout customization for a more visually appealing web design.

Approach

- First, create the HTML structure by creating a table using <tr> <th> <td> tags for designing calendar.
- Inside the <body>, create a <table> element to structure the calendar. Use <thead> for the table header containing day names (Sun, Mon, Tue, etc.).Style the header with background color and text color.
- Apply styles for the table, including `border-collapse`, `background`, and `color`, and add styling for specific cells using inline styles.
- Complete the table by inputting the days of the month within the `<td>` elements. Ensure a proper hierarchy and indentation for readability.

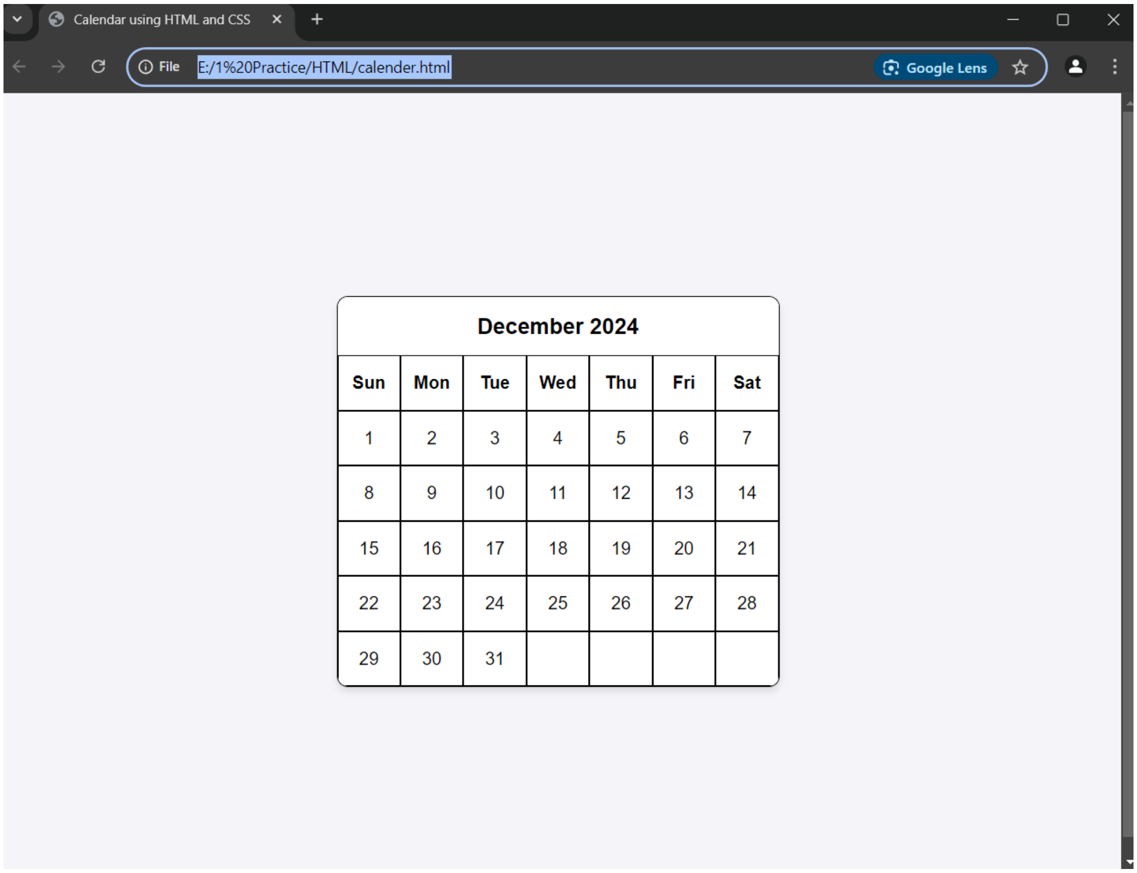
Syntax

Source Code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Calendar using HTML and CSS</title>
  <style>
  body {
    font-family: Arial, sans-serif;
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    background-color: #f4f4f9;
  }
  .calendar {
    width: 90%;
    max-width: 400px;
    background: #ffffff;
    border: 1px solid #000000;
    border-radius: 10px;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
    overflow: hidden;
  }
  .calendar_header {
    text-align: center;
    background: #ffffff;
    color: rgb(0, 0, 0);
    padding: 15px;
    font-size: 20px;
    font-weight: bold;
  }
  .calendar_body {
    display: grid;
    grid-template-columns: repeat(7, 1fr);
    grid-auto-rows: 50px;
    text-align: center;
    font-size: 16px;
  }
  .day, .dates {
    display: flex;
    align-items: center;
    justify-content: center;
    border: 1px solid #000000;
  }
  .day {
    background: #ffffff;
    font-weight: bold;
    color: #000;
  }
  .dates {
    background: #ffffff;
    transition: background 0.3s;
  }
  .dates:hover {
    background: #f1c40f;
    color: #ffffff;
    cursor: pointer;
  }
```

