Pandit Deendayal Energy University,

Gandhinagar

School of Technology

Department of Computer Science & Engineering

Introduction to Web Technology
(23CP306P)

Name: **Ravi Jamanbhai Makwana**

Enrolment No: **21BCP418**

Semester: **5**

Division: **6 (G12)**

Branch: **Computer Science Engineering**

INDEX

Practical-1

Aim: Design the front pages of a website using HTML and CSS properties

Hardware Requirement: HP Laptop

Software Requirement: Notepad Version 22H2 / VSCode

Knowledge Requirement: Basic understanding of HTML and CSS.

Theory: HTML is the standard markup language for Web pages. CSS is the language we use to style an HTML document. HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) are foundational tools utilized in web development for crafting and designing web pages. Let us delve into a thorough elucidation of HTML and CSS:

HTML (Hypertext Markup Language):

HTML serves as the foundational framework of web pages, dictating their structure and content through a system of tags, elements, and attributes. Each HTML element has a designated function and can encompass a range of content including text, images, links, and more.

Tags and Elements:

HTML relies on a set of tags and elements to structure the content on a web page.

Tags are enclosed in angle brackets (<>) and typically come in pairs, consisting of an opening tag and a closing tag.

Elements consist of a pair of tags along with the content they enclose. For example, <p> serves as an element for paragraphs, containing the actual text of the paragraph.

Inline and Block Elements:

HTML elements are categorized into two primary types: inline and block.

Inline elements, like <a>, do not start on a new line and only take up as much width as necessary. They are commonly used within text.

Block elements, represented by <p> and <div>, begin on a fresh line and occupy the full width of their container. They are employed for structuring the layout of a page.

Links and Images:

<a> is utilized to create hyperlinks, equipped with attributes like href (URL) and target (specifying where the linked content should open).

<img> is employed to showcase images, furnished with attributes such as src (image URL) and alt (alternative text for accessibility).

Lists:

HTML accommodates both unordered lists (<ul>) and ordered lists (<ol>).
List items are designated with <li> tags.

Tables and Definitions:

Tables are established using <table> elements. They may incorporate <caption> for labeling, <thead> for header rows, <th> for header cells, <tbody> for the primary content, and <tfoot> for footer rows.

Definition lists are crafted through <dl>, <dt> for definition titles, and <dd> for definition data.

Frames and iFrames:

HTML4 employed frames and framesets to partition a web page into multiple sections.
HTML5 introduced <iframe> for embedding external content within a page.

CSS (Cascading Style Sheets):

CSS enhances HTML by offering a means to oversee the presentation and arrangement of web pages. This segregation of content from its visual depiction simplifies the task of administering and upholding websites.

Selectors and Properties:

- CSS employs selectors to target HTML elements and implement styles on them.
- Properties are utilized to specify how an element's appearance should be, including aspects like color, font size, margins, and more.

Cascading and Specificity:

- CSS adheres to a cascading sequence, where styles from various origins (e.g., external stylesheets, internal styles, inline styles) can be enforced.
- Specificity dictates which style directives take precedence in situations where multiple rules apply to the same element.

Box Model:

- The box model stands as a fundamental principle in CSS. Each element is treated as a rectangular box encompassing content, padding, borders, and margins.
- CSS offers control over the dimensions and spacing of these constituents.

Layout and Positioning:

- CSS equips developers with tools to manage the arrangement of elements on a page.

Techniques such as floats, flexbox, and grid facilitate the creation of intricate layouts.
- Positioning properties (e.g., position, display, z-index) grant control over element placement.

Responsive Design:

- CSS plays a vital role in crafting responsive web designs that adapt seamlessly to diverse screen sizes and devices.
- Media queries are instrumental in applying styles based on specific screen attributes.

Transitions and Animations:

- CSS empowers the addition of transitions and animations to elements, enriching user interactions and engagement.

External Stylesheets:

- CSS can be integrated into web pages via inline <style> tags, internal styles within the <head> section, or through external stylesheets linked using the <link> element.

By merging HTML and CSS, web developers can craft visually captivating and meticulously structured websites, offering users an exceptional browsing experience. These technologies serve as the bedrock of contemporary web development, enabling the creation of dynamic and interactive web applications.

Code and Output:

1. Basic HTML- Code-

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

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

<body>
   <h1>Hello World</h1>
   <!-- <h2>Hello World</h2>
   <h3>Hello World</h3>
   <h4>Hello World</h4>
   <h5>Hello World</h5>
   <h6>Hello World</h6> -->
   <div class="webtech">
      <b>Paragraphs:</b>
```

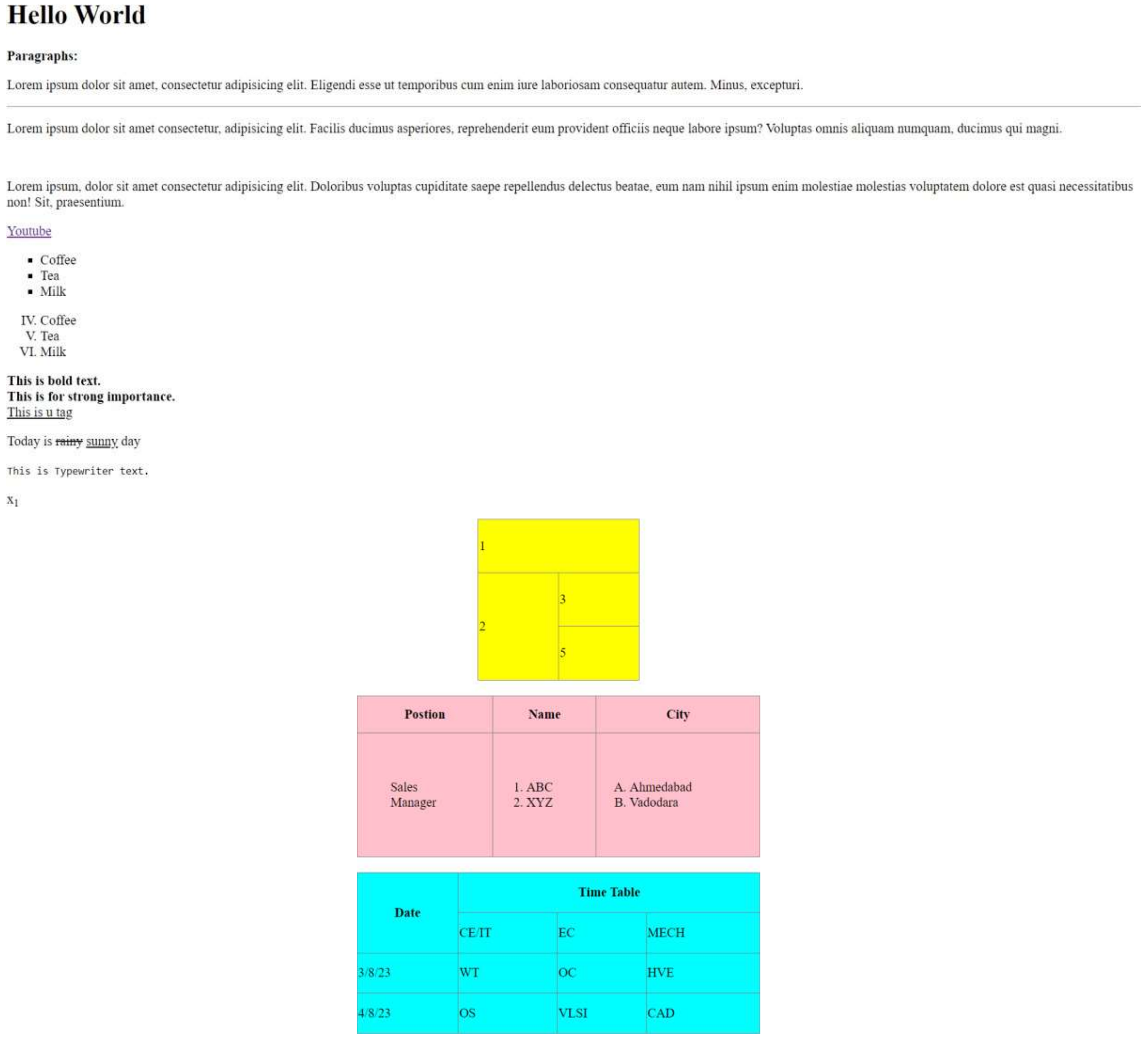
```
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Eligendi esse ut temporibus cum enim iure
    laboriosam consequatur autem. Minus, excepturi.</p>
  <hr>
  <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Facilis ducimus asperiores, reprehenderit eum
    provident officiis neque labore ipsum? Voluptas omnis aliquam numquam, ducimus qui magni.</p>
  <br>
  <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Doloribus voluptas cupiditate saepe repellendus
    delectus beatae, eum nam nihil ipsum enim molestiae molestias voluptatem dolore est quasi necessitatibus
    non! Sit, praesentium. </p>
</div>
<a href="https://www.youtube.com" target="_self">Youtube</a>
<br>
<!-- <a href="https://linkedin.com"><img src="pic_trulli.jpg" alt="Italian Trulli"></a> -->
<ul style="list-style-type: square;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
<ol type="I" start="4">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
<!-- Formatting tags -->
<!-- <b> or <strong> -->
<!-- <i> or <em> -->
<!-- <u>
  <mark>
  <ins>
  <del>
  <tt>
  <sub>
  <sup> -->
<!-- strong and bold are same but we use strong for SEO similar is the case for emphasize and italic -->
<b>This is bold text.</b>
<br>
<strong>This is for strong importance.</strong>
<br><u>This is u tag</u>
<p>Today is <del>rainy</del> <ins>sunny</ins> day</p>
```

```
<tt>This is Typewriter text.</tt>
<p>x<sub>1</sub></p>
<p>x<sup>2</sup></p>

<!-- HTML Table
<table>
<caption>
<thead>
<th>
<td>
<tr>
<tbody>
<tfoot>
th,thead and tfoot give bold text

Attributes of table -
border - compulsory
align
cell padding
cell spacing
bgcolor
width

Attributs of tr-
rowspan
colspan -->

<table border="1" align="center" rules="all" width="200px" height="200px"
  bgcolor="yellow">
    <tr>
        <td colspan="2">1</td>
    </tr>
    <tr>
        <td rowspan="2">2</td>
        <td>3</td>
    </tr>
    <tr>
        <td>5</td>
    </tr>
</table>
<br>
<table border="1" align="center" rules="all" width="500px" height="200px"
  bgcolor="pink">
    <th>Postion</th>
    <th>Name</th>
    <th>City</th>
```

```
        <tr>
            <td>
                <ul type="none">
                    <li>Sales</li>
                    <li>Manager</li>
                </ul>
            </td>
            <td>
                <ol type="1">
                    <li>ABC</li>
                    <li>XYZ</li>
                </ol>
            </td>
            <td>
                <ol type="A">
                    <li>Ahmedabad</li>
                    <li>Vadodara</li>
                </ol>
            </td>
    </table>
    <br>
    <table border="1" align="center" rules="all" width="500px" height="200px"
      bgcolor="aqua">
        <tr>
            <th rowspan="2">Date</th>
            <th colspan="3">Time Table</th>
        </tr>
        <tr>
            <td>CE/IT</td>
            <td>EC</td>
            <td>MECH</td>
        </tr>
        <tr>
            <td>3/8/23</td>
            <td>WT</td>
            <td>OC</td>
            <td>HVE</td>
        </tr>
        <tr>
            <td>4/8/23</td>
            <td>OS</td>
            <td>VLSI</td>
            <td>CAD</td>
        </tr>
    </table>
</body>
```

```
</html>
```

Output-



Hello World

Paragraphs:

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Eligendi esse ut temporibus cum enim iure laboriosam consequatur autem. Minus, excepturi.

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Facilis ducimus asperiores, reprehenderit eum provident officiis neque labore ipsum? Voluptas omnis aliquam numquam, ducimus qui magni.

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Doloribus voluptas cupiditate saepe repellendus delectus beatae, eum nam nihil ipsum enim molestiae molestias voluptatem dolore est quasi necessitatibus non! Sit, praesentium.

Youtube

- Coffee
- Tea
- Milk

IV. Coffee
V. Tea
VI. Milk

This is bold text.
This is for strong importance.
This is u tag

Today is ~~rainy~~ sunny day

`This is Typewriter text.`

x_1

1	
2	3
	5

Postion	Name	City
Sales Manager	1. ABC 2. XYZ	A. Ahmedabad B. Vadodara

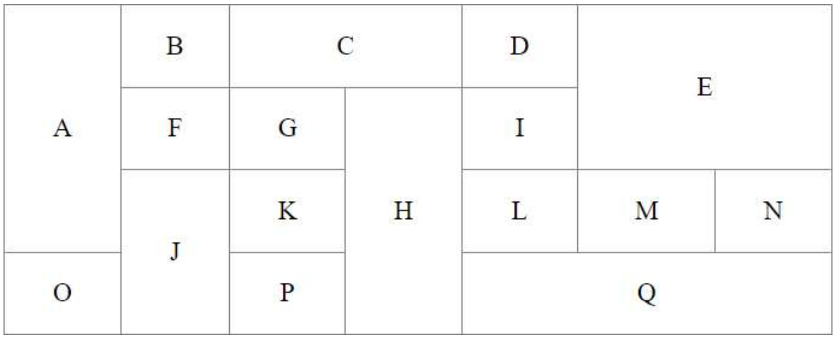
Date	Time Table		
	CE/IT	EC	MECH
3/8/23	WT	OC	HVE
4/8/23	OS	VLSI	CAD

2. Rowspan and colspan-

Code-

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

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Rowspan and Colspan</title>
</head>

<body>
  <table border="1" align="center" rules="all" width="500px" height="200px">
    <tr>
      <td rowspan="3" align="center">A</td>
      <td align="center">B</td>
      <td colspan="2" align="center">C</td>
      <td align="center">D</td>
      <td rowspan="2" colspan="2" align="center">E</td>
    </tr>
    <tr>
      <td align="center">F</td>
      <td align="center">G</td>
      <td rowspan="3" align="center">H</td>
      <td align="center">I</td>
    </tr>
    <tr>
      <td rowspan="2" align="center">J</td>
      <td align="center">K</td>
      <td align="center">L</td>
      <td align="center">M</td>
      <td align="center">N</td>
    </tr>
    <tr>
      <td align="center">O</td>
      <td align="center">P</td>
      <td colspan="3" align="center">Q</td>
    </tr>
  </table>
</body>

</html>
```

Output-



<table>
<tr><td rowspan="3">A</td><td>B</td><td colspan="2">C</td><td>D</td><td rowspan="2" colspan="2">E</td></tr>
<tr><td>F</td><td>G</td><td rowspan="3">H</td><td>I</td></tr>
<tr><td rowspan="2">J</td><td>K</td><td>L</td><td>M</td><td>N</td></tr>
<tr><td>O</td><td>P</td><td colspan="3">Q</td></tr>
</table>

3. Complex Table

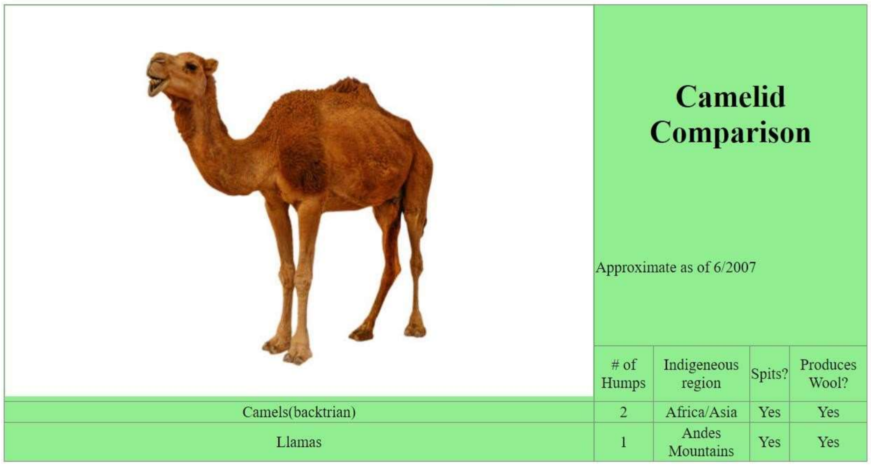
a)

Code-

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

<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Complex Table</title>
</head>

<body>
    <table border="1" align="center" rules="all" width="900px" height="200px"
bgcolor="#90EE90">
        <tr>
            <td rowspan="2"><img src="./camel.jpg" alt="camel.jpg"></td>
            <td colspan="4">
                <h1 align="center">Camelid Comparison</h1>
                <br>
                <br>
                <br>
                <br>
                <p>Approximate as of 6/2007</p>
            </td>
        </tr>
        <tr>
            <td align="center"># of Humps</td>
            <td align="center">Indigeneous region</td>
            <td align="center">Spits?</td>
            <td align="center">Produces Wool?</td>
        </tr>
        <tr>
            <td align="center">Camels(backtrian)</td>
            <td align="center">2</td>
            <td align="center">Africa/Asia</td>
            <td align="center">Yes</td>
            <td align="center">Yes</td>
        </tr>
        <tr>
            <td align="center">Llamas</td>
            <td align="center">1</td>
            <td align="center">Andes Mountains</td>
            <td align="center">Yes</td>
            <td align="center">Yes</td>
        </tr>
    </table>
</body>

</html>
```

Output-



Camelid Comparison

Approximate as of 6/2007

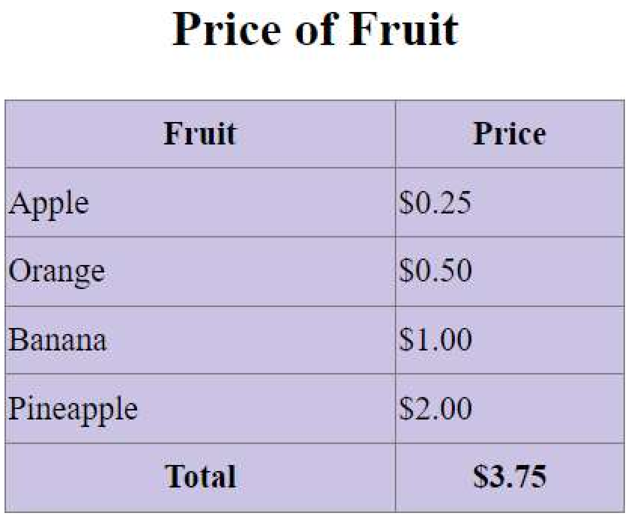
	# of Humps	Indigeneous region	Spits?	Produces Wool?
Camels(backtrian)	2	Africa/Asia	Yes	Yes
Llamas	1	Andes Mountains	Yes	Yes

b)
Code-

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

<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Complex Table 2</title>
</head>

<body>
   <h2 align="center">Price of Fruit</h2>
   <table border="1" align="center" rules="all" width="300px" height="200px"
bgcolor="#CBC3E3">
      <th>Fruit</th>
      <th>Price</th>
      <tr>
         <td>Apple</td>
         <td>$0.25</td>
      </tr>
      <tr>
         <td>Orange</td>
         <td>$0.50</td>
      </tr>
      <tr>
         <td>Banana</td>
         <td>$1.00</td>
      </tr>
      <tr>
         <td>Pineapple</td>
         <td>$2.00</td>
      </tr>
      <tr>
```

```
            <th align="center">Total</th>
            <th align="center">$3.75</th>
        </tr>
    </table>
</body>

</html>
```

