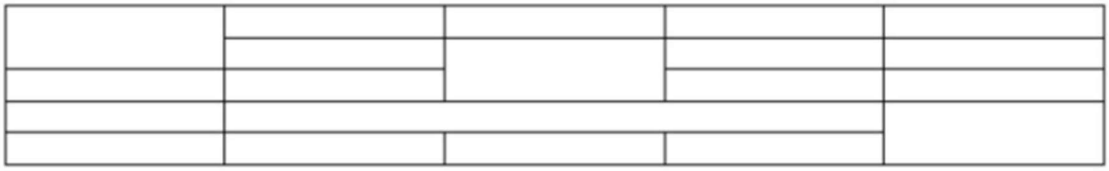
Name of the Student: Sai Balaji Section : CSE-2 Roll No.160121733130

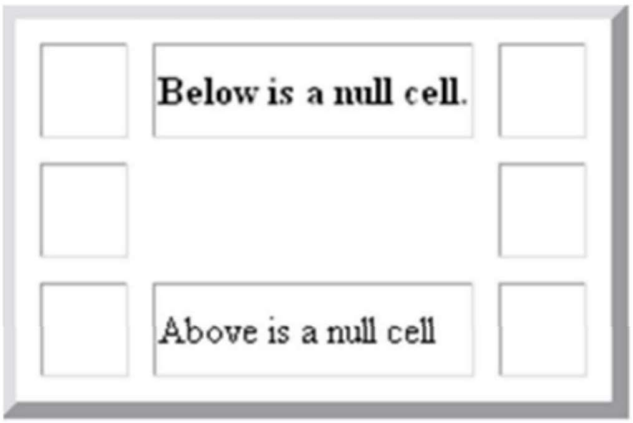
EXPERIMENT NUMBER- 01

Question-1:

AIM

To write the html code to display the following tables with some data.





Results of example code

DESCRIPTION

Tables are a useful way of organizing and presenting data in a structured format. HTML provides several elements for creating tables, including the <table>, <tr>, <td>, and< th> elements.
One common problem that tables can help solve is organizing and presenting data in a clear and concise manner. For example, if you have a large dataset of information, it can be challenging to make sense of itwithout some structure. Tables allow you to group related data together, making it easier to understand.

There are several HTML tags that are used to create tables, including:
<table>: This tag is used to create the table itself. It is a container tag that encloses all the other table elements, such as rows and cells.
<tr>: This tag stands for "table row" and is used to define a row of cells within the table.
<th>: This tag stands for "table header" and is used to define the header cells within a table. These cells are typically used to describe the data that is being presented in the table.
<td>: This tag stands for "table data" and is used to define the data cells within the table. These cells are used to display the actual data values.
<caption>: This tag is used to add a caption to the table, providing a brief description of the data being presented.

PROGRAM:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Question 1</title>
  </head>
  <body>
                <table align="center" border="1" cellspacing="0">
                  <tr>
                          <td rowspan="2">Kohli</td>
                          <td>XYZ </td>

                           <td> XYZ </td>
                          <td> XYZ </td>

                         <td> XYZ </td>
                  </tr>
                  <tr>
                          <td> XYZ </td>
```

```
            <td rowspan="2"> XYZ </td>
             <td> XYZ </td>
             <td> XYZ </td>
  </tr>
  <tr>


  <td> XYZ </td>
            <td> XYZ </td>
            <td> XYZ </td>
            <td> XYZ </td>
  </tr>
  <tr align="center">
            <td> XYZ </td>
            <td colspan="3"> XYZ </td>
            <td rowspan="2"> XYZ </td>
  </tr>

 <tr>
            <td> XYZ </td>
            <td> XYZ </td>
            <td> XYZ </td>
            <td> XYZ </td>
  </tr>
</table>

<br/>
<br/>


<table align="center" border="5" cellspacing="10">
  <tr align="center">
            <td>North<br/>West</td>
            <td><strong>Below is a null cell</strong></td>
            <td>North<br/>East</td>
  </tr>

 <tr align="center">
            <td>Mid<br/>West</td>
            <br/>
            <td>Mid<br/>East</td>
  </tr>
```

<tr align="center">

```
                        <td>South<br/>West</td>
                        <td>Above is a null cell</td>
                        <td>South<br/>East</td>
                    </tr>
                </table>
                <br/>
                <p align="center"><strong>Results of example code</strong></p>
                <table align="center" cellspacing="0" cellpadding="5">
                    <tr  align="center">
                        <td style="color: brown;background-color:
orange;"><strong>H<br/>T<br/>M<br/>L</strong></td>
                        <td style="color: orange;background-color: brown;">Mountain
Dragon<br/>WebDesigns</td>
                        <td style="color: brown;background-color: orange;"><pre>  </pre></td>
                    </tr>
                </table>
    </body>
</html>
```

OUTPUT:

<table>
<tr><td rowspan="2">zzz</td><td>HTML</td><td>HTML</td><td>HTML</td><td>HTML</td></tr>
<tr><td>HTML</td><td rowspan="2">HTML</td><td>HTML</td><td>HTML</td></tr>
<tr><td>HTML</td><td>HTML</td><td>HTML</td><td>HTML</td></tr>
<tr><td>HTML</td><td colspan="3">HTML</td><td rowspan="2">HTML</td></tr>
<tr><td>HTML</td><td>HTML</td><td>HTML</td><td>HTML</td></tr>
</table>

<table>
<tr><td></td><td>BELOW CELL IS NULL</td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td>ABOVE CELL IS NULL</td><td></td></tr>
</table>

<table>
<tr><td>H
T
M
L</td><td>Mountain Dragon
Web Designs</td></tr>
</table>

CONCLUSION:

The program illustrates the working and organizing the tabular data on a webpage using HTML tables. The first part displays a table containing several types of data, as well as rows and columns with cells that have been combined using the "rowspan" and "colspan" characteristics. To enhance the table's appearance and readability, borders and cell spacing have been added.The second part displays a different table with null and merged cells. Additionally, the table is designed with thicker borders and wider cell spacing.

Question-2:

AIM

To write the html program to display the following nested list.

Learning Web Development

I. Background Skills
    A. Unix Commands
    B. Vim Text Editor
II. HTML
    A. Minimal Page
    B. Headings
    C. Tags
    D. Lists
        i. Unordered
        ii. Ordered
        iii. Definition
        iv. Nested
    E. Links
        i. Absolute
        ii. Relative
    F. Images
III. CSS
    A. Anatomy
    B. Basic Selectors
        i. Element
        ii. Class
        iii. ID
        iv. Group
    C. The DOM
    D. Advanced Selectors
    E. Box Model
IV. Programming
    A. Python
    B. JavaScript
V. Database
    A. Flat File
    B. Relational

DESCRIPTION

The three forms of lists offered by HTML are ordered, unordered, and definition lists. These lists can be used to organise content organisation, making it simpler to read and comprehend.When information must be presented in a precise order, such as with numbered steps in a tutorial, ordered lists are employed. When the order of the items is not crucial, such as in a list of bullet points, unordered lists are utilised. When it's necessary to define terms, such as in a glossary, definition lists are employed. In HTML, lists are created

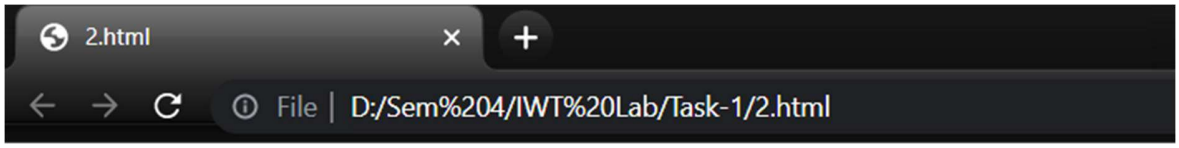
using the <ul>, <ol>, and <dl> tags for unordered, ordered, and definition lists, respectively. Within each list, individual list items are represented by the <li> tag. Lists can also be nested within each other to create more complex structures. The type attribute can be used to customize the appearance of the list markers.

PROGRAM

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Question 2</title>
  </head>
  <body>
    <h1>Learning Web Development</h1>
    <ol type="I">
      <li>Background Skills</li>
      <ol type="A">
        <li>Unix Commands</li>
        <li>Vim Text Editor</li>
      </ol>
      <li>HTML</li>
      <ol type="A">
        <li>Minimal Page</li>
        <li>Headings</li>
        <li>Tags</li>
        <li>Lists</li>
        <ol type="i">
          <li>Unordered</li>
  <li>Ordered</li>
          <li>Definition</li>
          <li>Nested</li>
        </ol>
```

```
            <li>Links</li>
            <ol type="i">

  <li>Absolute</li>
              <li>Relative</li>
            </ol>
   <li>Images</li>
          </ol>
          <li>CSS</li>
  <ol type="A">
            <li>Anatomy</li>
            <li>Basic Selectors</li>
            <ol type="i">
              <li>Element</li>
              <li>Class</li>
              <li>ID</li>
              <li>Group</li>
            </ol>
            <li>The DOM</li>
            <li>Advanced Selectors</li>
            <li>Box Model</li>
          </ol>
          <li>Programming</li>
          <ol type="A">
            <li>Python</li>
            <li>JavaScript</li>
          </ol>
          <li>Database</li>
          <ol type="A">
            <li>Flat File</li>
            <li>Relational</li>
          </ol>
        </ol>
    </body>
</html>
```

OUTPUT



Learning Web Development

I. Background Skills
- A. Unix Commands
- B. Vim Text Editor

II. HTML
- A. Minimal page
- B. Headings
- C. Tags
- D. Lists
  - i. Unordered
  - ii. Ordered
  - iii. Definition
  - iv. Nested
- E. Links
  - i. Absolute
  - ii. Relative
- F. Images

III. CSS
- A. Anatomy
- B. Basic Selectors
  - i. Element
  - ii. Class
  - iii. ID
  - iv. Group
- C. The DOM
- D. Advanced Selectors
- E. Box Model

IV. Programming
- A. Python
- B. JavaScript

V. Database
- A. Flat File
- B. Relational

CONCLUSION

The code exemplifies how to use nested lists in HTML. In HTML, lists are made using the ordered list (ol) and unordered list (ul) elements. The "type" attribute of the ordered list element can be changed to accommodate various numbering or lettering styles. Lists can also be nested inside of other lists to build a more sophisticated informational structure. To organise data about learning web programming, the code above combines sorted and nested unordered lists.

Question-3:

AIM

To write the html program to prepare your curriculum vitae.

DESCRIPTION

The given code presents a sample format for a Curriculum Vitae (CV) of an individual. This HTML document displays a curriculum vitae of a person. It contains details about the person's educational qualifications, IT-related skills, and strengths. The document uses various HTML tags such as <h1>, <pre>, <table>, <th>, <td>, , and to structure the content and format it appropriately. The <h1> tag is used to display the title of the document, while the <pre> tag is used to display the text in a preformatted manner. The <table>, <th>, and <td> tags are used to create tables and display tabular data.

PROGRAM

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
   <title>Curriculum Vitae</title>
</head>

<body bgcolor="pink" leftmargin="60">
   <h1 align="center"><u>CURRICULUM VITAE</u></h1>
   <br />
   <pre>SRIHAN REDDY          9059227066
HYDERABAD                srihanpasham@gmail.com</pre>
   <h4><u>Statement</u></h4>
   <code>A dedicated achiever who always gives his best on whatever he undertakes. He is committed to
completing anything he sets out to do. He aspires to be a researcher, analyst, developer, and
coder. He is a nature lover and enjoys reading and writing. He is a talented dancer and artist.</code>
   <br />
   <h4><u>Educational Qualifications</u></h4>
   <table border="5" bordercolor="white">
      <tr>
         <td><strong>Education</strong></td>
         <td><strong>School/University</strong></td>
         <td><strong>Percentage</strong></td>
         <td><strong>Passed</strong></td>
      </tr>
      <tr>
```

```
            <td>ICSE</td>
            <td>Johnson Grammar School</td>
            <td>90</td>
            <td>2019</td>
        </tr>
        <tr>
            <td>Intermediate</td>
            <td>Narayana Junior college</td>
            <td>95</td>
            <td>2021</td>
        </tr>
        <tr>
            <td>BE</td>
            <td>CBIT</td>
            <td>82</td>
            <td>2025</td>
        </tr>

    </table>
    <br />
    <h4><u>Technical Skills</u></h4>
    <table border="5" bordercolor="white">
        <tr>
            <td><strong>Skill</strong></td>
            <td><strong>Level</strong></td>
        </tr>
        <tr>
            <td> C-Language</td>
            <td>Intermediate</td>
        </tr>
        <tr>
            <td>Python</td>
            <td>Intermediate</td>
        </tr>
        <tr>
            <td>DSA</td>
            <td>Intermediate</td>
        </tr>
    </table>
    <br />
    <h4><u>Hobbies</u></h4>
    <ul>
        <li>Swimming</li>
        <li>Driving</li>
        <li>Playing games</li>
        <li>Coding</li>
```

```
    </ul>
    <p align="right">[Srihan Reddy]</p>
</body>

</html>
```

OUTPUT

CONCLUSION

The program includes a number of HTML elements that are used to format and organise the content, such as headings, paragraphs, tables, and lists. The strengths are displayed using the unordered list, whereas the educational credentials and IT skills are displayed using table elements. The content is center-aligned using the align attribute. The background colour of the body section can be changed using the bgcolor attribute.

EXPERIMENT NUMBER- 02

Question-1:

AIM:

To create a webpage with 3 frames. One is horizontal frame which displays the caption and two vertical frames where left frame contains 4 hyper links and right frame is empty.
For left frame: the hyper links unvisited color must be green, visited color must be red and active color must be violet. If the link is clicked, the resultant page should be displayed in right frame. For all three frames different light background color should be used. The web pages attached to the hyperlinks should have uniform styles.For attached web pages: foreground(text) color should be violet, bgcolor should be blue, font-face should be Times, order list type should be Roman.

DESCRIPTION

In this problem, we need to create a webpage with 3 frames, where one horizontal frame is used to display the caption, and two vertical frames are used to display the hyperlinks and resultant pages. The left frame will contain 4 hyperlinks with different colors for the unvisited, visited and active states. Clicking on the hyperlink should display the resultant page in the right frame. All three frames should have different background colors. The attached web pages should have uniform styles.

PROGRAM:

Index.xhtml:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
   <head>
      <title>Task-2(1)</title>
   </head>
   <frameset rows="30%, *">
      <frame name="f1" src="captions.xhtml" style="background-color: rgb(222, 142, 142);" />
      <frameset cols="50%,*">
         <frame name="f2" src="links.xhtml" style="background-color: rgb(199, 221, 153);" />
         <frame name="f3" style="background-color: rgb(134, 204, 214);" />
      </frameset>
   </frameset>
</html>
```

caption.xhtml:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
   <meta charset="utf-8">
   <head>
      <title>CAPTIONS</title>
   </head>
   <body style="background-color: rgb(100, 136, 167);">
      <p><strong> The Hyper Text Markup Language or HTML is the standard markup language for
documents designed to be displayed in a web browser.</strong></p>
   </body>
</html>
```

links.xhtml:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
   <head>
      <title>links</title>
   </head>
   <body align="center" link="green" alink="violet" vlink="red">
      <a href="link1.xhtml" target="f3">CLICK FOR LINK 1</a><br/><br/>
      <a href="link2.xhtml" target="f3">CLICK FOR LINK 2</a><br/><br/>
      <a href="link3.xhtml" target="f3">CLICK FOR LINK 3</a><br/><br/>
      <a href="link4.xhtml" target="f3">CLICK FOR LINK 4</a><br/><br/>

   </body>
</html>
```

link1.xhtml

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
   <head>
      <title>link1</title>
   </head>
   <body>
      <p><strong> It is often assisted by technologies such as Cascading Style Sheets (CSS) and scripting
languages such as JavaScript.</strong></p>

   </body>
</htm
```

link2.xhtml

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
   <head>
      <title>link2</title>
   </head>
   <body>
      <p><strong>Web browsers receive HTML documents from a web server or from local storage and
render the documents into multimedia web pages.</strong></p>

   </body>
</html>
```

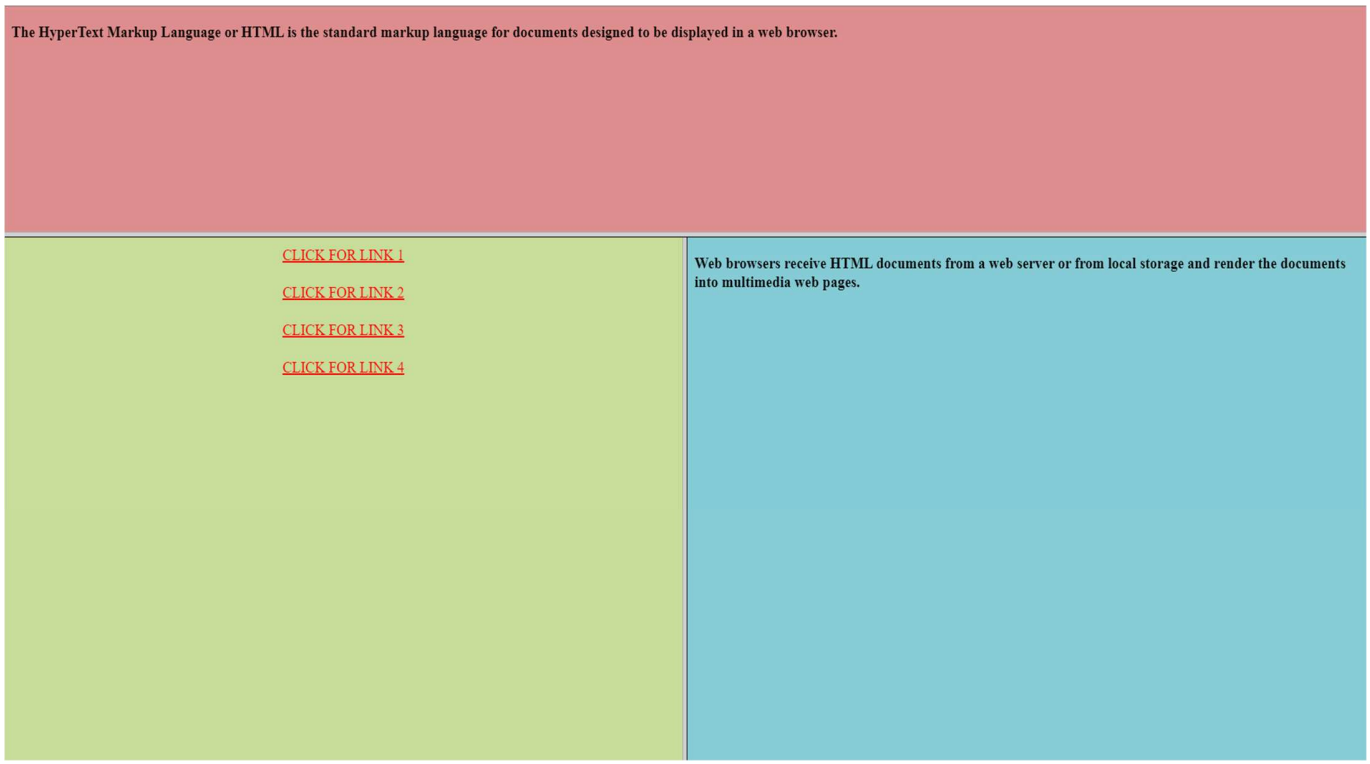
link3.xhtml

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
   <head>
      <title>link3</title>
   </head>
   <body>
      <p><strong>HTML elements are the building blocks of HTML pages.</strong></p>
   </body>
</html>
```

link4.xhtml

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
   <head>
      <title>link4</title>
   </head>
   <body>
      <p><strong> With HTML constructs, images and other objects such as interactive forms may be
embedded into the rendered page.</strong></p>

   </body>
</html>
```

OUTPUT



CONCLUSION

Thus, we have successfully designed a webpage with three frames using HTML, which contains hyperlinks with different colors for the unvisited, visited and active states, and the resultant pages are displayed in the right frame. The frames have different background colors, and the attached web pageshave uniform styles.

Question-2:

AIM

To design the following registration form

DESCRIPTION

- Create an HTML registration form that collects information from the user such as name, address, gender,state, district, city, pincode, courses, email id, DOB, and mobile number.
- The <form> tag is used to create the form element. Inside the form, various input elements such as <inputtype="text">, <input type="radio">, <select> and <input type="submit"> are used to collect user input.
- The <input type="text"> element is used to collect text input from the user. The <input type="radio"> element is used to create a radio button, which allows the user to select only one option from a group of options. The <select> element is used to create a dropdown list, which allows the user to select one optionfrom a list of options.