```
    </style>
</head>
<body>
    <div class="calendar">
        <div class="calendar_header">December 2024</div>
            <div class="calendar_body">
                <div class="day">Sun</div>
                <div class="day">Mon</div>
                <div class="day">Tue</div>
                <div class="day">Wed</div>
                <div class="day">Thu</div>
                <div class="day">Fri</div>
                <div class="day">Sat</div>
                <div class="dates">1</div>
                <div class="dates">2</div>
                <div class="dates">3</div>
                <div class="dates">4</div>
                <div class="dates">5</div>
                <div class="dates">6</div>
                <div class="dates">7</div>
                <div class="dates">8</div>
                <div class="dates">9</div>
                <div class="dates">10</div>
                <div class="dates">11</div>
                <div class="dates">12</div>
                <div class="dates">13</div>
                <div class="dates">14</div>
                <div class="dates">15</div>
                <div class="dates">16</div>
                <div class="dates">17</div>
                <div class="dates">18</div>
                <div class="dates">19</div>
                <div class="dates">20</div>
                <div class="dates">21</div>
                <div class="dates">22</div>
                <div class="dates">23</div>
                <div class="dates">24</div>
                <div class="dates">25</div>
                <div class="dates">26</div>
                <div class="dates">27</div>
                <div class="dates">28</div>
                <div class="dates">29</div>
                <div class="dates">30</div>
                <div class="dates">31</div>
                <div class="dates"></div>
                <div class="dates"></div>
                <div class="dates"></div>
                <div class="dates"></div>
            </div>
    </div>
</body>
</html>
```

Output


Calendar using HTML and CSS
File
E:/1%20Practice/HTML/calender.html
Google Lens

December 2024

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Conclusion : After completing this assignment students will understand how to work with internal CSS

Assignment 8

Aim : Using HTML, CSS create an image slider which works with radio button.

Theory :
The HTML <img> tag is used to embed images in a web page. It is an empty or self-closing tag, meaning it doesn’t have a closing tag. It allows to display images from various sources, such as files on a website or URLs from other websites.

Syntax:
<img src="./image.png" alt="image description" width="200" height="100">

HTML image Tag Attributes

- src Specifies the path to the image.
- alt Provides alternate text for the image, useful for informing users about the image and displaying in case of network issues.

- crossorigin Allows importing images from third-party sites with cross-origin access for use with canvas.

- height and width Specifies the height and width of the image.
- ismap Specifies an image as a server-side image map.
- loading Specifies whether a browser should defer loading of images until certain conditions are met or load an image immediately.
- longdesc Specifies a URL to a detailed description of an image.
- referrerpolicy Specifies which referrer information to use when fetching an image (e.g., no-referrer, no-referrer-when-downgrade, origin).

- size Specifies image sizes for different page layouts.
- srcset Specifies a list of image files to use in different situations.
- usemap Specifies an image as a client-side image map.

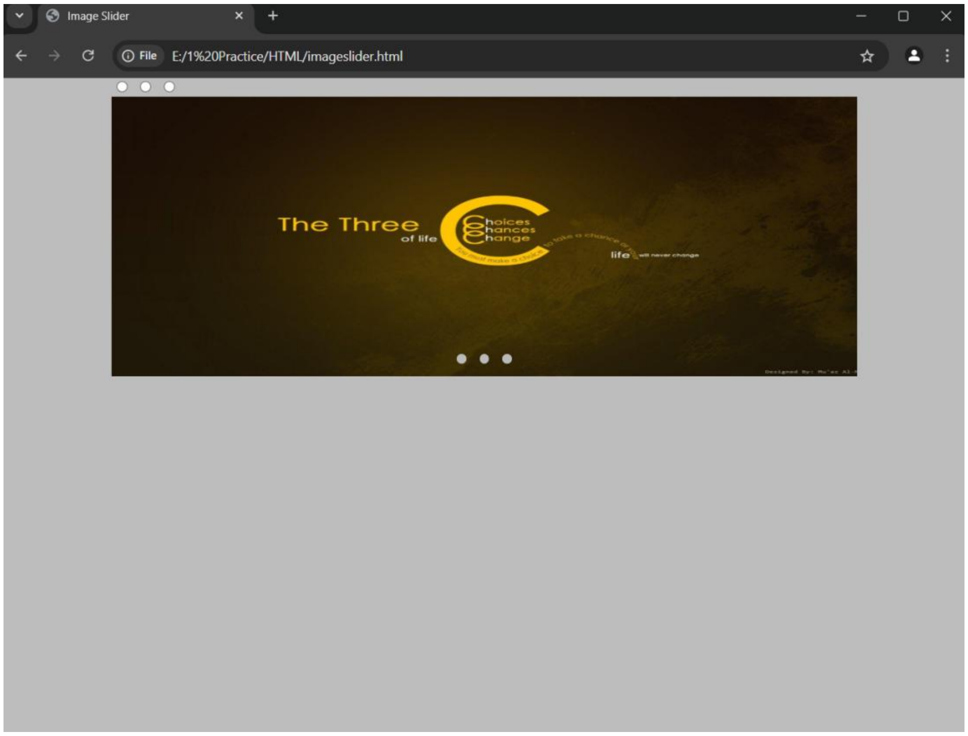
Source Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Image Slider</title>
    <style>
body {
font-family: Arial, sans-serif;
```

```
margin: 0;
padding: 0;
background-color: #bdbdbd;
height: 100vh;
margin: 0;
}
img{
    width: 400px;;
    height: 300px;
}
.slider {
position: relative;
width: 100%;
max-width: 50rem;
margin: auto;
overflow: hidden;
}
.slides {
display: flex;
transition: transform 0.5s ease-in-out;
}
.slide {
min-width: 100%;
box-sizing: border-box;
}
.slide img {
width: 100%;
display: block;
}
#slide1:checked ~ .slides {
transform: translateX(0);
}
#slide2:checked ~ .slides {
transform: translateX(-100%);
}
#slide3:checked ~ .slides {
transform: translateX(-200%);
}
.navigation {
    position: absolute;
    bottom: 10px;
    left: 50%;
```

```
        transform: translateX(-50%);
    }
    .navigation label {
        cursor: pointer;
        height: 10px;
        width: 10px;
        background-color: #bbb;
        border-radius: 50%;
        display: inline-block;
        margin: 0 5px;
        transition: background-color 0.3s;
    }
    </style>
    </head>
    <body>
        <div class="slider">
                <input type="radio" name="slider" id="slide1">
                <input type="radio" name="slider" id="slide2">
                <input type="radio" name="slider" id="slide3">
            <div class="slides">
                <div class="slide" id="s1">
                <img src="1.jpg" alt="Image 1">
                </div>
                <div class="slide" id="s2">
                <img src="2.jpg" alt="Image 2">
                </div>
                <div class="slide" id="s3">
                <img src="3.png" alt="Image 3">
                </div>
            </div>
            <div class="navigation">
                <label for="slide1"></label>
                <label for="slide2"></label>
                <label for="slide3"></label>
            </div>
            </div>
        </body>
    </html>
```

Output :



Conclusion : After completing this assignment students will understand how to create image slider using radio button & CSS

Assignment 9

Aim : Using HTML, CSS create display a text on top of an image using an overlay

Theory :
An overlay is a transparent layer of content that is placed on top of another element. It can be used to create different effects, such as a modal window, a tooltip, or a popover.

The overlay element should be positioned absolutely and have a higher z-index than the content element. This will ensure that the overlay is displayed on top of the content.

To create an overlay using CSS, follow these steps:

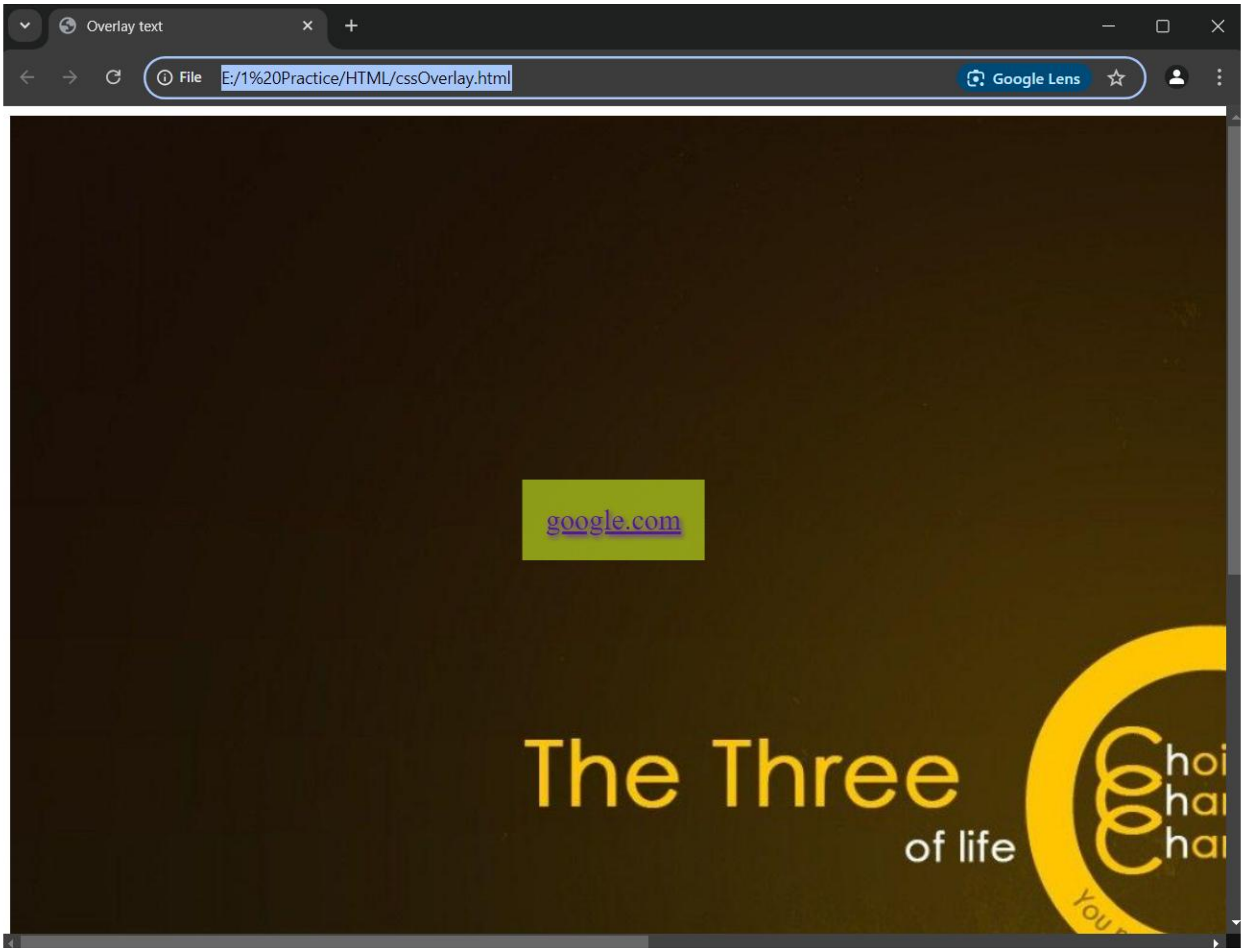
1. Create the Container: Use a container element to hold the content you want to overlay. A container can be div element, span element or even an image.
2. Set Up the Positioning: Make the container position: relative so that any absolutely positioned elements inside it will be positioned relative to this container.
3. Add the Overlay Element: Inside the container, add another element (the overlay) and set position: absolute to ensure it covers the entire container. Also make sure the overlay's top, left properties are set to 0 and width, height properties are set to 100%, so it fully covers the container.
4. Style the Overlay: Set the background color of the overlay using rgba() function to give a semi-transparent effect. And initially, set the opacity of the overlay to 0 to make it invisible by default.
5. Add Hover Effect: Use hover pseudo-class to the overlay container to change the overlay's opacity from 0 to 1 when user moves mouse over container.