Output-



Price of Fruit

Fruit	Price
Apple	$0.25
Orange	$0.50
Banana	$1.00
Pineapple	$2.00
Total	**$3.75**

4. IFrames-

Code-

radiant_holiday.html-

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Radiant Holiday</title>
</head>
<frameset rows="20%,10%,*,10%" />
<frame src="./hello.html" frameborder="0">
    <frame src="./nav.html" frameborder="-1">
        <frameset cols="25%,*,25%">
            <frame src="./list.html">
                <frame src="./text.html">
                    <frame src="./form.html">
        </frameset>
        <frame src="./footer.html" frameborder="0">
            </frameset>
```

```
</html>
```

hello.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 bgcolor="grey">
    <center>
        <h1>Radiant Holidays</h1>
    </center>
</body>
</html>
```

nav.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 bgcolor="magenta">
    <table width="100%">
        <thead width="100%">
            <th><a href="">Home</a></th>
            <th><a href="">About Us</a></th>
            <th><a href="">Gallery</a></th>
            <th><a href="">FAQ</a></th>
            <th><a href="">Contact Us</a></th>
        </thead>
    </table>
</body>

</html>
```

list.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 bgcolor="lightblue">
    <ul type="none">
        <li><a href="">Book Hotel</a></li>
        <li><a href="">Book Resort</a></li>
        <li><a href="">Holiday package</a></li>
        <li><a href="">Manange food</a></li>
    </ul>
</body>

</html>
```

text.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 bgcolor="pink">
    <div>Radiant Holidays is a captivating and enchanting experience that invites
individuals <br>
        to immerse themselves in a world of festive wonder and joy. It is a time when cities
and homes<br>
        come alive with shimmering lights, vibrant decorations, and a palpable sense of
warmth and togetherness. <br>
        During Radiant Holidays, families and friends gather to celebrate cherished
traditions, exchange heartfelt <br>
        gifts, and create lasting memories. Streets are adorned with intricately designed
displays, evoking a sense <br>
        of magic and delight that transcends age and background. Whether it's the sparkle of
twinkling lights,<br>
        the aroma of freshly baked treats, or the resonating laughter of children, Radiant
Holidays encapsulates the
        <br>
        essence of the holiday spirit. It's a season that encourages reflection, gratitude, and a
spirit of giving,
        reminding us all <br>
        of the beauty that can be found in the simple moments of connection and celebration.
    </div>
</body>
</html>
```

form.html-

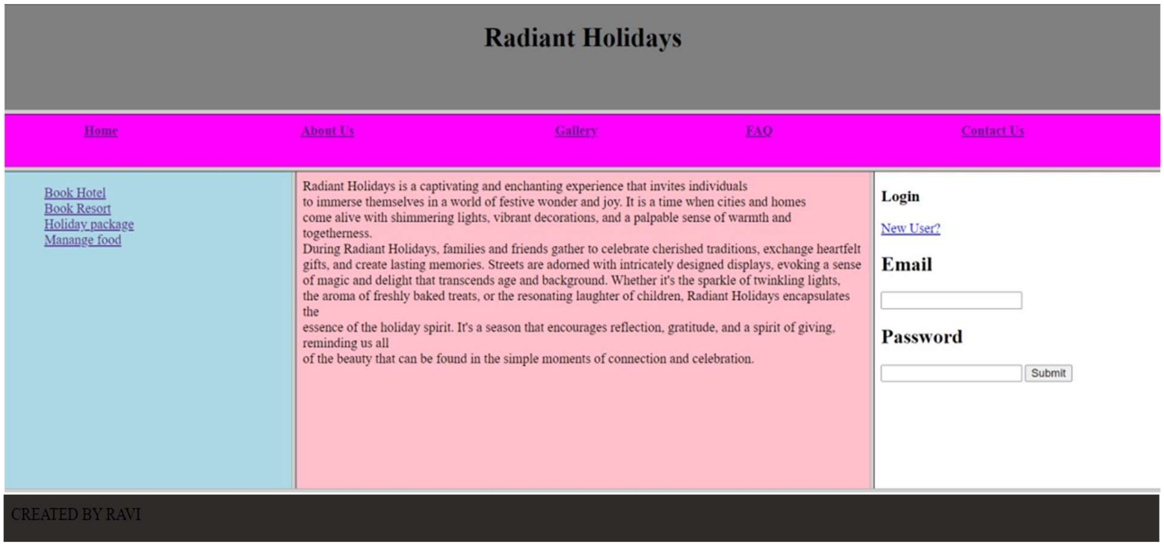
```
<!DOCTYPE html>
```

```
<html lang="en">

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

<body>
    <h3>Login</h3>
    <a href="#">New User?</a>
    <h2>Email</h2>
    <input type="textbox">
    <h2>Password</h2>
    <input type="textbox">
    <button>Submit</button>
</body>
</html>
```

footer.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: rgb(46, 43, 43);">
    <div>CREATED BY RAVI</div>
</body>

</html>
```

Output-



Radiant Holidays

Radiant Holidays is a captivating and enchanting experience that invites individuals
to immerse themselves in a world of festive wonder and joy. It is a time when cities and homes
come alive with shimmering lights, vibrant decorations, and a palpable sense of warmth and
togetherness.
During Radiant Holidays, families and friends gather to celebrate cherished traditions, exchange heartfelt
gifts, and create lasting memories. Streets are adorned with intricately designed displays, evoking a sense
of magic and delight that transcends age and background. Whether it's the sparkle of twinkling lights,
the aroma of freshly baked treats, or the resonating laughter of children, Radiant Holidays encapsulates
the
essence of the holiday spirit. It's a season that encourages reflection, gratitude, and a spirit of giving,
reminding us all
of the beauty that can be found in the simple moments of connection and celebration.

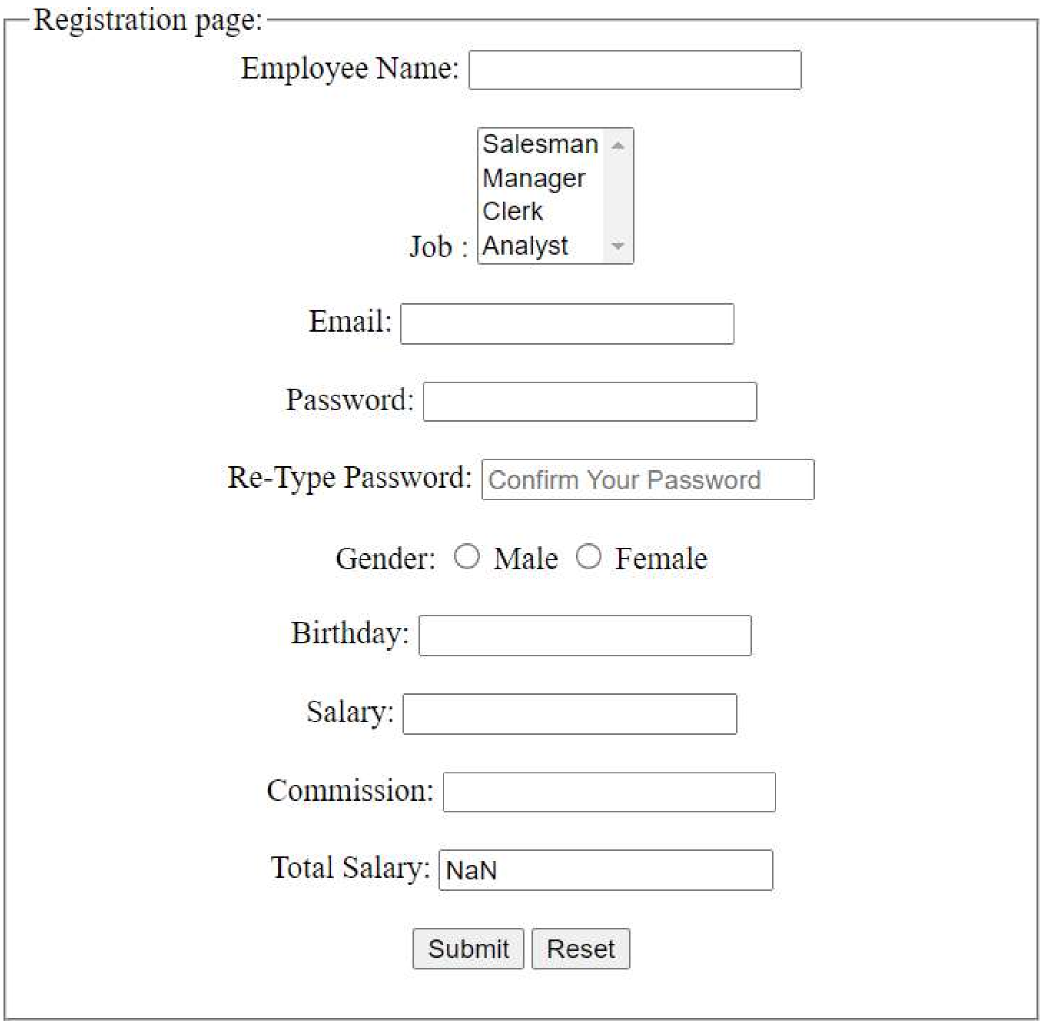
CREATED BY RAVI

5. HTML form-

Code-

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Registration Form</title>
</head>

<body>
    <div align="center">
        <form action="">
            <fieldset style="width:500px; height:500px">
                <legend>Registration page:</legend>
                <label for="fname">Employee Name:</label>
                <input type="text" name="fname" id="fname"><br><br>


                <label for="job">Job :</label>
                <select size="4">
                    <option value="Salesman">Salesman</option>
                    <option value="Manager">Manager</option>
                    <option value="Clerk">Clerk</option>
                    <option value="Analyst">Analyst</option>
                </select>

                <br><br>

                <label for="email">Email:</label>
                <input type="email" id="email" name="email" /> <br>
                <br>
                <label for="password">Password:</label>
                <input type="Password" id="pass" name="pass"> <br>
                <br>
                <label for="re-pass">Re-Type Password:</label>
                <input type="Password" id="repass" name="repass" placeholder="Confirm
           Your Password"> <br><br>

                <label for="gender">Gender:</label>
                <input type="radio" name="gender"> Male
                <input type="radio" name="gender"> Female
                <br><br>

                <label for="birthday">Birthday:</label>
                <input type="Birthday" id="birth" name="birth"> <br>
                <br>

                <label for="salary">Salary:</label>
                <input type="salary" id="sal" name="sal"> <br>
                <br>

                <label for="Commission">Commission:</label>
                <input type="Password" id="pass" name="pass"> <br>
                <br>

                <label for="Total Salary">Total Salary:</label>
                <input type="total" id="total" name="total" value="NaN"> <br>
                <br>
                <input type="submit" value="Submit" />
                <input type="reset" value="Reset" />
            </fieldset>
        </form>
    </div>
</body>

</html>
```

Output-



Registration page:

Employee Name:

Job : Salesman
Manager
Clerk
Analyst

Email:

Password:

Re-Type Password: Confirm Your Password

Gender: ○ Male ○ Female

Birthday:

Salary:

Commission:

Total Salary: NaN

6. **Questions-**

a. **Write HTML and CSS script to display two tables of 50%, 50% width having 6 columns and 6 rows.**

- **Add random data to table.**
- **These two tables must be horizontally adjacent to each other.**
- **Make responsive web page. Initially first table should have light blue color and second table should have light green color.**
- **When page comes to portrait mode, colors of tables should be reversed.**

Ans.
CODE-

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Responsive Tables</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
        }

        .container {
            display: flex;
        }

        .table-container {
            width: 50%;
            box-sizing: border-box;
            padding: 20px;
        }

        table {
            width: 100%;
            border-collapse: collapse;
        }

        th,
        td {
            padding: 8px;
            text-align: left;
            border-bottom: 1px solid #ddd;
        }

        th {
            background-color: #f2f2f2;
        }

        .table-1 {
            background-color: lightblue;
            margin-top: 100px;
        }

        .table-2 {
            background-color: lightgreen;
            margin-top: 100px;
        }
        @media screen and (max-width: 900px) {
            .container {
                flex-direction: column;
            }

            .table-container {
                width: 100%;
            }

            .table-1 {
                background-color: lightgreen;
            }

            .table-2 {
                background-color: lightblue;
            }
        }
    </style>
</head>

<body>
    <div class="container">
        <div class="table-container table-1">
            <h2>Table 1</h2>
            <table>
                <tr>
                    <th>First Name</th>
                    <th>Middle Name</th>
                    <th>Last Name</th>
                    <th>Roll No.</th>
                    <th>Branch</th>
                    <th>Passing Year</th>
                </tr>
                <tr>
                    <td>John</td>
                    <td>Doe</td>
                    <td>Smith</td>
                    <td>12345</td>
                    <td>Computer Science</td>
                    <td>2023</td>
                </tr>
                <tr>
                    <td>Jane</td>
                    <td>Lee</td>
                    <td>Brown</td>
                    <td>54321</td>
                    <td>Electrical Engineering</td>
                    <td>2022</td>
                </tr>
                <tr>
```

```html
            <td>Michael</td>
            <td>James</td>
            <td>Johnson</td>
            <td>98765</td>
            <td>Mechanical Engineering</td>
            <td>2024</td>
        </tr>
        <tr>
            <td>Susan</td>
            <td>Mary</td>
            <td>Clark</td>
            <td>67890</td>
            <td>Chemical Engineering</td>
            <td>2023</td>
        </tr>
        <tr>
            <td>David</td>
            <td>Paul</td>
            <td>Anderson</td>
            <td>45678</td>
            <td>Civil Engineering</td>
            <td>2022</td>
        </tr>
        <tr>
            <td>Emily</td>
            <td>Rose</td>
            <td>Taylor</td>
            <td>23456</td>
            <td>Information Technology</td>
            <td>2024</td>
        </tr>

    </table>
</div>
<div class="table-container table-2">
    <h2>Table 2</h2>
    <table>
        <tr>
            <th>Name</th>
            <th>Age</th>
            <th>City</th>
            <th>Country</th>
            <th>Occupation</th>
            <th>Salary</th>
        </tr>
        <tr>
            <td>John Doe</td>
            <td>32</td>
            <td>New York</td>
            <td>USA</td>
```

```
          <td>Engineer</td>
          <td>75,000</td>
        </tr>
        <tr>
          <td>Jane Smith</td>
          <td>28</td>
          <td>London</td>
          <td>UK</td>
          <td>Designer</td>
          <td>50,000</td>
        </tr>
        <tr>
          <td>Michael Johnson</td>
          <td>35</td>
          <td>Toronto</td>
          <td>Canada</td>
          <td>Doctor</td>
          <td>90,000</td>
        </tr>
        <tr>
          <td>Susan Davis</td>
          <td>30</td>
          <td>Sydney</td>
          <td>Australia</td>
          <td>Teacher</td>
          <td>60,000</td>
        </tr>
        <tr>
          <td>David Brown</td>
          <td>29</td>
          <td>Berlin</td>
          <td>Germany</td>
          <td>Software Developer</td>
          <td>65,000</td>
        </tr>
        <tr>
          <td>Emily Taylor</td>
          <td>31</td>
          <td>Tokyo</td>
          <td>Japan</td>
          <td>Marketing Manager</td>
          <td>70,000</td>
        </tr>
      </table>

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

Output-

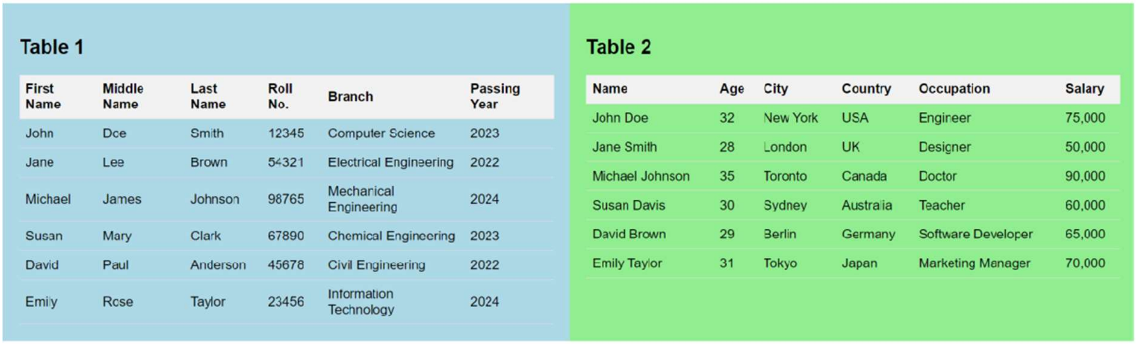


Table 1

First Name	Middle Name	Last Name	Roll No.	Branch	Passing Year
John	Dce	Smith	12345	Computer Science	2023
Jane	Lee	Brown	54321	Electrical Engineering	2022
Michael	James	Johnson	98765	Mechanical Engineering	2024
Susan	Mary	Clark	67890	Chemical Engineering	2023
David	Paul	Anderson	45678	Civil Engineering	2022
Emily	Rcse	Taylor	23456	Information Technology	2024

Table 2

Name	Age	City	Country	Occupation	Salary
John Doe	32	New York	USA	Engineer	75,000
Jane Smith	28	London	UK	Designer	50,000
Michael Johnson	35	Toronto	Canada	Doctor	90,000
Susan Davis	30	Sydney	Australia	Teacher	60,000
David Brown	29	Berlin	Germany	Software Developer	65,000
Emily Taylor	31	Tokyo	Japan	Marketing Menager	70,000

---

b. Write HTML and CSS script to design one login form having fields like username, password, submit button and reset button.

- **While hovering each control, it should be expanded around 20% horizontally and vertically with delay of 2 seconds.**
- **Expansion should take place in 4 seconds.**

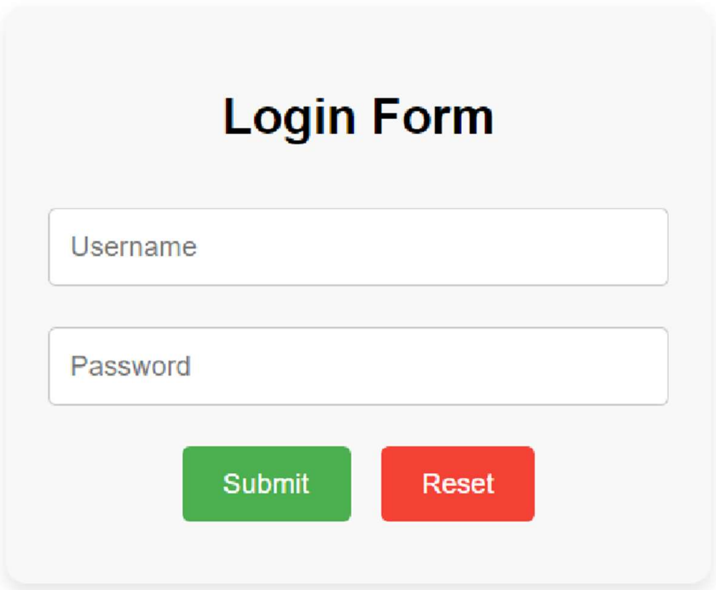
Ans.
Code-

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

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login Form</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      display: flex;
      justify-content: center;
      align-items: center;
      height:  100vh;
      margin: 0;
    }

    .form-container {
      width: 300px;
      background-color: #f8f8f8;
      padding: 20px;
      border-radius: 10px;
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
      text-align: center;
      transition: transform 4s ease;
```

```
        }

        .form-container:hover {
            transform: scale(1.2);
        }

        input[type="text"],
        input[type="password"] {
            width: 100%;
            padding: 10px;
            margin: 10px 0;
            box-sizing: border-box;
            border: 1px solid #ccc;
            border-radius: 4px;
        }

        input[type="submit"],
        input[type="reset"] {
            padding: 10px 20px;
            margin: 10px 5px;
            box-sizing: border-box;
            border: none;
            border-radius: 4px;
            cursor: pointer;
        }

        input[type="submit"] {
            background-color: #4caf50;
            color: #fff;
        }

        input[type="reset"] {
            background-color: #f44336;
            color: #fff;
        }
    </style>
</head>

<body>
    <div class="form-container">
        <h2>Login Form</h2>
        <form>
            <input type="text" name="username" placeholder="Username" required><br>
            <input type="password" name="password" placeholder="Password" required><br>
            <input type="submit" value="Submit">
            <input type="reset" value="Reset">
        </form>
    </div>
</body>
```

```
</html>
```