PROGRAM

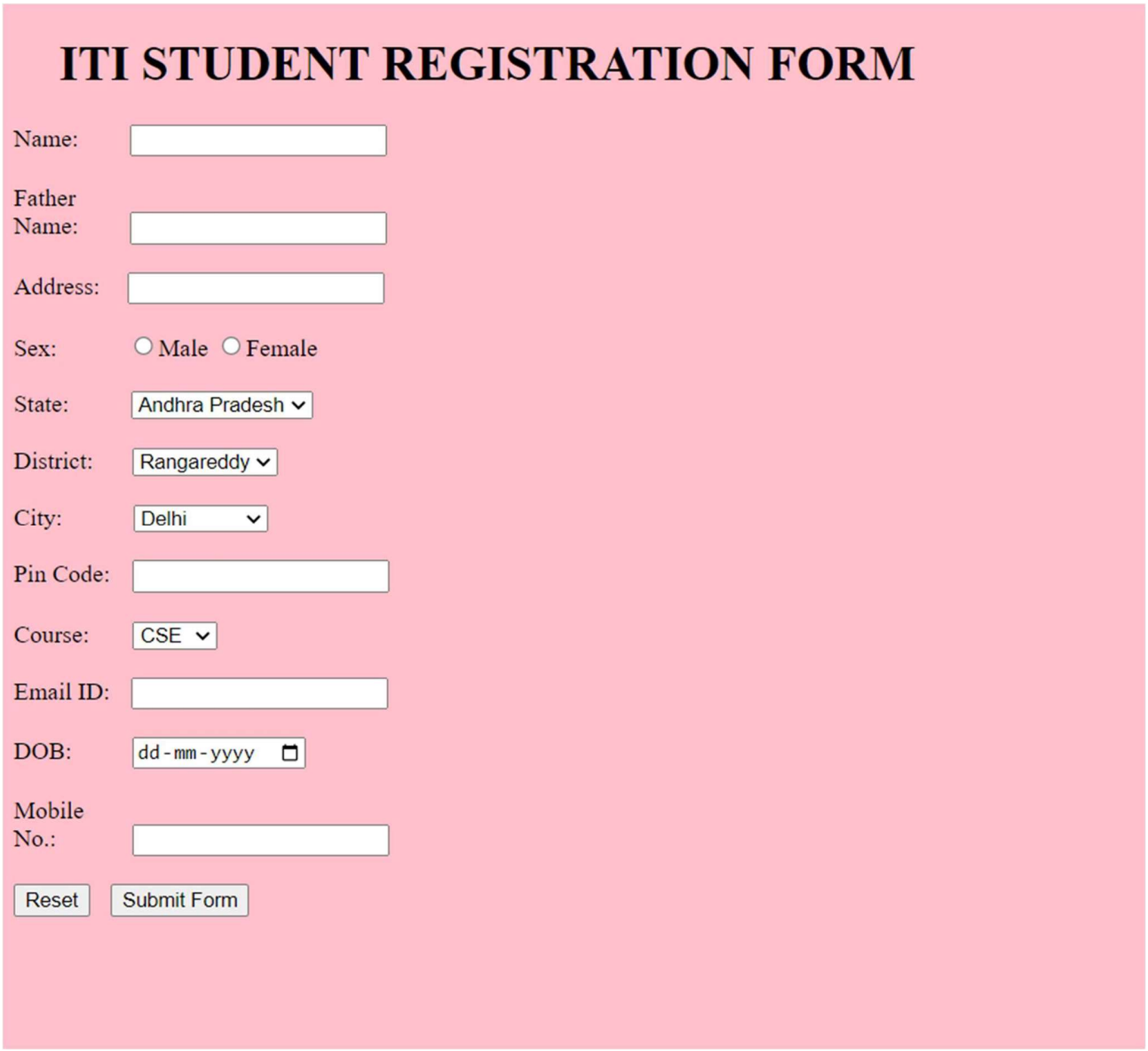
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Task-2(2)</title>
  </head>
  <body bgcolor="pink">
      <h1 style="margin-left:30px;" >ITI STUDENT REGISTRATION FORM</h1>
      <form action="" method="get">
            <label style="margin-right:30px;">Name:</label>
            <input type="text" name="Name"/>
            <br/>
            <br/>
            <label style="margin-right:30px;">Father<br/> Name:</label>
            <input type="text" name="Fathers Name"/>
            <br/>
            <br/>
            <label style="margin-right:14px;">Address:</label>



            <input type="text area" name="Address"/>
            <br/>
            <br/>
            <label style="margin-right:42px;">Sex:</label>
            <input type="radio" name="Sex" value="Male" />Male
            <input type="radio" name="Sex" value="Female" />Female
            <br/>
            <br/>
            <label style="margin-right:37px;">State:</label>
            <select name="State">
                <option value="Andhra Pradesh">Andhra Pradesh</option>
                  <option value="Bihar">Bihar</option>
                  <option value="Tamilnadu">Tamilnadu</option>
                  <option value="Telangana">Telangana</option>
            </select>
            <br/>
            <br/>
            <label style="margin-right:22px;">District:</label>
            <select name="District">
                <option value="Rangaredyy">Rangareddy</option>
                  <option value="Sangareddy">Sangareddy</option>
                  <option value="Adilabad">Adilabad</option>
```

```
        <option value="Khammam">Khammam</option>
</select>
<br/>
<br/>
<label style="margin-right:43px;">City:</label>
<select name="City">
      <option value="Delhi">Delhi</option>
         <option value="Bangalore">Bangalore</option>
         <option value="Hyderabad">Hyderabad</option>
         <option value="Chennai">Chennai</option>
</select>
<br/>
<br/>
<label style="margin-right:11px;">Pin Code:</label>
<input type="text" name="Pin Code"/>
<br/>
<br/>
<label style="margin-right:25px;">Course:</label>
<select name="Course">

          <option value="CSE">CSE</option>
         <option value="ECE">ECE</option>
         <option value="Mech">Mech</option>
         <option value="Civil">Civil</option>
</select>
<br/>
<br/>
<label style="margin-right:10px;">Email ID:</label>
<input type="email" name="Email ID"/>
<br/>
<br/>
<label style="margin-right:36px;">DOB:</label>
<input type="date" name="DOB"/>
<br/>
<br/>
<label style="margin-right:46px;">Mobile<br/> No.:</label>
<input type="text" name="Mobile NO."/>
<br/>
<br/>
<input style="margin-right:10px;" type="reset" value="Reset"/>
<input type="submit" value="Submit Form"/>
```

```
        </form>
    </body>
</html>
```

OUTPUT



CONCLUSION

The code demonstrates the creation of student's registration form.Name, Father's Name, Address, Sex, State, District, City, Pin Code, Course, Email ID, DOB, and Mobile Number are just a few of the entry fields on the form. The necessary data can be entered in the appropriate input fields, reset and submit buttons are used.Inline CSS is also used in the code to style some other elements, such as changing the background colour and giving the form labels some margins.

Question-3:

AIM

To design the following registration form using XHTML form

DESCRIPTION

This code creates a student registration form using HTML and CSS. The form contains various input fields such as text fields, radio buttons, select options, and date fields. Each input field has a label to describe the type of information to be entered. The form requires the student to enter their personal information, academic information, and contact information.The student must fill in all the required fields before submitting the form.

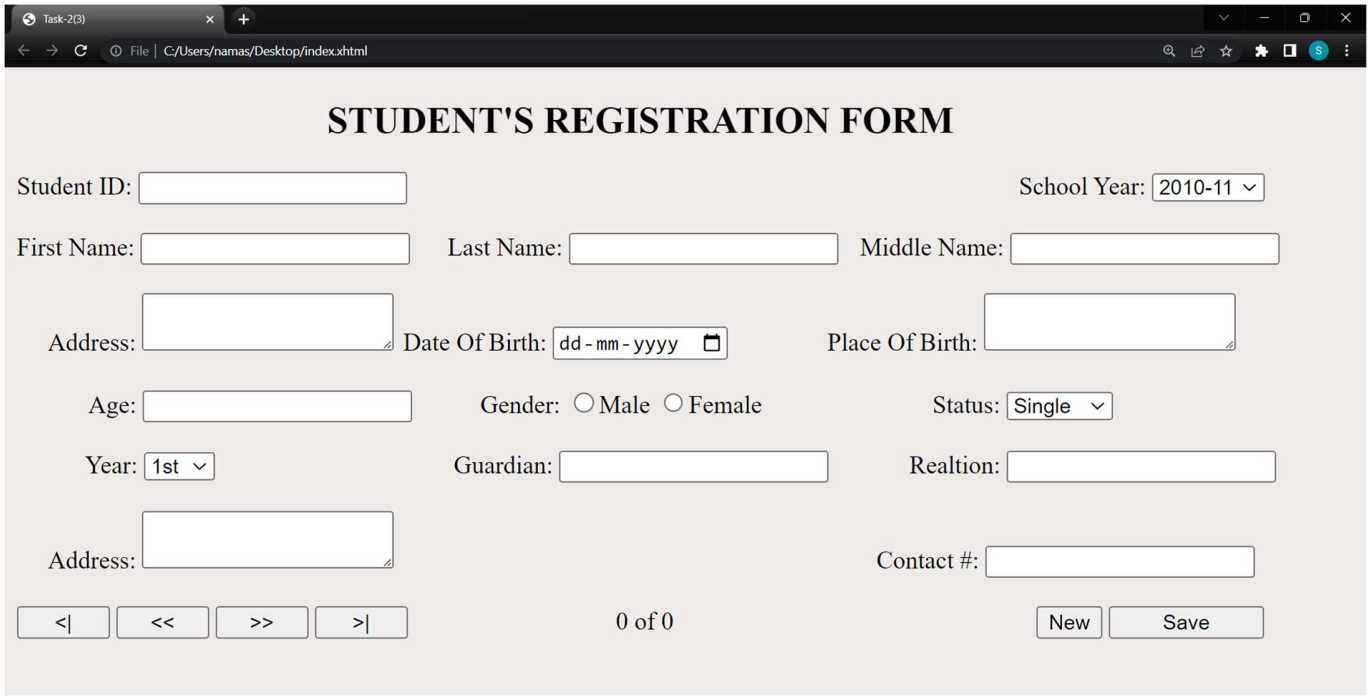
PROGRAM

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Task-2(3)</title>
  </head>
  <body style="background-color: #EEEBEA">
    <h2 style="margin-left:200px;" >STUDENT'S REGISTRATION FORM</h2>
      <form action="" method="get">
      <label>Student ID:</label>
            <input type="text" name="Studebt ID"/>
      <label style="margin-left:390px;">School Year:</label>
```

```
                    <select name="School Year">
                        <option value="2010-11">2010-11</option>
                        <option value="2011-12">2011-12</option>
                        <option value="2012-13">2012-13</option>
                        <option value="2013-14">2013-14</option>
    <option value="2014-15">2014-15</option>
                        <option value="2015-16">2015-16</option>
                </select>
<br/>
<br/>
<label>First Name:</label>
        <input type="text" name=" First Name"/>
<label style="margin-left:20px;">Last Name:</label>

  <input type="text" name=" Last Name"/>

<label style="margin-left:10px;">Middle Name:</label>

     <input type="text" name="Middle Name"/>
<br/>
<br/>
<label style="margin-left:20px;" for="Address">Address:</label>
        <textarea id="Address"name="Address"></textarea>
<label style="margin-left:2px;">Date Of Birth:</label>
        <input type="date" name="Date Of Birth"/>
<label style="margin-left:60px;" for="Place Of Birth">Place Of Birth:</label>
        <textarea id="Place Of Birth"name="Place Of Birth"></textarea>
<br/>
<br/>
<label style="margin-left:46px;">Age:</label>
        <input type="text" name=" Age"/>
<label style="margin-left:40px;">Gender:</label>
        <input type="radio" name="Sex" value="Male" />Male
        <input type="radio" name="Sex" value="Female" />Female
<label style="margin-left:106px;">Status:</label>
        <select name="Status">
          <option value="Single">Single</option>
                <option value="Married">Married</option>
          </select>
<br/>
<br/>
```

```
            <label style="margin-left:44px;">Year:</label>

                <select name="Year">
                        <option value="1st">1st</option>
                         <option value="2nd">2nd</option>
                         <option value="3rd">3rd</option>
                         <option value="4th">4th</option>
                 </select>
        <label style="margin-left:150px;">Guardian:</label>
        <input type="text" name="Guardian"/>
        <label style="margin-left:48px;">Realtion:</label>
                <input type="text" name="Realtion"/>
        <br/>
        <br/>
        <label style="margin-left:20px;" for="Address">Address:</label>

        <textarea id="Address"name="Address"></textarea>
        <label style="margin-left:307px;">Contact #:</label>

        <input type="text" name="Contact #"/>
        <br/>

        <br/>
        <footer>
          <button style="width: 60px;" type="button">&lt;|</button>
          <button style="width: 60px;" type="button">&lt;&lt;</button>
          <button style="width: 60px;" type="button">&gt;&gt;</button>
          <button style="width: 60px;" type="button">&gt;|</button>
          <label style="margin-left:130px;">0 of 0</label>
          <input style="margin-left:230px;" type="reset" value="New"/>
                <input style="width: 100px;" type="submit" value="Save"/>
        </footer>
      </form>
    </body>
  </html>
```

OUTPUT



CONCLUSION:

The above code is an HTML form for a student registration. It contains various input fields for the student's personal information, such as name, address, gender, age, and contact details, along with options to select school year, year of study, and marital status. The form also includes navigation buttons to move between records, a "New" button to reset the form, and a "Save" button to submit the data.

EXPERIMENT NUMBER- 03

Question-1:

AIM

To write a DTD (Document Type Definition) for given XML file.

DESCRIPTION

- The XML Document Type Declaration, commonly known as DTD, is a way to describe XML language precisely. DTDs check vocabulary and validity of the structure of XML documents against grammatical rules of appropriate XML language.
- An XML DTD can be either specified inside the document, or it can be kept in a separate document and then liked separately.

PROGRAM

Given XML file:

```
<stock>
   <new-car>
      <model>Fiat</model>
      <price>12000</price>
   </new-car>
   <used-car>
      <model>Fiat Bravo</model>
      <price>4000</price>
      <mileage>1000</mileage>
      <condition>Good</condition>
   </used-car>
   <used-car>
      <model>Ferrari</model>
      <price>400000</price>
      <mileage>100</mileage></used-car>
</stock>
```

DTD for above XML file:

```
<?xml version="1.0"?>
<!DOCTYPE stock [
<!ELEMENT stock (new-car,used-car+)>
<!ELEMENT new-car (model,price)>
```

```
<!ELEMENT used-car (model,price,mileage,condition?)>
<!ELEMENT model (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!ELEMENT mileage (#PCDATA)>
<!ELEMENT condition (#PCDATA)>
]>
```

RESULTS

Based on the provided XML file and the validated DTD, the XML file is valid according to the specifieddocument structure. This means that the file conforms to the rules and constraints outlined in the DTD, and can be processed and used by any software that is designed to read and interpret XML files.Overall, the XML file provides a structured and standardized way to store and share information about thecars in stock, making it easier to manage and process this data using software applications.

CONCLUSION

In conclusion, the given code shows a simple example of defining a DTD for an XML file. DTD is a powerful tool for defining the structure and constraints of an XML document, which makes it easier tovalidate and process XML data. By using a DTD, developers can ensure that the XML data is well-formed and conforms to a specific structure, which can improve data quality and interoperability.

Question-2:

AIM

To write a DTD (Document Type Definition) and XSD (XML Schema definition) for given XML file.

DESCRIPTION

- The XML Document Type Declaration, commonly known as DTD, is a way to describe XML language precisely. DTDs check vocabulary and validity of the structure of XML documents against grammatical rules of appropriate XML language.
- An XML DTD can be either specified inside the document, or it can be kept in a separate document and then liked separately.
- An XSD defines the structure of an XML document. It specifies the elements and attributes that can appear in an XML document and the type of data these elements and attributes can contain. This information is used to verify that each element or attribute in an XML document adheres to its description.
- XSD is defined in XML. It does not require intermediate processing by a parser. DTD is not defined in XML. You need separate parsers for DTD and XML.
- XSD is extensible. You can derive new elements from the existing elements. DTD is not extensible.

PROGRAM:

Given XML file:

```
<bookstore>
  <book price="730.54" ISBN="string" publicationdate="2016-02-27">
    <title>string</title>
    <author>
      <first-name>string</first-name>
      <last-name>string</last-name>
    </author>
    <genre>string</genre>
  </book>
  <book price="6738.774" ISBN="string">
    <title>string</title>
    <author>
      <first-name>string</first-name>
      <last-name>string</last-name>
    </author>
```

```
</book>
</bookstore>
```

Book.xml

```
<?xml version="1.0"?>
<!DOCTYPE bookstore SYSTEM "external.dtd">
<bookstore>
  <book price="730.54" ISBN="string" publicationdate="2016-02-27">
    <title>string</title>
    <author>
      <first-name>string</first-name>
      <last-name>string</last-name>
    </author>
    <genre>string</genre>
  </book>
  <book price="6738.774" ISBN="string">
    <title>string</title>
    <author>
      <first-name>string</first-name>
      <last-name>string</last-name>
    </author>
  </book>
</bookstore>
```

External.dtd

```
<!ELEMENT bookstore (book+)>
<!ELEMENT book (title,author,genre?)>
<!ATTLIST book price CDATA #REQUIRED>
<!ATTLIST book ISBN CDATA #REQUIRED>
<!ATTLIST book publicationdate CDATA "">
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (first-name,last-name)>
<!ELEMENT first-name (#PCDATA)>
<!ELEMENT last-name (#PCDATA)>
<!ELEMENT genre (#PCDATA)>
```

Book.xml

```
<bookstore xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="external.xsd">

 <bookstore>
   <book price="730.54" ISBN="string" publicationdate="2016-02-27">
      <title>string</title>
      <author>
         <first-name>string</first-name>
         <last-name>string</last-name>
      </author>

 <genre>string</genre>
   </book>
   <book price="6738.774" ISBN="string">
      <title>string</title>
      <author>
         <first-name>string</first-name>
         <last-name>string</last-name>
      </author>
   </book>
</bookstore>
```

External.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="bookstore" type="bookstoreType"/>
   <xs:complexType name="bookstoreType">
      <xs:sequence>
         <xs:element name="book" minOccurs="1" maxOccurs="unbounded"/>
   </xs:sequence>
   </xs:complexType>
   <xs:element name="book" type="bookType"/>
   <xs:complexType name="bookType" >
      <xs:sequence>
      <xs:element ref="title" minOccurs="1"/>
      <xs:element ref="author" minOccurs="1"/>
      <xs:element ref="genre" minOccurs="0" maxOccurs="1"/>
   </xs:sequence>

 <xs:attribute name="price" type="xs:decimal" use="required" />
```

```
<xs:attribute name="ISBN" type="xs:string" use="required"/>
  <xs:attribute name="publicationdate" type="xs:string" use="optional"/>
  </xs:complexType>

 <xs:element name="title" type="xs:string"/>
  <xs:element name="author" type="authorType"/>
  <xs:complexType name="authorType">
  <xs:sequence>
  <xs:element ref="first-name"/>
  <xs:element ref="last-name"/>
  </xs:sequence>
  </xs:complexType>
  <xs:element name="genre" type="xs:string"/>
  <xs:element name="first-name" type="xs:string"/>
  <xs:element name="last-name" type="xs:string"/>
</xs:schema>
```

RESULTS

- DTD:
  The DTD created for the given XML code defines the structure of the document and the allowed elementsand attributes within it. It enforces the rules that must be followed while creating the XML document. TheDTD for the provided XML code is successfully validated with an online DTD validator.

- XSD:
  The XSD created for the given XML code defines the data types, structure, and constraints for the elements and attributes used in the document. It provides a more detailed and precise way to describe thestructure of the XML document. The XSD for the provided XML code is successfully validated with an online XSD validator.

  Overall, both the DTD and XSD successfully validate the provided XML code and ensure that the document follows the predefined structure and rules. This helps in ensuring the integrity and consistencyof the data within the XML document.

CONCLUSION

In conclusion, DTD and XSD are two widely used schema languages for defining the structure and constraints ofXML documents. By creating and validating these schemas, we can ensure that the XML documents follow the predefined structure and rules, leading to more consistent and reliable data.

Question-3:

AIM

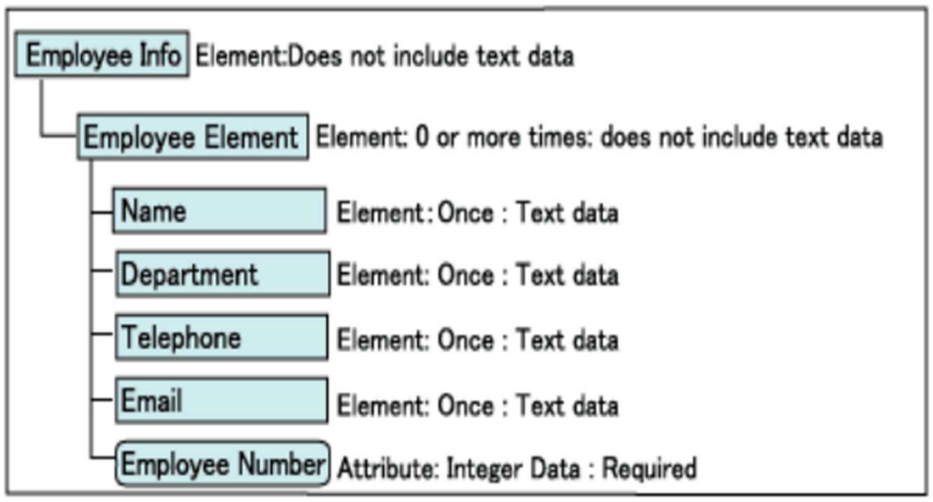
To write a External DTD (Document Type Definition) and XML for given structure.

DESCRIPTION

- XML (Extensible Markup Language) is a markup language similar to HTML, but without predefined tags to use. Instead, you define your own tags designed specifically for your needs. This is a powerful way to store data in a format that can be stored, searched, and shared. An XML document with correct syntax is called "Well Formed".
- An XML document validated against a DTD is both "Well Formed" and "Valid".DTD stands for Document Type Definition. A DTD defines the structure and the legal elements and attributes of an XML document.
- The XML Document Type Declaration, commonly known as DTD, is a way to describe XML language precisely. DTDs check vocabulary and validity of the structure of XML documents against grammatical rules of appropriate XML language.
- An XML DTD can be either specified inside the document, or it can be kept in a separate document and then liked separately.

PROGRAM

Given Structure



Employee.xml

```
<?xml version = "1.0" encoding = "UTF-8" standalone="no" ?>

<!DOCTYPE Employee_Info SYSTEM " Employee_Info.dtd">
```

```
<Employee_Info>
  <Employee_Element>
     <Name>Srihan Pasham </Name>
    <Department>CSE</Department>
    <Telephone>9059227066</Telephone>
    <Email>srihanpasham @gmail.com</Email>
    <Employee_Number integer="128"/>
  </Employee_Element>
</Employee_Info>
```

Employee_Info.dtd

```
<!ELEMENT employee (Employee_Element*)>
<!ELEMENT Employee_Element (Name,Department,Telephone,Email)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Department (#PCDATA)>
<!ELEMENT Telephone (#PCDATA)>
<!ELEMENT Email (#PCDATA)>
<!ELEMENT Employee_Number (#PCDATA)>
<!ATTLIST Employee_Element Employee_Number CDATA #REQUIRED>
```

RESULTS

There are no errors found in the provided DTD for the given XML structure. This means that the XML document adheres to the defined structure and can be considered valid according to the DTD.

CONCLUSION

In conclusion, the given problem statement describes the structure and content of an XML file that stores employee information, including their name, department, telephone number, and email address. The structurecan be represented using a DTD, which defines the rules for the elements and attributes that make up the file.

Question-4:

AIM

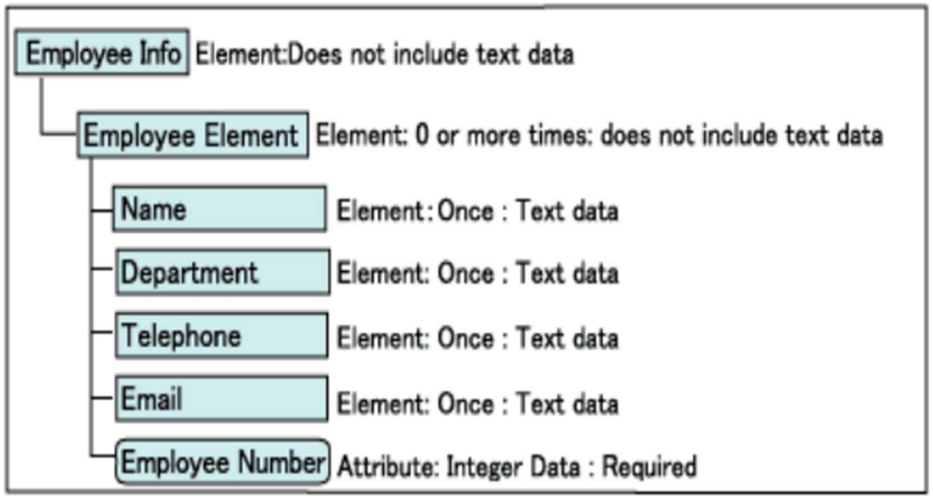
To write an XML Schema for given Structure.

DESCRIPTION

- An XSD defines the structure of an XML document. It specifies the elements and attributes that can appear in an XML document and the type of data these elements and attributes can contain. This information is used to verify that each element or attribute in an XML document adheres to its description.
- XSD is defined in XML. It does not require intermediate processing by a parser. DTD is not defined in XML. You need separate parsers for DTD and XML.
- XSD is extensible. You can derive new elements from the existing elements. DTD is not extensible.

PROGRAM

Given Structure



Employee.xml

```
<?xml version="1.0"?>
<Employee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="employee.xsd">
 <Employee_Element Employee_Number="123">
       <Name>Srihan Pasham </Name>
      <Department>CSE</Department>
     <Telephone>9059227066</Telephone>
     <Email>srihanpasham @gmail.com</Email>
```

```
</Employee_Element>
</Employee>
```

Employee.xsd

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Employee" type="employeeinfoType"/>
  <xs:complexType name="employeeinfoType">
    <xs:sequence>
      <xs:element ref="Employee_Element" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Employee_Element" type="employeeType"/>
  <xs:complexType name="employeeType">
    <xs:sequence>
      <xs:element ref="Name" minOccurs="1" maxOccurs="1"/>
      <xs:element ref="Department" minOccurs="1" maxOccurs="1"/>
      <xs:element ref="Telephone" minOccurs="1" maxOccurs="1"/>
      <xs:element ref="Email" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
    <xs:attribute name="Employee_Number" type="xs:long" use="required"/>
  </xs:complexType>
  <xs:element name="Name" type="xs:string"/>
  <xs:element name="Department" type="xs:string"/>
  <xs:element name="Telephone" type="xs:string"/>
  <xs:element name="Email" type="xs:string"/>
</xs:schema>
```

RESULTS

The result is a valid XML file that adheres to the specified structure defined by the XML schema. Any XML document that conforms to this schema can be validated against it and will be considered valid if itmeets all the requirements specified in the schema. The schema defines the elements, attributes, and data types that are allowed in the XML document and specifies the rules for their usage. By validating againstthe schema, it ensures that the document follows a consistent and standardized format, making it easier toprocess and exchange data between different systems.