Source Code

```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>Overlay text</title>
   <style>
      .img_part{
         position: relative;
         width: 100px;
         height: 100px;
      }
      .overlay {
         position: absolute;
         top: 50%;
```

```
            left: 50%;
            transform: translate(-50%, -50%);
            background-color: rgba(0, 0, 0, 0.7);
            color: white;
            padding: 20px;
            text-align: center;
            font-size: 24px;
            text-shadow: 2px 2px 4px rgb(255, 255, 255);
            animation: fade-in-out 2s infinite;        }
    </style>

</head>
<body>
    <section>
        <div class="img_part">
            <img src="1.jpg" alt="image">
        </div>
        <div class="overlay">
            <a href="https://www.google.com" target="_blank">
                google.com
            </a>
        </div>
    </section>
</body>
</html>
```

Output :



Conclusion : After completing this assignment students will understand working of overlay effect.

Assignment 10

Aim : Using HTML, CSS create a custom hover and focus effect for navigation items, using CSS transformations.

Theory :
Creating a horizontal navigation bar in HTML and CSS involves using an unordered list (<ul>) for navigation items and applying CSS to style them for a horizontal layout.

We will create a horizontal navigation bar using HTML and CSS, including styling for layout, spacing, and appearance, as well as adding interactivity with hover effects and ensuring responsiveness to media queries.

Approach to Create Horizontal Navigation Bar

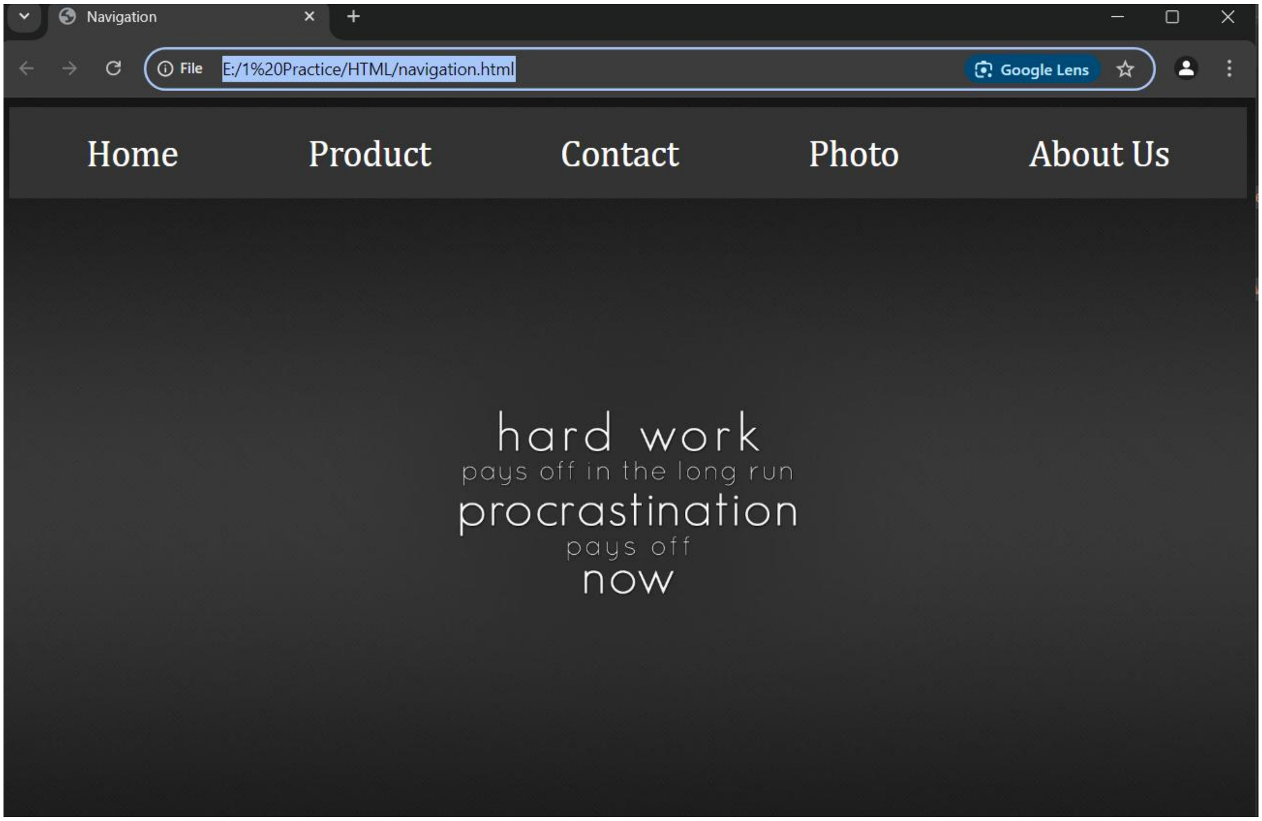
- Create a <nav> Element: The <nav> element will serve as the container for your navigation bar. Inside it, you'll use an unordered list (<ul>) with list items (<li>) for each navigation link.
- Use CSS Flexbox for Layout: CSS Flexbox allows you to arrange the list items horizontally and control their spacing and alignment.
- Add Styling for Links: Each navigation link (<a>) inside the list items will be styled for color, spacing, and hover effects to improve interactivity.
- Make the Navbar Responsive: Media queries will help make the navigation bar responsive, adjusting the layout on smaller screens by switching from a horizontal layout to a vertical one.

Source Code :

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Navigation</title>
   <style>
      body {
         font-family: Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;
         font-size: 30px;
         background-image: url(4.jpg);
         background-size: cover;
      }
     .nav_bar {
      background-color: #333;
```

```
        display: flex;
        justify-content: space-around;
        align-items: center;
        padding: 10px;
       }
        .nav_bar a {
            color: #fff;
            padding: 10px;
            text-decoration: none;
        }
        .nav_bar a:hover {
            background-color: #ffffff;
            color: #000000;
            transition: background-color 0.3s ease;
            cursor: pointer;
            box-shadow: 0 0 5px rgba(133, 130, 130, 0.2);
            box-sizing: border-box;
            border: 1px solid rgba(255, 255, 255, 0.2);
            border-radius: 5px;

        }
    </style>
</head>
<body>
    <section>
        <div class="nav_bar">
            <a href="#">Home</a>
            <a href="#">Product</a>
            <a href="#">Contact</a>
            <a href="#">Photo</a>
            <a href="#">About Us</a>
        </div>
    </section>

</body>
</html>
```

Output :



hard work
pays off in the long run
procrastination
pays off
now

Conclusion : After completing this assignment we can create navigation bar and style it using CSS.

Assignment 11

Aim : Write a java script program to accept a number form user and calculate and display its sum of digits.

Theory :

Javascript Data Types

JavaScript provides different data types to hold different types of values.

There are two types of data types in JavaScript.

Primitive data type

Non-primitive (reference) data type

JavaScript is a dynamic type language, means you don't need to specify type of the variable because it is dynamically used by JavaScript engine. You need to use var here to specify the data type. It can hold any type of values such as numbers, strings etc.

For example:

```
var N=10;          //holding number
var S="Pune";      //holding string
```

primitive data types

There are five types of primitive data types in JavaScript. They are as follows:

Data Type	**Description**
String	represents sequence of characters e.g. "hello"
Number	represents numeric values e.g. 100
Boolean	represents boolean value either false or true
Undefined	represents undefined value
Null	represents null i.e. no value at all

non-primitive data types

The non-primitive data types are as follows:

Data Type	**Description**
Object	represents instance through which we can access members
Array	represents group of similar values
RegExp	represents regular expression

JavaScript Operators

JavaScript operators are symbols that are used to perform operations on operands.

For example:

```
var sum=10+20;
```

Here, + is the arithmetic operator and = is the assignment operator.

There are following types of operators in JavaScript.

1. Arithmetic Operators
2. Comparison (Relational) Operators
3. Bitwise Operators
4. Logical Operators
5. Assignment Operators
6. Special Operators

JavaScript Arithmetic Operators

Arithmetic operators are used to perform arithmetic operations on the operands. The following operators are known as JavaScript 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===”A”= 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 Bitwise Operators

The bitwise operators perform bitwise operations on operands. The bitwise operators are as follows:

Operator	Description	Example
&	Bitwise AND	(10==20 & 20==33) = false
|	Bitwise OR	(10==20 | 20==33) = false
^	Bitwise XOR	(10==20 ^ 20==33) = false
~	Bitwise NOT	(~10) = -10
<<	Bitwise Left Shift	(10<<2) = 40
>>	Bitwise Right Shift	(10>>2) = 2
>>>	Bitwise Right Shift with Zero	(10>>>2) = 2

JavaScript Logical Operators

The following operators are known as 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 Special Operators

The following operators are known as JavaScript special operators.

Operator	Description
(?:)	Conditional Operator returns value based on the condition. It is like if-else.
,	Comma Operator allows multiple expressions to be evaluated as single statement.
delete	Delete Operator deletes a property from the object.
in	In Operator checks if object has the given property
instanceof	checks if the object is an instance of given type
new	creates an instance (object)
typeof	checks the type of object.
void	it discards the expression's return value.
yield	checks what is returned in a generator by the generator's iterator.

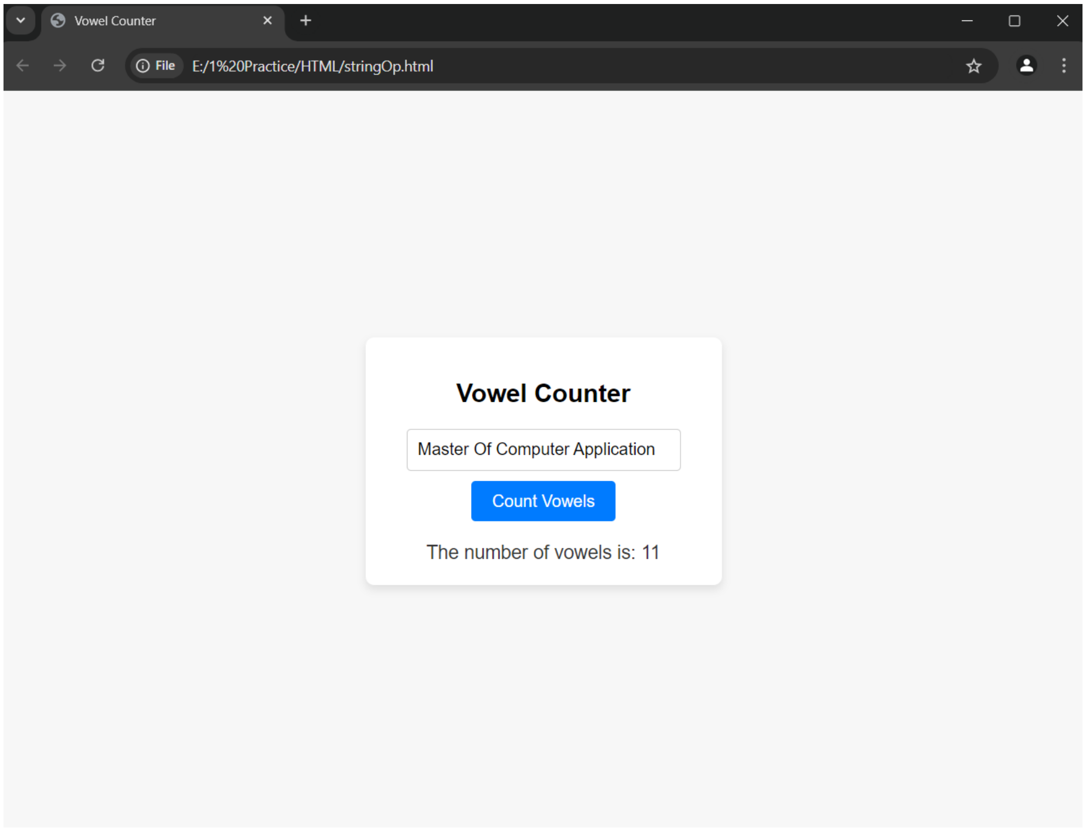
Source Code

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

```
<title>Sum of Digits</title>
<style>
    body {
        font-family: Arial, sans-serif;
        background-color: #f0f8ff;
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
        margin: 0;
    }
    .container {
        text-align: center;
        background: #ffffff;
        padding: 20px;
        border-radius: 8px;
        box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
        width: 300px;
    }
    input {
        padding: 10px;
        font-size: 16px;
        width: 80%;
        margin-bottom: 10px;
        border: 1px solid #ccc;
        border-radius: 4px;
    }
    button {
        padding: 10px 20px;
        font-size: 16px;
        color: #fff;
        background-color: #007bff;
        border: none;
        border-radius: 4px;
        cursor: pointer;
    }
    button:hover {
        background-color: #0056b3;
    }
    .result {
        margin-top: 20px;
        font-size: 18px;
```

```
            color: #333;
        }
    </style>
</head>
<body>
    <div class="container">
        <h2>Sum of Digits Calculator</h2>
        <input type="text" id="numberInput" placeholder="Enter a number" />
        <button onclick="calculateSum()">Calculate</button>
        <div class="result" id="result"></div>
    </div>

    <script>
        function calculateSum() {
            const input = document.getElementById("numberInput").value;
            if (!/^\d+$/.test(input)) {
                document.getElementById("result").innerText = "Please enter a valid
number.";
                return;
            }

            let sum = 0;
            for (let char of input) {
                sum += parseInt(char, 10);
            }

            document.getElementById("result").innerText = `The sum of digits is: ${sum}`;
        }
    </script>
</body>
</html>
```

Output :



Vowel Counter

Master Of Computer Application

The number of vowels is: 11

Conclusion : After completing this assignment students will understand the data types and operators provided by JavaScript.

Assignment 12

Aim : Write a java script program to accept a string from user and display the count of vowel characters from that string.

Theory :

The JavaScript string is an object that represents a sequence of characters.

There are 2 ways to create string in JavaScript

By string literal
By string object (using new keyword)

1) By string literal

The string literal is created using double quotes. The syntax of creating string using string literal is given below:

```
var stringname="string value";
```

for example

```
<script>
        var str="This is string literal";
        document.write(str);
</script>
```

Output: This is string literal

2) By string object (using new keyword)

The syntax of creating string object using new keyword is given below:

```
var stringname=new String("string literal");
```

Here, new keyword is used to create instance of string.

For example