Output-



---

c. **Write HTML and CSS script to take one square division and it should complete a square cycle by changing its color.**

Ans.
Code-

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Color Changing Square</title>
    <style>
        .square {
            width: 100px;
            height: 100px;
            background-color: red;
            animation: colorCycle 4s infinite;
            margin: 200px;
        }

        @keyframes colorCycle {
            0%,
            100% {
                background-color: red;
            }

            25% {
                background-color: blue;
            }

            50% {
                background-color: green;
            }

            75% {
                background-color: yellow;
            }
        }
    </style>
</head>

<body>
    <div class="square"></div>
</body>

</html>
```

Output-

d. **Write and HTML & CSS script to make a hoverable link button having blue border and white background initially. While hovering, border should disappear and entire button should get blue color with**

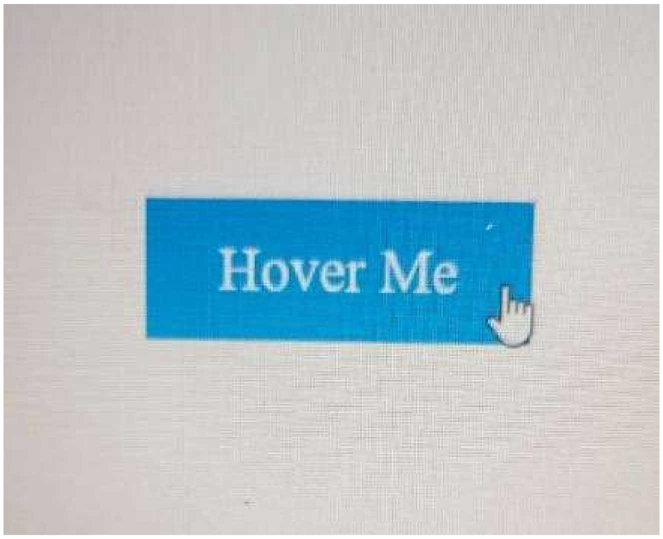
transition and pointer effect. Store blue color in CSS variable, since it is used frequently.

Ans.

Code-

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

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Hoverable Link Button</title>
  <style>
    :root {
      --button-background: #ffffff;
      /* Define blue color as a CSS variable */
    }

    .custom-button {
      display: inline-block;
      padding: 10px 20px;
      text-decoration: none;
      color: var(--button-background);
      background-color: var(--button-background);
      border: 2px solid #3498db;
      transition: background-color 0.3s, border-color 0.3s, color 0.3s;
      cursor: pointer;
      margin: 250px;
    }

    .custom-button:hover {
      background-color: #3498db;
      border: none;
      color: #ffffff;
    }
  </style>
</head>

<body>
  <a href="#" class="custom-button">Hover Me</a>
</body>

</html>
```

Output-



e. **Write and HTML & CSS script to design one banner of advertisement using <p> tag. That text of advertise should move continuously from left end to right end, then return back from right to left end. Use red and huge fonts for text. Advertise should begin after 1 second of loading page.**

Ans.
Code-

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

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Advertisement Banner</title>
  <style>
    .banner {
      font-size: 36px;
      color: red;
      white-space: nowrap;
      position: relative;
      overflow: hidden;
      width: 100%;
      background-color: #fff;
      text-align: center;
      animation: moveText 8s linear infinite;
    }

    @keyframes moveText {
      0% {
        transform: translateX(-100%);
```

```
            }

            50% {
                transform: translateX(0);
            }

            100% {
                transform: translateX(100%);
            }
        }
    </style>
</head>

<body>
    <div class="banner">
        <p>Discount!!!</p>
    </div>
</body>

</html>
```

<u>Output-</u>

Discount!!!

7.

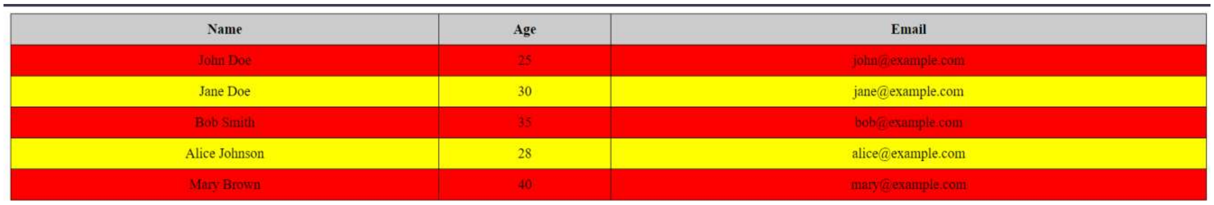
a. Write HTML and CSS script to create a table with 5 rows and 3 columns. Even no. of rows should be displayed in red color and odd no. of rows should be displayed in yellow color. Use external CSS file.

Ans.
Code-

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>7.1</title>
```

```
</head>

<body>
    <table class="table">
        <tr>
            <th>Name</th>
            <th>Age</th>
            <th>Email</th>
        </tr>
        <tr>
            <td>John Doe</td>
            <td>25</td>
            <td>john@example.com</td>
        </tr>
        <tr>
            <td>Jane Doe</td>
            <td>30</td>
            <td>jane@example.com</td>
        </tr>
        <tr>
            <td>Bob Smith</td>
            <td>35</td>
            <td>bob@example.com</td>
        </tr>
        <tr>
            <td>Alice Johnson</td>
            <td>28</td>
            <td>alice@example.com</td>
        </tr>
        <tr>
            <td>Mary Brown</td>
            <td>40</td>
            <td>mary@example.com</td>
        </tr>
    </table>
</body>

</html>
```

Output-



Name	Age	Email
John Doe	25	john@example.com
Jane Doe	30	jane@example.com
Bob Smith	35	bob@example.com
Alice Johnson	28	alice@example.com
Mary Brown	40	mary@example.com

b. Write following style in separate CSS file:

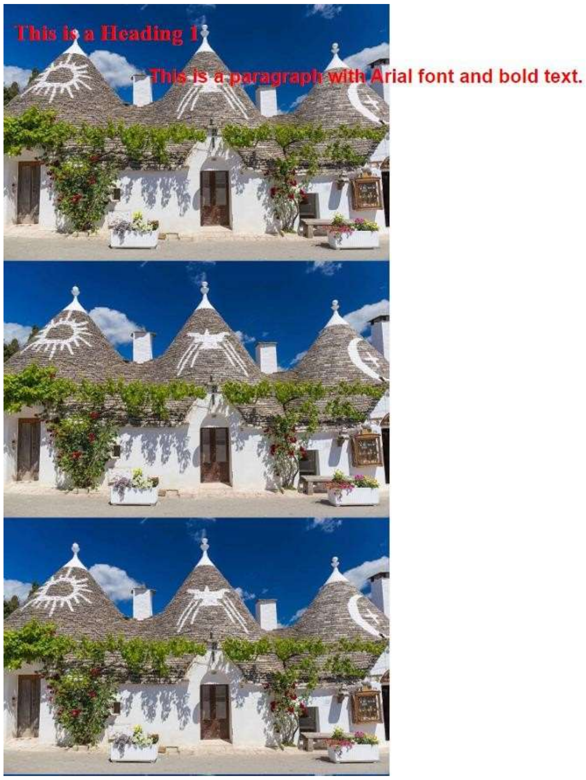
· Heading should have normal font style and size should be 120%.

· Define a class arial for paragraph with arial face and bold text and 3cm spacing before paragraph initialization.
Apply a background image hello.jpg repeating vertically only.

Ans.
Code-

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

<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <link rel="stylesheet" href="style.css">
   <title>Index 2</title>
</head>

<body>
   <h1>This is a Heading 1</h1>
   <p class="arial">This is a paragraph with Arial font and bold text.</p>
</body>

</html>

h1 {
   font-style: normal; color: red;
   font-size: 120%;
   }

.arial {
   font-family: Arial, sans-serif; font-weight: bold;
   color:red; margin-left: 3cm;
}

body {
   background-image: url("./pic_trulli.jpg"); background-repeat: repeat-y; background-size:
auto 250%;
}
```

Output-



This is a Heading 1

This is a paragraph with Arial font and bold text.

c. **Write HTML and CSS snippet for following effects:**

Link: after visited red color and before visited blue color.

Table with odd rows pink and even rows blue color.

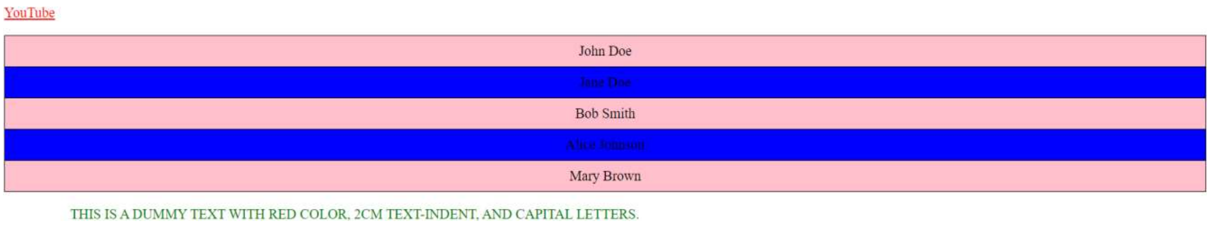
Text color-green, Text-indentation-2cm, Text case-Uppercase

Ans.
Code-

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

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

   <link rel="stylesheet" href="style.css">
   <title>7.3</title>
</head>

<body>
   <p><a href="https://www.youtube.com/" class="YouTube">YouTube</a></p>
```

```
    <table class="table">
        <tr>
            <td>John Doe</td>
        </tr>
        <tr>
            <td>Jane Doe</td>
        </tr>
        <tr>
            <td>Bob Smith</td>
        </tr>
        <tr>
            <td>Alice Johnson</td>
        </tr>
        <tr>
            <td>Mary Brown</td>
        </tr>
    </table>

    <p class="styled-text">This is a dummy text with red color, 2cm text-indent, and capital
letters.</p>
</body>

</html>

.YouTube:link { color: blue;
}

.YouTube:visited { color: red;
}

.table {
border-collapse: collapse; width: 100%;
}

.table td {
border: 1px solid black;



padding: 8px; text-align: center;
}

.table tr:nth-child(odd) { background-color: pink;
}

.table tr:nth-child(even) { background-color: blue;
}

.styled-text { color: green;
text-indent: 2cm;
text-transform: uppercase;
}
```

Output-



YouTube

John Doe

Jane Doe

Bob Smith

Alice Johnson

Mary Brown

THIS IS A DUMMY TEXT WITH RED COLOR, 2CM TEXT-INDENT, AND CAPITAL LETTERS.

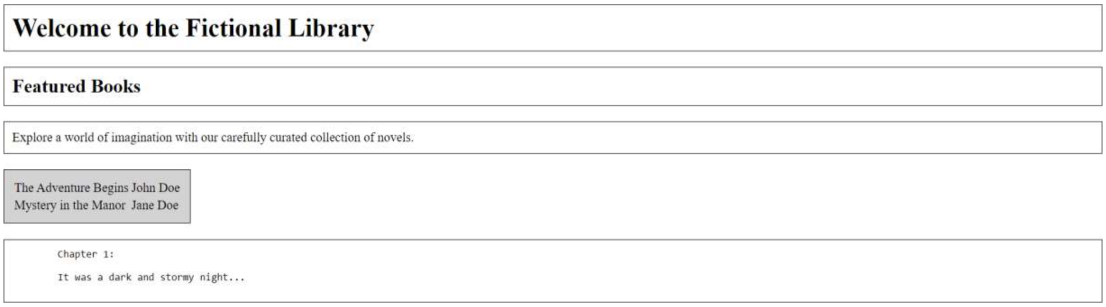
d. Write HTML and CSS script to check all pseudo classes on heading tags, paragraph tags, table tags and pre tag. Comment on which pseudo class works on which tag.

Ans.

Code-

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

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="style.css">
  <title>Index 4</title>
</head>

<body>
  <h1>Welcome to the Fictional Library</h1>
  <h2>Featured Books</h2>
  <p>Explore a world of imagination with our carefully curated collection of novels.</p>
  <table>
    <tr>
      <td>The Adventure Begins</td>
      <td>John Doe</td>
    </tr>
    <tr>
      <td>Mystery in the Manor</td>
      <td>Jane Doe</td>
    </tr>
  </table>
  <pre>
    Chapter 1:
    It was a dark and stormy night...
  </pre>
</body>

</html>

/* Basic styles */
h1, h2, p, table, pre { margin-bottom: 20px; padding: 10px;
  border: 1px solid black;
  }

  /* Pseudo-classes */ h1:hover {
  background-color: yellow;
  }

  h2:hover {
  background-color: blue;
  }

  p:hover {
  font-size: 20px;
  }

  table:hover {
  background-color: lightgray;
  }

  pre::selection {
  background-color: yellow; color: red;
  }
```

Output-



Welcome to the Fictional Library

Featured Books

Explore a world of imagination with our carefully curated collection of novels.

```
The Adventure Begins John Doe
Mystery in the Manor  Jane Doe
```

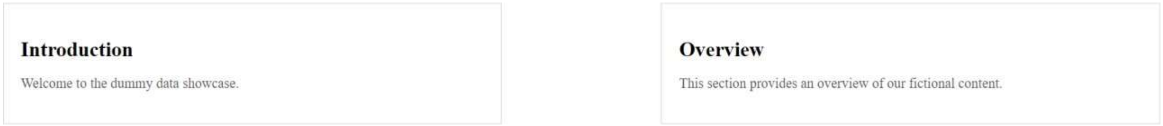
```
        Chapter 1:
        It was a dark and stormy night...
```

e. **Write HTML and CSS script to display two sections of 40% width using <div> tag. Both these sections are having title and description. Both these <div> should be adjacent with each other.**

Ans.
Code-

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

<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <link rel="stylesheet" href="style.css">
   <title>Index 5</title>
</head>

<body>
   <div class="section">
      <h2 class="section-title">Introduction</h2>
      <p class="section-description">Welcome to the dummy data showcase.</p>
   </div>
   <div class="section">
      <h2 class="section-title">Overview</h2>
      <p class="section-description">This section provides an overview of our fictional
content.</p>
   </div>
</body>

</html>

body {
   display: flex;
   justify-content: space-between; padding: 20px;
   }

   .section { width: 40%;
   padding: 20px;



   border: 1px solid #ccc;
   }

   .section-title {
   font-size: 1.5rem; margin-bottom: 10px;
   }

   .section-description { font-size: 1rem; color: #555;
   }
```

Output-



Introduction

Welcome to the dummy data showcase.

Overview

This section provides an overview of our fictional content.

f. **Write an HTML & CSS script to design a signup form having fields like username, password, choices of sports like Cricket, Hockey, Badminton & Tennis. Include gender field, submit and reset buttons. Follow below requirements.**

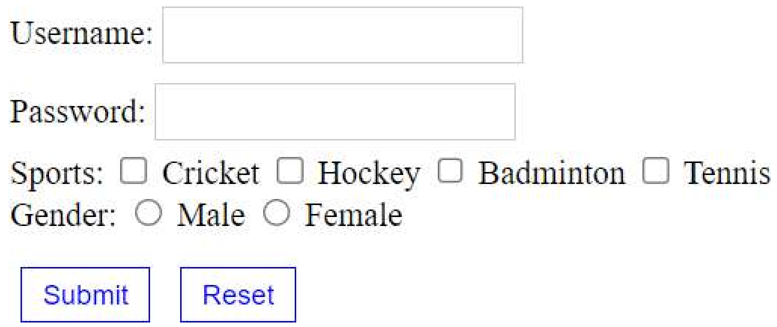
- **While hovering username and password, its background should turn yellow color and pink color respectively.**
- **While clicking sports and gender fields, text should turn to red color.**
- **Both submit and reset buttons' text should turn blue, while hovering.**
- **Use appropriate selectors.**

Ans.
Code-

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

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="style.css">
  <title>7.6</title>
</head>

<body>
  <form class="signup-form">
    <label for="username">Username:</label>
    <input type="text" id="username" class="username" /><br>

    <label for="password">Password:</label>
    <input type="password" id="password" class="password" /><br>

    <label>Sports:</label>
    <input type="checkbox" id="cricket" class="input-field" />
    <label for="cricket">Cricket</label>
    <input type="checkbox" id="hockey" class="input-field" />
    <label for="hockey">Hockey</label>
```

```
        <input type="checkbox" id="badminton" class="input-field" />
        <label for="badminton">Badminton</label>
        <input type="checkbox" id="tennis" class="input-field" />
        <label for="tennis">Tennis</label><br>

        <label>Gender:</label>
        <input type="radio" id="male" name="gender" class="input-field" />
        <label for="male">Male</label>
        <input type="radio" id="female" name="gender" class="input-field" />
        <label for="female">Female</label><br>

        <div class="button-container">
            <input type="submit" value="Submit" class="button" />
            <input type="reset" value="Reset" class="button" />
        </div>
    </form>
</body>

</html>

.username, .password { padding: 5px; margin: 5px 0;
    border: 1px solid #ccc;
    }

    .username:hover { background-color: yellow;
    }

    .password:hover { background-color: pink;
    }

    .input-field:checked + label { color: red;
    }

    .button-container { margin-top: 10px;
    }

    .button {
    background-color: transparent; border: 1px solid blue;



    color: blue; padding: 5px 10px; margin: 5px; cursor: pointer;
    }

    .button:hover { color: white;
    background-color: blue;
    }
```

Output-



Username:

Password:

Sports: □ Cricket □ Hockey □ Badminton □ Tennis
Gender: ○ Male ○ Female

g. **Follow below requirements using HTML & CSS by a separate CSS file.**

   Heading should have 120% font and right aligned.

   In paragraph, the text color should be green, with bold, italic and times new roman font having 50px size.

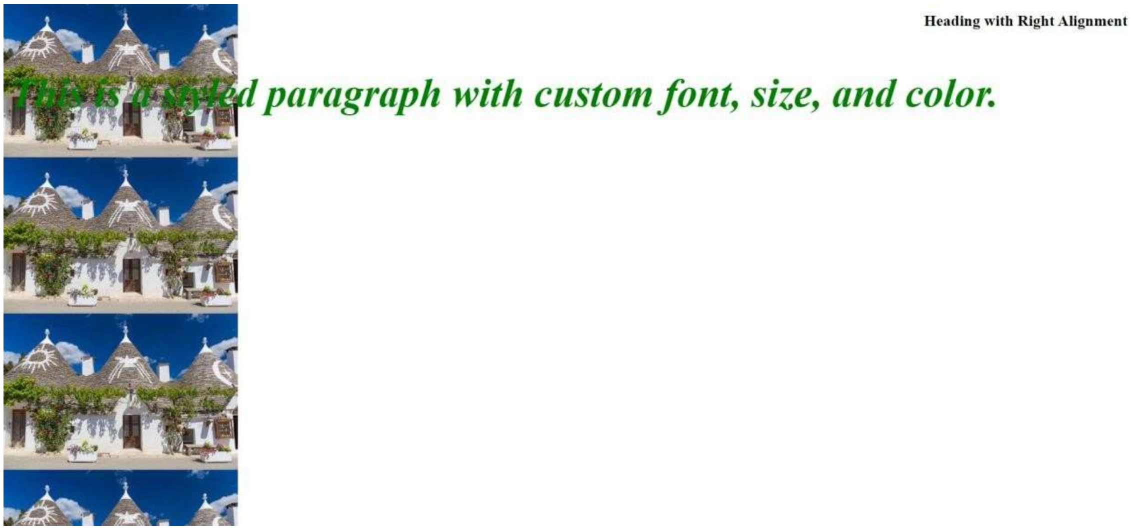
   Take a background image repeated vertically only.

Ans.
Code-

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

<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <link rel="stylesheet" href="style.css">
   <title>Index 7</title>
</head>

<body>
   <h1 class="heading">Heading with Right Alignment</h1>
   <p class="styled-paragraph">This is a styled paragraph with custom font, size, and
color.</p>
</body>

</html>

.heading {
   font-size: 120%; text-align: right;
```

```
    }

    /* Paragraph style */
    .styled-paragraph { color: green;
    font-weight: bold; font-style: italic;
    font-family: "Times New Roman", Times, serif; font-size: 50px;
    }

    /* Background image */ body {
    background-image: url("./pic_trulli.jpg"); background-repeat: repeat-y; background-size:
auto 100%;
    }
```

Output-



Heading with Right Alignment

This is a styled paragraph with custom font, size, and color.