CONCLUSION

In conclusion, XML schemas are used to define the structure and constraints of an XML document. They help ensure that XML documents conform to a specific format and contain the required data elements. By defining an XML schema for a given problem, we can ensure that XML documents conform to a specific structure and containthe required data elements, making it easier to validate, process and share the data.

EXPERIMENT NUMBER- 04

Question-1:

AIM

To install any three IDEs which are highly suitable for web development and one web server. Write the salient features, installation steps and uninstallation steps of chosen IDEs and web server.

DESCRIPTION

- An integrated development environment (IDE) is a software application that helps programmers develop software code efficiently. It increases developer productivity by combining capabilities such as software editing, building, testing, and packaging in an easy-to-use application. Just as writers use text editors and accountants use spreadsheets, software developers use IDEs to make their job easier.

- Web browsers communicate with web server using the Hypertext Transfer Protocol (HTTP). When you click a link on a web page, submit a form, or run a search, an HTTP request is sent from your browser to the target server.

- The request includes a URL identifying the affected resource, a method that defines the required action (for example to get, delete, or post the resource), and may include edits the fault of that open document reader additional information encoded in URL parameters (the field-value pairs sent via a query string), as POST data (data sent by the HTTP POST method), or in associated cookies.

- Web servers wait for client request messages, process them when they arrive, and reply to the web browser with an HTTP response message. The response contains a status line indicating whether or not the request succeeded (e.g. "HTTP/1.1 200 OK" for success).

PROGRAM

I. Visual Studio Code

Installation Steps: -

1. Open a terminal window by pressing "Ctrl+Alt+T" on your keyboard.
2. Add the Microsoft GPG key using the following command:
   wget -qO- https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor > packages.microsoft.gpg
   sudo install -o root -g root -m 644 packages.microsoft.gpg /usr/share/keyrings/
   sudo sh -c 'echo "deb [arch=amd64 signed-by=/usr/share/keyrings/packages.microsoft.gpg] https://packages.microsoft.com/repos/vscode stable main" > /etc/apt/sources.list.d/vscode.list'
3. Update the package list and install Visual Studio Code using the following commands:
   sudo apt update
   sudo apt install code
4. Wait for the installation process to complete.

Uninstallation Steps: -

1. Open a terminal window by pressing "Ctrl+Alt+T" on your keyboard.
2. Remove Visual Studio Code using the following command:
   sudo apt remove code
3. If you also want to remove the configuration files and any other data associated with Visual Studio Code, use the following command:
   sudo apt purge code

Features of Visual Studio Code: -

1. Built-in Git integration: VS Code has built-in support for Git, allowing you to easily manage and commit code changes without leaving the editor.

2. Language support: VS Code provides built-in support for many programming languages, including JavaScript, Python, C++, and more. It also has a large number of extensions available that provide support for additional languages.

3. Debugging: VS Code provides powerful debugging tools that make it easy to identify and

fix errors in your code.

4. IntelliSense: VS Code provides intelligent code completion suggestions as you type, making it easier to write code quickly and accurately.

5. Extensions: VS Code has a large and active community of developers who have created a wide range of extensions that add new functionality and features to the editor.

II. Atom

Installation Steps: -

1. Open a terminal window by pressing "Ctrl+Alt+T" on your keyboard.
2. Add the Atom repository to your system using the following command:
   sudo add-apt-repository ppa:webupd8team/atom
3. Update the package list and install Atom using the following commands:
   sudo apt update
   sudo apt install atom
4. Wait for the installation process to complete.

Uninstallation Steps: -

1. Open a terminal window by pressing "Ctrl+Alt+T" on your keyboard.
2. Remove Atom using the following command:
   sudo apt remove atom
3. If you also want to remove the configuration files and any other data associated with Atom, use the following command:
   sudo apt purge atom
4. Finally, remove any residual configuration files using the following command:
   rm -rf ~/.atom/
   This command will delete the .atom folder, which contains all the configuration files associated with Atom.

Features of Atom: -

1. Cross-platform support: Atom is available for Windows, macOS, and Linux, making it a versatile option for developers on any platform.

2. Package manager: Atom has a built-in package manager that allows you to easily install and manage packages for additional functionality and features.

3. Highly customizable: Atom is highly customizable, allowing you to tailor the editor to your specific needs and preferences. You can change the theme, layout, and key bindings, and even create your own packages to extend the editor.

4. Multiple panes: Atom allows you to split the editor into multiple panes, making it easy to work on multiple files or projects at once.

5. Autocomplete: Atom provides intelligent code completion suggestions as you type, making it easier to write code quickly and accurately.

III. Sublime Text

Installation Steps: -

1. Open a terminal window by pressing "Ctrl+Alt+T" on your keyboard.
2. Add the Sublime Text repository to your system using the following command:
   wget -qO - https://download.sublimetext.com/sublimehq-pub.gpg | sudo apt-key add –
3. Add the Sublime Text repository to your system's package sources using the following command:
   echo "deb https://download.sublimetext.com/ apt/stable/" | sudo tee /etc/apt/sources.list.d/sublime-text.list
4. Update the package list and install Sublime Text using the following commands:
   sudo apt update
   sudo apt install sublime-text
5. Wait for the installation process to complete.

Uninstallation Steps: -

1. Open a terminal window by pressing "Ctrl+Alt+T" on your keyboard.
2. Remove Sublime Text using the following command:
   sudo apt remove sublime-text
3. If you also want to remove the configuration files and any other data associated with Sublime Text, use the following command:
   sudo apt purge sublime-text
4. Finally, remove any residual configuration files using the following command
   rm -rf ~/.config/sublime-text-3/
   This command will delete the sublime-text-3 folder, which contains all the configuration files associated with Sublime Text.

Features of Sublime Text

1. Cross-platform support: Sublime Text is available for Windows, macOS, and Linux, making it a versatile option for developers on any platform.

2. Multiple cursors: Sublime Text allows you to work with multiple cursors simultaneously, making it easy to edit and manipulate large amounts of code.

3. Goto Anything: Sublime Text provides a "Goto Anything" feature that allows you to quickly jump to a file, symbol, or line of code using only a few keystrokes.

4. Split editing: Sublime Text allows you to split the editor into multiple panes, making it easy to work on multiple files or projects at once.

5. Command Palette: Sublime Text has a built-in Command Palette that allows you to quickly access all of the editor's commands and settings.

IV. Installation and Uninstallation steps for Apache2 web server: -

Installation Steps: -

1. Open a terminal window by pressing "Ctrl+Alt+T" on your keyboard.
2. Install Apache web server using the following command:
   sudo apt update
   sudo apt install apache2
3. Wait for the installation process to complete.
4. Once the installation is complete, you can verify that Apache is running by opening a web browser and navigating to *http://localhost/*. You should see the Apache default page displayed in the browser.

Uninstallation Steps: -

1. Open a terminal window by pressing "Ctrl+Alt+T" on your keyboard.
2. Stop the Apache service using the following command:
   sudo systemctl stop apache2
3. Remove the Apache package using the following command:
   sudo apt remove apache2
4. If you also want to remove the configuration files and any other data associated with Apache, use the following command:

sudo apt purge apache2

5. Finally, remove any residual configuration files using the following command:
   sudo rm -rf /etc/apache2/

Features of Apache2: -

1. Cross-platform support: Apache2 is available for Windows, macOS, and Linux, making it a versatile option for serving web content on any platform.

2. Scalability: Apache2 is highly scalable, able to handle large numbers of simultaneous connections and requests.

3. Security: Apache2 provides a wide range of security features, including SSL/TLS encryption, access control, and support for multiple authentication methods.

4. Virtual hosting: Apache2 allows you to serve multiple websites or domains from a single server using virtual hosting.

5. Module support: Apache2 supports a wide range of modules that extend its functionality, including support for PHP, Perl, and other scripting languages.

6. Caching: Apache2 provides built-in caching features that can help improve the performance of your web applications.

7. Logging: Apache2 provides detailed logging capabilities that can help you troubleshoot issues and analyze traffic to your web server.

8. Customization: Apache2 is highly customizable, allowing you to tailor the server to your specific needs and preferences.

RESULTS

After installing multiple IDEs, you will have access to a variety of tools and features that can help you write code more efficiently and effectively. Each IDE has its own strengths and weaknesses, so the best one for you will depend on your personal preferences, the programming languages you use, and the type of projects you work on.

Some of the benefits of using an IDE include:

- Code completion: IDEs can help you write code faster and with fewer errors by providing intelligent code completion suggestions as you type.

- Debugging: IDEs often include built-in debugging tools that can help you find and fix issues in your code.

- Syntax highlighting: IDEs typically provide syntax highlighting for the programming languages you work with, making it easier to read and understand your code.

- Project management: IDEs often provide tools for managing and organizing your projects, including version control integration and project templates.

- Refactoring: IDEs can help you refactor your code to make it cleaner, more maintainable, and easier to read.

CONCLUSIONS

In conclusion, installing Apache2 on your system can provide you with a robust and reliable web server that offers a range of features and capabilities. Whether you are looking to host a simple website or a complex web application, Apache2 can provide the scalability, security, and customization options you need to meet your requirements.

With its cross-platform support, module support, virtual hosting capabilities, and built-in caching and logging features, Apache2 is a popular choice for serving web content on a variety of platforms. Additionally, its highly customizable nature allows you to tailor the server to your specific needs and preferences, making it an ideal option for both beginners and experienced web developers alike.

In summary, installing Apache2 can provide you with a powerful web server that is flexible, reliable, and highly secure, making it a great choice for anyone looking to host web content or applications.

EXPERIMENT NUMBER- 04

Question-2:

AIM

Design a web page which consists of image and display your biodata.

DESCRIPTION

To create a web page consisting of our image and our bio data we can use table tag and create a table so that it contains a single row with two columns, one column containing our image and the other column containing our bio data it is better to place the image file in the same directory as that of the XHTML file.
We use <img> for inserting image by mentioning the source path in “src” attribute.

PROGRAM

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
 <head>
  <title>Biodata Template with Image</title>
  <link rel="stylesheet" type="text/css" href="style.css" />
 </head>
 <body>
  <table>
   <tr>
    <td class="col1">
     <img class="img" src="me.jpeg" alt="Your Photo" />
    </td>
    <td>
     <div class="container">
      <div class="header">
       <h1>Nama Sricharan</h1>
       <br />
      </div>
      <div class="sub">
       <h2>Computer Science Student</h2>
      </div>
      <div class="row">
       <div class="col">
```

```
      <h2>Personal Information</h2>
      <p>
        <strong>Date of Birth:</strong> 14/08/2003
      </p>
      <p>
        <strong>Address:</strong> Hyderabad, India
      </p>
      <p>
        <strong>Phone:</strong> 7742796678
      </p>
      <p>
        <strong>Email:</strong> namasricharan@gmail.com
      </p>
    </div>
    <div class="col">
      <h2>Education</h2>
      <p>
        <strong>University:</strong>Chaitanya Bharati Institute Of Technology
      </p>
      <p>
        <strong>Major:</strong> Computer Science
      </p>
      <p>
        <strong>GPA:</strong> 9.3
      </p>
    </div>
  </div>
  <div class="row">
    <div class="col">
      <h2>Skills</h2>
      <ul>
        <li>HTML</li>
        <li>CSS</li>
        <li>JavaScript</li>
        <li>C++</li>
        <li>Python</li>
      </ul>
    </div>
    <div class="col">
      <h2>Hobbies</h2>
      <ul>
        <li>Playing cricket</li>
        <li>Watching sports and movies</li>
        <li>Listening to music</li>
      </ul>
    </div>
```

```
          </div>
        </div>
      </td>
    </tr>
  </table>
  <table>
    <tr>
      <td class="col2">
        <h2>About</h2>
        <p> I am currently pursuing my B.Tech in computer science, I have a solid foundation in
programming, software development, and computer systems.I am proficient in programming languages
such as Java, Python, C++, and others, and have experience with software development tools and
methodologies. I have also gained knowledge in areas such as data structures, algorithms, databases and
MERN programming. </p>
      </td>
    </tr>
  </table>
 </body>
</html>
```

RESULT/OUTPUT

Nama Sricharan

Computer Science Student

Personal Information

Date of Birth: 14/08/2003

Address: Hyderabad, India

Phone: 7742796678

Email: namasricharan@gmail.com

Education

University: Chaitanya Bharati Institute Of Technology

Major: Computer Science

GPA: 9.3

Skills

HTML
CSS
JavaScript
C++
Python

Hobbies

Playing cricket
Watching sports and movies
Listening to music

About

I am currently pursuing my B.Tech in computer science, I have a solid foundation in programming, software development, and computer systems.I am proficient in programming languages such as Java, Python, C++, and others, and have experience with software development tools and methodologies. I have also gained knowledge in areas such as data structures, algorithms, databases and MERN programming.

CONCLUSIONS

We are able to display our photo and our biodata on the screen using table tag with a single row and two columns one column containing our picture and the other column containing the biodata.

EXPERIMENT NUMBER- 04

Question-3:

AIM

Design a web page which consists of image and display your biodata.

DESCRIPTION

Create Restaurant menu card sample given below or you can design your own menu card.

PROGRAM

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>MENU</title>
  <style>
     .menu{
        margin-top: 10%;
     }
     body{
        background-image: url("https://img.freepik.com/free-vector/black-carbon-fiber-texture-
pattern-background_1017-33436.jpg");
        background-repeat: no-repeat;
        background-size: cover;
     }
  </style>
</head>
<body>

   <div style="background-color: brown;margin: 0px; padding: 10px;;border-style: solid;border-
width: 0px;border-bottom-width: 15px;border-color: white;border-bottom-right-radius:40%
;border-bottom-left-radius:40% ;">
   <h1 align="center" > <span style="color: white;font-size: 90px;padding: 0px;">MENU CARD
<br/><span style="font-size: 60px;">Restaurant XYZ <br/>Milky Way Street 88 - 8888
Galaxy</span></span></h1>
   </div>
```

```
<div style="margin-top: 1%;margin-left: 33%;margin-bottom: a;">
<table align="left" >

<tr>
  <th align="left" style="font-size: 70px;" ><span style="color: white;"> BURGERS
</span></th>
</tr>

<tr>
  <td style="color: white;font-size:xx-large;">Cabernet Franc </td>
  <td ><pre>        </pre></td>
  <td style="color: white;font-size:xx-large;">$5.50 </td>
  <td ><pre>                                        </pre></td>
  <td rowspan="4"><img style="border-radius: 150px;border-width: 10px;border-style:
solid;border-color: brown;width: 250px;height: 250px;" src="burger.png" alt="image
error"/></td>
</tr>

<tr>
  <td style="color: white;font-size:xx-large;">Cabernet Souvignon</td>
  <td ><pre>        </pre></td>
  <td style="color: white;font-size:xx-large;">$6.00</td>
</tr>

<tr>
  <td style="color: white;font-size:xx-large;">Chianti</td>
  <td ><pre>        </pre></td>
  <td style="color: white;font-size:xx-large;">$5.50</td>
</tr>

<tr>
  <td style="color: white;font-size:xx-large;">Dormfelder</td>
  <td ><pre>        </pre></td>
  <td style="color: white;font-size:xx-large;">$3.50</td>
</tr>



<tr>
  <th style="font-size: 70px;" ><span style="color: white;"> HOTDOGS </span></th>
</tr>

<tr>
```

```
            <td style="color: white;font-size:xx-large;">Cabernet Franc</td>
            <td ><pre>        </pre></td>
            <td style="color: white;font-size:xx-large;">$5.50</td>
            <td ><pre>                                        </pre></td>
        <td rowspan="4" ><img style="border-radius: 150px;border-width: 10px;border-style:
solid;border-color: brown;width: 250px;height: 250px;" src="img.jpg" alt="image error"/></td>
    </tr>

    <tr>
        <td style="color: white;font-size:xx-large;">Cabernet Souvignon</td>
        <td ><pre>        </pre></td>
        <td style="color: white;font-size:xx-large;">$6.00</td>
    </tr>
    <tr>
        <td style="color: white;font-size:xx-large;">Chianti</td>
        <td ><pre>        </pre></td>
        <td style="color: white;font-size:xx-large;">$5.50</td>
    </tr>

    <tr>
        <td style="color: white;font-size:xx-large;">Dormfelder</td>
        <td ><pre>        </pre></td>
        <td style="color: white;font-size:xx-large;">$3.50</td>
    </tr>

    <tr>
        <th align="left" style="font-size:70px;" ><span style="color: white;"> DRINKS
</span></th>
    </tr>

    <tr>
        <td style="color: white;font-size:xx-large;">Cabernet Franc</td>
        <td ><pre>        </pre></td>
        <td style="color: white;font-size:xx-large;">$5.50</td>
        <td ><pre>                                        </pre></td>
        <td rowspan="4"><img style="border-radius: 150px;border-width: 10px;border-style:
solid;border-color: brown;height: 250px;width: 250px;" src="diabolo-mockail-440-x400-
c795884.jpg" alt="image error"/></td>
    </tr>

    <tr>
        <td style="color: white;font-size:xx-large;">Cabernet Souvignon</td>
```

```
        <td ><pre>        </pre></td>
        <td style="color: white;font-size:xx-large;">$6.00</td>
    </tr>

    <tr>
        <td style="color: white;font-size:xx-large;">Chianti</td>
        <td ><pre>        </pre></td>
        <td style="color: white;font-size:xx-large;">$5.50</td>
    </tr>

    <tr>
        <td style="color: white;font-size:xx-large;">Dormfelder</td>
        <td ><pre>        </pre></td>
        <td style="color: white;font-size:xx-large;">$3.50</td>
    </tr>
 </table>

<p>Registered under Pepsico 2006 <br/>California 2023</p>
</div>
</body>
</html>
```

RESULT/OUTPUT

CONCLUSIONS

A restaurant menu webpage created using HTML table tags is an effective way to showcase a restaurant's menu in an organized and structured format. The use of table tags helps to clearly present information such as the name, description, and price of each menu item, making it easy for customers to navigate and choose what they would like to order. The inclusion of images of some of the dishes or drinks also adds visual appeal to the menu card and gives customers a better idea of what they can expect.

EXPERIMENT NUMBER- 04

Question-4:

AIM

Design your class time table using table tags.

DESCRIPTION

The design of a Class Time Table represents a table with the periods, recess and time stamp details as each cell of the respective table's row in HTML. First create a table with a sufficient number of rows and columns to accommodate the table's contents using the <TR> and <TD> tags under the <TABLE> tag. Pronounce the partitions in a single cell using ROWSPAN and COLSPAN attributes for respective rows and adjust the spacing using CELLSPACING attribute to fit the contents in the cell.

Here, <TABLE> tag creates a table, <TR> creates a row for that table and data tis inserted using <TD> tag. For combining any number of rows we use the ROWSPAN attribute with mention of the number of rows to be spanned, and similarly we use COLSPAN for column spanning.

