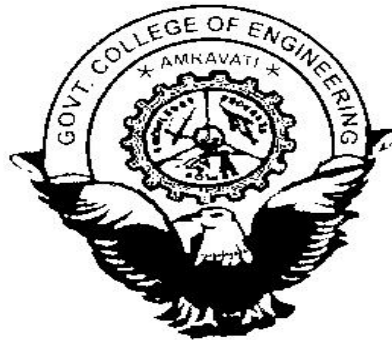


DEPARTMENT OF INFORMATION TECHNOLOGY

Web & Internet Technology Laboratory

Laboratory Manual

VI Sem B.Tech



FACULTY NAME

Prof. Bhushan Wakode

DEPARTMENT OF INFORMATION TECHNOLOGY

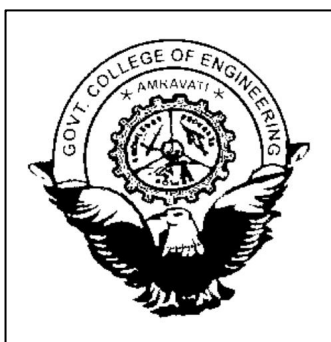
**GOVERNMENT COLLEGE OF ENGINEERING,
AMRAVATI**

GOVERNMENT COLLEGE OF ENGINEERING, AMRAVATI

(An Autonomous Institute of Govt. of Maharashtra)

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE



This is to certify that this lab record contains the bonafide lab work of Vaishnavi Bhoyar having ID **19007019** of semester VI of B.Tech in **Information Technology** during academic year 2021-22 for the,

Course Title:-**web and Internet Technology lab**

Course Code:-**ITU629**

Date:-

Faculty
(Prof. Bhushan Wakode)

INDEX

Course Web & Internet Technology Laboratory

Course Code :ITU629

Sr. No.	Name of Practical	Date of Performance	Pg no	Remark
1	Creating a web page using HTML tags.			
2	Creating a web page with all types of Cascading Style Sheet.			
3	Client side scripts for validating web page form controls using DHTML.			
4	Write program in Java to create three-tier application using JSP and databases <ul style="list-style-type: none">• For conducting online examination• For displaying student marklist			
5	Write programs using XML Schema – XSLT/XLS			
6	Write program using AJAX			
7	Implementing python's Flask frame work to host a site on internet.			
8	Design the following static web pages required for an online book store web site. <ul style="list-style-type: none">1) HOME PAGE: The static home page must contain three frames.2) LOGIN PAGE3) CATOLOGUE PAGE: The catalogue page should contain the details of all the books available in the web site in a table.4) REGISTRATION PAGE			
9	Write <i>JavaScript</i> to validate the following fields of the Registration page. <ul style="list-style-type: none">1. First Name (Name should contains alphabets and the length should not be less than 6 characters).			
	2. Password (Password should not be			

	<p>less than 6 characters length).</p> <p>3. E-mail id (should not contain any invalid and must follow the standard pattern name@domain.com)</p> <p>4. Mobile Number (Phone number should contain 10 digits only).</p> <p>5. Last Name and Address (should not be Empty).</p>			
10	Develop and demonstrate the usage of inline, internal and external style sheet using CSS.			
11	Develop and demonstrate JavaScript with POP-UP boxes and functions .			
12	Write an HTML page that contains a selection box with a list of 5 countries. When the user selects a country, its capital should be printed next in the list. Add CSS to customize the properties of the font of the capital (color,bold and font size).			
13	Write an HTML page including any required JavaScript that takes a number from text field in the range of 0 to 999 and shows it in words. It should not accept four and above digits, alphabets and special characters.			
14	Develop and demonstrate PHP Script for the following problems: a) Write a PHP Script to find out the Sum of the Individual Digits. b) Write a PHP Script to check whether the given number is Palindrome or not			
15	Create an XML document that contains 10 users information. Write a Java Program, which takes User Id as input and returns the user details by taking the user information from XML document using DOM parser or SAX parser.			
16	Modify the above PHP program to use an xml instead of database			

Signature of Faculty

Experiment No. 1

Aim : Creating a web page using HTML tags.