Conclusion:

In this hands-on session, we gained fundamental insights into web development using HTML and CSS. HTML played a pivotal role in organizing web content through elements like headings, paragraphs, lists, links, and images. CSS, on the other hand, took charge of refining and presenting this content.

We delved into the crucial aspect of responsive design, which entails ensuring that web pages seamlessly adjust to various devices and screen dimensions. Furthermore, we ventured into the realm of enhancing user interaction by incorporating transitions and animations for dynamic web elements.

Collectively, this practical session furnished us with a robust groundwork for crafting websites that are both user-friendly and aesthetically engaging.

References:

- https://www.w3schools.com/html/
- https://www.w3schools.com/css/

Practical 2

Aim: Create interactive webpages using Javascript.

Hardware Requirement: HP Laptop

Software Requirement: Notepad Version 22H2 / VSCode

Knowledge Requirement: Basic understanding of HTML, CSS, Javascript.

Theory:

JavaScript is a versatile and widely-used programming language primarily known for its role in web development. It enables the creation of interactive and dynamic content on websites, making it an essential tool for modern web applications.

JavaScript is a high-level, interpreted programming language that is primarily used for front-end web development. It was first introduced by Netscape in 1995 and has since become one of the core technologies for building websites and web applications.

One of JavaScript's key features is its ability to add interactivity to web pages. It enables developers to create dynamic elements that respond to user actions, such as form validation, image sliders, and interactive maps. This interactivity greatly enhances the user experience, making websites more engaging and user-friendly.

JavaScript operates on the client-side, meaning it runs within the user's web browser. This allows for rapid responses and reduces the need for constant communication with the server. This client-side nature is what distinguishes JavaScript from server-side languages like Python or Ruby.

Moreover, JavaScript is highly versatile. While it is predominantly employed for front-end development, its usage has expanded into server-side scripting with the advent of platforms like Node.js. This flexibility allows developers to utilize JavaScript for a wide range of applications, from creating dynamic web pages to building entire web applications.

Code and Output:

1. DOM

Properties:Code-

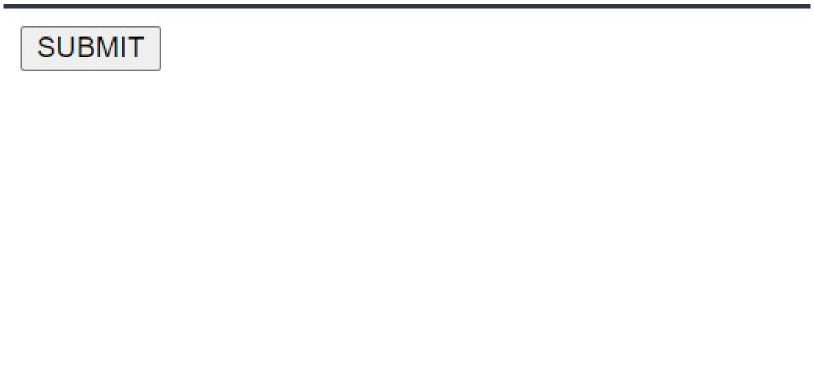
```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>DOM</title>
</head>
```

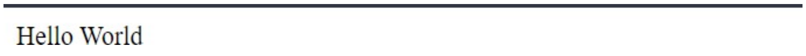
```
<body>
    <input type="button" onclick="Hello()" value="SUBMIT">
    <script src="./firstjs.js"></script>
</body>
</html>
```

firstjs.js

```
function Hello()
{
    document.write("Hello World");
}
```

Output-





```
Hello World
```

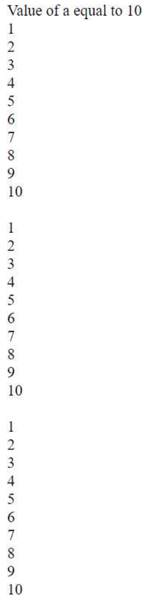
2. Conditionals and loops:

Code-

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Javascript</title>
</head>
```

```
<body>
   <script>
      //conditionals and loops
      a = 10
      if (a > 10) {
         document.write("Value of a greater than 10 <br>");
      }
      else if (a == 10) {
         document.write("Value of a equal to 10 <br>");
      }
      else {
         document.write("Value of a lesser than 10 <br>");
      }

      //for loop
      for (i = 0; i < a; i++) {
         document.write((i + 1) + "<br>");
      }
      document.write("<br>");

      //while loop
      x = 0
      while (x < a) {
         document.write((x + 1) + "<br>");
         x++;
      }
      document.write("<br>");

      //do-while loop
      x = 0
      do {
         document.write((x + 1) + "<br>");
         x++;
      } while (x < a)
      document.write("<br>");

   </script>
   <!-- <button onclick="alertbox();">Alert box</button>
   <button onclick="confirmbox()">Confirm box</button>
   <button onclick="promptbox()">Prompt box</button> -->
</body>

</html>
```

Output-



```
Value of a equal to 10
1
2
3
4
5
6
7
8
9
10

1
2
3
4
5
6
7
8
9
10

1
2
3
4
5
6
7
8
9
10
```

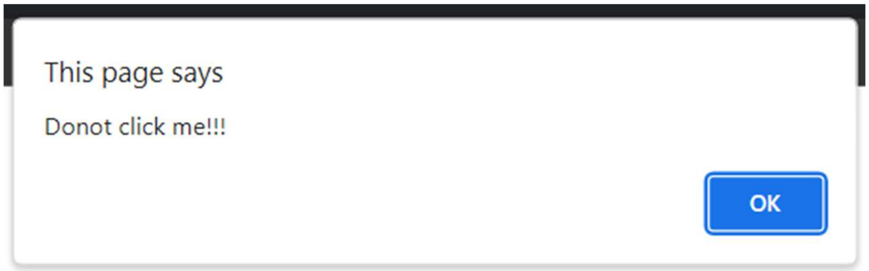
3. Alert box, Prompt box,

Code-

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

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

<body>
    <script>
        function alertbox() {
                      alert("Donot click
                      me!!!")
        }

        function confirmbox() {
            var txt;
            if (confirm("Press a button!")) {
                document.write("You pressed OK!")
            } else {
                document.write("You pressed cancel!!!")
            }
        }
        function promptbox() {
            let text;
            let person = prompt("Please enter your name:", "Ravi Jamanbhai Makwana");
            if (person == null || person == "") {
                document.write("You clicked cancel!!");
            } else {
                document.write("Hello " + person + "! How are you today?");
            }
        }

    </script>
    <button onclick="alertbox();">Alert box</button>
    <button onclick="confirmbox()">Confirm box</button>
    <button onclick="promptbox()">Prompt box</button>
</body>

</html>
```

Output-

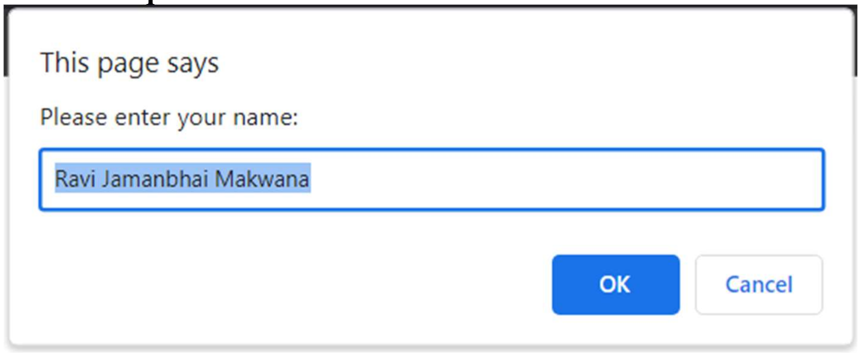
a. **Alert box:**



This page says

Donot click me!!!

b. **Prompt box:**



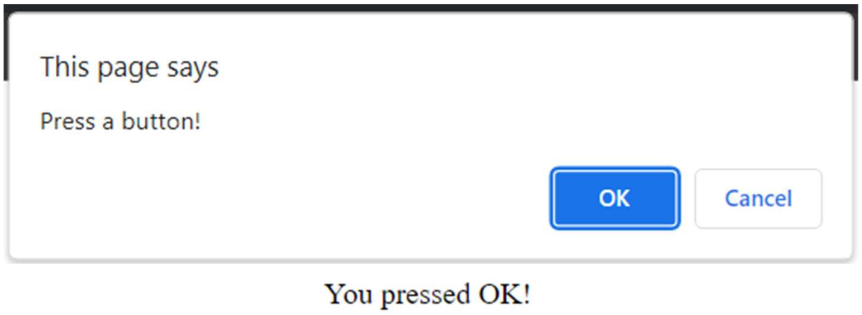
This page says

Please enter your name:

Ravi Jamanbhai Makwana

Hello Ravi Jamanbhai Makwana! How are you today?

c. **Confirm Box-**



This page says

Press a button!

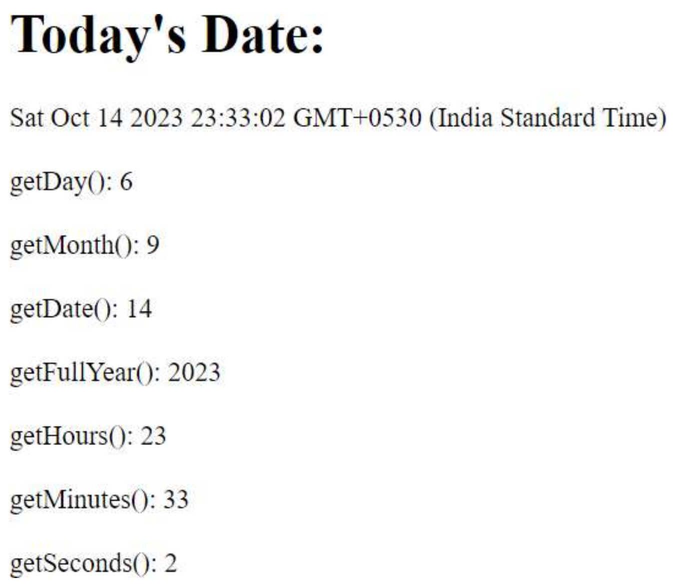
You pressed OK!

4. Javascript Date:

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>Today's Date:</h1>
    <script>
        var currentDate = new Date();
        document.write(currentDate + "<br><br>");
        document.write("getDay(): " + currentDate.getDay() + "<br><br>");
        document.write("getMonth(): " + currentDate.getMonth() + "<br><br>");
        document.write("getDate(): " + currentDate.getDate() + "<br><br>");
        document.write("getFullYear(): " + currentDate.getFullYear() + "<br><br>");
        document.write("getHours(): " + currentDate.getHours() + "<br><br>");
        document.write("getMinutes(): " + currentDate.getMinutes() + "<br><br>");
        document.write("getSeconds(): " + currentDate.getSeconds() + "<br><br>");
    </script>
</body>

</html>
```

Output-



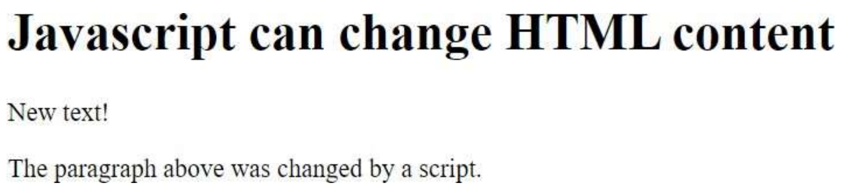
```
Today's Date:

Sat Oct 14 2023 23:33:02 GMT+0530 (India Standard Time)

getDay(): 6

getMonth(): 9

getDate(): 14

getFullYear(): 2023

getHours(): 23

getMinutes(): 33

getSeconds(): 2
```

5. Javascript can change HTML

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>Javascript can change HTML content</h1>
   <p id="p1">Hello World!</p>

   <script>

      document.getElementById("p1").innerHTML = "New text!";

   </script>

   <p>The paragraph above was changed by a script.</p>

</body>

</html>
```

Output-



Javascript can change HTML content

New text!

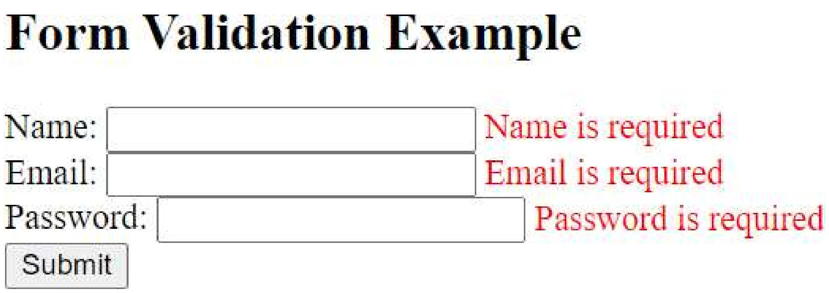
The paragraph above was changed by a script.

6. Form Validation:

Code-

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Form Validation Example</title>
    <style>
        .error {
            color: red;
        }
    </style>
</head>

<body>

    <h2>Form Validation Example</h2>

    <form id="myForm" onsubmit="return validateForm()" method="post">

        <label for="name">Name:</label>
        <input type="text" id="name" name="name">
        <span id="nameError" class="error"></span><br>

        <label for="email">Email:</label>
        <input type="text" id="email" name="email">
        <span id="emailError" class="error"></span><br>

        <label for="password">Password:</label>
        <input type="password" id="password" name="password">
        <span id="passwordError" class="error"></span><br>
    <input type="submit" value="Submit">

</form>

<script>
    function validateForm() {
        var name = document.getElementById("name").value;
        var email = document.getElementById("email").value;
        var password = document.getElementById("password").value;
        var nameError = document.getElementById("nameError");
        var emailError = document.getElementById("emailError");
        var passwordError = document.getElementById("passwordError");

        // Reset error messages
        nameError.innerHTML = "";
        emailError.innerHTML = "";
        passwordError.innerHTML = "";

        var isValid = true;

        if (name === "") {
            nameError.innerHTML = "Name is required";
            isValid = false;
        }

        if (email === "") {
            emailError.innerHTML = "Email is required";
            isValid = false;
        } else if (!isValidEmail(email)) {
            emailError.innerHTML = "Invalid email address";
            isValid = false;
        }

        if (password === "") {
            passwordError.innerHTML = "Password is required";
            isValid = false;
        } else if (password.length < 6) {
            passwordError.innerHTML = "Password must be at least 6 characters long";
            isValid = false;
        }

        return isValid;
    }

    function isValidEmail(email) {
        var emailRegex = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/;
        return emailRegex.test(email);
        }
    </script>