PROGRAM

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
   <link rel="stylesheet" href="styles.css">
   <title>Timetable CSE-2</title>
</head>

<body style="font-family:Cambria">
   <h2 align='center'>
      <strong> CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY(A)</strong>
   </h2>
   <h2 align='center'>DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING</h2>
   <h3 align="center">
      <strong> IV Semester Class Time Table for the A.Y. 2022-23</strong>
   </h3>
   <pre
```

```
        style='margin-left:17%;'>Class: BE-CSE
    2                                                    w.e.f: 27-03-
    2023                                                 Room No: C213</pre>
      <div>
        <table border='1' align='center' cellspacing='0px'>
          <table width="1200px" border="2">
            <thead align="center">
              <tr bgcolor="LightGray">
                <th bgcolor="LightGray">Period</th>
                <th>I</th>
                <th>II</th>
                <th>III</th>
                <th rowspan="0">12:15 - 1:00</th>
                <th>IV</th>
                <th>V</th>
                <th>VI</th>
              </tr>
              <tr bgcolor="LightGray">
                <th>Day</th>
                <th>9:10 - 10:10</th>
                <th>10:10 - 11:10</th>
                <th>11:15 - 12:15</th>
                <th>1:00 - 2:00</th>
                <th>2:00 - 3:00</th>
                <th>3:05 - 4:00</th>
              </tr>
            </thead>
            <tbody align="center">
              <tr>
                <th bgcolor="LightGray">MON</th>
                <td>CAMP</td>
                <td>IWT</td>
                <td>EEA</td>
                <td rowspan="6">
                  <strong>L</strong>
                  <br>
                  <strong>U</strong>
                  <br>
                  <strong>N</strong>
                  <br>
                  <strong>C</strong>
                  <br>
                  <strong>H</strong>
                </td>
                <td>MFDS</td>
                <td>
```

```
            <i>Mentoring</i>
        </td>
        <td>
            <i>Fit India</i>
        </td>
    </tr>
    <tr>
        <th bgcolor="LightGray">TUE</th>
        <td>MFDS</td>
        <td>CAMP</td>
        <td>DAA</td>
        <td>DBMS</td>
        <td colspan="2">DAA Lab@ Lab-1 (B1)
            <hr>DBMS Lab@Lab-2 (B2)
            <hr>IWT Lab@Lab-3 (B3)
        </td>
    </tr>
    <tr>
        <th bgcolor="LightGray">WED</th>
        <td colspan="2">DAA Lab@ Lab-1 (B2)
            <hr>DBMS Lab@Lab-2 (B3)
            <hr>IWT Lab@Lab-3 (B1)
        </td>
        <td>DBMS</td>
        <td>EEA</td>
        <td>DAA</td>
        <td>MFDS</td>
    </tr>
    <tr>
        <th bgcolor="LightGray">THU</th>
        <td>EEA</td>
        <td colspan="2">DAA Lab@ Lab-1 (B3)
            <hr>DBMS Lab@Lab-2 (B1)
            <hr>IWT Lab@Lab-3 (B2)
        </td>
        <td>CAMP</td>
        <td>IWT</td>
        <td>
            <i>Library</i>
        </td>
    </tr>
    <tr>
        <th bgcolor="LightGray">FRI</th>
        <td colspan="2">MFDS Lab@TPO-1 (B1,B2)
            <hr>IWT Lab@Lab-5 (B3)
        </td>
```

```
                <td>DAA</td>
                <td colspan="2">MFDS Lab@TPO-1 (B3)
                  <hr>IWT Lab@Lab-5(B1),Lab-6(B2)
                </td>
                <td>DBMS</td>
              </tr>
              <tr>
                <th bgcolor="LightGray">SAT</th>
                <td colspan="3"></td>
                <td colspan="3"></td>
              </tr>
            </tbody>
          </table>
        </div>
        <br>
        <div>
          <table width="1200px" border="2">
            <thead align="center">
              <tr bgcolor="LightGray">
                <th>CODE</th>
                <th>Subject</th>
                <th>Faculty</th>
                <th>Mobile No.</th>
              </tr>
              <tr>
                <th colspan="4">Theory</th>
              </tr>
            <tbody align="left">
              <tr>
                <td>20MTC13</td>
                <td>Mathematical Foundation for Data Science & Security</td>
                <td>Dr. Swatmaram</td>
                <td>9849167507</td>
              </tr>
              <tr>
                <td>20CSC12</td>
                <td>Design and Analysis of Algorithms</td>
                <td>Smt. G.Mamatha</td>
                <td>9491870706</td>
              </tr>
              <tr>
                <td>20CSC13</td>
                <td>Computer Architecture and Microprocessor</td>
                <td>Dr. Premkumar Chitthaluru</td>
                <td>7060509299</td>
              </tr>
```

```
<tr>
   <td>20CSC14</td>
   <td>Data Base Management System</td>
   <td>Sri. B.Sateesh</td>
   <td>9866801591</td>
</tr>
<tr>
   <td>20CSC15</td>
   <td>Internet & Web Technologies</td>
   <td>Smt. E.Swathi</td>
   <td>8143369095</td>
</tr>
<tr>
   <td>20MBC01</td>
   <td>Engineering Economics & Accountacy</td>
   <td>Dr. Sowmya Kethi Reddi</td>
   <td>9642802757</td>
</tr>
<tr>
   <th colspan="4" align="center">Practicals</th>
</tr>
<tr>
   <td>20MTC14</td>
   <td>Mathematical Foundation for Data Science & Security Lab</td>
   <td>Dr. Swatmaram</td>
   <td>9849167507</td>
</tr>
<tr>
   <td>20CSC16</td>
   <td>Design and Analysis of Algorithms Lab</td>
   <td>Smt. G.Mamatha</td>
   <td>9491870706</td>
</tr>
<tr>
   <td>20CSC17</td>
   <td>Data Base Management System Lab</td>
   <td>Sri. B.Sateesh</td>
   <td>9866801591</td>
</tr>
<tr>
   <td>20CSC18</td>
   <td>Internet & Web Technologies Lab</td>
   <td>Smt. E.Swathi <br>Smt. G. Shanmukhi Rama </td>
   <td>8143369095 <br>9949438284 </td>
</tr>
<tr>
```

```
                <th colspan="2">Class In-Charge</th>
                <td>Smt. E.Swathi</td>
                <td>8143369095</td>
              </tr>
              <tr>
                <th colspan="2">Sessional Marks/Attendance In-charge</th>
                <td>Sri. B.Sateesh</td>
                <td>9866801591</td>
              </tr>
            </tbody>
            </thead>
          </table>
        </div>
        <br>
        <div>
          <table width="1200px" border="2">
            <thead align="center">
              <tr bgcolor="LightGray">
                <th>Mentor Name</th>
                <th colspan="2">Roll Numbers</th>
                <th>Mobile No.</th>
              </tr>
            <tbody align="left">
              <tr>
                <td>Dr. Ravi Uyyala</td>
                <td colspan="2">160121733071 - 094</td>
                <td>9618511774</td>
              </tr>
              <tr>
                <td>Smt. D.Naga Jyothi</td>
                <td colspan="2">160121733095 - 118</td>
                <td>8008023453</td>
              </tr>
              <tr>
                <td>Smt. T.Sushmitha</td>
                <td colspan="2">160121733119 - 134,213,307-313</td>
                <td>9502150355</td>
              </tr>
            </tbody>
            </thead>
          </table>
        </div>
        <br>
        <div class="A">
          <table class="B" style="text-align: left;">
            <tbody>
```

```
            <tr>
               <th>Prepared by</th>
            </tr>
            <tr>
               <td>Smt. D. Naga Jyothi</td>
            </tr>
            <tr>
               <td>Smt. G. Mamatha</td>
            </tr>
            <tr>
               <td>Smt. M. Madhu Latha</td>
            </tr>
         </tbody>
      </table>
   </div>
</body>

</html>
```

RESULT/OUTPUT

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY(A)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

IV Semester Class Time Table for the A.Y. 2022-23

Class: BE-CSE 2 w.e.f: 27-03-2023 Room No: C213

Period	I	II	III	12:15 - 1:00	IV	V	VI
Day	9:10 - 10:10	10:10 - 11:10	11:15 - 12:15		1:00 - 2:00	2:00 - 3:00	3:05 - 4:00
MON	CAMP	IWT	EEA		MFDS	*Mentoring*	*Fit India*
TUE	MFDS	CAMP	DAA		DBMS	DAA Lab@ Lab-1 (B1) DBMS Lab@Lab-2 (B2) IWT Lab@Lab-3 (B3)	
WED	DAA Lab@ Lab-1 (B2) DBMS Lab@Lab-2 (B3) IWT Lab@Lab-3 (B1)		DBMS	LUNCH	EEA	DAA	MFDS
THU	EEA	DAA Lab@ Lab-1 (B3) DBMS Lab@Lab-2 (B1) IWT Lab@Lab-3 (B2)			CAMP	IWT	*Library*
FRI	MFDS Lab@TPO-1 (B1,B2) IWT Lab@Lab-5 (B3)		DAA		MFDS Lab@TPO-1 (B3) IWT Lab@Lab-5(B1),Lab-6(B2)		DBMS
SAT							

CODE	Subject	Faculty	Mobile No.
Theory			
20MTC13	Mathematical Foundation for Data Science & Security	Dr. Swatmaram	9849167507
20CSC12	Design and Analysis of Algorithms	Smt. G.Mamatha	9491870706
20CSC13	Computer Architecture and Microprocessor	Dr. Premkumar Chitthaluru	7060509299
20CSC14	Data Base Management System	Sri. B.Sateesh	9866801591
20CSC15	Internet & Web Technologies	Smt. E.Swathi	8143369095
20MBC01	Engineering Economics & Accountacy	Dr. Sowmya Kethi Reddi	9642802757
Practicals			
20MTC14	Mathematical Foundation for Data Science & Security Lab	Dr. Swatmaram	9849167507
20CSC16	Design and Analysis of Algorithms Lab	Smt. G.Mamatha	9491870706
20CSC17	Data Base Management System Lab	Sri. B.Sateesh	9866801591
20CSC18	Internet & Web Technologies Lab	Smt. E.Swathi Smt. G. Shanmukhi Rama	8143369095 9949438284
Class In-Charge		Smt. E.Swathi	8143369095
Sessional Marks/Attendance In-charge		Sri. B.Sateesh	9866801591

Mentor Name	Roll Numbers	Mobile No.
Dr. Ravi Uyyala	160121733071 - 094	9618511774
Smt. D.Naga Jyothi	160121733095 - 118	8008023453
Smt. T.Sushmitha	160121733119 - 134,213,307-313	9502150355

CONCLUSIONS

The above HTML code can be used as a basic template to create a Time Table for various layouts such as colleges calendars, event schedules time table, personal study time table and so on. By customizing the input fields and adding additional input elements, more information can be displayed to make it optimal.

EXPERIMENT NUMBER- 05

Question-1:

AIM

Display multiplication table for the given number.

DESCRIPTION

In Javascript we use “prompt” function to take the input from the user. We use “var” keyword just to declare a variable and the basic syntax for the looping is as same as C language. “console.log” is used to print the output on the console or we could also use “document.write()” to display the output on the screen.

PROGRAM

```
var n=prompt("Enter the number:")
for(i=1;i<=10;i++)
{
   console.log(n+'x'+i+'='+n*i)
}
```

RESULT/OUTPUT

```
Enter the number:4
4x1=4
4x2=8
4x3=12
4x4=16
4x5=20
4x6=24
4x7=28
4x8=32
4x9=36
4x10=40
```

CONCLUSION

We understood the functionality of “prompt” in the JavaScript and were able to display the multiplication table according to the users choice using control statements.

EXPERIMENT NUMBER- 05

Question-2:

AIM

Create a Java script code to generate a multiplication table asking the user to enter number of rows and columns. If user enters nothing or 0 then display multiplication table with 10 rows and 10 columns.

DESCRIPTION

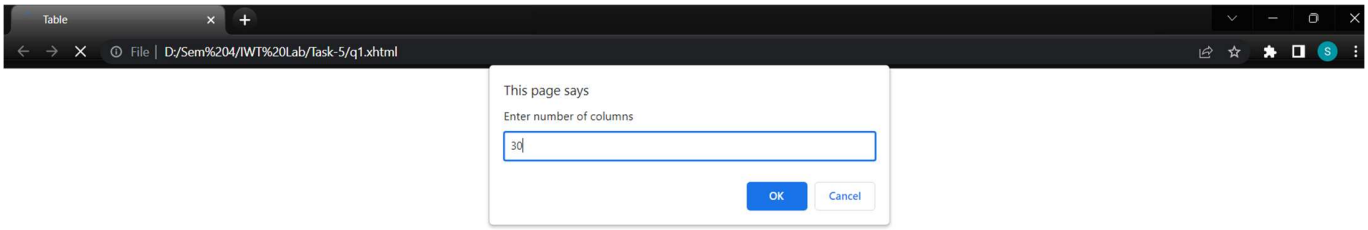
For creating a multiplication table according to the inputs given by the user regarding the number of rows and columns we use “script” tag to write the JavaScript code inside HTML code. For displaying the table on the output screen we use “table” tag and the respective “td” and “tr” tags, the logic of getting multiplication table is written inside the script tag with the for loops, functions and necessary prompt and if statements. Finally we use “document.write()” to display the output on the screen.

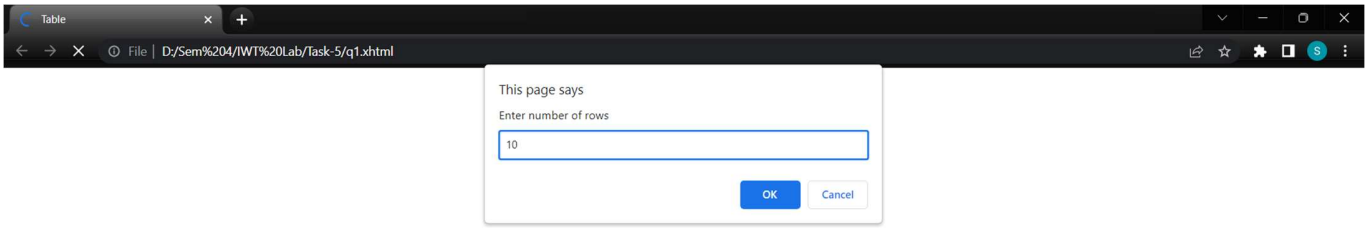
PROGRAM

```
<html>
<head>
 <title>Multiplication Table</title>
 <script type="text/javascript">
   var rows = prompt("How many rows for your multiplication table?");
   var cols = prompt("How many columns for your multiplication table?");
   if(rows == "" || rows==0 || rows == null)
       rows = 10;
   if(cols== "" || cols==0|| cols== null)
       cols = 10;
   createTable(rows, cols);
   function createTable(rows, cols)
   {
    var j=1;
    var mt = "<table border='1' width='500' cellspacing='0'cellpadding='5'>";
    for(i=1;i<=rows;i++)
    {
     mt = mt + "<tr>";
     while(j<=cols)
     {
      mt = mt + "<td>" + i*j + "</td>";
      j = j+1;
     }
      mt = mt + "</tr>";
      j = 1;
```

```
    }
    mt = mt + "</table>";
    document.write(mt);
    }
  </script>
</head>
<body>
</body>
</html>
```

RESULT/OUTPUT





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
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52	54	56	58	60
3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54	57	60	63	66	69	72	75	78	81	84	87	90
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80	84	88	92	96	100	104	108	112	116	120
5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	105	110	115	120	125	130	135	140	145	150
6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96	102	108	114	120	126	132	138	144	150	156	162	168	174	180
7	14	21	28	35	42	49	56	63	70	77	84	91	98	105	112	119	126	133	140	147	154	161	168	175	182	189	196	203	210
8	16	24	32	40	48	56	64	72	80	88	96	104	112	120	128	136	144	152	160	168	176	184	192	200	208	216	224	232	240
9	18	27	36	45	54	63	72	81	90	99	108	117	126	135	144	153	162	171	180	189	198	207	216	225	234	243	252	261	270
10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200	210	220	230	240	250	260	270	280	290	300

CONCLUSION

We understood how to link a HTML code within JavaScript code. We were able to display the multiplication table according to the user given number of rows and columns.

EXPERIMENT NUMBER- 05

Question-3:

AIM

Write a java script function which converts Fahrenheit to Celsius.

DESCRIPTION

To create a function in JavaScript we use a keyword “function” along with the name of the function, parameter passed in it. A “return” statement returns back the value so that we could display the output using console.log(). Converting Fahrenheit to Celsius is based on a formula (F-32)*(5/9).

PROGRAM

```
var n=prompt("Enter the temperature in Fahrenheit:")
function convert(n)
{
   return (n-32)*(5/9)
}
result=convert(n)
console.log(result)
```

RESULT/OUTPUT

```
Enter the temperature in Fahrenheit:67
19.444444444444446
```

CONCLUSION

We understood the way we use functions in JavaScript and the functionality of a return statement by converting a user given Fahrenheit temperature to Celsius.

EXPERIMENT NUMBER- 05

Question-4:

AIM

Write a JavaScript program which accept a number as input and insert dashes (-) between each two even numbers. For example if you accept 025468 the output should be 0-254-6-8.

DESCRIPTION

We should insert a '-' in between two digits in a given number , so we check whether the two consecutive numbers are divisible by two or not and accordingly insert '-'in between them using "splice" operation. Splice operation can be used to replace the items in an array or remove the items in an array according to the number of arguments passed inside the function. "Split" function is used to divide each digit of the given number in to a single entity and place it in an array and at the end we "join" the modified array and display it using console.log().

CODE

```
var n=prompt("Enter the number:")
p=n.split('')
i=0
while(i<p.length)
{
   if(p[i]%2==0 && p[i+1]%2==0)
   {
      p.splice(i+1,0,'-')
   }
   i++
}
console.log(p.join(''))
```

RESULT/OUTPUT

```
Enter the number:025468
0-254-6-8
```

CONCLUSION

We understood the functionalities of two inbuilt JavaScript functions which are "split" and "splice" where one is used to split each element to an entity and the other is used to update or remove an element in an array or string.

EXPERIMENT NUMBER- 05

Question-5:

AIM

Write a JavaScript program to find the most frequent item of an array.

DESCRIPTION

We need to find the most occurred item in an given array so we check the frequency of every element in the array using for loop and store the frequency in another array arr1[]. Finally we return the maximum value of the arr1[] and corresponding item. “prompt” function and for loops are used for taking inputs and for iterations respectively.

PROGRAM

```
var n=prompt("Enter the length:")
var arr1=[]
for(i=0;i<n;i++)
{
   var t=prompt("Enter Element:")
   arr1.push(t)
}
var m=0
var k=0
for(i=0;i<arr1.length;i++)
{
   var c=0
   for(j=i;j<arr1.length;j++)
   {
      if(arr1[i]==arr1[j])
      {
         c+=1
      }
   }
   if(c>m)
   {
      m=c
      k=i
   }
}
console.log(arr1[k]+' '+'frequency is:'+m)
```

RESULT/OUTPUT

```
Enter the length:13
Enter Element:3
Enter Element:a
Enter Element:a
Enter Element:a
Enter Element:2
Enter Element:3
Enter Element:a
Enter Element:3
Enter Element:a
Enter Element:2
Enter Element:4
Enter Element:9
Enter Element:3
a frequency is:5
```

CONCLUSION

We found the frequency of the most repeated item by taking the elements of the array from the user along with the length of the array.

EXPERIMENT NUMBER- 06

Question-1:

AIM

To Validate form fields using javascript. In this assignment you have to validate form fields using javascript.

DESCRIPTION

Here, we need to create a form and validate it using javascript. If any of the things went wrong an error message should be shown below that field.

PROGRAM

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
 <head>
  <title>Validation Form</title>
  <style>
   input[type="text"],
   input[type="email"],
   input[type="date"],
   textarea,
   select {
    width: 300px;
   }

   h3 {
    text-align: center;
    margin-right: 60px;
   }

   tr {
    vertical-align: top;
   }

   td {
    padding-left: 100px;
   }

   form {
```

```
      margin-left: 210px;
      margin-right: 190px;
      border:2px solid black;
    }

    input[type="submit"] {
      margin-left: 100px;
      background-color: rgb(82, 148, 223);
      color: white;
      padding: 5px;
      font-style: bold;
      margin-bottom: 5px;
    }

    .error {
      display: none;
      color: red;
      margin-top: 5px;
    }
    div{
      background-color: rgb(150, 196, 150);
      border:4px solid rgb(56, 146, 56);
      padding: 5px;
    }
  </style>
</head>
<body>
  <form action="">
    <h3>Personal details</h3>
    <table cellspacing="10px">
      <tr>
        <td>Your name</td>
        <td>Email</td>
      </tr>
      <tr>
        <td>
          <input type="text" class="name" />
          <p class="error"></p>
        </td>
        <td>
          <input class="email" type="text" />
          <p class="error"></p>
        </td>
      </tr>
      <tr>
        <td>Gender</td>
```

```
  <td>Date of birth</td>
 </tr>
 <tr>
  <td>
   <input type="radio" value="1" name="radioBtn" />male
   <input type="radio" value="2" name="radioBtn" />female
   <p class="error"></p>
  </td>
  <td>
   <input type="date" />
   <p class="error"></p>
  </td>
 </tr>
 <tr>
  <td>Mobile</td>
  <td>Address</td>
 </tr>
 <tr>
  <td>
   <input type="text" class="mobile" />
   <p class="error"></p>
  </td>
  <td>
   <textarea name="" id="" cols="30" rows="3"></textarea>
   <p class="error"></p>
  </td>
 </tr>
 <tr>
  <td>City</td>
  <td>Country</td>
 </tr>
 <tr>
  <td>
   <select name="" class="city">
    <option value="0">Select your city</option>
    <option value="1">Hyderabad</option>
    <option value="2">Nizamabad</option>
    <option value="3">Secunderabad</option>
   </select>
   <p class="error"></p>
  </td>
  <td>
   <input class="country" type="text" />
   <p class="error"></p>
  </td>
 </tr>
```

```
        <tr>
          <td>Expertise</td>
          <td>Group</td>
        </tr>
        <tr>
          <td>
            <input type="checkbox" name="radioBtn2" />HTML
            <input type="checkbox" name="radioBtn2" />CSS
            <input type="checkbox" name="radioBtn2" />Javascript
            <input type="checkbox" name="radioBtn2" />jQuery
            <p class="error"></p>
          </td>
          <td>
            <select multiple="multiple">
              <option value="1">Family</option>
              <option value="2">Friend</option>
              <option value="3">Co-worker</option>
              <option value="4">Neighbor</option>
            </select>
            <p class="error"></p>
          </td>
        </tr>
      </table>
      <input type="submit" value="Save" />
    </form>


  <div id="success" style="display: none">
    <h2>Congratulations!</h2>
    <p>You have successfully validated the form</p>
    <a href="./form_validation.xhtml" class="link">Show the form again</a>
  </div>
  <script src="validate.js"></script>
 </body>
</html>
```

Validate.js

```
//  var nameInput=document.querySelector('.name')
// var emailInput=document.querySelector('input[type="email]')
// var mobileInput=document.querySelector('.mobile')
// var buttonInput=document.querySelector('button[type="submit"]')
// buttonInput.addEventListener('click',function(event){
// event.preventDefault();
// name_given=nameInput.value
// email=emailInput.value
// mobile=mobileInput.value
```