Theory:

What is HTML ?

HTML (Hypertext Markup Language) is a text-based approach to describing how content contained within an HTML file is structured. This markup tells a web browser how to display the text, images and other forms of multimedia on a webpage.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as `` and `<input />` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

Tags used in HTML:

- **Body tag**

Description

The HTML **<body> tag** defines the main content of the HTML document or the section of the HTML document that will be directly visible on your web page. This tag is also commonly referred to as the `<body>` element.

Syntax

In HTML, the syntax for the **<body> tag** is:

```
<body>
</body>
```

- **Title tag**

Description

The HTML `<title>` tag is used for indicating the title of the HTML document. The body title is placed between the `<head>` and the `</head>` tags.

HTML document title is visible via browser's title bar.

Example

```
<head>
  <title>Title comes here</title>
```

```
</head>
```

- **Head tag**

Description

The HTML `<head>` tag is used for indicating the head section of the HTML document. Tags included inside head tags are not displayed on browser window.

Example

```
<html>

  <head>

    <title>HTMLs head Tag</title>

  </head>

  <body>

    actual content goes here

  </body>

</html>
```

- **Font tag**

Fonts play a very important role in making a website more user friendly and increasing content readability. Font face and color depends entirely on the computer and browser that is being used to view your page but you can use HTML `` tag to add style, size, and color to the text on your website. You can use a `<basefont>` tag to set all of your text to the same size, face, and color.

The font tag is having three attributes called **size**, **color**, and **face** to customize your fonts. To change any of the font attributes at any time within your webpage, simply use the `` tag. The text that follows will remain changed until you close with the `` tag. You can change one or all of the font attributes within one `` tag.

```
<body>

  <font size = "1">Font size = "1"</font><br />

  <font size = "2">Font size = "2"</font><br />

</body>
```

Bold tag

The `` tag should be used to markup text as bold without conveying any extra importance, for example in article abstracts, where the beginning of an article is set in bold text. It should *not* be used to convey extra importance. To convey extra importance, use the `` tag. To emphasize text, use the `` tag.

Syntax

The `` tag is written as `` with the text to be bold inserted between the start and end tags.

Example:

```
<body>
<b>bold text</b>
</body>
```

- **<h> tag**

Description

The HTML `<h1>` to `<h6>` tag is used to define headings in an HTML document. `<h1>` defines largest heading and `<h6>` defines smallest heading.

Example

```
<body>

  <h1>Around the World</h1>

  <h2>Asian Countries</h2>

  <h3>India</h3>

</body>
```

- **Image tag**

Description

The HTML `` tag is used to put an image in an HTML document.

Example

```
<body>

  <img src = "https://www.tutorialspoint.com/images/html.gif"

    alt = "HTML Tutorial" height = "150" width = "140" />

</body>
```

- **P tag**

Description

The HTML `<p>` tag defines a paragraph of text.

Example

```
<head>

  <title>HTML p Tag</title>

</head>

<body>

  <p>This paragraph is defined using the HTML p tag</p>

</body>
```

<A> tag

The <a> tag defines a hyperlink, which is used to link from one page to another.

The most important attribute of the <a> element is the href attribute, which indicates the link's destination.