</body>

</html>
```

Output-



Form Validation Example

Name: Name is required

Email: Email is required

Password: Password is required

Conclusion-

In this lab session, we delved into fundamental JavaScript principles crucial for crafting dynamic web pages. This encompassed employing message boxes such as alert(), confirm(), and prompt() for presenting messages and gathering user input. Additionally,we tackled the intricacies of managing dates and times utilizing the Date object.
Moreover, we delved into DOM manipulation techniques, demonstrating how to alter HTML content and locate specific HTML elements through the document object. Lastly,we applied JavaScript to validate form inputs, guaranteeing that user-provided information adheres to predefined criteria.

References-

- https://www.w3schools.com/js/

Practical 3

Aim: Install the LAMP stack

Hardware Requirement: Computer: HP Laptop

Software Requirement:

1. Operating System: XAMPP is compatible with Windows, macOS, and Linux. Ensure your system meets the following OS-specific requirements: Windows: XP, Vista, 7, 8, 10
macOS: 10.6 or later
Linux: Any modern distribution with a 32- or 64-bit architecture.

2. Web Browser: You will need a web browser for testing your web applications. Popular choices include Google Chrome, Mozilla Firefox, and Microsoft Edge.

3. XAMPP Software: Download and install the XAMPP software from the official website (https://www.apachefriends.org/). Make sure to download the version that matches your operating system.

4. Text Editor or IDE: A text editor or integrated development environment (IDE) for writing and editing your web application code. Popular options include Visual Studio Code, Sublime Text, and PHP Storm.

5. Database Management Tool: If your web application uses databases, you may need a database management tool such as phpMyAdmin (included in XAMPP) or MySQL Workbench.

Knowledge Requirement:

1. Fundamental grasp of web technologies, which include HTML, CSS, JavaScript, and PHP.
2. Familiarity with web servers, databases, and server-side scripting.

3. Understanding of your operating system (whether it's Windows, macOS, or Linux) and its command-line interface.
4. Awareness of best practices for ensuring security while configuring and using web servers.

Theory:

The LAMP stack is a popular open-source software stack commonly used for web development and hosting. LAMP is an acronym that stands for:

1. Linux: The "L" in LAMP refers to the Linux operating system, which serves as the foundation of the stack. Linux is an open-source, Unix-like operating system that provides a stable and secure environment for web servers. It is known for its scalability, versatility, and robust security features.

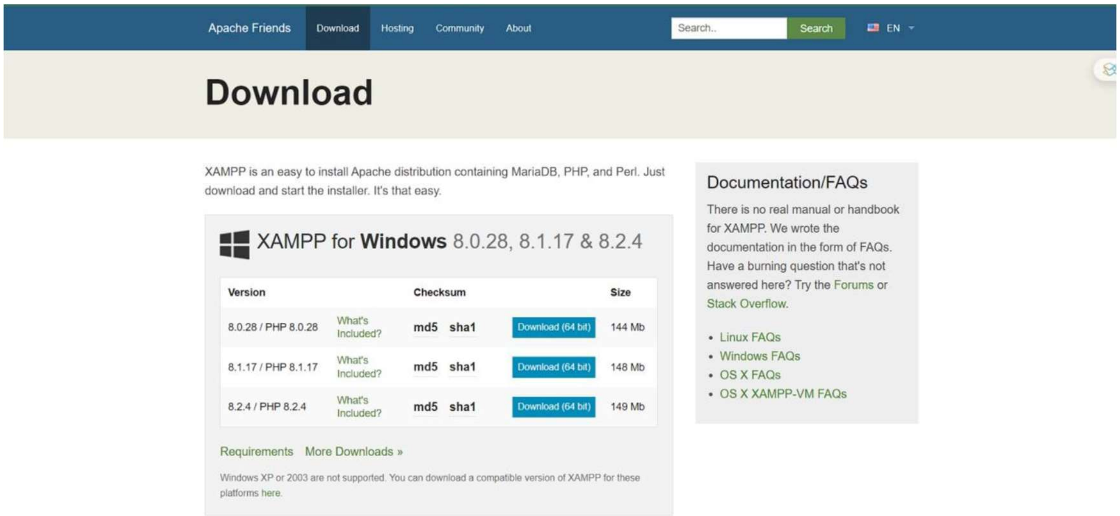
2. Apache: The "A" represents the Apache HTTP Server. Apache is a widely used, opensource web server software that is renowned for its reliability and performance. It serves as the web server component of the LAMP stack, handling HTTP requests and delivering web content to clients. Apache is highly customizable and supports a range of features, including URL rewriting, authentication, and SSL encryption.

3. MySQL: The "M" stands for MySQL, a popular open-source relational database management system (RDBMS). MySQL is known for its speed, scalability, and ease of use. It is used to store, manage, and retrieve data for web applications. MySQL is compatible with various programming languages and can be integrated seamlessly into web applications to provide dynamic data storage and retrieval.

4. PHP: The "P" signifies PHP, which is a widely used open-source server-side scripting language. PHP (Hypertext Preprocessor) is embedded within web pages and executed on the web server. It allows developers to create dynamic and interactive web applications by generating content on the server before sending it to the client's web browser. PHP can interact with databases, handle forms, and perform a wide range of server-side tasks. In summary, the LAMP stack is a robust, open-source solution for web development and hosting. It combines the Linux operating system, the Apache web server, the MySQL database, and the PHP scripting language to create a versatile, scalable, and cost-effective environment for building and deploying web applications. Its popularity in the web development community is a testament to its reliability and adaptability.

Steps and Output:

1. Install XAMPP



Download

XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy.

XAMPP for **Windows** 8.0.28, 8.1.17 & 8.2.4

Version		Checksum		Size
8.0.28 / PHP 8.0.28	What's Included?	**md5 sha1**	Download (64 bit)	144 Mb
8.1.17 / PHP 8.1.17	What's Included?	**md5 sha1**	Download (64 bit)	148 Mb
8.2.4 / PHP 8.2.4	What's Included?	**md5 sha1**	Download (64 bit)	149 Mb

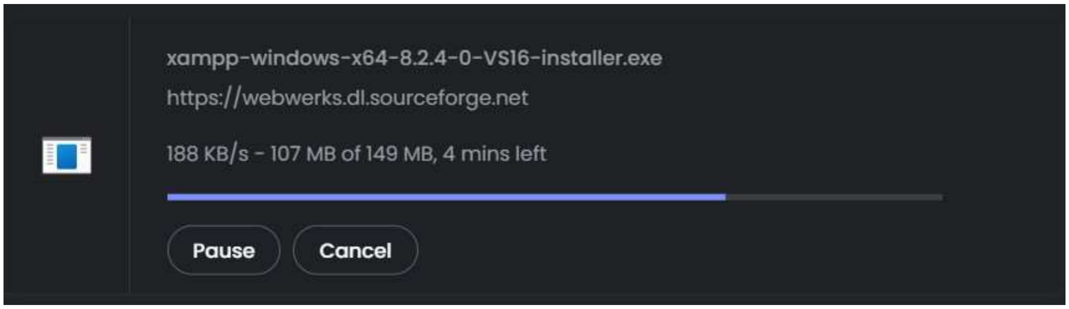
Requirements More Downloads »

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms here.

Documentation/FAQs

There is no real manual or handbook for XAMPP. We wrote the documentation in the form of FAQs. Have a burning question that's not answered here? Try the Forums or Stack Overflow.

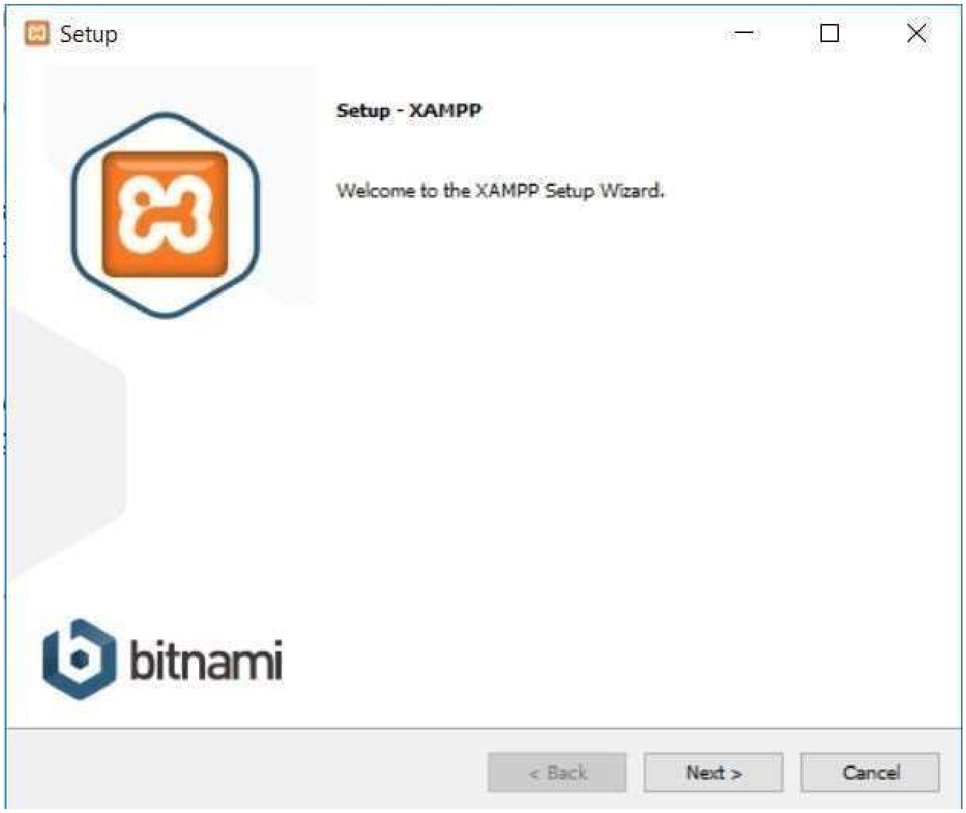
- Linux FAQs
- Windows FAQs
- OS X FAQs
- OS X XAMPP-VM FAQs



xampp-windows-x64-8.2.4-0-VS16-installer.exe

https://webwerks.dl.sourceforge.net

188 KB/s - 107 MB of 149 MB, 4 mins left



Setup

Setup - XAMPP

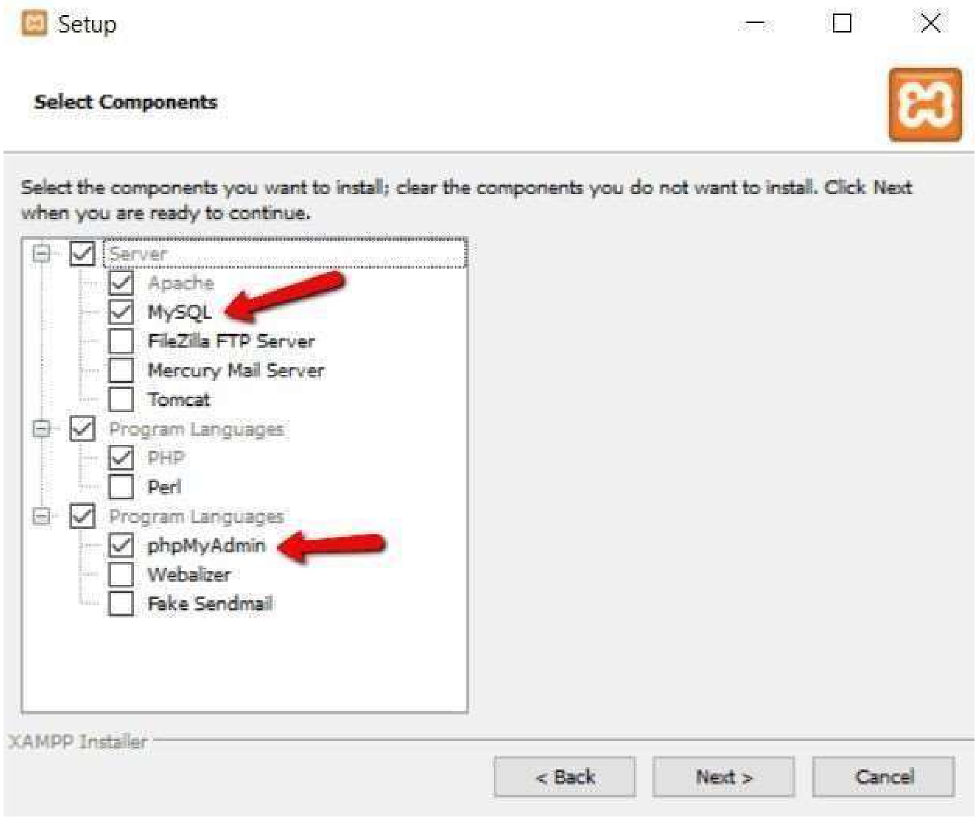
Welcome to the XAMPP Setup Wizard.

Just click Next to continue.

On the next screen, you can choose which components to install. To install XAMPP and WordPress, you do not need all the components. In addition to the required components, all you need are:

MySQL

phpMyAdmin



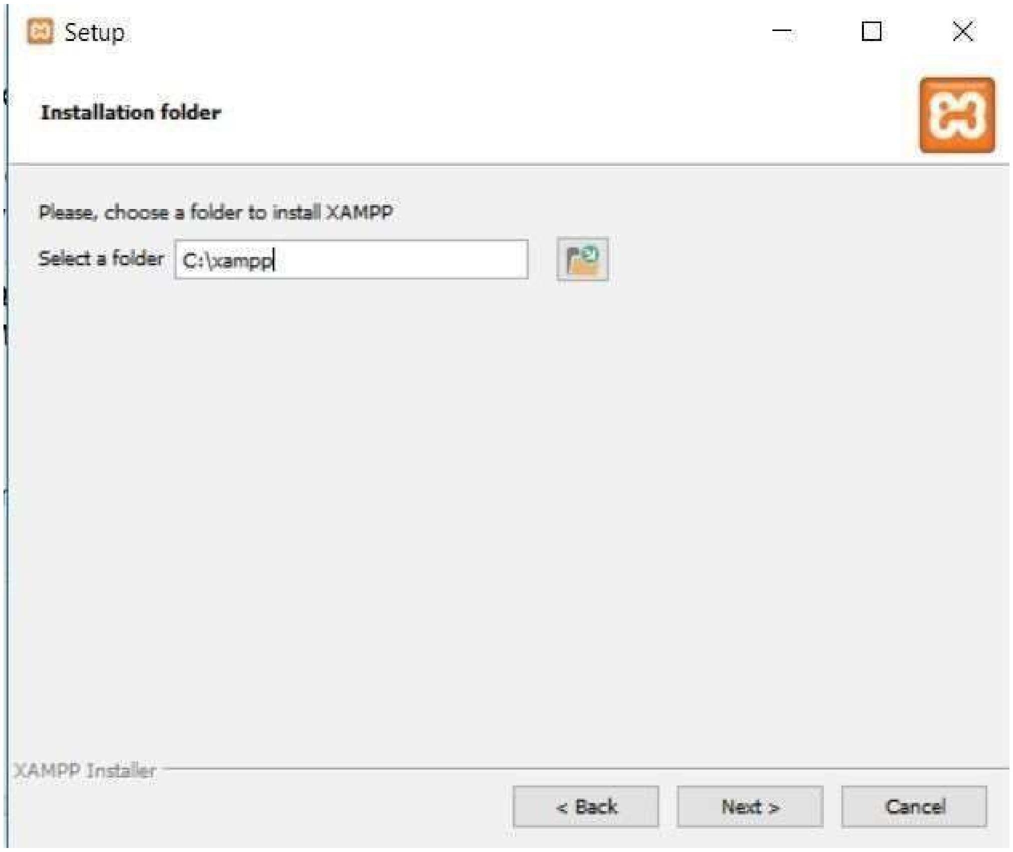
Select Components

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- ☑ Server
  - ☑ Apache
  - ☑ MySQL
  - □ FileZilla FTP Server
  - □ Mercury Mail Server
  - □ Tomcat
- ☑ Program Languages
  - ☑ PHP
  - □ Perl
- ☑ Program Languages
  - ☑ phpMyAdmin
  - □ Webalizer
  - □ Fake Sendmail

Again, this is all that is needed to install WordPress locally. Uncheck everything else and click Next.

You can now select which folder to install XAMPP in. I always leave it as the default:

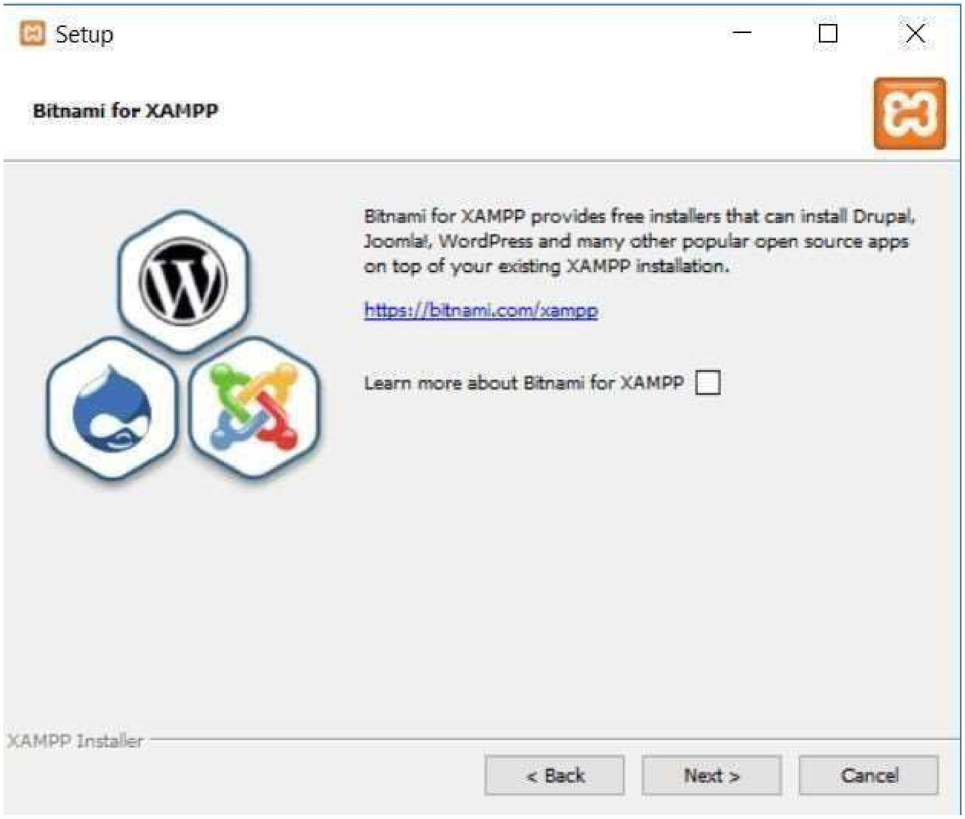


Installation folder

Please, choose a folder to install XAMPP

Select a folder C:\xampp

On the next screen, you'll get a prompt asking you to install Bitnami for XAMPP. You do not need this to install XAMPP and WordPress, so just uncheck the box:



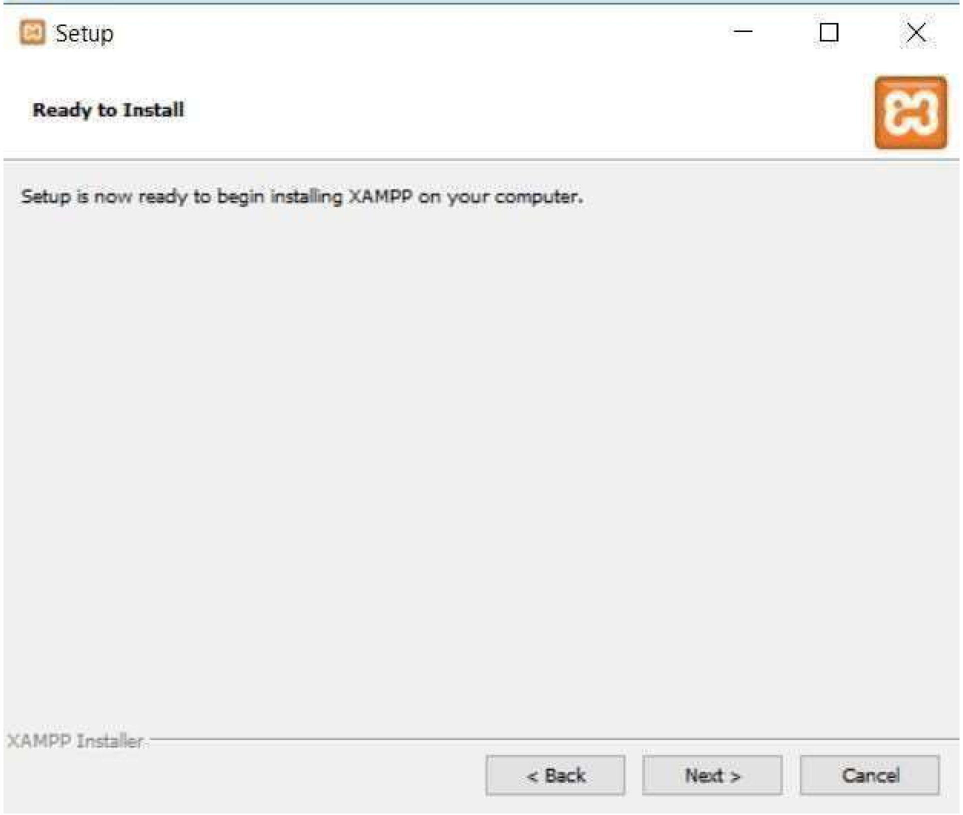
Bitnami for XAMPP

Bitnami for XAMPP provides free installers that can install Drupal, Joomla!, WordPress and many other popular open source apps on top of your existing XAMPP installation.

https://bitnami.com/xampp

Learn more about Bitnami for XAMPP □

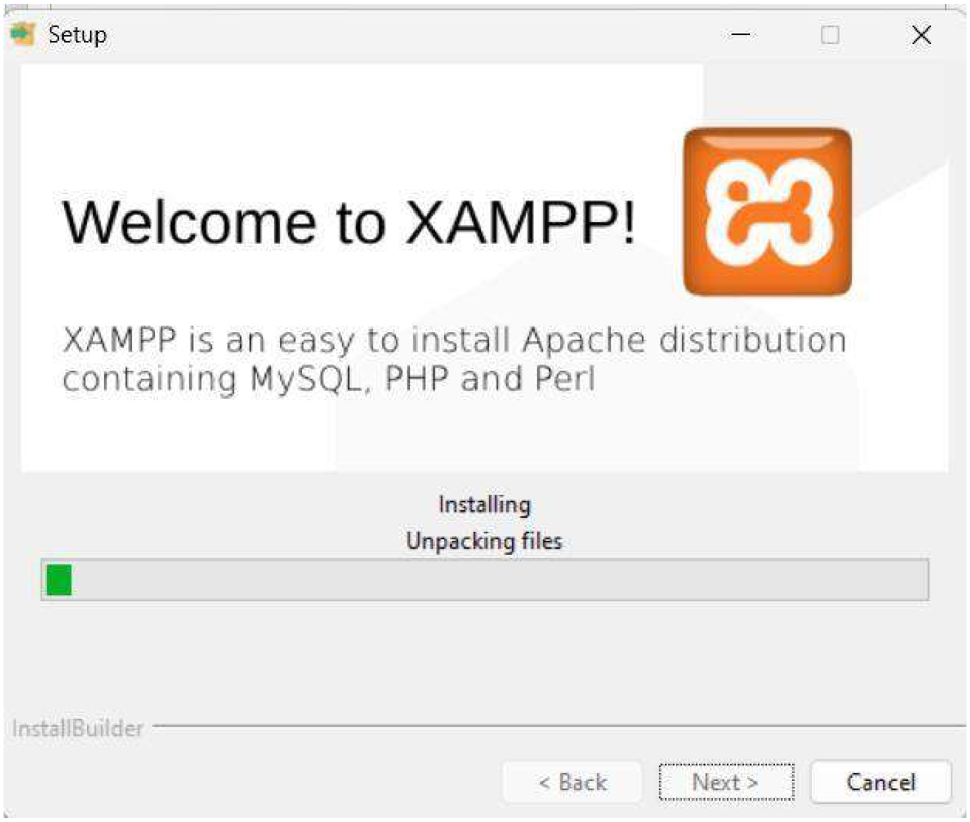
Once you click Next, you are ready to run the install:



Ready to Install

Setup is now ready to begin installing XAMPP on your computer.

The install process might take a few minutes:



Welcome to XAMPP!

XAMPP is an easy to install Apache distribution containing MySQL, PHP and Perl

Installing

Unpacking files

Once it finishes, you can launch the control panel to start working with XAMPP:

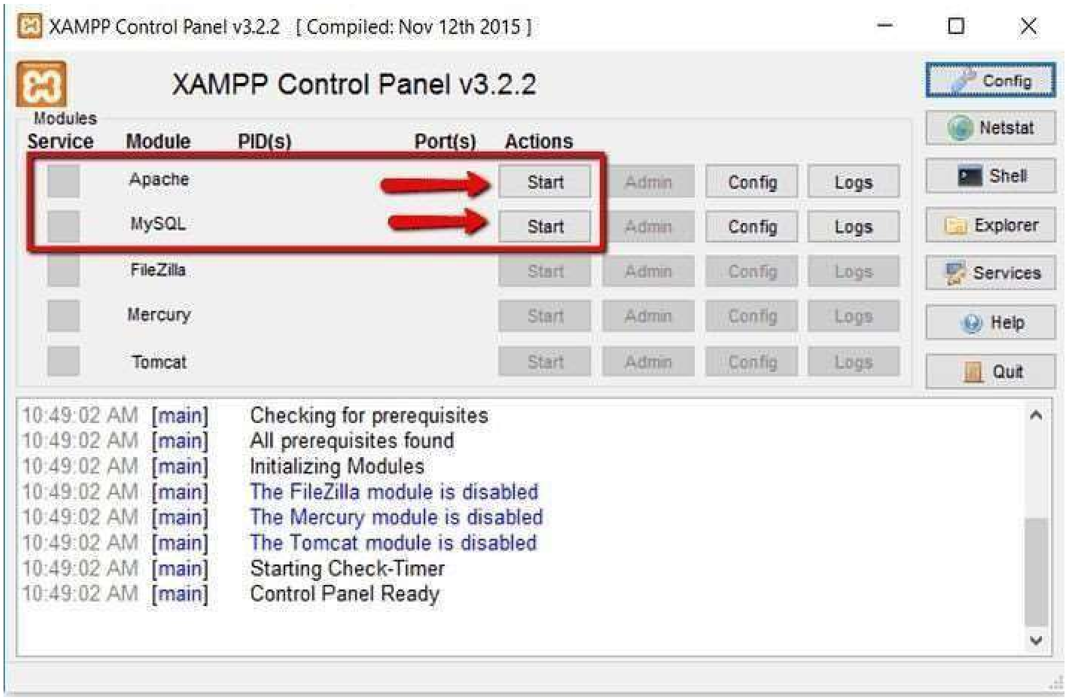
2. XAMPP Control Panel (Starting Apache and MySQL)

To install XAMPP and WordPress properly, you’ll need to run two modules:

Apache

MySQL

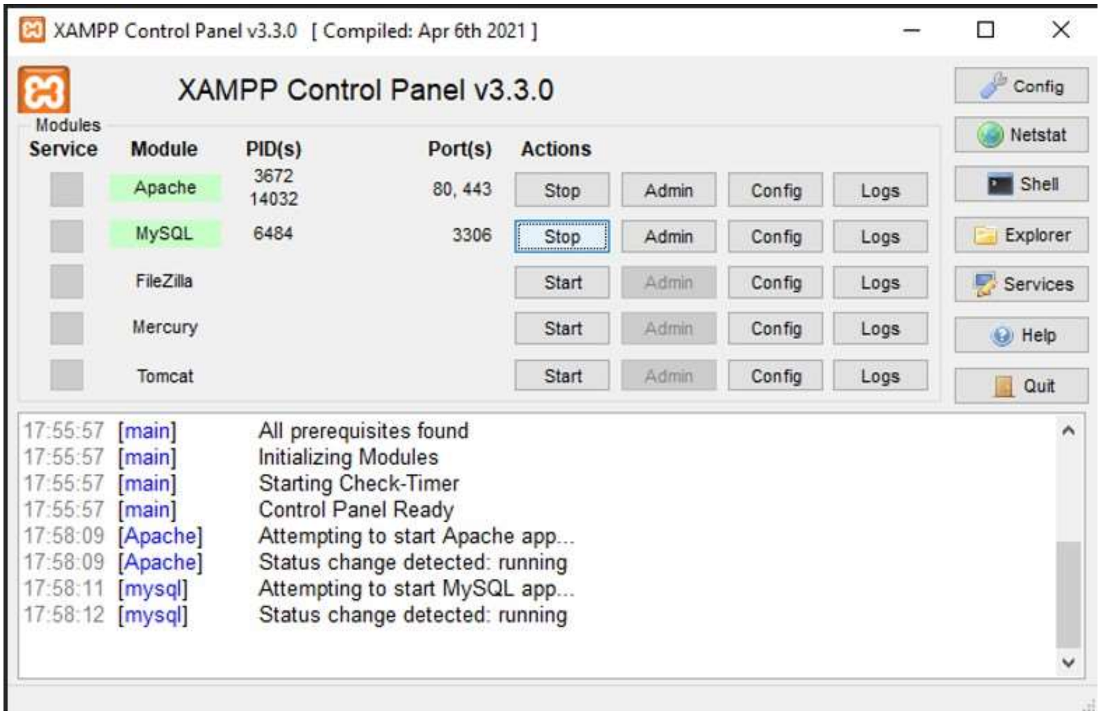
You can start both modules from the XAMPP control panel:


XAMPP Control Panel v3.2.2 [Compiled: Nov 12th 2015]
XAMPP Control Panel v3.2.2
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions			
	Apache			Start	Admin	Config	Logs
	MySQL			Start	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
10:49:02 AM  [main]   Checking for prerequisites
10:49:02 AM  [main]   All prerequisites found
10:49:02 AM  [main]   Initializing Modules
10:49:02 AM  [main]   The FileZilla module is disabled
10:49:02 AM  [main]   The Mercury module is disabled
10:49:02 AM  [main]   The Tomcat module is disabled
10:49:02 AM  [main]   Starting Check-Timer
10:49:02 AM  [main]   Control Panel Ready
```