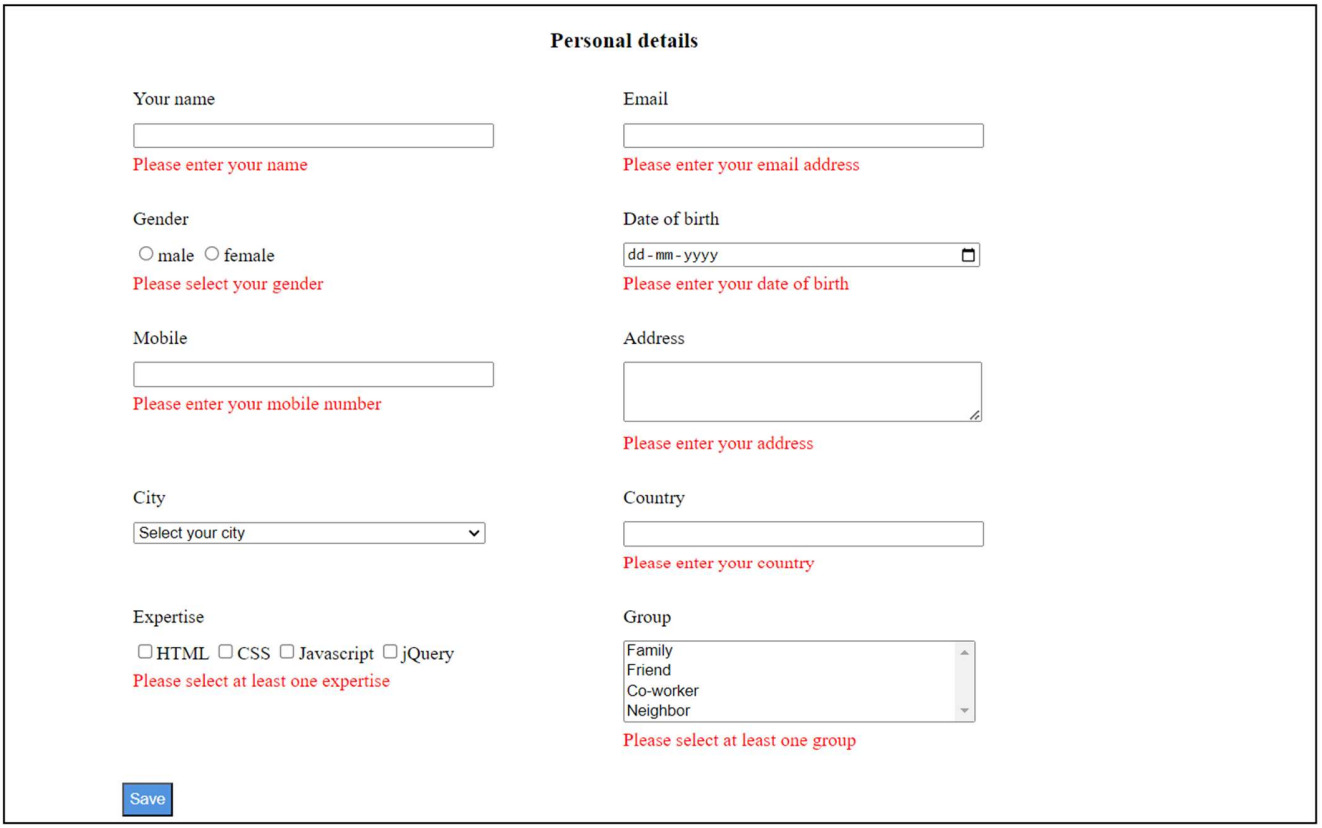
```
// alert('hi')
// if(!name_given || !email || !mobile){
//     alert('please fill the fields..')
// }else{
//     alert(`${name_given}\n${email}\n${mobile}`)
// }

const form = document.querySelector('form');

// Add an event listener for form submission
form.addEventListener('submit', function(event) {
  event.preventDefault(); // Prevent form submission
  // Reset error messages
  const errorMessages = document.querySelectorAll('.error');
  errorMessages.forEach(function(errorMessage) {
    errorMessage.style.display = 'none';
  });

  // Validate form fields
  const nameInput = document.querySelector('.name');
  const emailInput = document.querySelector('.email');
  const mobileInput = document.querySelector('.mobile');
  const genderInputs = document.querySelectorAll('input[name="radioBtn"]');
  const dobInput = document.querySelector('input[type="date"]');
  const addressInput = document.querySelector('textarea');
  const cityInput = document.querySelector('.city');
  const countryInput = document.querySelector('.country');
  const expertiseInputs = document.querySelectorAll('input[name="radioBtn2"]');
  const groupInput = document.querySelector('select[multiple="multiple"]');
  const link=document.querySelector('.link')

  let isValid = true; // Track form validity

  // Validate name
  if (nameInput.value.trim() === '') {
    showErrorMessage(nameInput, 'Please enter your name');
    isValid = false;
  } else if (nameInput.value.length < 3) {
    showErrorMessage(nameInput, 'Name should be at least 3 characters long');
    isValid = false;
  }

  // Validate email
  if (emailInput.value.trim() === '') {
    showErrorMessage(emailInput, 'Please enter your email address');
    isValid = false;
```

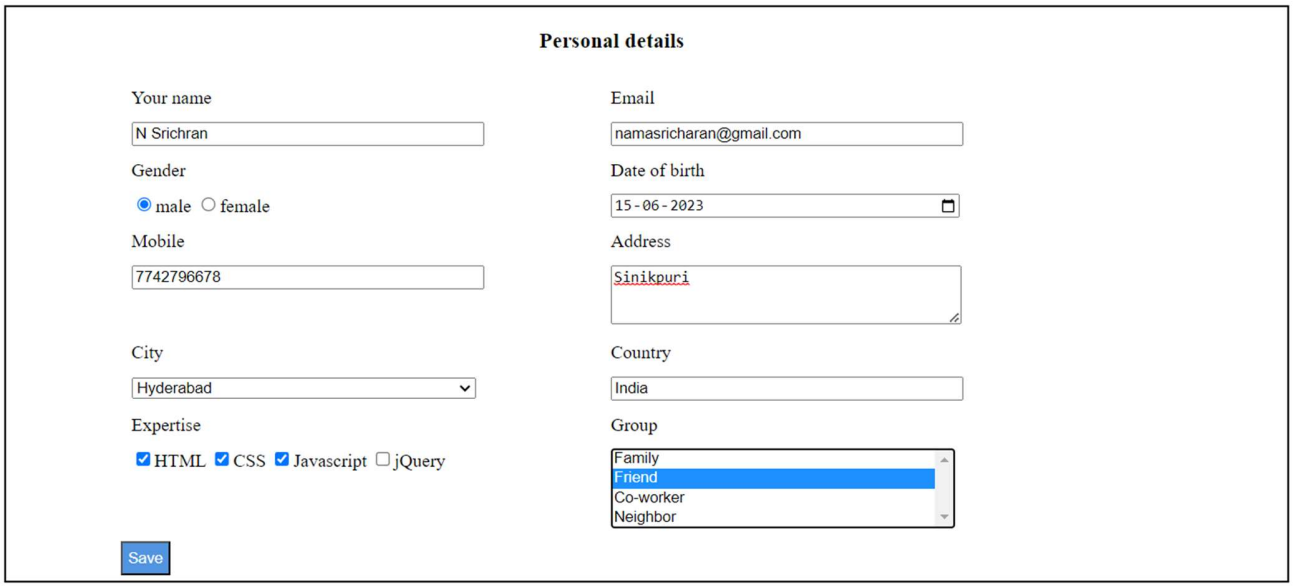
```
} else if (!validateEmail(emailInput.value)) {
  showErrorMessage(emailInput, 'Please enter a valid email address');
  isValid = false;
}

// Validate mobile number
if (mobileInput.value.trim() === '') {
  showErrorMessage(mobileInput, 'Please enter your mobile number');
  isValid = false;
} else if (!validateMobileNumber(mobileInput.value)) {
  showErrorMessage(mobileInput, 'Mobile number should be numeric and 10 digits');
  isValid = false;
}

// Validate gender
let genderSelected = false;
genderInputs.forEach(function(genderInput) {
  if (genderInput.checked) {
    genderSelected = true;
  }
});
if (!genderSelected) {
  showErrorMessage(genderInputs[0],'Please select your gender')
  // const genderError = genderInputs[0].parentElement.querySelector('.error');
  // genderError.textContent = 'Please select your gender';
  // genderError.style.display = 'block';
  isValid = false;
}

// Validate date of birth
if (dobInput.value.trim() === '') {
  showErrorMessage(dobInput, 'Please enter your date of birth');
  isValid = false;
}

// Validate address
if (addressInput.value.trim() === '') {
  showErrorMessage(addressInput, 'Please enter your address');
  isValid = false;
}

// Validate city
if (cityInput.value === '') {
  showErrorMessage(cityInput, 'Please select your city');
  isValid = false;
}
```

```
  // Validate country
  if (countryInput.value.trim() === '') {
    showErrorMessage(countryInput, 'Please enter your country');
    isValid = false;
  }

  // Validate expertise
  let expertiseSelected = false;
  expertiseInputs.forEach(function(expertiseInput) {
    if (expertiseInput.checked) {
      expertiseSelected = true;
    }
  });
  if (!expertiseSelected) {
    const expertiseError = expertiseInputs[0].parentElement.querySelector('.error');
    expertiseError.textContent = 'Please select at least one expertise';
    expertiseError.style.display = 'block';
    isValid = false;
  }

  // Validate group
  if (groupInput.selectedOptions.length === 0) {
    showErrorMessage(groupInput, 'Please select at least one group');
    isValid = false;
  }

  // If form is valid, hide the form and show success message
  if (isValid) {
    form.reset();
    form.style.display = 'none';
    document.querySelector('#success').style.display = 'block';

  }
});

// Function to validate email address
function validateEmail(email) {
  const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
  return emailRegex.test(email);
}

// Function to validate mobile number
function validateMobileNumber(mobile) {
  const mobileRegex = /^\d{10}$/;
  return mobileRegex.test(mobile);
```

```
}

// Function to show error message
function showErrorMessage(input, message) {
  const errorMessage = input.parentElement.querySelector('.error');
  errorMessage.textContent = message;
  errorMessage.style.display = 'block';
}
```

OUTPUT





CONCLUSION

We have successfully validated the form using javascript.

Question-2:

AIM

To write a javascript code to display a table to order a pizza.

DESCRIPTION

Here, we will create a table using xhtml table tag.then we give input tags to one of the column where the data will be given dynamically during the execution. By using javascript we compute the total cost. At last we will return the user of confirmation of their order.

PROGRAM

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
 <head>
  <title>Pizza Order Form</title>
  <style>
   input[type='button'],input[type='reset'],.total{
    margin-top: 20px;
   }
  </style>
 </head>
 <body>
  <form action="">
     <h1>Pizza order form</h1>
     <table border="1px">
        <tr>
           <th>Item name</th>
           <th>Price</th>
           <th>Quantity</th>
        </tr>
        <tr>
           <td>Chicken Pizza</td>
           <td>100</td>
           <td>
              <input class='cp' type="text"/>
           </td>
        </tr>
        <tr>
          <td>Panner Pizza</td>
          <td>80</td>
```

```
            <td>
               <input class="pp" type="text"/>
            </td>
          </tr>
          <tr>
            <td>Veg Pizza</td>
            <td>70</td>
            <td>
               <input class="vp" type="text"/>
            </td>
        </tr>
        </table>
        <input onclick="calculate_total_cost();" type="button" value="total cost"/>
        <input class='total' onfocus="this.blur()" type="text"/><br/>
        <input onclick="confirm_order();" type="button" value="confirm order"/>
        <input type="reset" value="cancel order"/>
    </form>
    <script src="pizza_js.js"></script>
  </body>
</html>
```

Pizza_js.js

```
function calculate_total_cost(){
    var qtyCP=document.querySelector('.cp').value;
    var qtyPP=document.querySelector('.pp').value;
    var qtyVP=document.querySelector('.vp').value;
    document.querySelector('.total').value=100*qtyCP+80*qtyPP+70*qtyVP;
}
function confirm_order(){
    var qtyCP=document.querySelector('.cp').value;
    var qtyPP=document.querySelector('.pp').value;
    var qtyVP=document.querySelector('.vp').value;
    if(!qtyCP && !qtyPP && !qtyVP){
        alert('please select atlest one quantity')
    }else{
        confirm('Your order is confirmed,reach you in 10minutes')
    }
}
```

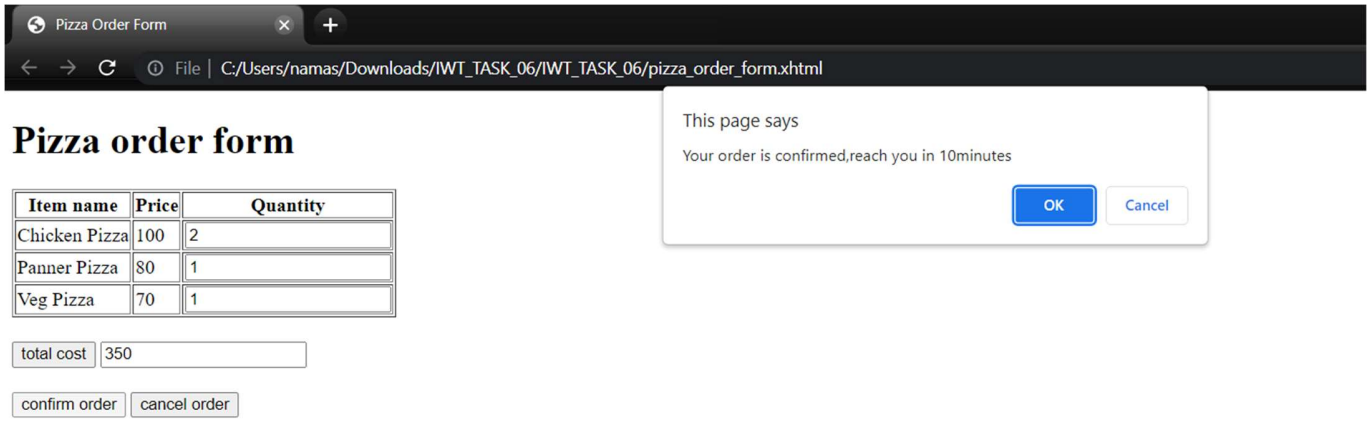
OUTPUT

Pizza order form

Item name	Price	Quantity
Chicken Pizza	100	2
Panner Pizza	80	1
Veg Pizza	70	1

total cost 350

confirm order cancel order



CONCLUSION

We have successfully created a pizza order form and validated using javascript.

Question-3:

AIM

To Write a javascript code for designing a basic calculator.

DESCRIPTION

Here, we design a calculator using table in xhtml. To compute the expressions here we use javascript code.

PROGRAM

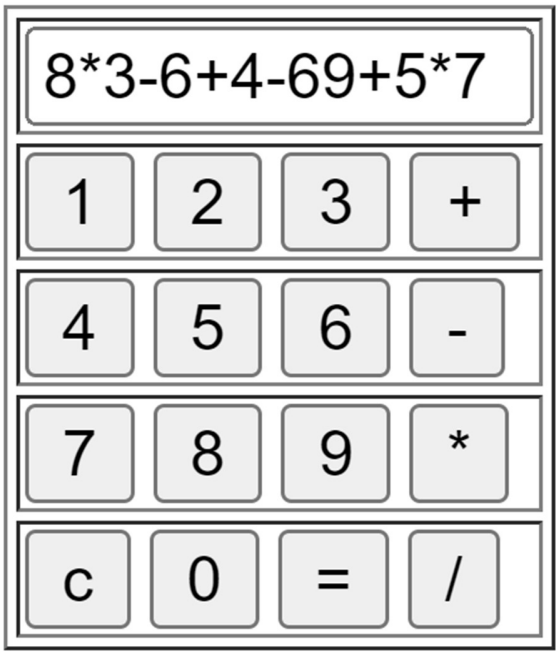
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
   <title>calculator</title>
   <style>
      .input{
         width: 100px;
      }
   </style>
</head>
<body>
   <form action="">
      <table border="1px">
         <tr>
            <td>
               <input class='input' type="text" />
            </td>
         </tr>
         <tr>
            <td>
               <input onclick='display(1);' type="button" class='btn1' value="1" />
               <input onclick='display(2);' type="button" class='btn2' value="2" />
               <input onclick='display(3);' type="button" class='btn3' value="3" />
               <input onclick="display('+');" type="button" class='btn+' value="+" />
            </td>
         </tr>
         <tr>
            <td>
               <input onclick='display(4);' type="button" class='btn4' value="4" />


}
```

```
<input onclick='display(5);' type="button" class='btn5' value="5" />
                <input onclick='display(6);' type="button" class='btn6' value="6" />
                <input onclick='display("-");' type="button" class='btn-' value="-" />
            </td>
        </tr>
        <tr>
            <td>
                <input onclick='display(7);' type="button" class='btn7' value="7" />
                <input onclick='display(8);' type="button" class='btn8' value="8" />
                <input onclick='display(9);' type="button" class='btn9' value="9" />
                <input onclick='display("*");' type="button" class='btn' value="*" />
            </td>
        </tr>
        <tr>
            <td>
                <input type="reset" class='btn1' value="c" />
                <input onclick='display(0);' type="button" class='btn1' value="0" />
                <input onclick='perform("=");' type="button" class='btnEql' value="=" />
                <input onclick='display("/");' type="button" class='btn1' value="/" />
            </td>
        </tr>
    </table>
</form>
<script src="calci.js"></script>
</body>
</html>
```

calci.js

```
function display(value) {
    document.querySelector('.input').value += value;
}
function perform() {
    var inputField = document.querySelector('.input');
    var expression = inputField.value;
    var result = eval(expression);
    inputField.value = result;
}
```

OUTPUT



CONCLUSION

We have successfully designed a basic calculator using HTML,CSS and Javascript.

Question-4:

AIM

To write a program to handle Mouse Events.

DESCRIPTION

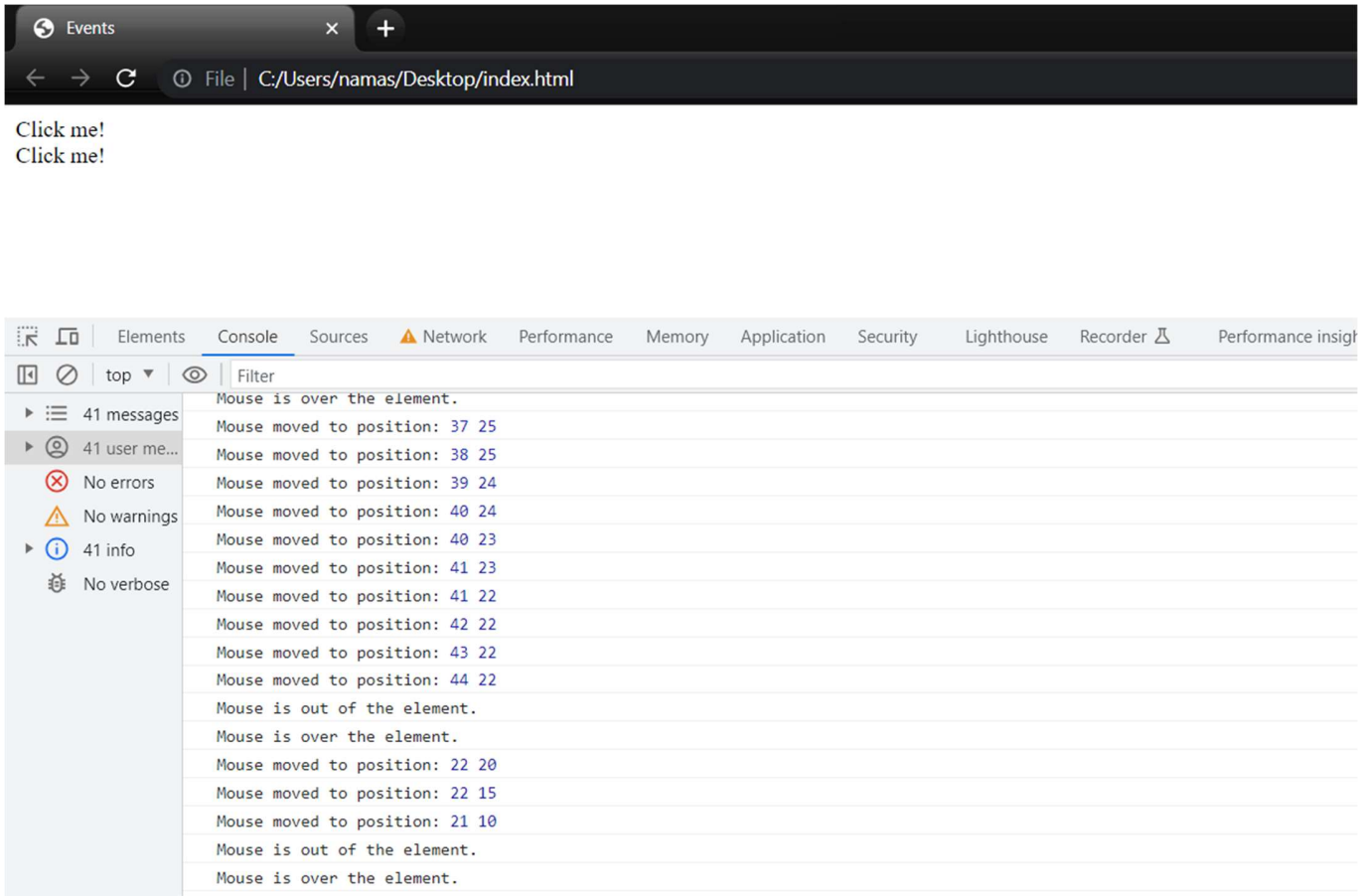
Here, we write a program to handle mouse events using javascript.