```
<script>
        var stringname=new String("hello javascript string");
        document.write(stringname);
</script>
```

Output: hello javascript string

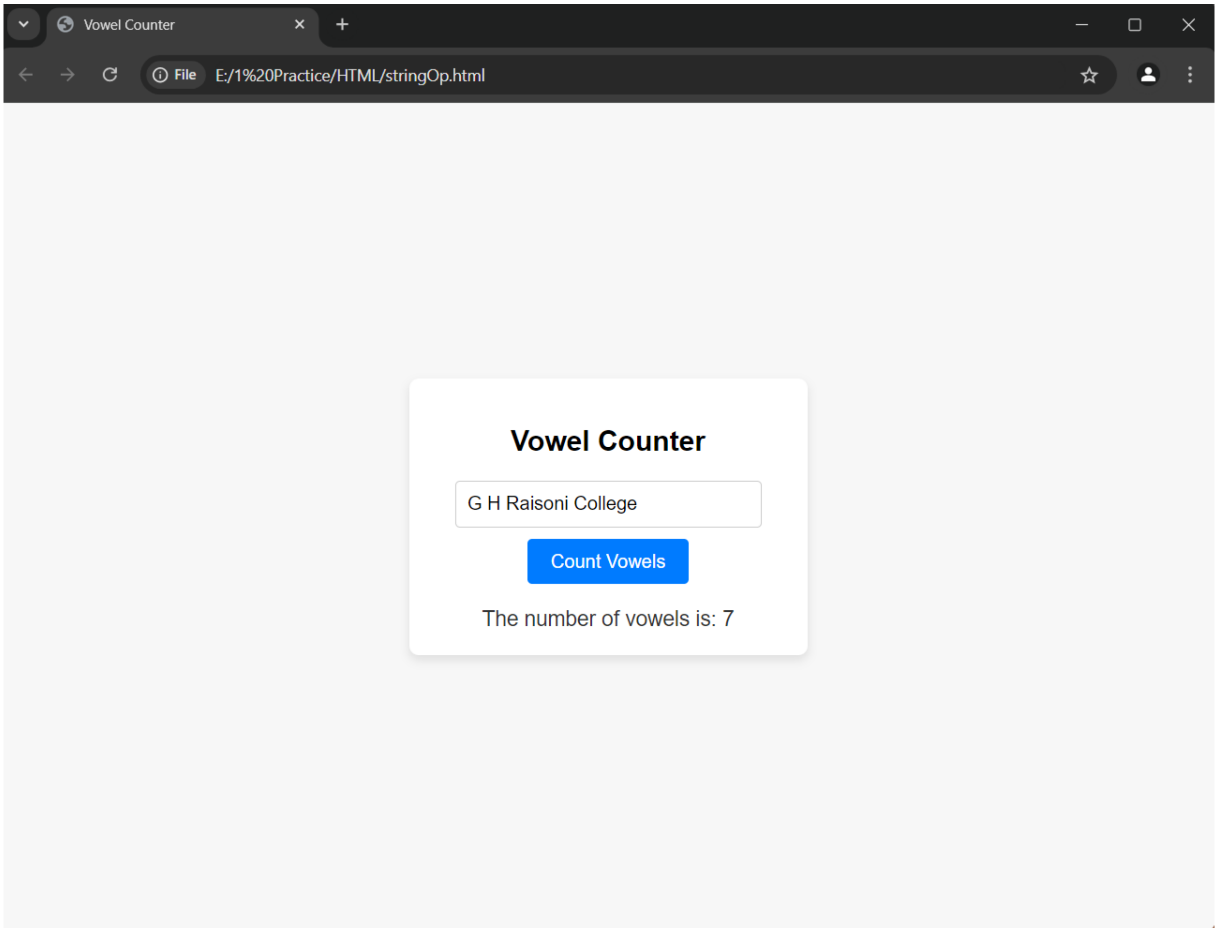
JavaScript String Methods

MethodsMethod Name	**Description**
charAt()	It provides the char value present at the specified index.
charCodeAt()	It provides the Unicode value of a character present at the specified index.
concat()	It provides a combination of two or more strings.
indexOf()	It provides the position of a char value present in the given string.
lastIndexOf()	It provides the position of a char value present in the given string by searching a character from the last position.
search()	It searches a specified regular expression in a given string and returns its position if a match occurs.
match()	It searches a specified regular expression in a given string and returns that regular expression if a match occurs.
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.
toLowerCase()	It converts the given string into lowercase letter.
toLocaleLowerCase()	It converts the given string into lowercase letter on the basis of host?s current locale.
toUpperCase()	It converts the given string into uppercase letter.
toLocaleUpperCase()	It converts the given string into uppercase letter on the basis of host?s current locale.
toString()	It provides a string representing the particular object.
valueOf()	It provides the primitive value of string object.
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.

Source Code

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Vowel Counter</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f7f7f7;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            margin: 0;
        }
        .container {
            text-align: center;
            background: #ffffff;
            padding: 20px;
            border-radius: 8px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
            width: 300px;
        }
        input {
            padding: 10px;
            font-size: 16px;
            width: 80%;
            margin-bottom: 10px;
            border: 1px solid #ccc;
            border-radius: 4px;
        }
        button {
            padding: 10px 20px;
            font-size: 16px;
            color: #fff;
            background-color: #007bff;
            border: none;
            border-radius: 4px;
            cursor: pointer;
        }
        button:hover {
            background-color: #0056b3;
        }
        .result {
            margin-top: 20px;
            font-size: 18px;
            color: #333;
        }
    </style>
</head>
<body>
    <div class="container">
        <h2>Vowel Counter</h2>
        <input type="text" id="stringInput" placeholder="Enter a string" />
        <button onclick="countVowels()">Count Vowels</button>
        <div class="result" id="result"></div>
    </div>

    <script>
        function countVowels() {
            const input = document.getElementById("stringInput").value;
            if (!input) {
                document.getElementById("result").innerText = "Please enter a string.";
                return;
            }

            const vowels = 'aeiouAEIOU';
            let count = 0;

            for (let char of input) {
                if (vowels.includes(char)) {
                    count++;
                }
            }

            document.getElementById("result").innerText = `The number of vowels is:
${count}`;
        }
    </script>
</body>
</html>
```

Output :



Vowel Counter

G H Raisoni College

The number of vowels is: 7

Conclusion : String operation is demonstrated using above program.

Assignment 13

Aim : Write a JavaScript Program to read a number from user, store its factors into the array and display that array. (Handle on Click Event).

Theory :

JavaScript Events

The change in the state of an object is known as an Event. In html, there are various events which represents that some activity is performed by the user or by the browser. When javascript code is included in HTML, js react over these events and allow the execution. This process of reacting over the events is called Event Handling. Thus, js handles the HTML events via Event Handlers.

For example, when a user clicks over the browser, add js code, which will execute the task to be performed on the event.

Event Handler Uses:

It can be used directly within HTML elements by adding special attributes to those elements. They can also be used within the <script> tags or in external JavaScript files.

Some of the HTML events and their event handlers are:

Mouse events:

Event Performed	**Event Handler**	**Description**
click	onclick	When mouse click on an element
mouseover	onmouseover	When the cursor of the mouse comes over the element
mouseout	onmouseout	When the cursor of the mouse leaves an element
mousedown	onmousedown	When the mouse button is pressed over the element
mouseup	onmouseup	When the mouse button is released over the element
mousemove	onmousemove	When the mouse movement takes place.

Keyboard events:

Event Performed	Event Handler	Description
Keydown & Keyup	onkeydown & onkeyup	When the user press and then release the key