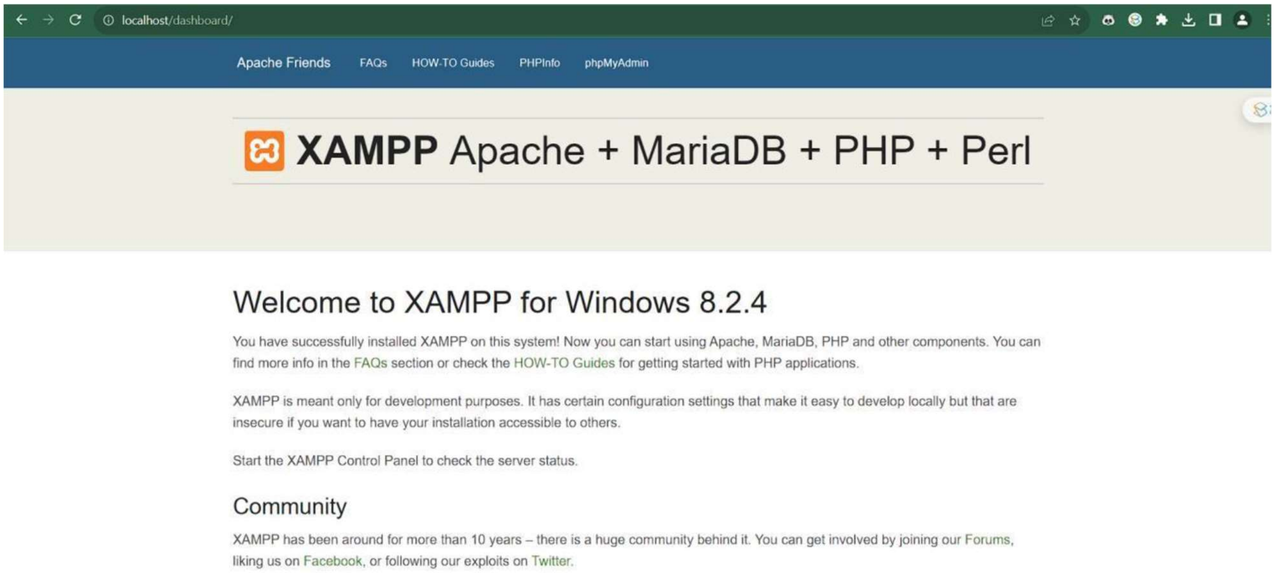
Once you launch them, you should see their status turn to green:


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions			
	Apache	3672 14032	80, 443	Stop	Admin	Config	Logs
	MySQL	6484	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
17:55:57  [main]     All prerequisites found
17:55:57  [main]     Initializing Modules
17:55:57  [main]     Starting Check-Timer
17:55:57  [main]     Control Panel Ready
17:58:09  [Apache]   Attempting to start Apache app...
17:58:09  [Apache]   Status change detected: running
17:58:11  [mysql]    Attempting to start MySQL app...
17:58:12  [mysql]    Status change detected: running
```

And now you should be able to test that your local server is working by going to http://localhost/ in your web browser of choice:



Welcome to XAMPP for Windows 8.2.4

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.

XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others.

Start the XAMPP Control Panel to check the server status.

Community

XAMPP has been around for more than 10 years – there is a huge community behind it. You can get involved by joining our Forums, liking us on Facebook, or following our exploits on Twitter.

Conclusion:

The setup of the XAMPP stack, incorporating Apache, MySQL, PHP, and Perl, offers a simple and user-friendly method for swiftly establishing a local web server environment. It presents a convenient foundation for web development, testing, and hosting web applications. With XAMPP installed, users obtain access to vital web server elements and can easily oversee databases, execute server-side scripts, and craft dynamic websites. In summary, XAMPP streamlines the process of establishing a local development environment and is an invaluable asset for web developers, making it an excellent choice for those seeking to create and test web applications without the intricacies of configuring each server component individually.

References:

https://www.apachefriends.org/download_success.html

Practical: 4

Aim: Implement the server-side scripting using PHP language

Hardware Requirement:

1. **Computer:** A desktop or laptop computer with sufficient processing power and memory to run web applications. XAMPP is not resource-intensive, so most modern computers should suffice.
2. **Storage Space:** Adequate storage space for your web development projects and the XAMPP software. A few gigabytes of free space should be more than enough.
3. **RAM:** At least 2GB of RAM for smooth operation. More RAM may be required if you plan to run resource-intensive applications.

Software Requirement:

1. **Operating System:** XAMPP is compatible with Windows, macOS, and Linux. Ensure your system meets the following OS-specific requirements:
   - Windows: XP, Vista, 7, 8, 10
   - macOS: 10.6 or later
   - Linux: Any modern distribution with a 32- or 64-bit architecture.

2. **Web Browser:** You will need a web browser for testing your web applications. Popular choices include Google Chrome, Mozilla Firefox, and Microsoft Edge.

3. **XAMPP Software:** Download and install the XAMPP software from the official website (https://www.apachefriends.org/). Make sure to download the version that matches your operating system.

4. **Text Editor or IDE:** A text editor or integrated development environment (IDE) for writing and editing your web application code. Popular options include Visual Studio Code, Sublime Text, and PHP Storm.

5. **Database Management Tool:** If your web application uses databases, you may need a database management tool such as phpMyAdmin (included in XAMPP) or MySQL Workbench.

6. **Optional: Version Control System:** Consider using a version control system (e.g., Git) for tracking changes in your code.

7. **Optional:** Content Management System (CMS): If you plan to use a CMS like WordPress, Joomla, or Drupal, make sure to meet their specific software requirements.

Knowledge Requirement:

1. Basic Programming Concepts:

- Variables, data types, and operators.
- Control structures (if statements, loops).
- Functions and procedures.

2. PHP Syntax:

- Familiarity with PHP syntax, including variables, arrays, loops, and functions.

3. Web Development Basics:

- Understanding of HTML and CSS.
- Knowledge of how web servers work.

4. Server-Side Scripting:

- Understanding of the difference between server-side and client-side scripting.

5. PHP Data Types:

- Scalars (int, float, string, boolean).
- Composite data types (arrays, objects).
- Special data types (NULL).

6. Functions and Control Structures:

- How to define and call functions.
- Conditional statements (if, else, switch).
- Looping (for, while, foreach).

7. Super Globals:

- Knowledge of PHP's superglobal arrays like `$_GET`, `$_POST`, `$_SESSION`, and `$_COOKIE`.

8. Error Handling:

- Understanding of error reporting and handling mechanisms.

9. Working with Databases:

- Connecting to databases using PHP (e.g., MySQL, PostgreSQL, SQLite).
- Executing SQL queries safely.

10. Form Handling:

- Handling form submissions and user input validation.

11. Sessions and Cookies:

- Working with sessions and cookies for user state management.

12. File Handling:

- Reading and writing files, and working with directories.

13. Security Best Practices:

- Awareness of common security vulnerabilities (e.g., SQL injection, XSS) and how to mitigate them.

14. Object-Oriented Programming (OOP):

- Basics of OOP in PHP, including classes, objects, inheritance, and encapsulation.

15. MVC (Model-View-Controller) Architecture:

- Understanding the concept of separating the application into models, views, and controllers.

16. API Integration:

- How to work with RESTful and other types of APIs in PHP.

17. Frameworks and Libraries:

- Familiarity with popular PHP frameworks (e.g., Laravel, Symfony, CodeIgniter) and libraries.

18. Composer:

- Dependency management using Composer, a PHP package manager.

19. Debugging and Profiling:

- Using debugging tools and techniques to find and fix issues in your code.

20. Version Control:

- Using version control systems like Git to manage your PHP projects.

21. Server Configuration:

- Knowledge of server configurations, such as PHP.ini settings.

22. Caching:

- Implementing caching mechanisms to improve application performance.

23. Web Security:

- Knowledge of best practices for securing PHP applications and servers.

24. Regular Expressions:

- Understanding and using regular expressions for pattern matching.

25. Web Application Deployment:

- Deploying PHP applications on web servers.

26. Performance Optimization:

- Techniques for optimizing PHP code and improving application performance.

27. API Documentation:

- How to create and document APIs in PHP.

28. Testing and Test-Driven Development (TDD):

- Strategies for testing PHP code and implementing TDD practices.

29. Continuous Integration/Continuous Deployment (CI/CD):

- Integrating PHP projects into CI/CD pipelines for automated testing and deployment.

Theory:

Introduction to PHP

PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP. PHP is an acronym for "PHP: Hypertext Preprocessor" PHP is a widely-used, open-source scripting language PHP scripts are executed on the server.

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial- scale=1.0">
<title>First Practical</title>
</head>
<body>
<?php
echo "My first PHP Script";
?>

</body>
</html>
```

Output:

```
My first PHP Script
```

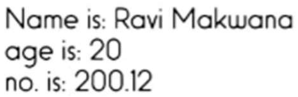
Variable Declaration in PHP

Rules for Declaring PHP Variables:

In PHP, variable names are denoted by a dollar sign ($) followed by the variable's name. These names are subject to certain rules: they can consist of alphanumeric characters and underscores (A-z, 0-9, _), but they must start with a letter or an underscore (_) character. Spaces are not allowed within variable names. It's essential to note that variable names in PHP are case-sensitive, meaning that $name and $NAME are considered distinct variables. Furthermore, they cannot begin with numbers or special symbols. Adhering to these conventions is crucial when working with PHP variables to ensure proper functionality and avoid errors in your code.

Code:

```
<?php
$str = "Ravi Makwana";
$age = 20;
$no = 200.12;
echo "Name is: $str <br/>"; echo "age is: $age <br/>"; echo "no is: $no <br/>"
?>
```

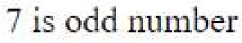
Output:	 Name is: Ravi Makwana age is: 20 no. is: 200.12

Control Statements in PHP

PHP If-Else Statement

The if-else statement executes one block of code if a specified condition is true and another block if the condition is false.