Program:

```
<html>
<head><title>PG1</title></head>
<body background color="pink">
<marquee behavior="Alternate" bgcolor="yellow"><h1><FONT face="Algerian"
COLOR="PURPLE" SIZE="355">GRAPHICS DESIGNING</FONT></h1></marquee>

<center></center>
<p><center><FONT COLOR="BLACK" size="350">

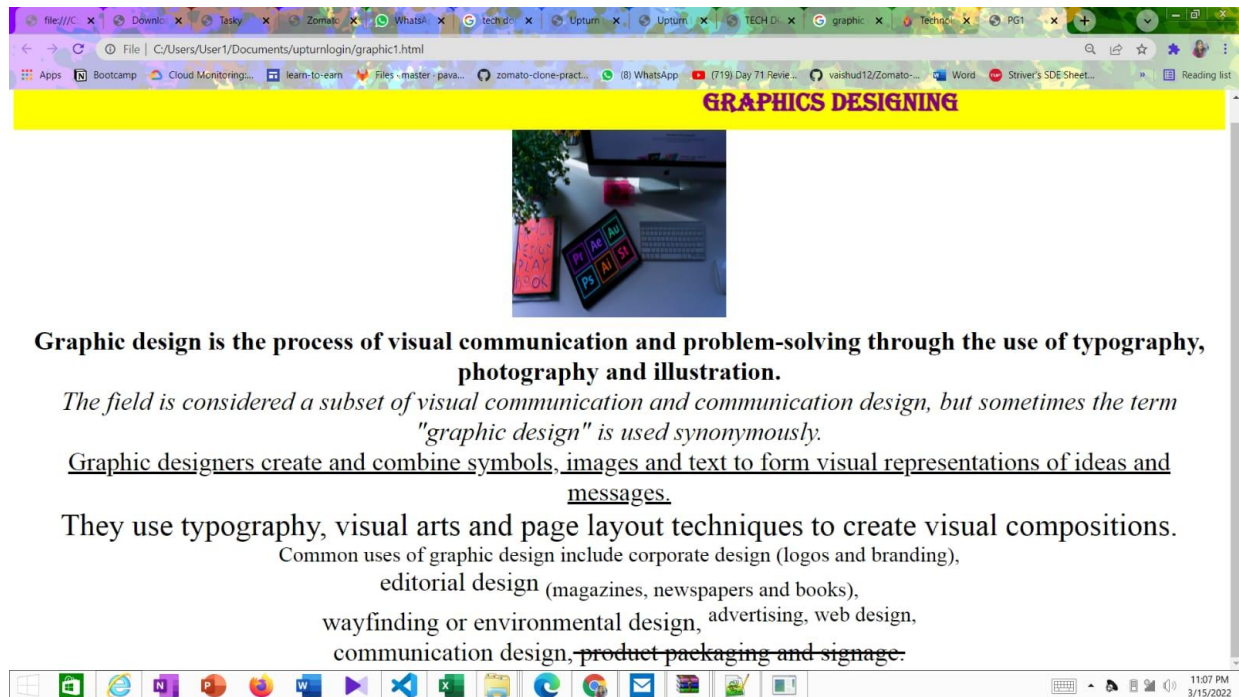
<B>Graphic design is the process of visual communication and problem-solving through the use of
typography, photography and illustration.</B><BR>

<i>The field is considered a subset of visual communication and communication design, but
sometimes the term "graphic design" is used synonymously.</i><br>
<u>Graphic designers create and combine symbols, images and text to form visual representations of
ideas and messages.</u><br>

<big>They use typography, visual arts and page layout techniques to create visual
compositions.</big><br>

<small>Common uses of graphic design include corporate design (logos and branding),</small><br>
editorial design <sub>(magazines, newspapers and books),</sub><br>
wayfinding or environmental design, <sup>advertising, web design,</sup><br>
communication design,<strike> product packaging and signage.</strike><br></center></font>
</h2>

</body>
</html>
```

Graphic design is the process of visual communication and problem-solving through the use of typography, photography and illustration.

The field is considered a subset of visual communication and communication design, but sometimes the term "graphic design" is used synonymously.

Graphic designers create and combine symbols, images and text to form visual representations of ideas and messages.

They use typography, visual arts and page layout techniques to create visual compositions.

Common uses of graphic design include corporate design (logos and branding),

editorial design (magazines, newspapers and books),

wayfinding or environmental design, advertising, web design,

communication design, ~~product packaging and signage.~~

Conclusion: Thus, we have created a web page using html.

Experiment No. 02

AIM: Creating a web page with all types of Cascading Style Sheet.

Theory:

Styling HTML with CSS

CSS stands for Cascading Style Sheets.

CSS describes how HTML elements are to be displayed on screen, paper, or in other media.

CSS saves a lot of work. It can control the layout of multiple web pages all at once.

CSS can be added to HTML elements in 3 ways:

- **Inline** - by using the style attribute in HTML elements
- **Internal** - by using a `<style>` element in the `<head>` section
- **External** - by using an external CSS file

The most common way to add CSS, is to keep the styles in separate CSS files. However, here we will use inline and internal styling, because this is easier to demonstrate, and easier for you to try it yourself.