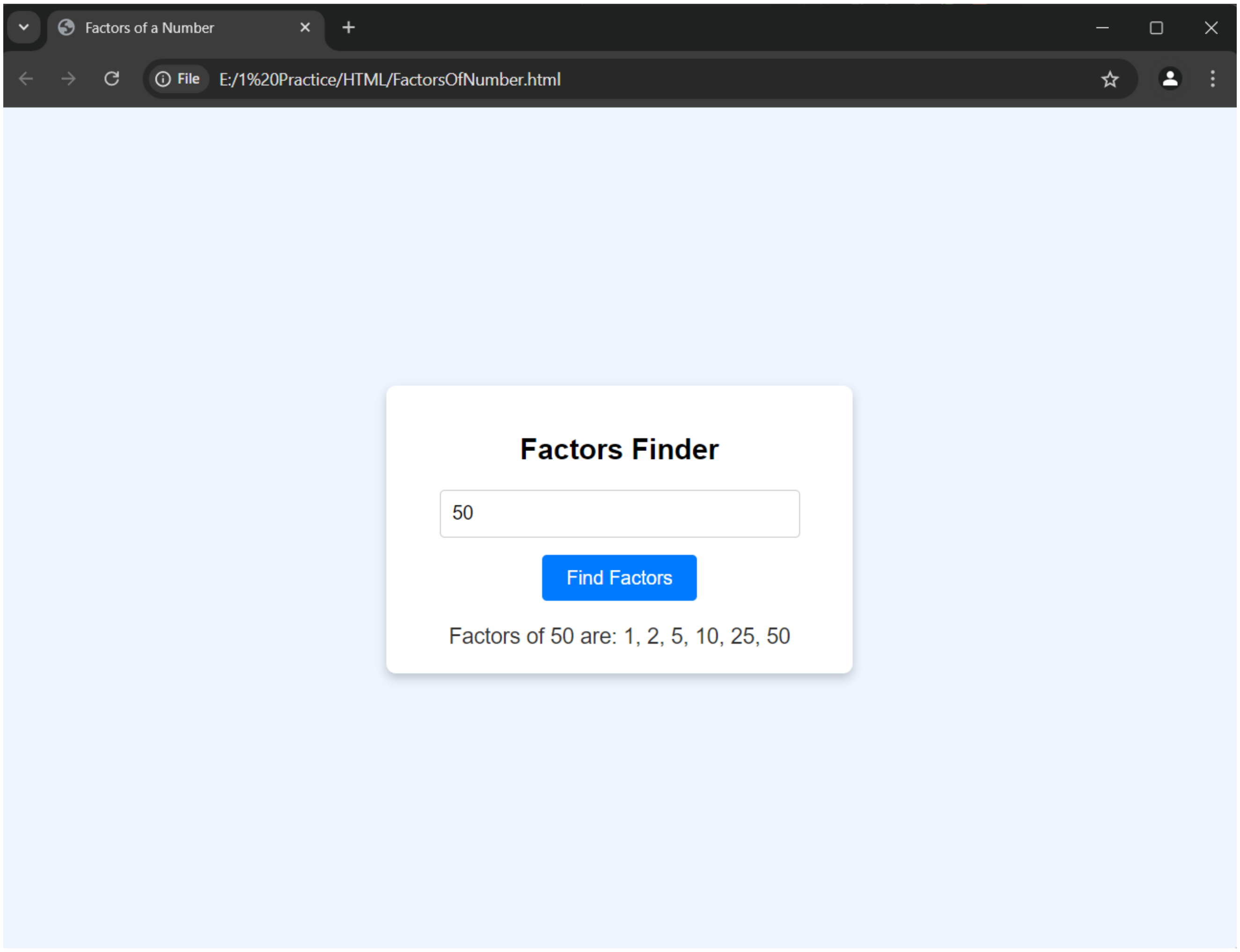
Form events:

Event Performed	Event Handler	Description
focus	onfocus	When the user focuses on an element
submit	onsubmit	When the user submits the form
blur	onblur	When the focus is away from a form element
change	onchange	When the user modifies or changes the value of a form element

Source Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Factors of a Number</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #eef5ff;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            margin: 0;
        }
        .container {
            background: #ffffff;
            padding: 20px;
            border-radius: 8px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
            text-align: center;
            width: 350px;
        }
        input {
            padding: 10px;
            font-size: 16px;
            width: 80%;
            margin-bottom: 15px;
            border: 1px solid #ccc;
            border-radius: 4px;
        }
        button {
            padding: 10px 20px;
            font-size: 16px;
            color: #fff;
            background-color: #007bff;
            border: none;
            border-radius: 4px;
            cursor: pointer;
        }
        button:hover {
            background-color: #0056b3;
        }
        .result {
            margin-top: 20px;
            font-size: 18px;
            color: #333;
        }
    </style>
</head>
<body>
    <div class="container">
        <h2>Factors Finder</h2>
        <input type="text" id="numberInput" placeholder="Enter a number" />
        <button onclick="findFactors()">Find Factors</button>
        <div class="result" id="result"></div>
    </div>

    <script>
        function findFactors() {
            const input = document.getElementById("numberInput").value;
            const resultDiv = document.getElementById("result");

            // Validate the input
            if (!/^\d+$/.test(input) || parseInt(input) <= 0) {
                resultDiv.innerText = "Please enter a valid positive number.";
                return;
            }

            const number = parseInt(input);
            const factors = [];

            // Find factors
            for (let i = 1; i <= number; i++) {
                if (number % i === 0) {
                    factors.push(i);
                }
            }

            // Display factors
            resultDiv.innerText = `Factors of ${number} are: ${factors.join(", ")}`;
        }
    </script>
</body>
</html>
```

Output:



Factors Finder

50

Factors of 50 are: 1, 2, 5, 10, 25, 50

Conclusion : Event handling using HTML, CSS & JavaScript is demonstrated.

Assignment 14

Aim : Write a JavaScript Program to design a digital clock & Calculator

Theory : Form basics

To create a form in HTML, you use the <form> element:

```
<form action="/signup" method="post" id="signup">
</form>
```

Code language: HTML, XML (xml)

The <form> element has two important attributes: action and method.

The action attribute specifies a URL that will process the form submission. In this example, the action is the /signup URL.

The method attribute specifies the HTTP method to submit the form with. Usually, the method is either post or get.

Generally, you use the get method when you want to retrieve data from the server and the post method when you want to change data on the server.

JavaScript uses the HTMLFormElement object to represent a form. The HTMLFormElement has the following properties that correspond to the HTML attributes:

action – is the URL that processes the form data. It is equivalent to the action attribute of the <form> element.

method – is the HTTP method which is equivalent to the method attribute of the <form> element.

The HTMLFormElement element also provides the following useful methods:

submit() – submit the form.

reset() – reset the form.

Referencing forms

To reference the <form> element, you can use DOM selecting methods such as getElementById():

```
const form = document.getElementById('subscribe');
```

Code language: JavaScript (javascript)

An HTML document can have multiple forms. The document.forms property returns a collection of forms (HTMLFormControlsCollection) on the document:

```
document.forms
```

Code language: JavaScript (javascript)

To reference a form, you use an index. For example, the following statement returns the first form of the HTML document:

```
document.forms[0]
```

Code language: CSS (css)

Submitting a form

Typically, a form has a submit button. When you click it, the browser sends the form data to the server. To create a submit button, you use <input> or <button> element with the type submit:

```
<input type="submit" value="Subscribe">
```

Code language: HTML, XML (xml)

Or

```
<button type="submit">Subscribe</button>
```

Code language: HTML, XML (xml)

If the submit button has focus and you press the Enter key, the browser also submits the form data.

When you submit the form, the submit event is fired before the request is sent to the server. This gives you a chance to validate the form data. If the form data is invalid, you can stop submitting the form.

To attach an event listener to the submit event, you use the addEventListener() method of the form element as follows:

```
const form  = document.getElementById('signup');

form.addEventListener('submit', (event) => {
   // handle the form data
});
```

Code language: JavaScript (javascript)

To stop the form from being submitted, you call the preventDefault() method of the event object inside the submit event handler like this:

```
form.addEventListener('submit', (event) => {
   // stop form submission
   event.preventDefault();
});
```

Code language: PHP (php)

Typically, you call the event.preventDefault() method if the form data is invalid. To submit the form in JavaScript, you call the submit() method of the form object:

```
form.submit();
```

Code language: CSS (css)

Note that the form.submit() does not fire the submit event. Therefore, you should always validate the form before calling this method.

Accessing form fields

To access form fields, you can use DOM methods like getElementsByName(), getElementById(), querySelector(), etc.

Also, you can use the elements property of the form object. The form.elements property stores a collection of the form elements.

Syntax

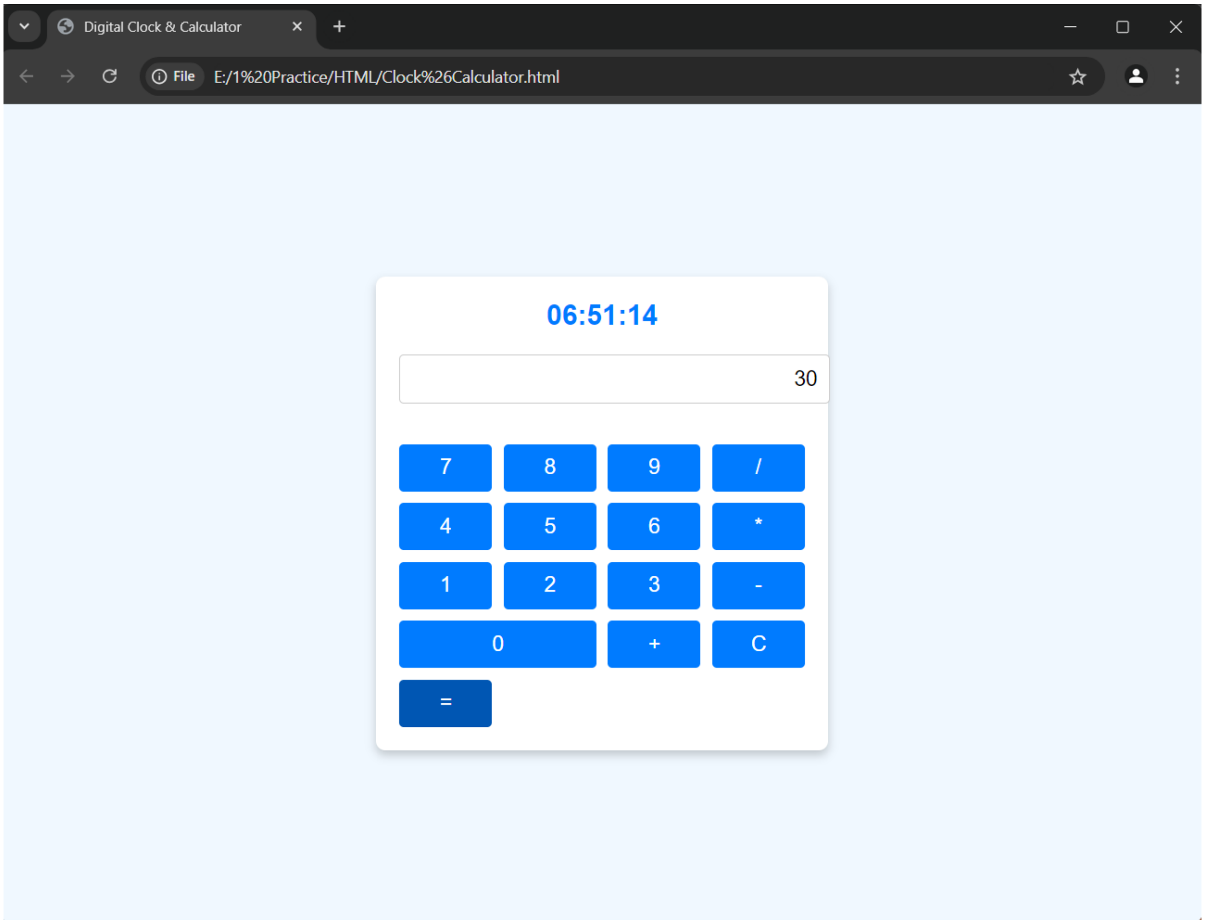
Source Code Code :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Digital Clock & Calculator</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f0f8ff;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }
    .container {
      background: #ffffff;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
      text-align: center;
      width: 350px;
    }
    .clock {
      font-size: 24px;
      font-weight: bold;
      margin-bottom: 20px;
      color: #007bff;
    }
    .calculator {
      display: grid;
      grid-template-columns: repeat(4, 1fr);
      gap: 10px;
      margin-top: 20px;
    }
    input {
      font-size: 18px;
      padding: 10px;
      width: 100%;
      margin-bottom: 15px;
      border: 1px solid #ccc;
      border-radius: 4px;
      text-align: right;
    }
    button {
      font-size: 18px;
      padding: 10px;
      background-color: #007bff;
      color: white;
      border: none;
      border-radius: 4px;
      cursor: pointer;
    }
    button:hover {
      background-color: #0056b3;
    }
    .wide {
      grid-column: span 2;
    }
  </style>
</head>
<body>
  <div class="container">
    <!-- Digital Clock -->
    <div class="clock" id="digitalClock">00:00:00</div>
```

```
      <!-- Calculator -->
      <input type="text" id="calcDisplay" readonly placeholder="0" />
      <div class="calculator">
        <button onclick="appendNumber('7')">7</button>
        <button onclick="appendNumber('8')">8</button>
        <button onclick="appendNumber('9')">9</button>
        <button onclick="setOperation('/')">/</button>
        <button onclick="appendNumber('4')">4</button>
        <button onclick="appendNumber('5')">5</button>
        <button onclick="appendNumber('6')">6</button>
        <button onclick="setOperation('*')">*</button>
        <button onclick="appendNumber('1')">1</button>
        <button onclick="appendNumber('2')">2</button>
        <button onclick="appendNumber('3')">3</button>
        <button onclick="setOperation('-')">-</button>
        <button class="wide" onclick="appendNumber('0')">0</button>
        <button onclick="setOperation('+')">+</button>
        <button onclick="clearDisplay()">C</button>
        <button onclick="calculate()">=</button>
      </div>
    </div>

    <script>
      // Digital Clock
      function updateClock() {
        const now = new Date();
        const hours = String(now.getHours()).padStart(2, '0');
        const minutes = String(now.getMinutes()).padStart(2, '0');
        const seconds = String(now.getSeconds()).padStart(2, '0');
        document.getElementById("digitalClock").innerText =
  `${hours}:${minutes}:${seconds}`;
      }
      setInterval(updateClock, 1000);
      updateClock();

      // Calculator Logic
      let currentInput = '';
      let currentOperation = '';
      let previousValue = '';

      function appendNumber(number) {
        currentInput += number;
```

```
            document.getElementById("calcDisplay").value = currentInput;
        }

        function setOperation(operation) {
            if (currentInput === '') return;
            previousValue = currentInput;
            currentOperation = operation;
            currentInput = '';
        }

        function clearDisplay() {
            currentInput = '';
            currentOperation = '';
            previousValue = '';
            document.getElementById("calcDisplay").value = '0';
        }

        function calculate() {
            if (currentInput === '' || previousValue === '' || currentOperation === '')
    return;
            let result;
            const num1 = parseFloat(previousValue);
            const num2 = parseFloat(currentInput);

            switch (currentOperation) {
                case '+':
                    result = num1 + num2;
                    break;
                case '-':
                    result = num1 - num2;
                    break;
                case '*':
                    result = num1 * num2;
                    break;
                case '/':
                    result = num2 !== 0 ? num1 / num2 : 'Error';
                    break;
                default:
                    return;
            }

            document.getElementById("calcDisplay").value = result;
            currentInput = result.toString();
            previousValue = '';
            currentOperation = '';
        }
    </script>
</body>
</html>
```

Output:


Digital Clock & Calculator
File
E:/1%20Practice/HTML/Clock%26Calculator.html
06:51:14
30
7
8
9
/
4
5
6
*
1
2
3
-
0
+
C
=

Conclusion : Digital clock and calculator is designed using HTML, CSS & JavaScript

Assignment 15

Aim :

To understand the basic features, functionalities, and applications of Power BI for data visualization and analysis.

Theory :

Power BI is a powerful business analysis tool developed by Microsoft that allows users to connect, transform, and visualize data. The tool supports a range of data sources and enables interactive dashboards and reports.

Some key components of Power BI include:

- **Power BI Desktop**: The primary application for creating reports and visualizations.
- **Power BI Service**: An online platform for sharing reports.
- **Power BI Mobile**: Applications for accessing reports on mobile devices.

Key features include:

1. **Data Connections**: Integrate multiple data sources like Excel, SQL Server, and APIs.
2. **Data Transformation**: Use Power Query to clean and shape data.
3. **Interactive Visualizations**: Create dashboards with slicers, filters, and charts.
4. **AI Insights**: Leverage AI-powered tools for trend analysis and forecasting.