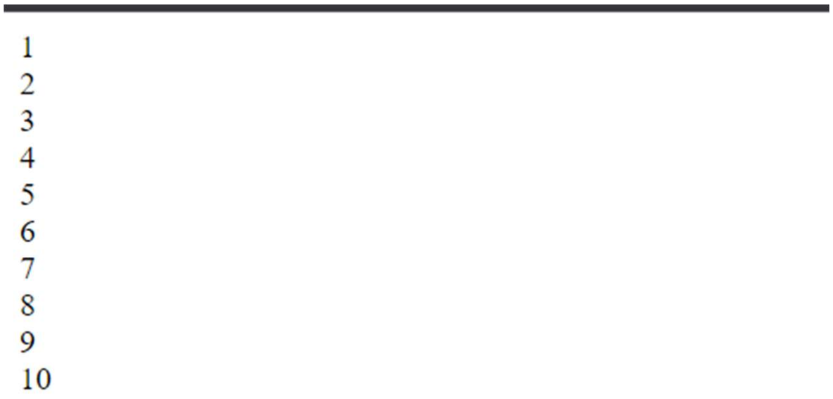
Code:	<?php
	$num=7; if($num%2==0){
	echo "$num is even number";
	}else{
	echo "$num is odd number";
	}

	?>
Output:	 7 is odd number

PHP For Loop

The for loop is used to execute a block of code for a specified number of times when the number of iterations is known.

Code:	<?php
	for($n=1;$n<=10;$n++){ echo "$n<br/>";
	}
	?>

Output:	 1 2 3 4 5 6 7 8 9 10

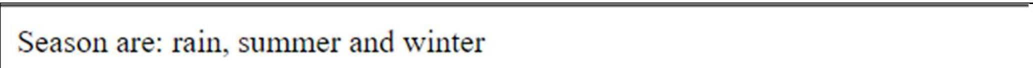
PHP Arrays

PHP arrays are used to store multiple values of similar types in a single variable.

There are three types of arrays in PHP:

Indexed Array:

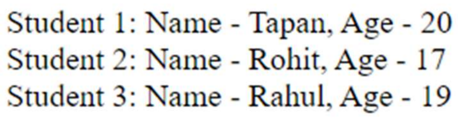
Code:	<?php $season=array("rain","summer","winter"); echo "Season are: $season[0], $season[1] and $season[2]"; ?>
Output:	

	 Season are: rain, summer and winter

Associative Array

Code:	<?php $salary=array("Tapan"=>"500000","Rohit"=>"550000", "Rahul"=>"600000"); echo "Ravi salary: ".$salary["Tapan"]."<br/>"; echo "Rohit salary: ".$salary["Rohit"]."<br/>"; echo "Rahul salary: ".$salary["Rahul"]."<br/>"; ?>
Output:	Ravi salary: 500000 Rohit salary: 550000 Rahul salary: 600000

Multidimensional Array

Code:	<?php $studentData = array(array("name" => "Tapan", "age" => 20), array("name" => "Rohit", "age" => 17), array("name" => "Rahul", "age" => 19)); echo "Student 1: Name - " . $studentData[0]["name"] . ", Age - " . $studentData[0]["age"] . " "; echo "Student 2: Name - " . $studentData[1]["name"] . ", Age - " . $studentData[1]["age"] . " "; echo "Student 3: Name - " . $studentData[2]["name"] . ", Age - " . $studentData[2]["age"]; ?>
Output:	 Student 1: Name - Tapan, Age - 20 Student 2: Name - Rohit, Age - 17 Student 3: Name - Rahul, Age - 19

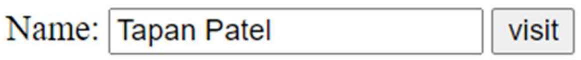
PHP Form Handling

PHP Get Form

The get request is the default form request and sends data as part of the URL. It's not secure for sensitive data.

HTML Code:	<form action="first.php" method="get"> Name: <input type="text" name="name" /> <input type="submit" value="visit" /> </form>
PHP Code:	<?php $name=$_GET["name"]; echo "Welcome, $name"; ?>

Output:

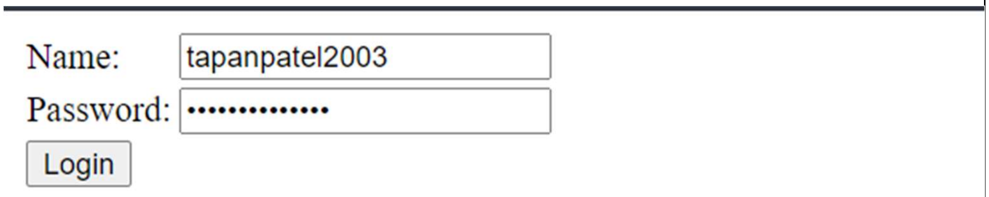

Name: Tapan Patel

PHP Post Form

The post request is used for sensitive or large data as it's not visible in the URL.

HTML Code:

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
    <form action="first.php" method="post">
        <table>
            <tr>
                <td>Name:</td>
                <td><input type="text" name="name" required /></td>
            </tr>
            <tr>
                <td>Password:</td>
                <td><input type="password" name="password" required
/></td>
            </tr>
            <tr>
                <td colspan="2"><input type="submit" value="Login"
/></td>
            </tr>
        </table>
    </form>
</body>
</html>
```

PHP Code:	<?php $name=$_POST["name"]; $password=$_POST["password"]; echo "Welcome: $name, your password is: $password"; ?>
Output:	 Name: tapanpatel2003 Password: ••••••••••••••

Form Validation

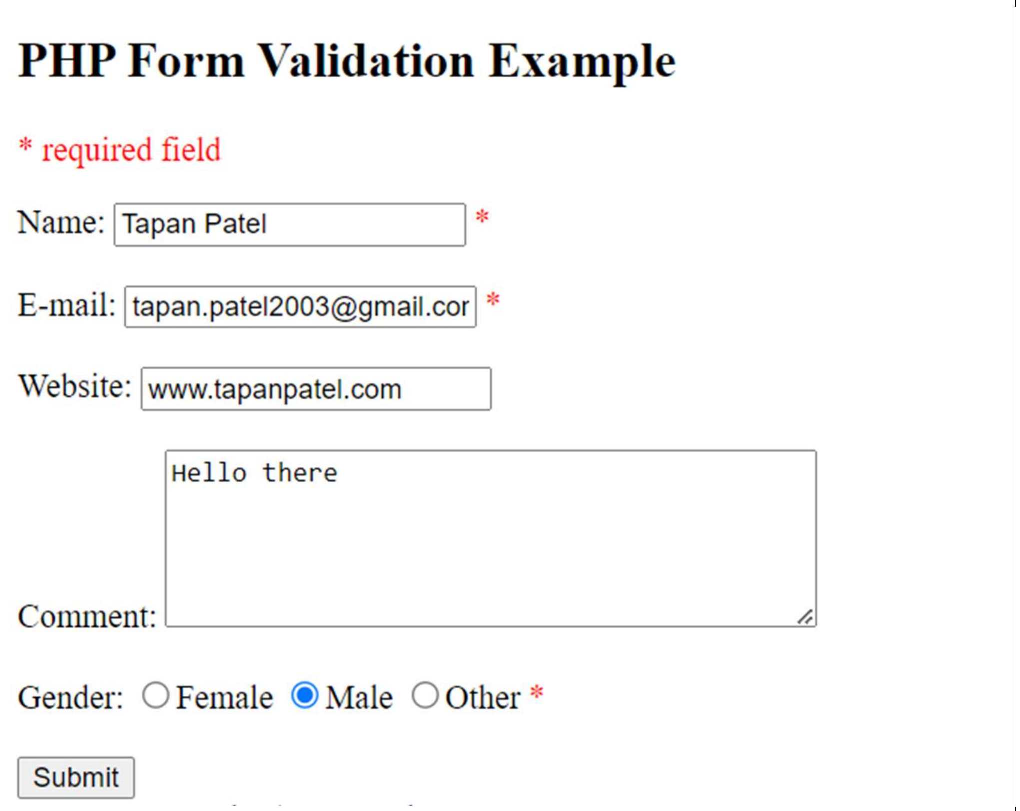
To ensure the data is safe and valid, you should validate and sanitize form input. Here's an example of form validation:

Code:

```
<!DOCTYPE html>
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>
<?php

$nameErr = $emailErr = $genderErr = $websiteErr = "";
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {
        $name = test_input($_POST["name"]);
    }
    if (empty($_POST["email"])) {
        $emailErr = "Email is required";
    } else {
        $email = test_input($_POST["email"]);
    }
    if (empty($_POST["website"])) {
        $website = "";
    } else {
        $website = test_input($_POST["website"]);
    }
    if (empty($_POST["comment"])) {
        $comment = "";
    } else {
```

```
                $comment = test_input($_POST["comment"]);
        }
        if (empty($_POST["gender"])) {
                $genderErr = "Gender is required";
        } else {
                $gender = test_input($_POST["gender"]);
        }
}


        function test_input($data) {
              $data = trim($data);
        $data = stripslashes($data);
        $data = htmlspecialchars($data);return
        $data;
}
?>
<h2>PHP Form Validation Example</h2>
<p><span class="error">* required field</span></p>
<form method="post" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]);?>">Name:
<input type="text" name="name">
<span class="error">* <?php echo $nameErr;?></span>
<br><br>
E-mail: <input type="text" name="email">
<span class="error">* <?php echo $emailErr;?></span>
<br><br>
Website: <input type="text" name="website">
<span class="error"> <?php echo $websiteErr;?></span>
<br><br>
Comment: <textarea name="comment" rows="5" cols="40"></textarea>
<br><br>
Gender:
<input type="radio" name="gender" value="female">Female
<input type="radio" name="gender" value="male">Male
<input type="radio" name="gender" value="other">Other
<span class="error">* <?php echo $genderErr;?></span>
<br><br>
<input type="submit" name="submit" value="Submit">
</form>
<?php
echo "<h2>Your Input:</h2>";echo
$name;
echo "<br>";echo
$email;
echo "<br>";
echo $website;echo
"<br>"; echo
$comment;echo
"<br>"; echo
$gender;
?>
</body>
</html>
```

Output:



PHP Form Validation Example

* required field

Name: Tapan Patel *

E-mail: tapan.patel2003@gmail.cor *

Website: www.tapanpatel.com

Comment: Hello there

Gender: ○ Female ● Male ○ Other *

PHP Cookie

PHP cookies are like digital sticky notes that websites can leave on your computer. They store small pieces of information, like your preferences or login status, so the website can remember you and provide a better experience next time you visit.

PHP setcookie() Function

PHP setcookie() function is used to set cookie with HTTP response. Once cookie is set, you can access it by $_COOKIE superglobal variable.

Syntax:

bool setcookie (string $name [, string $value [, int $expire = 0 [, string $path [, string $domain [, bool $secure = false [, bool $httponly = false]]]]]])

Example:

setcookie("CookieName", "CookieValue"); // Defining name and value only

setcookie("CookieName", "CookieValue", time() + 1 * 60 * 60); // Using expiry in 1 hour

PHP $_COOKIE Superglobal

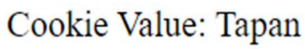
The $_COOKIE superglobal variable is used to retrieve cookies.

Setcookie.php	<?php setcookie("user", "Tapan"); ?>

Displaycookie.php

```
<html>
<body>
<?php
if (!isset($_COOKIE["user"])) {
        echo "Sorry, the cookie is not found!";
} else {
        echo "Cookie Value: " . $_COOKIE["user"];
}
?>
</body>
</html>
```

Output:



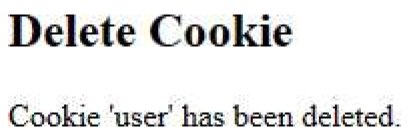
```
Cookie Value: Tapan
```

Deleting a Cookie

You can delete a cookie by setting its expiration date in the past.

Code:

```
<?php
 if (isset($_COOKIE["user"])) { setcookie("user", "", time() -
        3600, "/");
        $message = "Cookie 'user' has been deleted.";
} else {
        $message = "Cookie 'user' does not exist.";
}
?>
<!DOCTYPE html>
<html>
<head>
        <title>Delete Cookie</title>
</head>
<body>
<h2>Delete Cookie</h2>
<p><?php echo $message; ?></p>
</body>
</html>
```

Output:	 Delete Cookie Cookie 'user' has been deleted.

PHP Session

PHP sessions are like short-term memory for websites. When you visit a website, it creates a session to remember you while you're on the site. It's like a temporary storage space for information specific to your visit, such as your login status or items in your shopping cart. Sessions help websites provide a personalized experience during your visit, and they forget everything once you leave.

PHP Session Start

The session_start() function is used to start a new or resume an existing session. It returns the existing session if it already exists.

Syntax:

```
bool session_start()
```

Example:

```
session_start();
```

PHP $_SESSION

The $_SESSION superglobal is an associative array used to set and get session variable values.

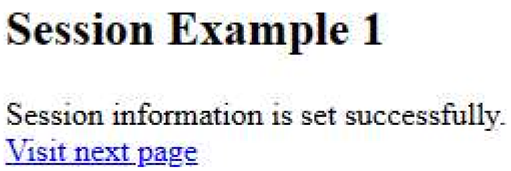
Session1.php

```
<?php session_start();
?>
<!DOCTYPE html>
<html>
<head>
	<title>Session 1</title>
</head>
<body>
<h2>Session Example 1</h2>
<?php
$_SESSION["user"] = "Tapan";
echo "Session information is set successfully.<br/>";
?>
<a href="session2.php">Visit next page</a>
</body>
</html>
```

Session2.php

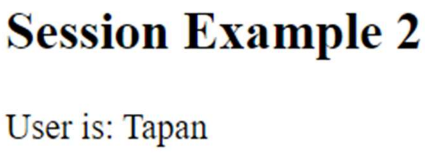
```
<?php session_start();
?>
<!DOCTYPE html>
<html>
<head>
	<title>Session 2</title>
</head>
<body>
<h2>Session Example 2</h2>
<?php
echo "User is: " . $_SESSION["user"];
?>
</body>
</html>
```

Output:



Session Example 1

Session information is set successfully.
Visit next page



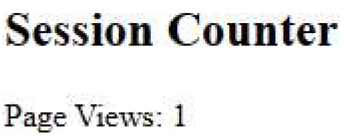
Session Example 2

User is: Tapan

Code:

```
<?php session_start();
if (!isset($_SESSION['counter'])) {
        $_SESSION['counter'] = 1;
} else {
        $_SESSION['counter']++;
}
?>
<!DOCTYPE html>
<html>
<head>
        <title>Session Counter</title>
</head>
<body>
<h2>Session Counter</h2>
<?php
echo "Page Views: " . $_SESSION['counter'];
?>
</body>
</html>
```

Output:



Session Counter

Page Views: 1

Destroying a Session

You can destroy all session variables completely using session_destroy().

Code:	<?php session_start();
	session_destroy();
	echo "Session Destroed";
	?>

Output:	 Session Destroed

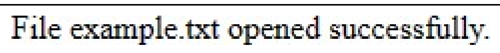
PHP File Handling

PHP file handling is how websites deal with files. They can open files, read from them, write to them, and even delete them. It's like how you work with documents on your computer but done by the website's code.

PHP Open File - fopen()

The fopen() function is used to open a file.

Code:	<?php
	$filename = "example.txt";
	$file = fopen($filename, "r");
	if ($file) {
	        echo "File $filename opened successfully.";fclose($file);
	} else {
	        echo "Failed to open the file.";
	}
	?>

Output:	 File example.txt opened successfully.

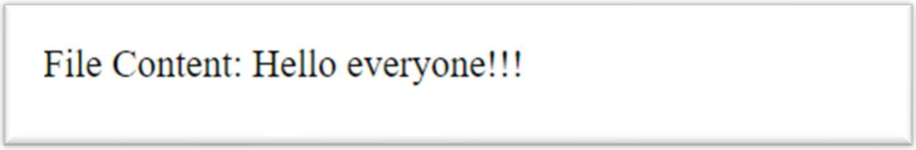
PHP Read File - fread()

The fread() function is used to read the content of a file.

Code:

```
<?php
$filename = "example.txt";

if (file_exists($filename)) {
        $fileContent = file_get_contents($filename);echo "File
        content:\n" . $fileContent;
} else {
        echo "File does not exist.";
}
?>
```

Output:



```
File Content: Hello everyone!!!
```

PHP Write File - fwrite()

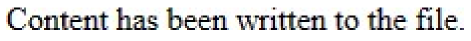
The fwrite() function is used to write the content of a string into a file.

Code:

```
<?php
$filename = "example.txt";
$content = "Good Bye!";
```

```
$file = fopen($filename, "w");if ($file) {
        fwrite($file, $content);
        fclose($file);
        echo "Content has been written to the file.";
} else {
        echo "Failed to open the file for writing.";
}
?>
```

Output:


Content has been written to the file.

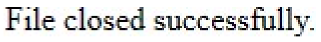
PHP Close File - fclose()

The fclose() function is used to close an open file pointer.

Code:

```
<?php
$filename = "example.txt";
$file = fopen($filename, "r");

if ($file) {
        fclose($file);
        echo "File closed successfully.";
} else {
        echo "Failed to open the file.";
}
?>
```

Output:


File closed successfully.

PHP Delete File - unlink()

The unlink() function is used to delete a file.

Code:

```php
<?php
$filename = "example.txt";

  if (file_exists($filename)) {if
          (unlink($filename)) {
                  echo "File deleted successfully.";
          } else {
                  echo "Failed to delete the file.";
          }
} else {
          echo "File does not exist.";
}
?>
```

Output:

```
File deleted successfully.
```

PHP File Upload

PHP allows you to upload single and multiple files through few lines of code only.

PHP file upload features allows you to upload binary and text files both. Moreover, you can have the full control over the file to be uploaded through PHP authentication and file operation functions.

PHP $_FILES

The $_FILES global contains all the information about a file being uploaded, including its name, type, size, temp name, and errors.

$_FILES['filename']['name'] : returns file name.

$_FILES['filename']['type']: returns MIME type of the file.

$_FILES['filename']['size'] : returns size of the file (in bytes).

$_FILES['filename']['tmp_name'] : returns temporary file name of the file which was stored on the server.

$_FILES['filename']['error'] : returns error code associated with this file

move_uploaded_file() function

The move_uploaded_file() function moves the uploaded file to a new location. The move_uploaded_file() function checks internally if the file is uploaded thorough the POST request. It moves the file if it is uploaded through the POST request.

Syntax

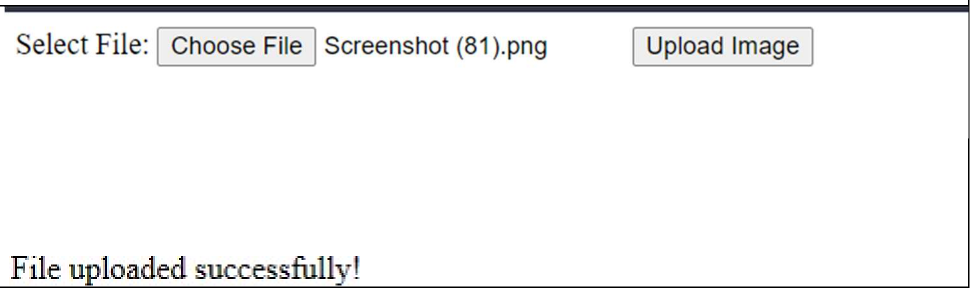
bool move_uploaded_file (string $filename , string $destination)

PHP File Upload Example

Uploadfile.html	<!DOCTYPE html> <html> <head> <title>File Upload Example</title> </head> <body> <form action="uploadfiles.php" method="post" enctype="multipart/form-data"> Select File: <input type="file" name="fileToUpload" /> <input type="submit" value="Upload Image" name="submit" /> </form> </body> </html>
Uploadfiles.php	<?php $uploadDirectory = "C:/xampp/htdocs/web-tech-lab/File Handling/Files";