PROGRAM

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Events</title>
  </head>
  <body>
    <div id="myElement">Click me!</div>
    <div id="myElement">Click me!</div>
    <script>
  // Function to handle mouse click event
  function handleClick(event) {
    console.log('Mouse clicked at position:', event.clientX, event.clientY);
  }
  // Function to handle mouse move event
  function handleMouseMove(event) {
    console.log('Mouse moved to position:', event.clientX, event.clientY);
  }
  // Function to handle mouse over event
  function handleMouseOver(event) {
    console.log('Mouse is over the element.');
  }
  // Function to handle mouse out event
  function handleMouseOut(event) {
    console.log('Mouse is out of the element.');
  }
  // Get the element on which the mouse events will be handled
  var element = document.getElementById('myElement');

  // Attach event listeners to the element
  element.onclick = handleClick;
  element.onmousemove = handleMouseMove;
```

```
    element.onmouseover = handleMouseOver;
    element.onmouseout = handleMouseOut;
      </script>

    </body>
</html>
```

OUTPUT



CONCLUSION

We have successfully handled mouse events and displayed the positions in console.

EXPERIMENT NUMBER- 07

Question-1:

AIM

A program to demonstrate absolute and relative positioning of elements

DESCRIPTION

1. Relative Positioning:
- When an element is set to `position: relative`, it remains within the normal document flow.
- The element is positioned relative to its original position in the document or to its nearest positioned ancestor (an ancestor with a position value of `relative`, `absolute`, or `fixed`).
- You can use properties like `top`, `bottom`, `left`, and `right` to move the element from its original position, relative to its normal flow.
- Other elements in the document still recognize the space occupied by the relatively positioned element, even if it has been moved.

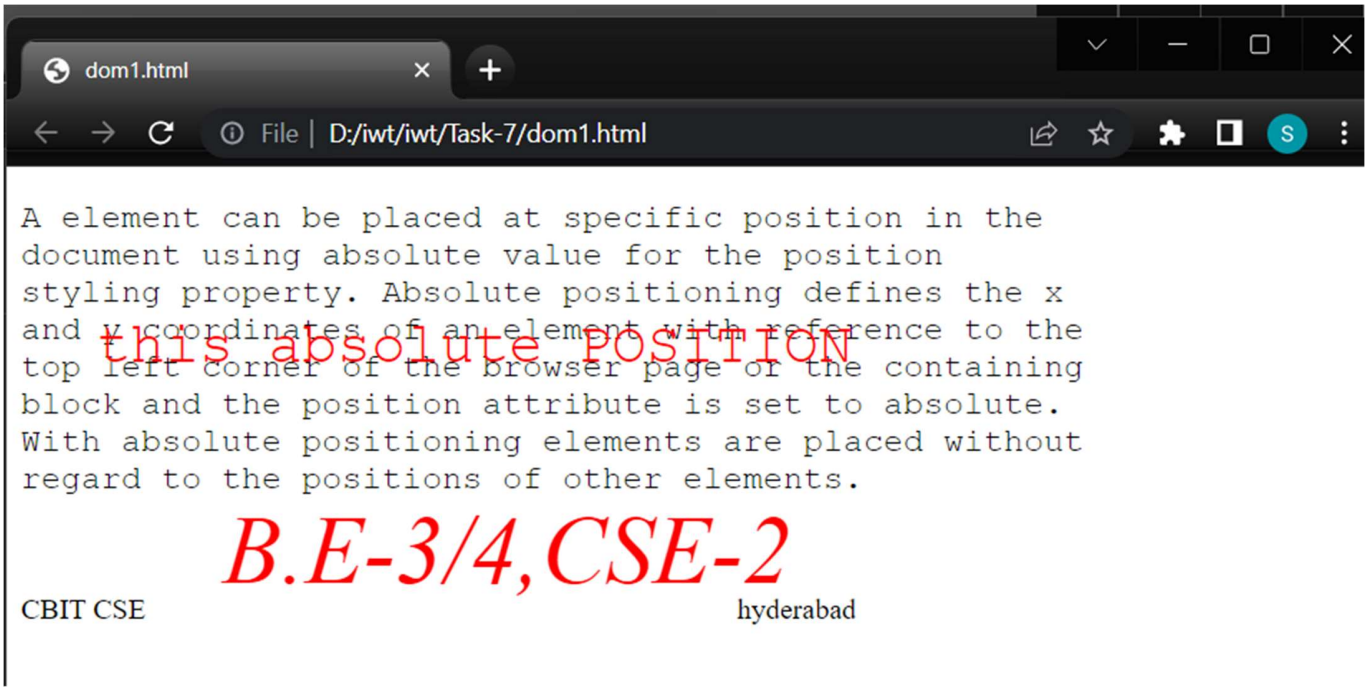
2. Absolute Positioning:
- When an element is set to `position: absolute`, it is removed from the normal document flow and positioned relative to its nearest positioned ancestor (an ancestor with a position value of `relative`, `absolute`, or `fixed`).
- The element's position is determined by using the `top`, `bottom`, `left`, and `right` properties. These properties specify the distance between the element and the edges of its nearest positioned ancestor.
- Absolute positioning allows you to precisely place an element anywhere on the page, overriding its normal flow.
- Other elements in the document may overlap the absolutely positioned element, and they won't be affected by its presence in terms of layout and spacing.

PROGRAM

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
   <style type="text/css">
      .regtext {
         font-family: 'Courier New', Courier, monospace;

         font-size: 14pt;
```

```
            width: 600px
        }

        .abstext {
            position: absolute;
            top: 50px;
            left: 50px;
            font-family: 'Courier New', Courier, monospace;
            width: 500px;
            color: red;
            font-size: 24pt;
        }
    </style>
</head>

<body>
    <p class="regtext">
        A element can be placed at specific position in the document using absolute
        value for the position styling property. Absolute positioning defines the x and y
        coordinates of an element with reference to the top left corner of the browser page or the
        containing block and the position attribute is set to absolute. With absolute positioning
        elements are placed without regard to the positions of other elements.
    </p>
    <p class="abstext">
        this absolute POSITION
    </p>
    <p>
        CBIT CSE<span style="position:relative;top:-20px;left:30px;
    font-family:Times;font-size:40pt;
    font-style:italic;color:red;">
            B.E-3/4,CSE-2</span>hyderabad
    </p>

</body>


</html>
```

OUTPUT



CONCLUSION

The program successfully demonstrates the concepts of absolute and relative positioning of elements in web development. This experiment highlights the flexibility and power of CSS positioning properties, offering web developers versatile tools to control the layout and presentation of elements on their websites. The use of the position property in CSS allows us to control the precise placement of elements on a web page.

Question-2:

AIM

A program to display a image when mouse is clicked and hide that image when mouse released.

DESCRIPTION

1. Mouse Events:
   - JavaScript is used to handle mouse events and manipulate the image's display property.
   - When the mouse is clicked, a `mousedown` event is triggered.
   - On the `mousedown` event, a JavaScript function is called that selects the image element and changes its `display` property to "block", making the image visible.

2. Image Hide:
   - When the mouse is released, a `mouseup` event is triggered.
   - On the `mouseup` event, another JavaScript function is called that selects the image element and changes its `display` property back to "none", hiding the image.

3. Event Listeners:
   - Event listeners are added to the HTML elements to listen for the mouse events.
   - The image container or any suitable HTML element can be used as the target for the event listeners.
   - When the mouse is clicked, the `mousedown` event listener calls the function to display the image.
   - When the mouse is released, the `mouseup` event listener calls the function to hide the image.

PROGRAM

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">



  <head>
    <title>Mouse Click Image Display</title>
    <style>
      #image {
        display: none;
        width: max-content;
```

```
            }
        </style>
        <script>
            function showImage() {
                var image = document.getElementById("image");
                image.style.display = "block";
            }

            function hideImage() {
                var image = document.getElementById("image");
                image.style.display = "none";
            }
        </script>
    </head>

    <body>
        <div>
            <h1>Click on mouse to display image</h1>
        </div>
        <div id="imageContainer" style="width: 600px; height: 600px;">
            <img id="image" src="Cricket.jpg" alt="cricket" />
        </div>

        <script>

            var imageContainer = document.getElementById("imageContainer");
            imageContainer.addEventListener("mousedown", showImage);
            imageContainer.addEventListener("mouseup", hideImage);
        </script>
    </body>

</html>
```

OUTPUT

Click on mouse to display image

CONCLUSION

This experiment successfully demonstrates how to show an image when the mouse is clicked and hide it when the mouse is released. By utilizing JavaScript event listeners and manipulating the CSS display property, we can dynamically control the visibility of elements on a web page based on user interactions. This functionality can be useful for creating interactive image galleries, tooltips, pop-ups, or any scenario where you want to reveal or hide content based on user input.

Question-3:

AIM

A program to demonstrate z-index property.(or) a program to demonstrate stacking of elements using z-index property

DESCRIPTION

1. Z-Index Property:
   - The `z-index` property determines the stacking order of elements on the z-axis (depth axis) in a positioned context.
   - A higher `z-index` value brings an element closer to the viewer, making it appear in front of elements with lower `z-index` values.
   - By default, elements have a `z-index` value of `auto`, which means they are stacked in the order they appear in the HTML structure.

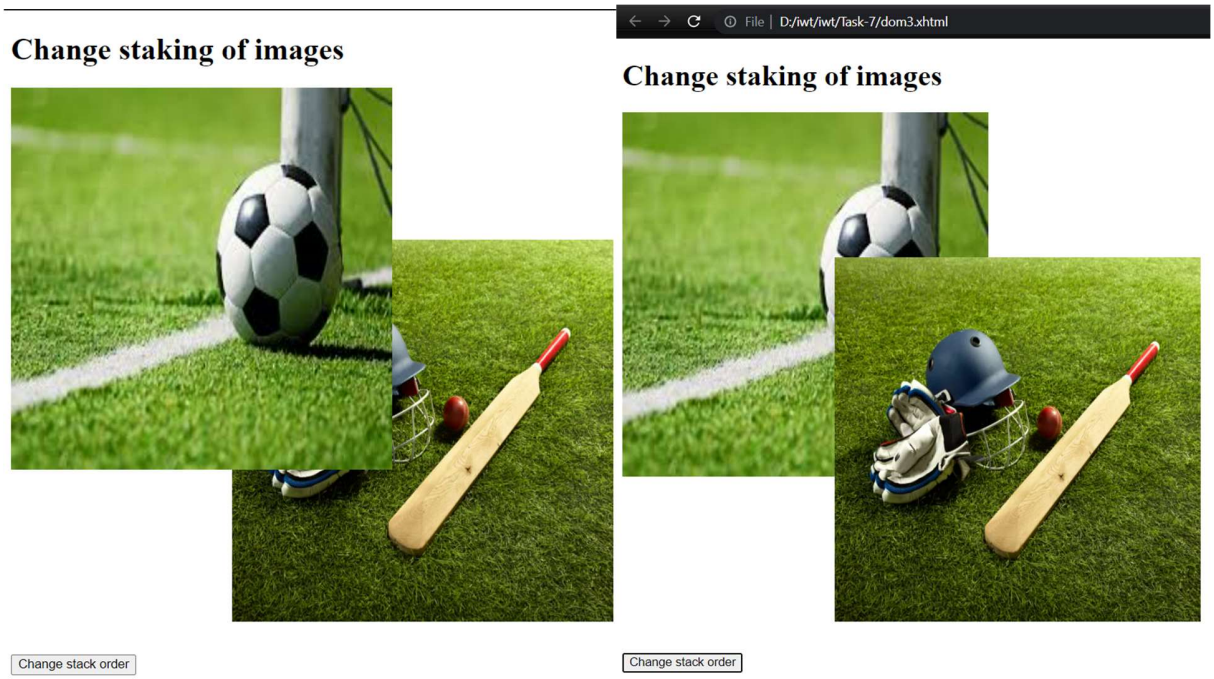
2. Stacking Elements:
   - In the program, different elements are assigned different `z-index` values to control their stacking order.
   - Higher `z-index` values are assigned to elements that should appear on top of other elements.

PROGRAM

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
   <style>
      #img1 {
         position: absolute;
         left: 240px;
         top: 240px;
         z-index: -1
      }
   </style>
</head>

<body>
   <h1>Change staking of images</h1>
```

```
    <div>
      <img id="img1" src="Cricket.jpg" width="400" height="400" />
      <img id="img2" src="football.jpg" width="400" height="400" />

    </div>
    <div style="position: absolute; bottom: 50px;">
<button type="button" onclick="myFunction()">Change stack order</button>
    </div>
    <script>
      function myFunction() {
        document.getElementById("img1").style.zIndex = "1";
      }
    </script>
</body>



</html>
```

OUTPUT



CONCLUSION

This experiment demonstrates how the z-index property can be used to control the stacking order of elements in a web page. By assigning different z-index values to elements, we can determine which elements appear in front of or behind others. By using the z-index property wisely, web developers can create complex layouts and control the visual hierarchy of elements on their web pages.

Question-4:

AIM

A program to insert the rows into a table dynamically.

DESCRIPTION

1. HTML Structure:
   - The program consists of an HTML file with the necessary structure.
   - It includes a table element `<table>` with an id attribute (`id="myTable"`).
   - The table already has a header row `<tr>` with column headers `<th>` and an initial row of data `<tr>`.

2. JavaScript Function:
   - The program includes a JavaScript function called `addRow()`.
   - This function is triggered when a button is clicked.

3. Accessing the Table:
   - Inside the `addRow()` function, the table element is accessed using `document.getElementById("myTable")`.
   - The `getElementById` method allows us to get a reference to the table element based on its unique id.

4. Creating and Inserting Rows:
   - The `insertRow()` method is called on the table element to create a new row.
   - The new row is then inserted into the table using the `insertRow()` method.
   - The `insertCell()` method is called on the new row to create cells within the row.
   - The `innerHTML` property is used to set the content of each cell to some text (e.g., "New Cell 1", "New Cell 2").

5. Triggering the Function:
   - The program includes a button element `<button>` with an `onclick` attribute that triggers the `addRow()` function when clicked.

PROGRAM

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

```
<head>
   <title>Manipulating Tables</title>
   <script type="text/javascript">
      function addRow(tableId, c) {
         var tabEle = document.getElementById(tableId);
         var newRow = tabEle.insertRow(tabEle.rows.length);
         var newCell;
         for (var i = 0; i < c.length; i++) {
            newCell = newRow.insertCell(newRow.cells.length);
            newCell.innerHTML = c[i];
         }
         document.getElementsByClassName('input1')[0].value=""
         return newRow;
      }

   </script>
</head>

<body>
   <table id="tblPeople" border="1">
      <tr>
         <th>First Name</th>
         <th>Middle Name</th>
         <th>Last Name</th>
      </tr>
   </table>
   <hr>
   <form>
      First Name: <input class="input1" type="text" name="FirstName" />
      Middle Name: <input class="input1" type="text" name="MiddleName" />
      Last Name: <input class="input1" type="text" name="LastName" />
      <br />
      <br />
      <center>
         <input type="button" value="Add Name" onclick="addRow('tblPeople',
      [this.form.FirstName.value,this.form.MiddleName.value,
      this.form.LastName.value]
      );" />

      </center>
   </form>
   <hr>
</body>

</html>
```

OUTPUT

First Name	Middle Name	Last Name
Zaid	Ahmed	Khan
Mahendra	Singh	Dhoni
Virat	King	Kohli

First Name: Middle Name: Last Name:

Add Name

CONCLUSION

This experiment demonstrates how to dynamically insert rows into an HTML table using JavaScript. By utilizing the insertRow() and insertCell() methods, we can programmatically add new rows and cells to the table. By adding rows dynamically, we can enhance the functionality and flexibility of our web applications, providing a seamless user experience.

Question-5:

AIM

A program to track the mouse coordinates.

DESCRIPTION

1. Event Listeners:

- The program uses JavaScript event listeners to track the mouse movements and capture the coordinates.
- Specifically, it utilizes the `mousemove` event to detect when the mouse moves within a specified area.

2. Mouse Coordinate Tracking:

- When the `mousemove` event is triggered, a JavaScript function is called to track the mouse coordinates.
- The function accesses the event object's `clientX` and `clientY` properties to obtain the X and Y coordinates of the mouse relative to the viewport.

PROGRAM

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<body>
   <h2 onclick="showCoords(event)">Click this heading to get the
      x (horizontal) and y (vertical) coordinates of the mouse pointer when it was clicked.</h2>
   <p><strong>Tip:</strong> Try to click different places in the heading.</p>
   <p id="demo"></p>
   <script>
      function showCoords(event) {
         var x = event.clientX;
         var y = event.clientY;
         var coords = "X coords: " + x + ", Y coords: " + y;
         document.getElementById("demo").innerHTML = coords;
      }
   </script>
</body>

</html>
```

OUTPUT

Click this heading to get the x (horizontal) and y (vertical) coordinates of the mouse pointer when it was clicked.

Tip: Try to click different places in the heading.

X coords: 310, Y coords: 64

CONCLUSION

This experiment demonstrates how to track and display the mouse coordinates in a web page using JavaScript. By listening to the mousemove event, we can capture the X and Y coordinates of the mouse cursor and update the displayed values dynamically. Understanding how to access and utilize mouse events and their associated properties enables developers to create responsive and interactive user interfaces.

Question-6:

AIM

A program to move a image slowly with the help of either setTimeout() or setInterval().

DESCRIPTION

1. HTML Structure:
   - The program consists of an HTML file with the necessary structure.
   - It includes an `<img>` tag with the source attribute (`src`) pointing to the image file you want to move.
   - The image is positioned within a container element (e.g., a `<div>`), which will be used to control its movement.

2. CSS Styling:
   - The CSS styling is used to position the container element and apply any necessary styling to the image.
   - You can set the position of the container to `relative` or `absolute` to enable movement.

3. JavaScript Function:
   - The program includes a JavaScript function that will be responsible for moving the image.
   - This function will use either `setTimeout()` or `setInterval()` to create a delayed or repetitive movement effect.

PROGRAM

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
   <title> Moving Image </title>
   <script type="text/javascript" src="mover.js">
   </script>
</head>

<body onload="initImage()">
   <h1>This is a demo to show moving image dynamically</h1>
   <div id='image' style="position: absolute; left:0px; top:60px; margin-top: 25px;">
      <img src="Cricket.jpg" style="width:300px ; height: 300px;" />
   </div>
</body>

</html>
```

Mover.js

```
var dom,
```

```
  x,
  y,
  finalx = window.innerWidth - 300;

function initImage() {
  dom = document.getElementById("image").style;

  var x = parseInt(dom.left);
  var y = parseInt(dom.top);

  moveImage(x, y);
}

function moveImage(x, y) {
  if (x != finalx) {
    x++;

    dom.left = x + "px";
    dom.top = y + "px";

    setTimeout("moveImage(" + x + "," + y + ")", 1);
  }
}
```

OUTPUT

This is a demo to show moving image dynamically **This is a demo to show moving image dynamically**

CONCLUSION

This experiment demonstrates how to move an image slowly using JavaScript's setTimeout() function. By incrementing the position of the image element and updating its CSS property, we create a smooth animation effect. Using setTimeout() (or alternatively, setInterval()) enables us to control the timing of each step in the animation, allowing the image to move gradually and smoothly over time.

EXPERIMENT NUMBER- 08

Name of the Student: Sai Balaji Section : CSE-2 Roll No.160121733130

Question-1:

AIM

Steps to install Django and create a Django application.

DESCRIPTION

Django is a high-level web framework for building web applications using the Python programming language. It follows the Model-View-Controller (MVC) architectural pattern, also known as the Model-View-Template (MVT) pattern, which promotes clean and reusable code.

PROGRAM

```
# Step 1: Set up Python
# Ensure that Python is installed on your system.
python --version

# Step 2: Install virtualenv
pip install virtualenv

# Step 3: Create a Virtual Environment
# Navigate to the directory where you want to create your virtual environment.
virtualenv myenv

# Step 4: Activate the Virtual Environment
# On Windows
myenv\Scripts\activate

# On macOS/Linux
source myenv/bin/activate

# Step 5: Install Django
pip install django
# check whether Django is successfully installed or not
django-admin -version

# Step 6: Create a Django Project
django-admin startproject myproject

# Step 7: Change to the Project Directory
cd myproject

# Step 8: Create a Django Application
python manage.py startapp myapp
# Step 9: Configure the Application in settings.py
```

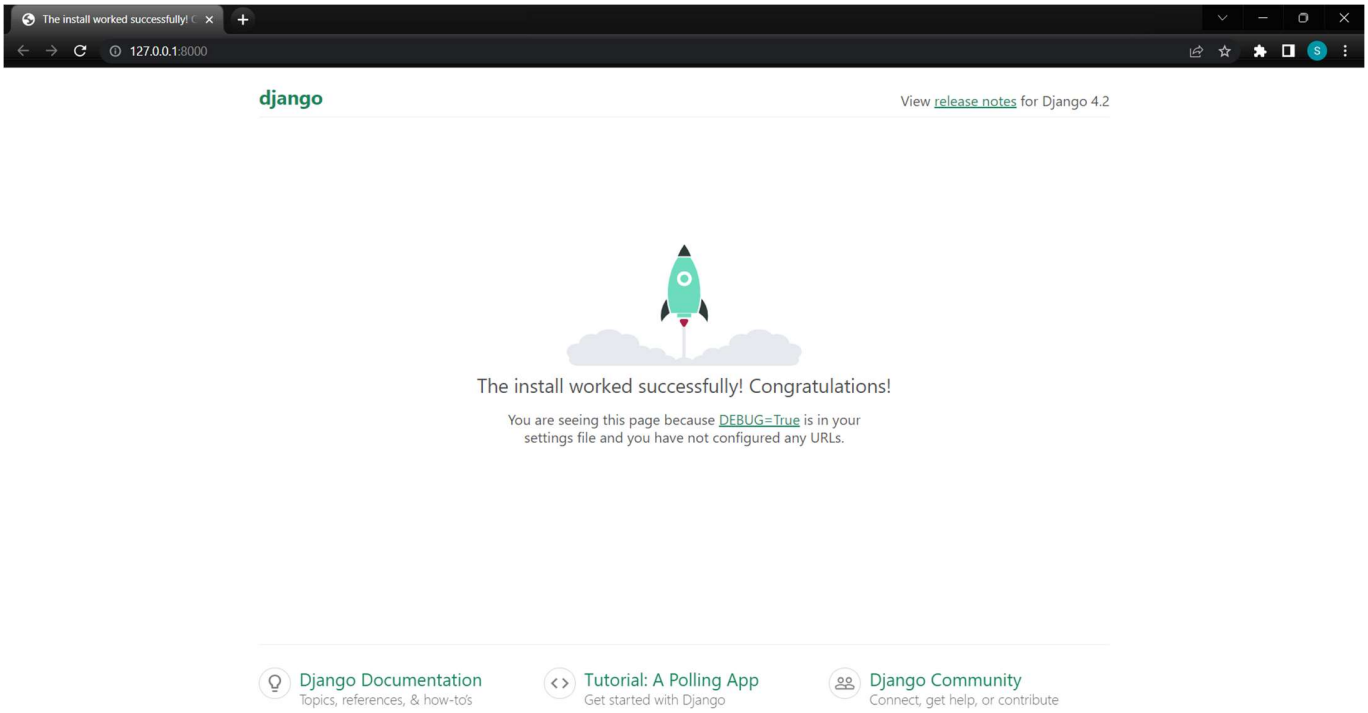
Open the **settings.py** file located in the project directory. Find the **INSTALLED_APPS** list and add your application to it

```
INSTALLED_APPS = [
    ...
    'myapp',
    ...
]
```

Step 10: Start the Development Server

```
python manage.py runserver
```

OUTPUT



CONCLUSION

This experiment provided step-by-step instructions on how to install Django and create a Django application. By following these steps, you can set up a Django project, define views and URLs, and run the development server to start building web applications.

Question-2:

AIM

Write a Django application to print a welcome message with the name passed from view.

DESCRIPTION

In Django, views are the components responsible for handling requests from clients and returning appropriate responses. A view function or class defines the logic and behavior of a particular web page or API endpoint. When a URL is accessed, Django uses the URL routing system to map the URL to a specific view, which then processes the request and generates a response.