Syntax

In Power BI, expressions are used for calculations and data analysis. Below are common syntax examples:

1. Creating a Measure:

```
Total Sales = SUM(Sales[Amount])
```

2. Conditional Statement:

```
Profit Margin = IF(Sales[Amount] > 1000, "High", "Low")
```

3. Date Calculations:

```
Sales Last Year = CALCULATE(SUM(Sales[Amount]),
SAMEPERIODLASTYEAR(Sales[Date]))
```

Source Code:

Provide a small Power BI project example:

1. Sample Data Source: Excel file with the following columns:
   a. Date
   b. Product
   c. Sales Amount
   d. Region
2. Steps to Create Visualization:

a. Import the Excel file into Power BI.
b. Use the "Table" visual to display raw data.
c. Create a measure for "Total Sales":

Total Sales = SUM(Sales[Amount])

d. Use the "Bar Chart" visual to show sales by product.
e. Add a slicer for regions.

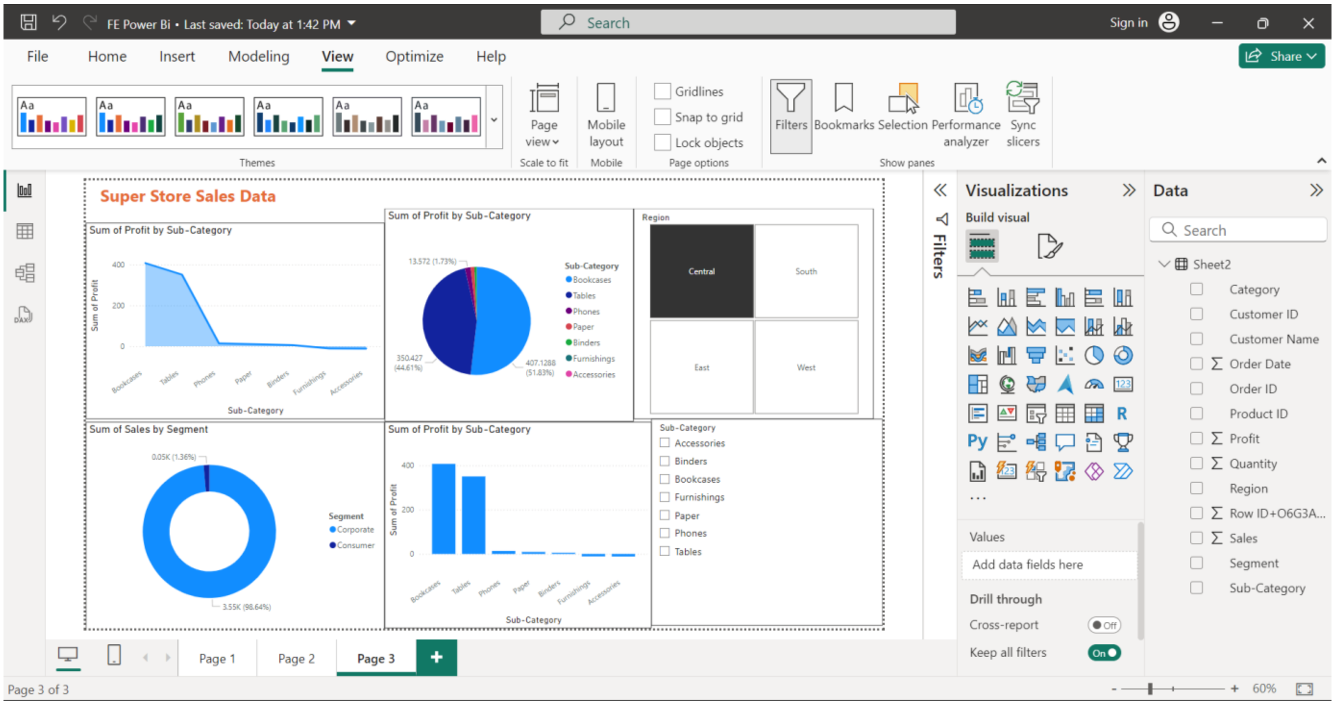
Order ID	**Order Date**	**Ship Date**	**Ship Mode**	**Customer ID**	**Customer Name**	**Segment**	**Region**	**Product ID**	**Category**	**Sub-Category**	**Sales**	**Quantity**	**Profit**
CA-2019-160304	43466	43472	Standard Class	BM-11575	Brendan Murry	Corporate	East	FUR-BO-10004709	Furniture	Bookcases	73.94	1	28.2668
CA-2019-160304	43467	43472	Standard Class	BM-11575	Brendan Murry	Corporate	East	FUR-BO-10004709	Furniture	Bookcases	173.94	3	38.2668
CA-2019-160304	43467	43472	Standard Class	BM-11575	Brendan Murry	Corporate	East	TEC-PH-10000455	Technology	Phones	231.98	2	67.2742
CA-2019-125206	43468	43470	First Class	LR-16915	Lena Radford	Consumer	West	OFF-ST-10003692	Office Supplies	Storage	114.46	2	28.615
US-2019-116365	43468	43473	Standard Class	CA-12310	Christine Abelman	Corporate	Central	TEC-AC-10002217	Technology	Accessories	30.08	2	-5.264
US-	4346	434	Standa	CA-	Christin	Corpor	Centr	TEC-	Technol	Accessor	165.6	3	-6.21

2019-116365	8	73	rd Class	12310	e Abelman	ate	al	AC-10002942	ogy	ies			
US-2019-116365	43468	43473	Standard Class	CA-12310	Christine Abelman	Corporate	Central	TEC-PH-10002890	Technology	Phones	180.96	5	13.572
CA-2019-105207	43468	43473	Standard Class	BO-11350	Bill Overfelt	Corporate	Central	FUR-TA-10000617	Furniture	Tables	1592.85	7	350.427
CA-2019-105207	43468	43473	Standard Class	BO-11350	Bill Overfelt	Corporate	Central	OFF-BI-10004364	Office Supplies	Binders	11.88	2	5.346
US-2019-164630	43469	43474	Standard Class	EB-13975	Erica Bern	Corporate	South	TEC-CO-10000971	Technology	Copiers	959.968	4	119.996
CA-2019-158211	43469	43473	Standard Class	BP-11185	Ben Peterman	Corporate	East	OFF-AR-10004078	Office Supplies	Art	4.672	1	0.584
CA-2019-158211	43469	43473	Standard Class	BP-11185	Ben Peterman	Corporate	East	OFF-BI-10002026	Office Supplies	Binders	104.58	6	-80.178
CA-2019-134474	43470	43472	Second Class	AJ-10795	Anthony Johnson	Corporate	South	TEC-AC-10001714	Technology	Accessories	191.472	6	40.6878
CA-2019-134474	43470	43472	Second Class	AJ-10795	Anthony Johnson	Corporate	South	OFF-AR-10003958	Office Supplies	Art	5.248	2	0.5904
CA-2019-134474	43470	43472	Second Class	AJ-10795	Anthony Johnson	Corporate	South	TEC-PH-10002923	Technology	Phones	59.184	2	5.1786
CA-2019-101938	43472	43477	Standard Class	DW-13480	Dianna Wilson	Home Office	West	OFF-AR-10003696	Office Supplies	Art	34.58	1	10.0282
CA-2019-158806	43472	43476	Standard Class	NM-18520	Neoma Murray	Consumer	Central	FUR-FU-10004270	Furniture	Furnishings	23.076	3	-10.9611

CA-2019-158806	43472	43476	Standard Class	NM-18520	Neoma Murray	Consumer	Central	OFF-PA-10004621	Office Supplies	Paper	25.92	5	9.072
US-2019-100461	43473	43477	Standard Class	JO-15145	Jack O'Briant	Corporate	Central	FUR-BO-10002545	Furniture	Bookcases	1565.88	6	407.1288

DATA USED FOR DASHBOARD

Output:-



Conclusion:

This assignment introduced Power BI's capabilities for data connection, transformation, and visualization. Through the use of , users can perform advanced calculations and create dynamic insights. The practical steps provided a hands-on understanding of developing interactive dashboards.