```
if (isset($_POST["submit"]) &&
isset($_FILES["fileToUpload"])) {
        $targetPath = $uploadDirectory . basename($_FILES["fileToUpload"]["name"]);
          if (file_exists($targetPath)) {
                echo "Sorry, the file already exists. Please renamethe file and try
again.";
          } else {
                if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"],
$targetPath)) {
                          echo "File uploaded successfully!";
                  } else {
                        echo "Sorry, there was an error uploading yourfile. Please try
again.";
                  }
          }
} else {
          echo "Please select a file to upload.";
}
?>
```

Output:



File uploaded successfully!

Conclusion:

In this manual, our focus has been on delving into the realm of web technology, with a primary emphasis on harnessing the power of PHP to craft dynamic and captivating web pages. PHP stands out as a formidable scripting language, enabling web developers to create a myriad of compelling features and functionalities for websites. Its widespread popularity and open-source nature have cemented its status as the go-to choice for a multitude of web development projects.

Throughout this guide, we've journeyed through the essential building blocks of PHP. We began by acquainting ourselves with the fundamentals, understanding what PHP is and how it facilitates the creation of engaging web content. Subsequently, we explored the realm of variables, learning how to store and manipulate data efficiently. We also delved into decision-making processes and repetitive actions through if-else statements and loops. Furthermore, we delved into the world of data organization with arrays and data retrieval from web forms. Additionally, we addressed the intricacies of managing user data with cookies and sessions, which enhance web experiences by preserving user-specific information. Finally, we explored file handling, mastering the techniques of reading, writing, and manipulating data files in the web development context. These topics collectively equip you with the necessary knowledge to create feature-rich and secure web pages using PHP.

References:

https://www.w3schools.com/php/

Practical: 5

Aim: Create a web page that retrieves and displays info from the XML file

Requirement: Laptop /Computer

Software Requirement: -

- Windows 10/11 / Browser (Chrome, Brave, Firefox) / Winget

- PHP / XAMPP / VS code / Notepad

Knowledge Requirement: HTML, CSS, PHP

Ajax2.html

```
<!DOCTYPE html>

<html>
  <style>
    table,th,
    td {
      border: 1px solid black;

      border-collapse: collapse;
    }

    th,
    td {
      padding: 5px;
    }
  </style>

  <body>
    <button type="button" onclick="loadXMLDoc()">Get my CD collection</button>

    <br /><br />

    <table id="demo"></table>

    <script>
      function loadXMLDoc() {
        var xmlhttp = new XMLHttpRequest();

        xmlhttp.onreadystatechange = function () {
          if (this.readyState == 4 && this.status == 200) {
            myFunction(this);
          }
        };

        xmlhttp.open("GET", "cd_catalog.xml", true);

        xmlhttp.send();
      }

      function myFunction(xml) {
        var i;

        var xmlDoc = xml.responseXML;

        var table = "<tr><th>Artist</th><th>Title</th></tr>";var

        x = xmlDoc.getElementsByTagName("CD");

        for (i = 0; i < x.length; i++) {table
          +=
            "<tr><td>" +
            x[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeVal
            ue +"</td><td>" +
            x[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue
            + "</td></tr>";
        }

        document.getElementById("demo").innerHTML = table;
      }
    </script>
  </body>
</html>
```

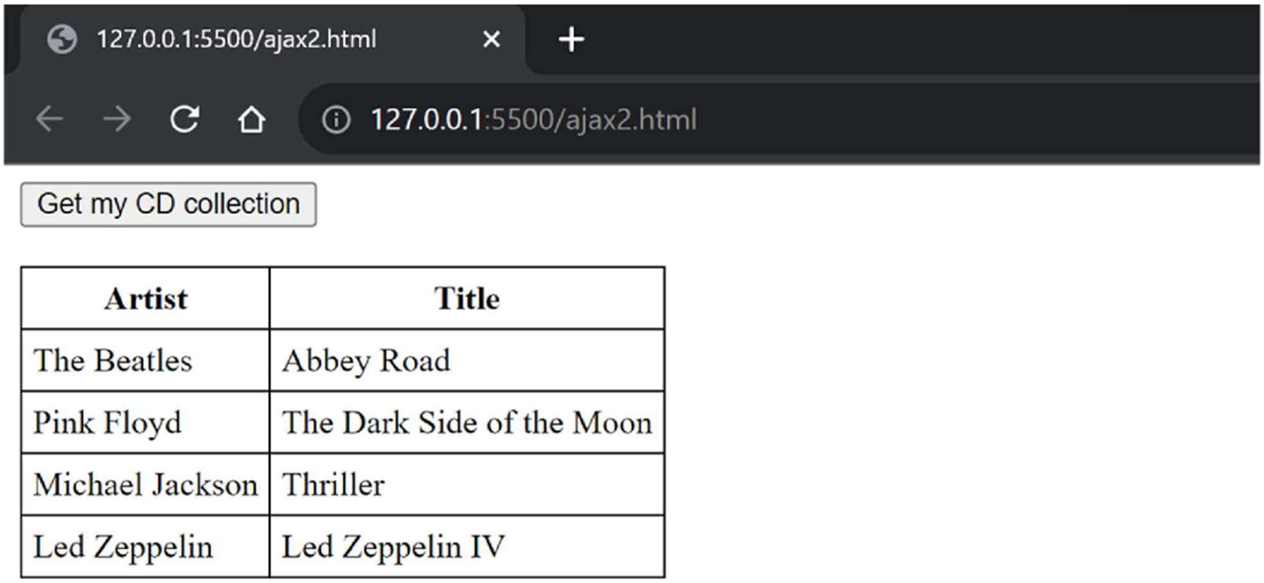
Cd_catalog.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<CATALOG>
  <CD>
    <ARTIST>The Beatles</ARTIST>
    <TITLE>Abbey Road</TITLE>
  </CD>
  <CD>
    <ARTIST>Pink Floyd</ARTIST>
    <TITLE>The Dark Side of the Moon</TITLE>
  </CD>
```

```
<CD>
  <ARTIST>Michael Jackson</ARTIST>
  <TITLE>Thriller</TITLE>
</CD>
```

```
    <CD>
      <ARTIST>Led Zeppelin</ARTIST>
      <TITLE>Led Zeppelin IV</TITLE>
    </CD>
  </CATALOG>
```

OUTPUT:



Artist	Title
The Beatles	Abbey Road
Pink Floyd	The Dark Side of the Moon
Michael Jackson	Thriller
Led Zeppelin	Led Zeppelin IV

Practical: 6

Aim: **Create a web page that retrieves and displays info from the Json file**

Requirement: Laptop /Computer

Software Requirement: -

- Windows 10/11 / Browser (Chrome, Brave, Firefox) / Winget

- VS code / Notepad

Knowledge Requirement: HTML, CSS, JS

Index.html

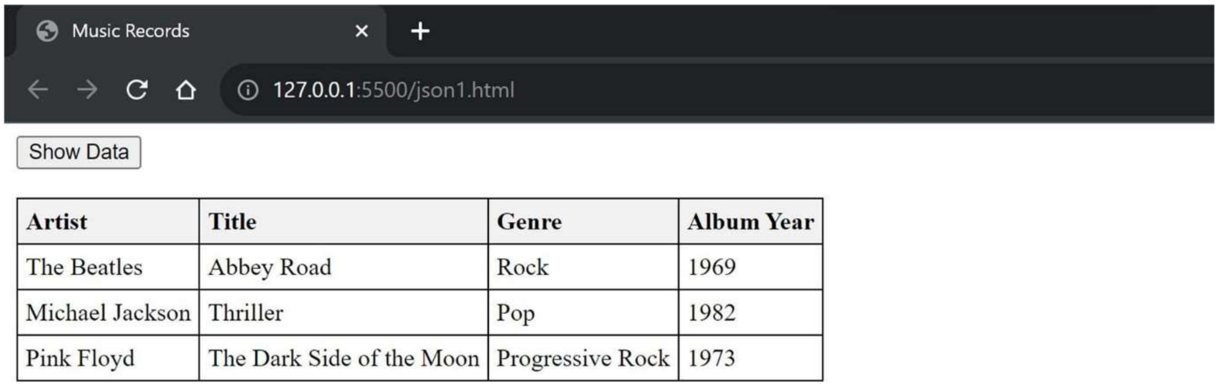
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Music Records</title>
    <style>
      table,th,
      td {
        border: 1px solid black;
        border-collapse: collapse;
      }
      th,
      td {
        padding: 5px;
        text-align: left;
      }
      th {
        background-color: #f2f2f2;
      }
    </style>
  </head>
  <body>
    <button type="button" onclick="loadMusicData()">Show Data</button>

    <br /><br />

    <table id="musicTable">
      <thead>
        <tr>
```

```
          <th>Artist</th>
          <th>Title</th>
          <th>Genre</th>
          <th>Album Year</th>
        </tr>
      </thead>
      <tbody>
        <!-- Data will be inserted here -->
      </tbody>
    </table>

    <script>
      function loadMusicData() {
        var xhr = new XMLHttpRequest();
        xhr.onreadystatechange = function () {
          if (this.readyState == 4 && this.status == 200) {var
            musicData = JSON.parse(this.responseText);
            populateTable(musicData);
          }
        };
        xhr.open("GET", "music_records.json", true);
        xhr.send();
      }

      function populateTable(data) {
        var table = document
          .getElementById("musicTable")
          .getElementsByTagName("tbody")[0];
        table.innerHTML = "";

        data.forEach(function (record) {
          var row = table.insertRow();
          row.insertCell(0).textContent = record.artist;
          row.insertCell(1).textContent = record.title;
          row.insertCell(2).textContent = record.genre;
          row.insertCell(3).textContent = record.albumYear;
        });
      }
    </script>
  </body>
</html>
```

music_records.json

```
[
   {
     "artist": "The
     Beatles","title":
     "Abbey Road",
     "genre": "Rock",
     "albumYear": "1969"
   },
   {
     "artist": "Michael
     Jackson","title":
     "Thriller",
     "genre": "Pop",
     "albumYear":
     "1982"
   },
   {
     "artist": "Pink Floyd",
     "title": "The Dark Side of the
     Moon","genre": "Progressive
     Rock", "albumYear": "1973"
   }
]
```

OUTPUT:



Artist	Title	Genre	Album Year
The Beatles	Abbey Road	Rock	1969
Michael Jackson	Thriller	Pop	1982
Pink Floyd	The Dark Side of the Moon	Progressive Rock	1973

Practical: 7

Aim: **Implement the web applications using PHP and add the AJAX feature into it.**

Requirement: Laptop /Computer

Software Requirement: -

- Windows 10/11 / Browser (Chrome, Brave, Firefox) / Winget

- PHP / XAMPP / VS code / Notepad

Knowledge Requirement: HTML, CSS, PHP, XML

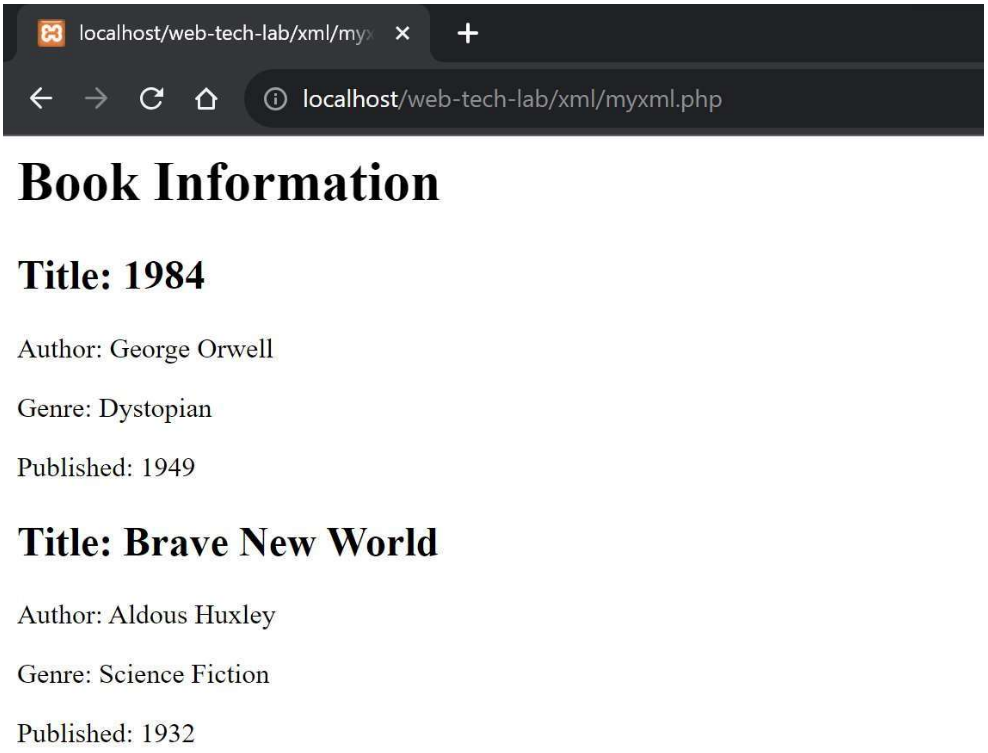
myxml.php

```
<?php
// Load and parse the XML file
$xml = simplexml_load_file('data.xml');

// Check if the XML was loaded successfullyif ($xml
=== false) {
        echo '<p>Failed to load XML data.</p>';
} else {
        // Output the information
        echo '<h1>Book Information</h1>';
        foreach ($xml->book as $book) {
                echo '<div class="book">';
                echo '<h2>Title: ' . htmlspecialchars($book->title) . '</h2>';echo '<p>Author: ' .
                htmlspecialchars($book->author) . '</p>';echo '<p>Genre: ' .
                htmlspecialchars($book->genre) . '</p>'; echo '<p>Published: ' .
                htmlspecialchars($book->published) .
'</p>';
                echo '</div>';
        }
}
?>
```

data.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<library>
    <book>
        <title>1984</title>
        <author>George Orwell</author>
        <genre>Dystopian</genre>
        <published>1949</published>
    </book>
    <book>
        <title>Brave New World</title>
        <author>Aldous Huxley</author>
        <genre>Science Fiction</genre>
        <published>1932</published>
    </book>
</library>
```

OUTPUT:


localhost/web-tech-lab/xml/my
localhost/web-tech-lab/xml/myxml.php

Book Information

Title: 1984

Author: George Orwell

Genre: Dystopian

Published: 1949

Title: Brave New World

Author: Aldous Huxley

Genre: Science Fiction

Published: 1932

Practical: 8

Aim: Implement the web applications using JSP.

Requirement: Laptop /Computer

Software Requirement: -

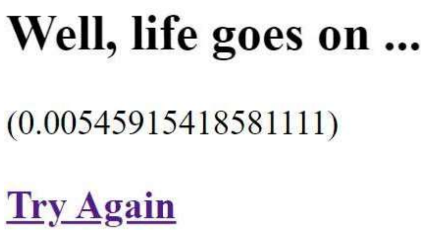
- Windows 10/11 / Browser (Chrome, Brave, Firefox) / Winget

- JSP / XAMPP / VS code / Notepad

Knowledge Requirement: HTML, CSS, JSP

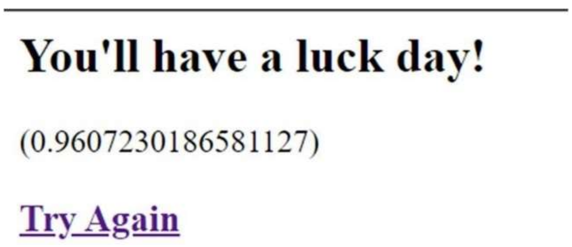
page.jsp

```
<html>
    <head>
        <title>JSP-1</title>
    </head>

    <body>
        <% double num=Math.random(); if (num> 0.95) { %>
        <h2>You'll have a luck day!</h2>
        <p>(<%= num %>)</p>
        <% } else { %>
        <h2>Well, life goes on ...</h2>
        <p>(<%= num %>)</p>
        <% } %>
        <a href="<%= request.getRequestURI() %>">
            <h3>Try Again</h3>
        </a>
    </body>
</html>
```



Well, life goes on ...

(0.005459154185811111)

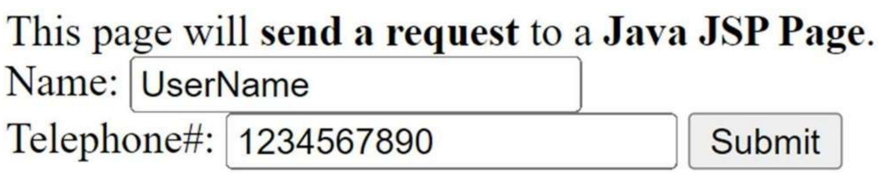


You'll have a luck day!

(0.9607230186581127)

Index.html

```
<meta charset="ISO-8859-1" />
<title>This page will send a request to a Java JSP Page.</title>

<a> This page will <b>send a request</b> to a <b>Java JSPPage</b>. </a>
<p style="margin-top: 0; margin-bottom: 0"></p>

<form action="login.jsp">
    Name: <input type="text" name="name" />
    <br />
    Telephone#: <input type="text" name="telephone" />
    <input type="submit" value="Submit" />
</form>
```



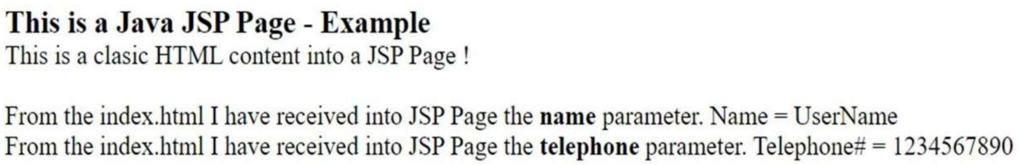
This page will **send a request** to a **Java JSP Page**.

Name: UserName

Telephone#: 1234567890

Login.jsp

```
<title>This is a Java JSP Page - Example</title>

<b><big>This is a Java JSP Page - Example</big></b>
<br>
This is a clasic HTML content into a JSP Page !

<br><br>
From the index.html I have received into JSP Pagethe <b>name</b>
parameter. Name =
<%= request.getParameter("name") %>

<br>
From the index.html I have received into JSP Pagethe <b>telephone</b>
parameter. Telephone# =
<%= request.getParameter("telephone") %>
```



This is a Java JSP Page - Example
This is a clasic HTML content into a JSP Page !

From the index.html I have received into JSP Page the **name** parameter. Name = UserName
From the index.html I have received into JSP Page the **telephone** parameter. Telephone# = 1234567890