PROGRAM

Step 1: Create a Django Project and Application

```
django-admin startproject first
cd first
python manage.py startapp firstApp
```

Step 2: Define the View
Open the file welcome_firstApp/views.py and add the following code:

```
from django.shortcuts import render

# Create your views here.

def hello(request):
    return render(request, 'index.html', {'name': 'Sricharan'})
```

Step 3: Create the Template
Create a new directory named templates inside the first project directory. Inside the templates directory, create a file named index.html and add the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1>Hello {{name}}</h1>
</body>
</html>
```

Make the following changes in settings.py of first project

```
from pathlib import Path
import os

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent
TEMPLATE_DIR = os.path.join(BASE_DIR, 'templates')

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'firstApp',
]

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [TEMPLATE_DIR,],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

Step 4: Configure the URL Routing
Create a new file named urls.py inside the firstApp directory. Add the following code to urls.py:

```
from django.urls import path
from . import views

urlpatterns = [

    path('', views.hello, name='hello'),
]
```

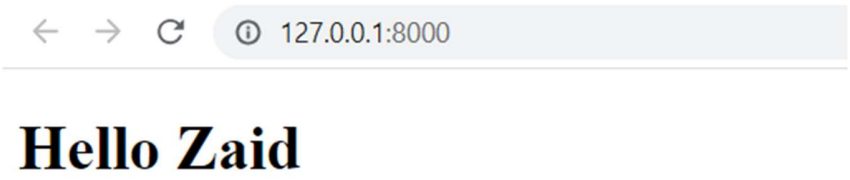
Step 5: Update the Project-level URL Configuration
Open the file first/urls.py and update the code as follows:

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('firstApp.urls')),
]
```

Step 6: Run the Development Server

Start the development server by running the command **python manage.py runserver**. Then, access the application in your web browser at http://127.0.0.1:8000/.

OUTPUT



CONCLUSION

This experiment demonstrates how to create a simple Django application that displays a welcome message with a name passed from a view. By following the steps outlined above, you can build a functioning web application that dynamically generates personalized welcome messages.
Django's ability to handle dynamic routing and template rendering allows for easy customization of web content based on user input.

Question-3:

AIM

Write a Django application to print addition of two numbers. Use a HTML form to read the numbers to be added and display the result in another HTML page.

DESCRIPTION

In Django, views are the components responsible for handling requests from clients and returning appropriate responses. A view function or class defines the logic and behavior of a particular web page or API endpoint. When a URL is accessed, Django uses the URL routing system to map the URL to a specific view, which then processes the request and generates a response.

PROGRAM

addNum/addApp/urls.py

```
from django.urls import path
from addApp import views

urlpatterns = [

    path('', views.addNum, name="addNum"),
    path('result', views.result, name="result"),
]
```

addNum/addApp/views.py

```
from django.shortcuts import render
from django.http import HttpResponse

# Create your views here.


Def addNum(request):
    return render(request, ‘add.html’)


def result(request):
    var1 = int(request.GET[“val1”])
    var2 = int(request.GET[“val2”])
    res = var1+var2

    return render(request, ‘result.html’, {‘result’: res})
```

addNum/addNum/urls.py

```
from django.contrib import admin
from django.urls import path
from django.conf.urls import include
from addApp import views

urlpatterns = [

    path('admin/', admin.site.urls),

    path('', include('addApp.urls')),

]
```

addNum/addNum/settings.py

```
from pathlib import Path
import os

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent
TEMPLATE_DIR = os.path.join(BASE_DIR, 'templates')

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'addApp',
]

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [TEMPLATE_DIR,],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
```

```
        },
    },
]
```

addNum/templates/add.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>Add two Numbers</title>

    </head>
    <body>

        <p>I am html file to add numbers</p>
        <form action="result">
            num1<input type="text" name="val1"/>
            num2<input type="text" name="val2"/>
            <button name="submit" value="add">Submit</button>
        </form>
    </body>

</html>
```

addNum/templates/result.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>I am result page</title>
    </head>
    <body>

        <p>hello this is result page.</p>
        <span>{{result}}</span>
    </body>
</html>
```

Run the Development Server

Start the development server by running the command **python manage.py runserver**. Then, access the application in your web browser at http://127.0.0.1:8000/.

OUTPUT

CONCLUSION

This experiment showcases how to create a simple Django application that performs the addition of two numbers using HTML forms and Django views. By following the outlined steps, we can build a functioning web application that takes user input, performs the addition operation, and displays the result on a separate HTML page.

This example highlights the use of Django's template system, view functions, and URL routing to create a seamless user experience. It demonstrates the power and simplicity of Django's web development framework in handling form submissions and rendering dynamic content.

EXPERIMENT NUMBER- 09

Question-1:

AIM

Exploration of web frameworks(AngularJS, JQuery, Flask, Web2Py, FuelPhP)

DESCRIPTION

Web frameworks provide developers with a structured and efficient way to build web applications. These frameworks come with pre-defined libraries, tools, and components that simplify common web development tasks, such as routing, handling requests and responses, managing databases, and rendering dynamic content.

PROGRAM

Exploration of Web Frameworks:

1. AngularJS:

AngularJS is a JavaScript-based front-end framework developed by Google. It is designed to build dynamic and responsive single-page applications (SPAs). AngularJS provides a powerful set of features such as two-way data binding, dependency injection, and templating, making it easier to develop complex web applications. It follows the Model-View-Controller (MVC) architecture and provides a robust framework for building interactive user interfaces.

AngularJS provides a robust framework for building dynamic SPAs. It offers features like data binding, which automatically synchronizes data between the model and view, reducing the need for manual DOM manipulation. AngularJS also includes a powerful dependency injection system, making it easier to manage and test application components. It provides a rich set of directives for handling user interactions, form validation, and routing, making it a comprehensive solution for front-end development.

2. jQuery:

jQuery is a popular JavaScript library that simplifies HTML document manipulation and event handling. It allows developers to manipulate the Document Object Model (DOM), handle events, create animations, and perform AJAX requests with ease. jQuery provides a concise and efficient way to write JavaScript code and is widely used for enhancing the functionality and interactivity of web pages.

jQuery simplifies client-side scripting by providing a concise syntax and a wide range of built-in functions. It excels in DOM manipulation, allowing developers to easily select elements, modify their attributes or content, and handle events. jQuery also provides AJAX functions for seamless communication with the server, making it easier to load and update data without refreshing the entire web page. Its extensive plugin ecosystem allows for additional functionality and customization.

3. Flask:

Flask is a lightweight and flexible web framework for Python. It provides a minimalistic approach to web development, allowing developers to choose their preferred components and extensions. Flask is easy to learn and use, making it suitable for small to medium-sized web applications. It offers routing, request handling, template rendering, and database integration, enabling developers to quickly build web applications using Python.

Flask is known for its simplicity and flexibility. It has a small core framework and allows developers to choose the components they need, making it highly customizable. Flask provides routing capabilities to map URLs to functions, enabling clean and structured URL handling. It supports various template engines for rendering dynamic HTML content. Flask also has excellent integration with databases through its SQLAlchemy extension, which offers powerful ORM features for database management.

4. Web2Py:

Web2Py is a full-stack web framework written in Python. It aims to provide simplicity and ease of use, allowing developers to quickly build scalable and secure web applications. Web2Py follows the Model-View-Controller (MVC) architectural pattern and provides features such as automatic database administration, form validation, and secure authentication. It also includes a web-based integrated development environment (IDE) for streamlined development.

Web2Py aims to provide a complete web development framework that is easy to use and secure. It includes an administration interface for managing databases, user authentication, and access control. Web2Py's form validation system simplifies data validation and helps prevent common security vulnerabilities. It also provides an abstraction layer for interacting with various databases, making it convenient to work with different database systems. Additionally, Web2Py supports internationalization and localization features out of the box.

5. FuelPHP:

FuelPHP is a modular and flexible web framework built on PHP. It emphasizes simplicity, performance, and security. FuelPHP follows the Model-View-Controller (MVC) pattern and provides features like routing, caching, input validation, and database ORM (Object-Relational Mapping). It aims to offer a balance between simplicity and extensibility, making it suitable for both small and large-scale web applications.

FuelPHP is known for its modular and flexible architecture. It allows developers to create reusable components called "packages," which can be easily integrated into applications. FuelPHP provides a robust ORM called "Oil" for database management, offering features like query building and relationships. It has built-in caching capabilities for optimizing application performance. FuelPHP also includes security features such as input validation, cross-site scripting (XSS) prevention, and output encoding to ensure secure coding practices.

CONCLUSION

In conclusion, the exploration of web frameworks reveals the diverse range of options available for web application development.

- AngularJS stands out for its ability to build dynamic single-page applications (SPAs) with powerful features like two-way data binding and dependency injection.
- jQuery simplifies client-side scripting by providing a concise syntax and extensive functionality for DOM manipulation and AJAX interactions.
- Flask offers a lightweight and flexible framework for Python developers, allowing customization and integration of preferred components and extensions.
- Web2Py provides simplicity, security, and scalability, with features like automatic database administration and a web-based integrated development environment (IDE).
- FuelPHP emphasizes modularity and flexibility, enabling developers to build applications with reusable components and features like ORM, caching, and security.

Each framework has its own strengths and target use cases, catering to the specific needs and preferences of developers. Exploring these frameworks can help developers choose the most suitable toolset for their web development projects.

Question-2:

AIM

Develop a data driven web application using SQLite database.

DESCRIPTION

To develop a data-driven web application using a SQLite database with Django:
1. Install Django and create a new project and app.
2. Define models in your app's models.py file to represent the database structure.
3. Run database migrations to create the necessary tables.
4. Create views that interact with the SQLite database to handle requests and generate responses.
5. Configure URLs and create templates to map URLs to views and render data-driven web pages.

PROGRAM

crud/crud/settings.py

```
from pathlib import Path
import os

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent
TEMPLATE_DIR = os.path.join(BASE_DIR, 'templates')

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'addApp',
]

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [TEMPLATE_DIR,],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
```

```
            ],
        },
    },
]
```

```
from django.contrib import admin
from django.urls import path, include
```

crud/crud/urls.py

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('crudApp.urls'))
]
```

crud/crudApp/admin.py

```
from django.contrib import admin
from crudApp.models import Student, Branch

# Register your models here.
admin.site.register(Student)
admin.site.register(Branch)
```

crud/crudApp/forms.py

```
from django import forms
from .models import Student


class StudentForm(forms.ModelForm):
    class Meta:
        model = Student
        fields = ['name', 'age', 'address']
```

crud/crudApp/models.py

```
from django.db import models

# Create your models here.


class Student(models.Model):
    name = models.CharField(max_length=50)
    age = models.IntegerField()
    address = models.CharField(null=True, max_length=100)
```

```
    def __str__(self):
        return self.name


class Branch(models.Model):
    pass
```

crud/crudApp/urls.py

```
from django.urls import path
from . import views


urlpatterns = [

    path('insert/', views.insert, name='insert'),
    path('', views.show, name='show'),
    path('show', views.show, name='show'),
    path('update/<int:id>', views.update, name="update"),
    path('delete/<int:id>', views.delete, name='delete'),
]
```

crud/crudApp/views.py

```
from django.shortcuts import render
from django.http import HttpResponse, HttpResponseRedirect
from .models import Student
from .forms import StudentForm


# Create your views here.


def insert(request):
    form = StudentForm(request.POST or None)
    if form.is_valid():
        form.save()
        return HttpResponseRedirect('/show')
    return render(request, 'insert.html', {'form': form})


def show(request):
    stu = Student.objects.all()
    return render(request, 'show.html', {'stu': stu})
```

```
def delete(request):
    pass


def update(request, id):
    stu = Student.objects.get(id=id)
    form = StudentForm(request.POST, instance=stu)
    if (form.is_valid()):
        form.save()
        return HttpResponseRedirect('/show')
    return render(request, 'update.html', {'student': stu, 'form': form})


def delete(request, id):
    stu = Student.objects.get(id=id)
    stu.delete()
    return HttpResponseRedirect('/show')
```

crud/templates/base.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body style="background-color:ivory">
    <h3>CRUD operations</h3>
    {%block content%}
    {%endblock %}

</body>
</html>
```

crud/templates/insert.html

```
<!DOCTYPE html>
{% extends 'base.html' %}

{%block content%}
<form method="POST" action="{% url 'insert' %}">
    {% csrf_token %}
```

```
    name:<input type="text" name="name" value="{{student.name}}" /><br><br>
    age:<input type="text" name="age" value="{{student.age}}" /><br><br>
    address:<input type="text" name="address" value="{{student.address}}" /><br><br>

    <button type="submit">Add Record</button>
    <a href="{% url 'show'%}">Show Details</a>
</form>
{%endblock%}
```

crud/templates/show.html

```
{% extends 'base.html'%}

{%block content%}
<h4>Student Details</h4>
<br>
<a href="insert">Add New</a>

<br><br>
<table border="1">
    <thead>
        <tr>
            <td>id</td>
            <td>name</td>
            <td>age</td>
            <td>address</td>
            <td>action</td>

        </tr>
    </thead>
    <tbody>
        {%for student in stu %}
        <tr>
            <td>{{student.id}}</td>
            <td>{{student.name}}</td>
            <td>{{student.age}}</td>
            <td>{{student.address}}</td>
            <td><a href="update/{{student.id}}">Edit</a><br><a
href="delete/{{student.pk}}">Delete</a></td>

        </tr>
        {%endfor%}
    </tbody>

</table>
{%endblock%}
```

crud/templates/update.html

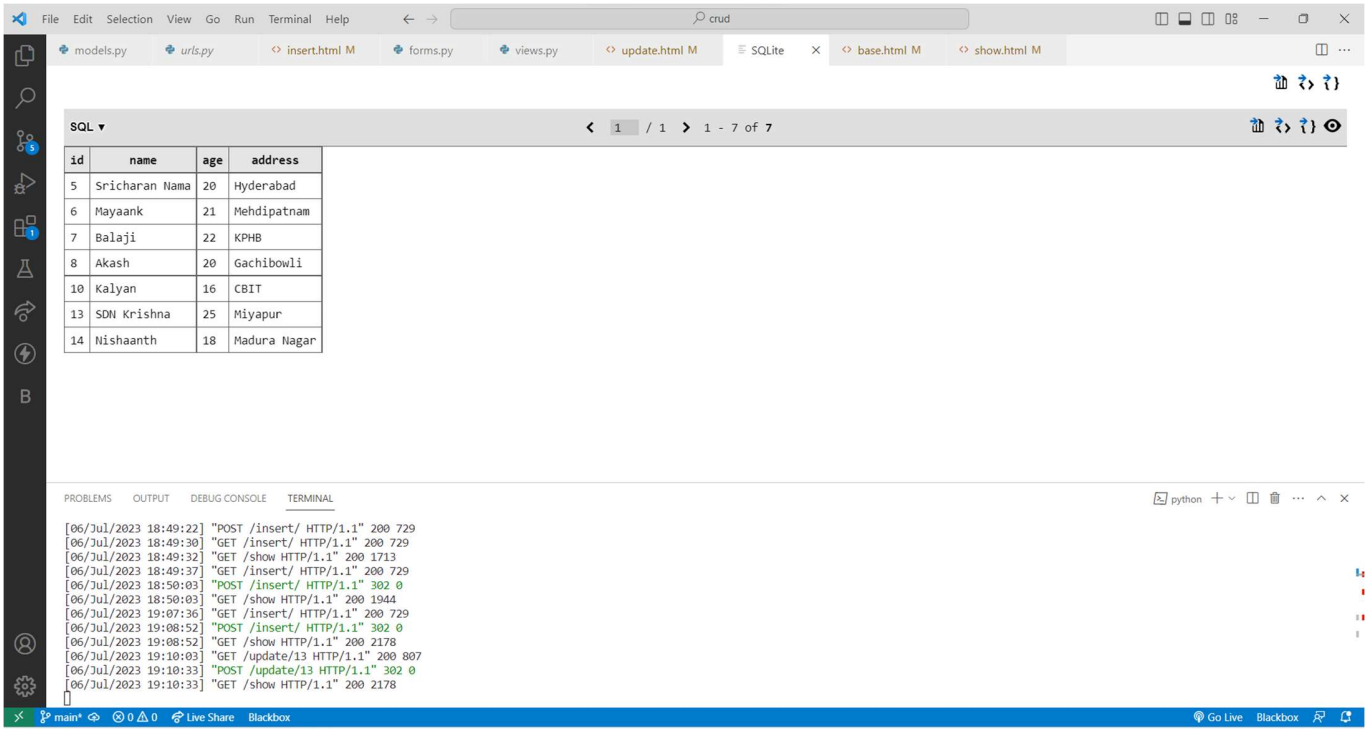
```
{% extends 'base.html'%}

{%block content%}
<h4>Update Record</h4>
<form action="/update/{{student.id}}" method="POST">
   {%csrf_token%}
   id:<input type="text" name="id" value="{{student.id}}" /><br><br>
   name:<input type="text" name="name" value="{{student.name}}" /><br><br>
   age:<input type="text" name="age" value="{{student.age}}" /><br><br>
   address:<input type="text" name="address" value="{{student.address}}" /><br><br>
   <button type="submit">Update</button>
   <a href="{% url 'show'%}">Show Details</a>

</form>
{%endblock%}
```

Run the Development Server

Start the development server by running the command **python manage.py runserver**. Then, access the application in your web browser at http://127.0.0.1:8000/.

OUTPUT

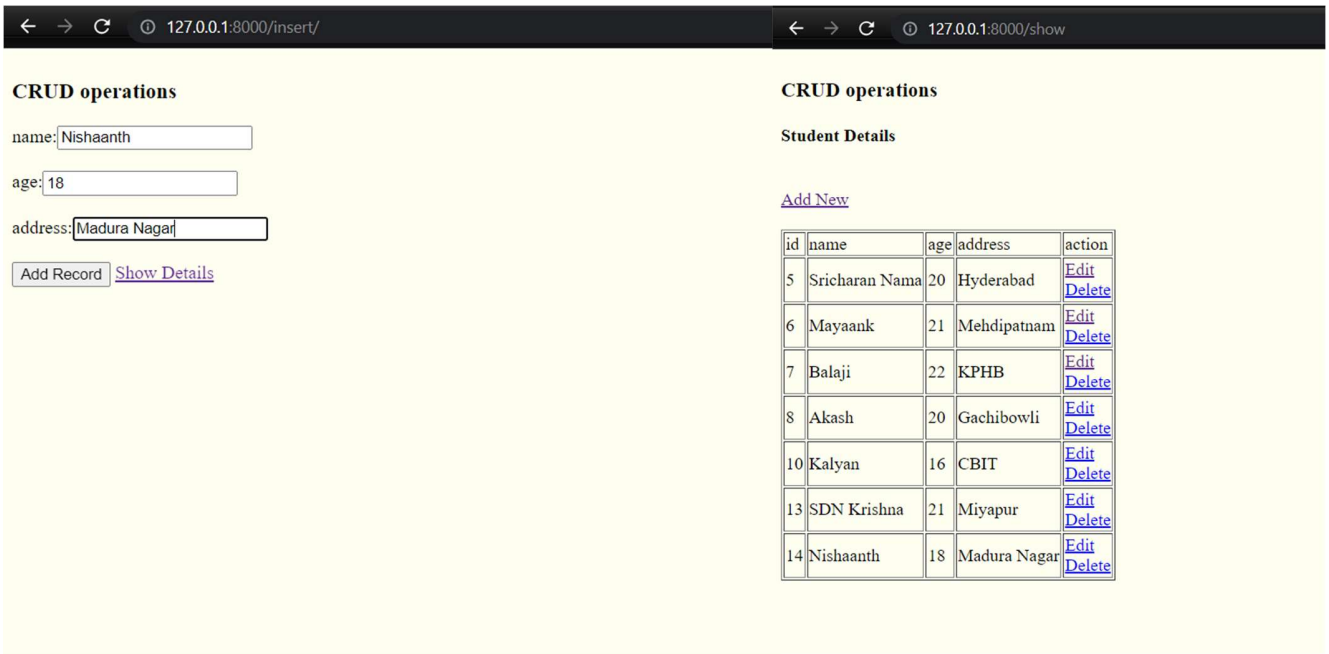
SQLite



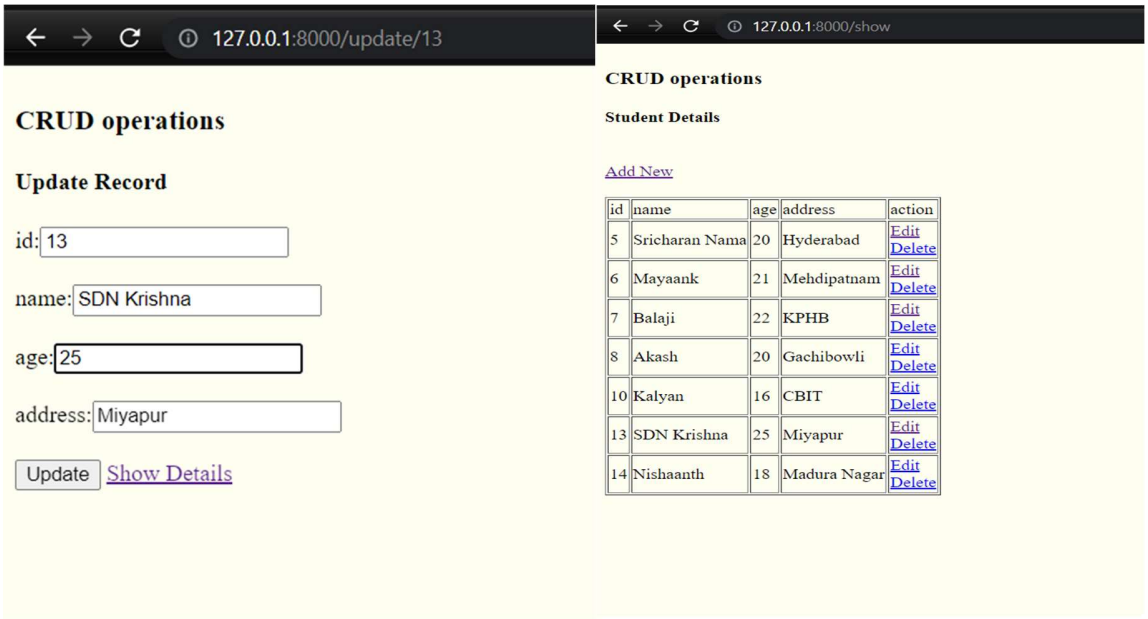
SQL ▾ 1 / 1 1 - 7 of 7

id	name	age	address
5	Sricharan Nama	20	Hyderabad
6	Mayaank	21	Mehdipatnam
7	Balaji	22	KPHB
8	Akash	20	Gachibowli
10	Kalyan	16	CBIT
13	SDN Krishna	25	Miyapur
14	Nishaanth	18	Madura Nagar

Webpage



id	name	age	address	action
5	Sricharan Nama	20	Hyderabad	Edit Delete
6	Mayaank	21	Mehdipatnam	Edit Delete
7	Balaji	22	KPHB	Edit Delete
8	Akash	20	Gachibowli	Edit Delete
10	Kalyan	16	CBIT	Edit Delete
13	SDN Krishna	21	Miyapur	Edit Delete
14	Nishaanth	18	Madura Nagar	Edit Delete



CONCLUSION

In conclusion, developing a data-driven web application using a SQLite database with Django involves installing Django, creating a project and app, defining models, running database migrations, creating views, configuring URLs, and designing templates. This approach allows you to leverage Django's powerful features EXPERIMENT NUMBER- 09

EXPERIMENT NUMBER- 10

Question-1:

AIM

Create a form validation application in Django.

DESCRIPTION

Step 1: Set up the Django Project and App.
Step 2: Define the Form.
Step 3: Create the View.
Step 4: Create Templates.
Step 5: Configure URLs

PROGRAM

validation/validation/settings.py

```
from pathlib import Path
import os

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'validationapp',
]

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR,'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
```

```
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

validation/validation/urls.py

```
from django.contrib import admin
from django.urls import path,include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('',include('validationapp.urls')),
]
```

validation/validationapp/urls.py

```
from django.urls import path
from . import views

urlpatterns = [
    path(“,views.valid),
]
```

validation/validationApp/views.py

```
from django.shortcuts import render
from . import forms
def valid(request):
    form = forms.SignUp()
    if request.method=='POST':
        form=forms.SignUp(request.POST)
        html='We have received this form again'
    else:
        html='welcome for the first time'
    return render(request,'signup.html',{'html':html,'form':form})
```

validation/templates/signup.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Document</title>
```

```
</head>
<body>
   <pre><center>
      <h1>Registration Form</h1>
      <h2>{{html}}</h2>
      <form action="" method="post">
         {% csrf_token %}
         <table>
            {{form.as_table}}
         </table>
         <input type="submit" name="register">
      </form>
   </center>
   </pre>
</body>
</html>
```

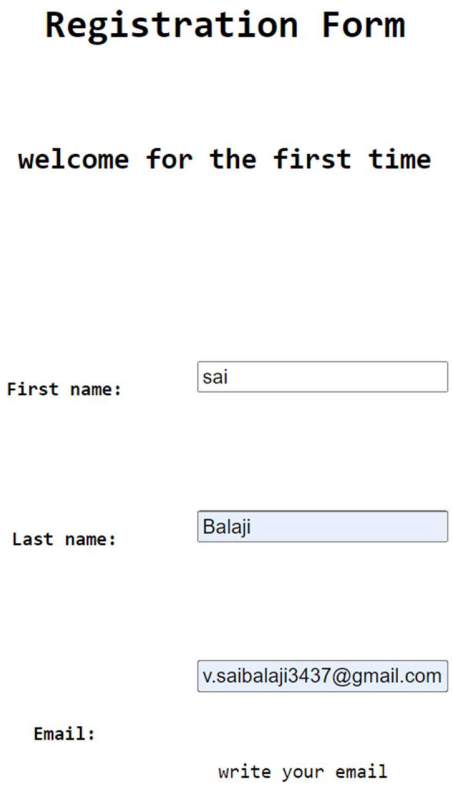
validation/validationapp/forms.py

```
from django import forms


class SignUp(forms.Form):
   first_name=forms.CharField(initial='First Name',)
   last_name=forms.CharField()
   email=forms.EmailField(help_text="write your email",)
   Address=forms.CharField(required=False,)
   Technology=forms.CharField(initial='Django',disabled=True,)
   age=forms.IntegerField()
   password=forms.CharField(widget=forms.PasswordInput)
   re_password=forms.CharField(help_text='enter your password',widget=forms.PasswordInput)


   def clean_password(self):
      password=self.cleaned_data['password']
      if len(password)<4:
         raise forms.ValidationError("Password is too short")
      return password
```

Run the Development Server

Start the development server by running the command **python manage.py runserver**. Then, access the application in your web browser at http://127.0.0.1:8000/

Output



Address: kukatpally

Technology: Django

Age: 19

Password: ••••••••

••••••••

Re password: enter your password

Submit

Registration Form

We have received this form again

First name: sai

Last name: Balaji

v.saibalaji3437@gmail.com

Email: write your email

Address: kukatpally

Conclusion

In conclusion, we have successfully created a simple form validation application in Django. The application allows users to fill out a form with their name, email, and age. When the form is submitted, it validates the input data.

Question-2:

AIM

Create session handling application in Django**.**

DESCRIPTION

Step 1: Set up the Django Project and App.

Step 2: Configure Settings for Sessions.

Step 3: Create the Views.

Step 4: Create Templates.

Step 5: Create and Configure URLs.

Step 6: Run server.

PROGRAM

session/session/settings.py

```
from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent
TEMPLATE_DIR = os.path.join(BASE_DIR, 'templates')

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'sessionsapp',
]

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
```

```
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

session/session/urls.py

```
from django.contrib import admin
from django.urls import path,include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('',include('sessionsapp.urls'))
]
```

session/sessionsapp/urls.py

```
from django.urls import path
from . import views

urlpatterns = [
    path('create/',views.create_session),
    path('access/',views.access_session),
    path('delete/',views.delete_session)

]
```

session/sessionsapp/views.py

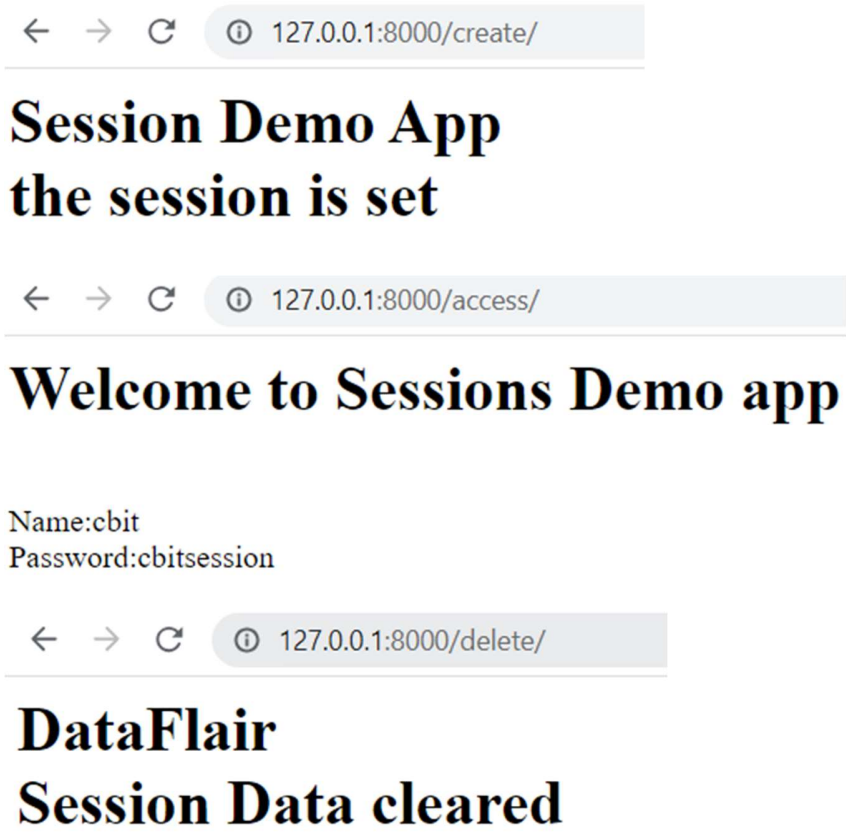
```
from django.shortcuts import render,redirect
from django.http import HttpResponse

def create_session(request):
    request.session['name']='cbit'
    request.session['password']='cbitsession'
    return HttpResponse("<h1>Session Demo App<br> the session is set</h1>")
def access_session(request):
    response="<h1>Welcome to Sessions Demo app</h1><br>"
    if request.session.get('name'):
        response+="Name:{0} <br>".format(request.session.get('name'))
    if request.session.get('password'):
        response+="Password:{0} <br>".format(request.session.get('password'))
```

```
        return HttpResponse(response)
    else:
        return redirect('/create')
def delete_session(request):
    try:
        del request.session['name']
        del request.session['password']
    except KeyError:
        pass
    return HttpResponse("<h1>DataFlair<br>Session Data cleared</h1>")
```

Run the Development Server

Start the development server by running the command **python manage.py runserver**. Then, access the application in your web browser at http://127.0.0.1:8000/

Output



Conclusion

Overall, the session handling application showcases the power and simplicity of Django's built-in session management capabilities. Sessions enable the application to store user-specific data and maintain state throughout a user's interaction with the web application. This functionality is crucial for creating personalized and interactive web experiences.

It is worth noting that while this example demonstrated basic session handling, you can build more complex session-based features by storing and retrieving various data types, managing session expiration, and securing sensitive session data using Django's session-related settings and features.

Question-3

AIM: To create a responsive website using Bootstrap.

DESCRIPTION:

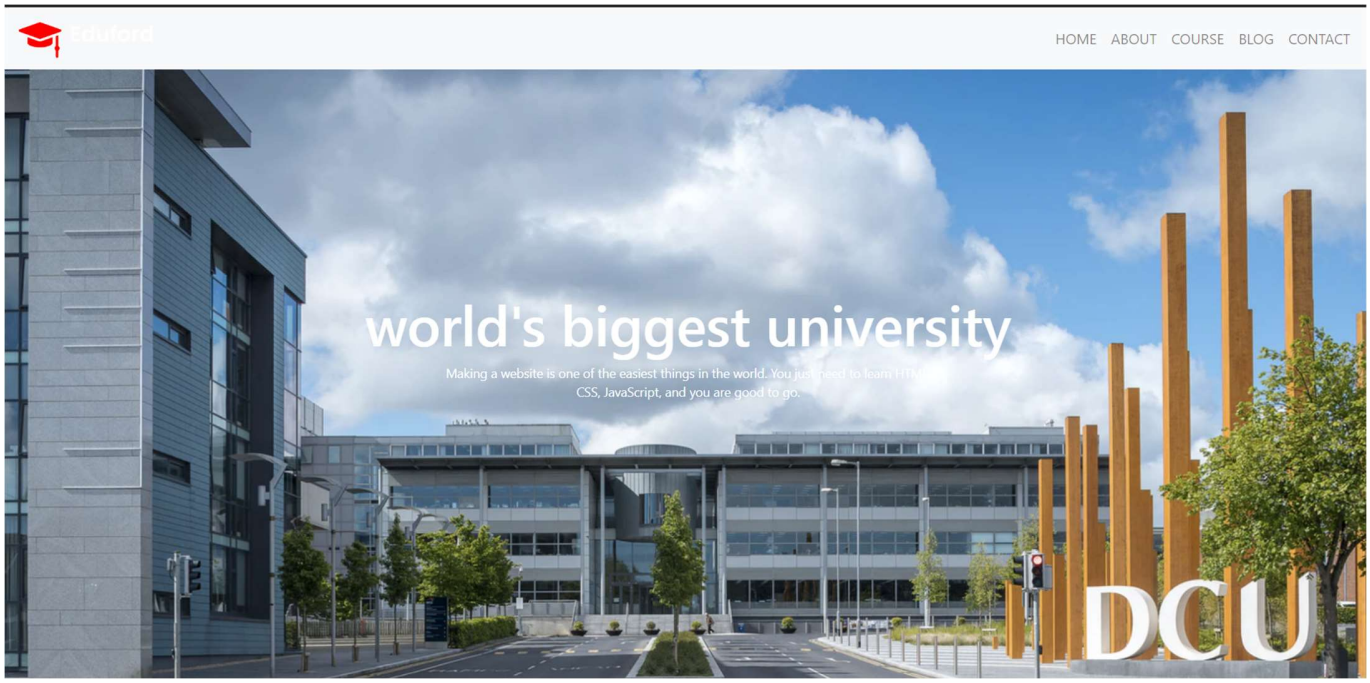
A responsive website built with Bootstrap combines the power of a flexible grid system and a rich set of pre-designed components to create a seamless user experience across devices and screen sizes. With Bootstrap's responsive framework, the website automatically adjusts its layout, content, and functionality to provide optimal viewing and interaction on desktops, tablets, and mobile devices. By utilizing responsive design principles, the website ensures that users can access and navigate the site with ease, regardless of the device they're using. Bootstrap's extensive library of responsive CSS classes and JavaScript plugins simplifies the process of creating a visually appealing and user-friendly website that adapts effortlessly to different screen resolutions.

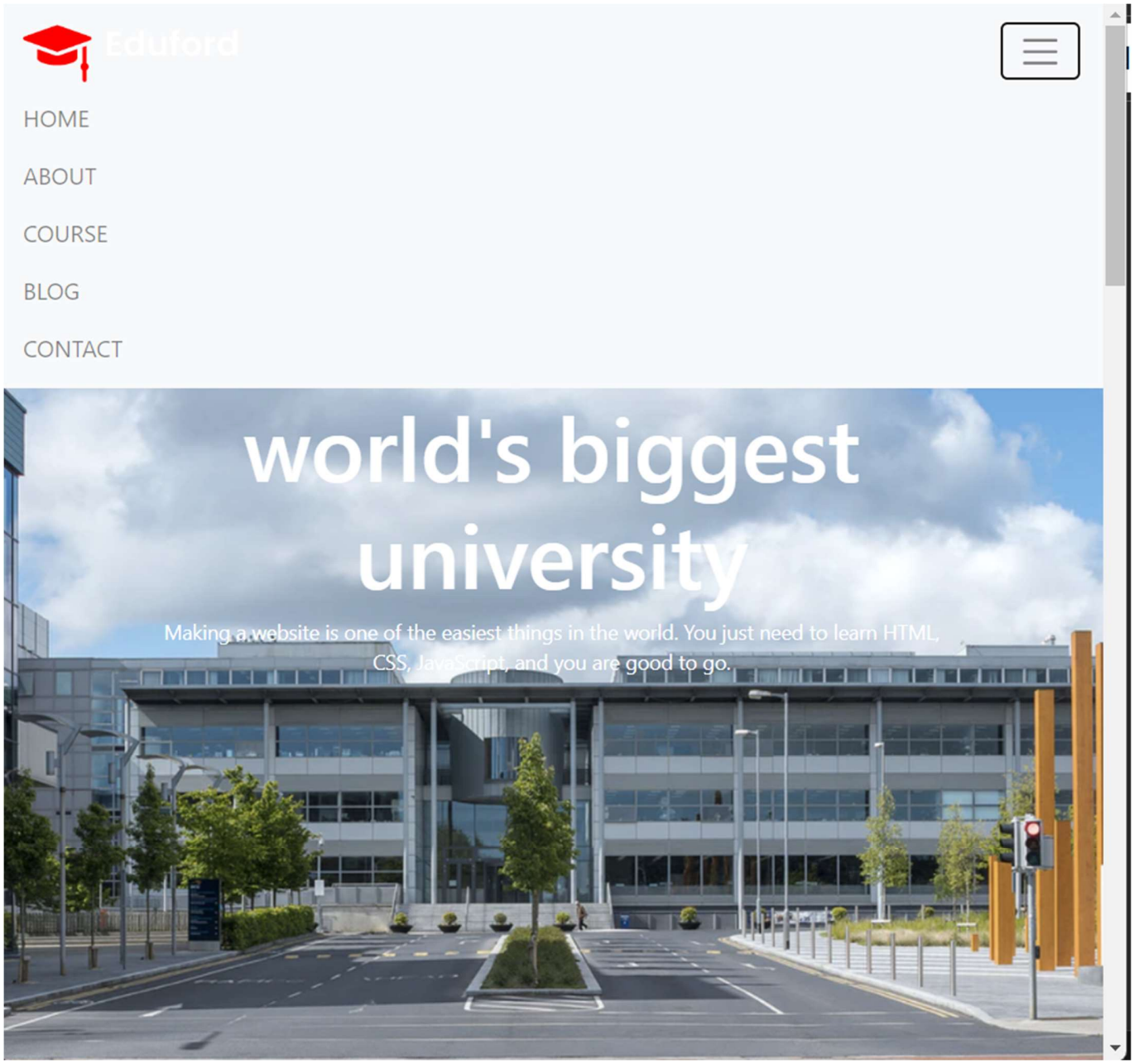
CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>web design</title>
   <!-- Include Bootstrap CSS -->
   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
   <link rel="stylesheet" href="style.css">
</head>
<body>
   <section class="header">
      <nav class="navbar navbar-expand-lg navbar-light bg-light">
         <a class="navbar-brand" href="index.html"><img src="images/logo.png" alt="image"></a>
         <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav"
            aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
         </button>
         <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav ml-auto"> <!-- Add ml-auto to right-align the links -->
               <li class="nav-item"><a class="nav-link" href="#">HOME</a></li>
               <li class="nav-item"><a class="nav-link" href="#">ABOUT</a></li>
               <li class="nav-item"><a class="nav-link" href="#">COURSE</a></li>
               <li class="nav-item"><a class="nav-link" href="#">BLOG</a></li>
               <li class="nav-item"><a class="nav-link" href="#">CONTACT</a></li>
            </ul>
         </div>
      </nav>
```

```
        <div class="text-box">
          <h1>world's biggest university</h1>
          <p>Making a website is one of the easiest things in the world. You just need to learn HTML,<br>
            CSS, JavaScript, and you are good to go.</p>
            <!-- <a href="#" class="hero-btn btn btn-primary btn-lg">Visit us to know more</a> -->
        </div>
    </section>
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>
```

OUTPUT:





CONCLUSION:

In conclusion, Bootstrap is a powerful framework for building responsive websites. Its extensive features and mobile-first approach make it an excellent choice for creating visually appealing and user-friendly web experiences.

Question 4:

AIM: Build and application on Ajax, Node.js and JSON

DESCRIPTION:

Ajax:

Ajax, which stands for Asynchronous JavaScript and XML, is a web development technique that allows for dynamic, asynchronous communication between a web browser and a server without the need for page reloads. It enables seamless, responsive user experiences by sending and receiving data in the background, updating specific parts of a webpage dynamically. Ajax utilizes a combination of JavaScript, XML or JSON, and HTTP requests to achieve this functionality, making it a powerful tool for creating interactive and real-time web applications.

Node.js:

Node.js is an open-source, cross-platform runtime environment for executing JavaScript code outside of a web browser. It is designed to build scalable network applications that can handle a high volume of concurrent connections. Node.js is event-driven and non-blocking, which means that it can handle multiple requests at the same time without having to create a new thread for each request. This makes it very efficient for handling real-time applications such as chat servers, web sockets, and streaming media.

JSON:

JSON, short for JavaScript Object Notation, is a lightweight data interchange format widely used in modern programming. It provides a simple and flexible way to structure and transmit data between different systems. JSON uses a human-readable text format, making it easy for both humans and machines to understand and work with. Its syntax consists of key-value pairs enclosed in curly braces, with values representing strings, numbers, booleans, arrays, or nested objects. JSON's popularity stems from its compatibility with a wide range of programming languages and its support for complex data structures. It is extensively employed in web APIs, configuration files, and data storage applications.

PROGRAM:

Ajax:

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

<div id="demo">
<h1>The XMLHttpRequest Object</h1>
```

```
<button type="button" onclick="loadDoc()">Change Content</button>
</div>
<script>
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML =
      this.responseText;
    }
  };
  xhttp.open("GET", "electronics_ajax.txt", true);
  xhttp.send();
}
</script>

</body>
</html>
```

Node.js and json:

```
const http = require('http');
const hostname = '127.0.0.1';
const port = 3000;
const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/html');
  res.end(`<!DOCTYPE html>
  <html lang="en" dir="ltr">
    <head>
      <meta charset="utf-8">
      <title>PHN_TASK_3</title>
    </head>
    <body>
      <div align='center'  style='margin-left:15%;margin-right:15%;margin-top:15%;margin-
bottom:15%;background-color:#d4cb6c'>
        <table cellspacing='15'>
          <tr>
            <td><a style='text-decoration:none;' href="#" id="1" ></a></td>
            <td><a style='text-decoration:none;' href="#" id="2"></a></td>
            <td><a style='text-decoration:none;' href="#" id="3"></a></td>
            <td><a style='text-decoration:none;' href="#" id="4"></a></td>
          </tr>
        </table>
        <h1>Welcome!</h1>
        <p>Welcome to RRR shopping website.We provide high quality and wide range of products.We have
3000+ products overall,no more delay <br><b>START YOUR SHOPPING!!</b></p>

 <img width='200px' height="200px" src="https://5.imimg.com/data5/OV/KM/CR/ANDROID-
```

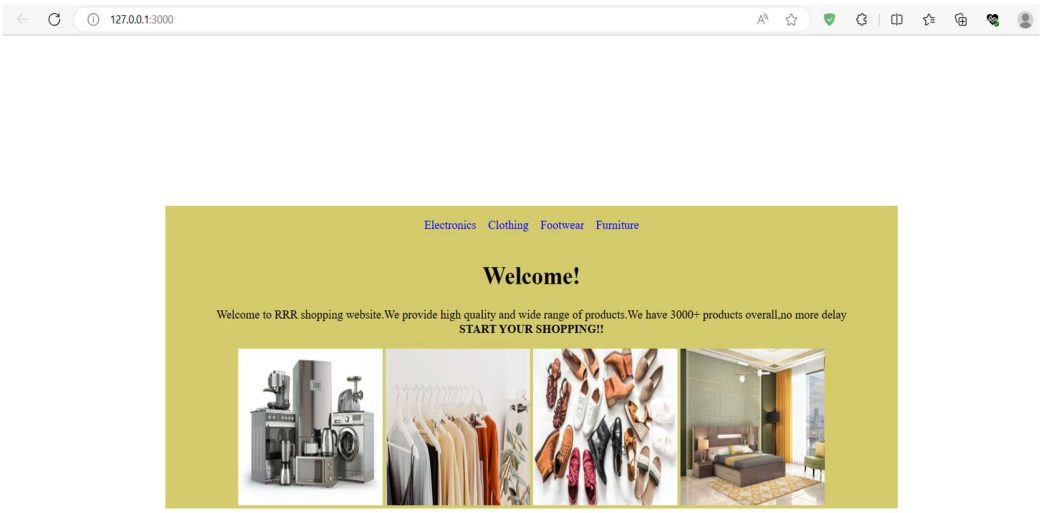
```
37090388/product-jpeg-500x500.jpg">
      <img width='200px' height='200px' src="https://cdn.shopify.com/s/files/1/0070/7032/files/start-a-
clothing-line-retail.jpg?v=1635875038" alt="">
      <img width='200px' height='200px'
src="https://images.hindustantimes.com/img/2022/12/22/550x309/istockphoto-1279108197-
170667a_1671687926903_1671687937504_1671687937504.jpg">
      <img width='200px' height='200px'
src="https://img.staticmb.com/mbcontent//images/uploads/2022/3/home-furniture-ideas.jpg">
    </div>
  </body>
  <script>
    var shop={'elec':'Electronics','cloth':'Clothing','foot':'Footwear','furr':'Furniture'}
    document.getElementById('1').innerHTML=shop.elec
    document.getElementById('2').innerHTML=shop.cloth
    document.getElementById('3').innerHTML=shop.foot
    document.getElementById('4').innerHTML=shop.furr
  </script>
 </html>
 `);
});
server.listen(port, hostname, () => {
 console.log(`Server running at http://${hostname}:${port}/`);
});
```

RESULT/OUTPUT:

The XMLHttpRequest Object

Change Content

Node.js and json:



CONCLUSION:

I executed a basic application on AJAX, Node.js, JSON and understood their respective applications.