Inline Cascading Style Sheet:

An inline CSS is used to apply a unique style to a single HTML element.

An inline CSS uses the style attribute of an HTML element.

This example sets the text color of the `<h1>` element to blue:

```
<h1 style="color:blue;">This is a Blue Heading</h1>
```

Internal Cascading Style Sheet:

An internal CSS is used to define a style for a single HTML page.

An internal CSS is defined in the `<head>` section of an HTML page, within a `<style>` element:

External Cascading Style Sheet:

An external style sheet is used to define the style for many HTML pages.

With an external style sheet, you can change the look of an entire web site, by changing one file!

To use an external style sheet, add a link to it in the `<head>` section of the HTML page:

Procedure:

STEP 1 : Start

STEP 2 : Open Notepad.

STEP 3 : Write HTML code using different HTML tags with Cascading Style Sheet.

STEP 4 : Save the notepad document with .html extension.

STEP 5 : Open (.html) file in web browser to see web page.

STEP 6 : Stop

Code:

```
<!doctype html>

<html lang="en">

  <head>

    <!-- Required meta tags -->

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Ecommerce Product Card</title>

    <!--CSS link-->

    <link rel="stylesheet" href="style.css">

    <!--Font CDN Link-->

    <link      rel="stylesheet"      href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css">

  </head>

  <body>

    <div class="container">

      <div class="navbar_container">

        <div class="logo_container">

        </div>

        <div class="navlinks" id="navlinks_container">

          <div class="navlinks">

            <a href="#" class="active">Home</a>

            <a href="#">Services</a>

            <a href="#">Blog</a>

            <a href="#">About</a>
```

```
<a href="#">FAQ</a>

</div>

</div>

<div class="button">

  <a href="" class="search"><i class="fa fa-search"></i></a>

</div>

</div>

<div class="banner">

  <div class="left">

    <div class="left_inner_container">

      <h2><span>Subscribe Now</span><br>to Our Newsletter</h2>

      <p class="info">Lorem ipsum dolor sit amet consectetur adipisicing elit. Doloribus id adipisci, deleniti ab
pariatur delectus, totam tempore accusamus beatae cupiditate culpa voluptas.</p>

      <div class="box">

        <form>

          <input type="text" placeholder="Name">

          <input type="email" placeholder="Email">

          <button class="button2" type="button" name="button">Send<i class="fa fa-caret-right"></i></button>

        </form>

      </div>

    </div>

  </div>

</div>

<div class="right">

  <div class="right_inner_container">

  </div>

</div>

</div>

<div class="bottomline"><div class="minline"></div></div>

</div>

</body>

</html>
```

Css:

```
*{  
  
  margin: 0;  
  
  padding: 0;  
  
  box-sizing: border-box;  
  
}  
  
body{  
  
  font-family: 'Be vietnam', sans-serif;  
  
  overflow: hidden;  
  
}  
  
html{  
  
  font-size: 62.5%;  
  
}  
  
.container{  
  
  background: #20364b;  
  
  position: relative;  
  
}  
  
.navbar_container{  
  
  width: 100%;  
  
  position: absolute;  
  
  height: 10rem;  
  
  background: transparent;  
  
  display: flex;  
  
  justify-content: space-around;  
  
  align-items: center;  
  
  padding: 0% 10% 0% 10%;  
  
}  
  
.logo_container{  
  
  width: 15%;  
  
  display: flex;  
  
  justify-content: center;  
  
  align-items: center;  
  
}  
  
.navlinks{
```

```
font-size: 2rem;

width: 100%;

text-align: right;
}

.navlinks a{

text-decoration: none;

font-size: 1.5rem;

color: #fff;

padding-right: 40px;

font-weight: 600;
}

.navlinks a.active,

.navlinks a:hover{

color: #eab31d;
}

.search{

background: #fdc11f;

width: 40px;

height: 40px;

border-radius: 50%;
}

.search i{

position: relative;

font-size: 16px;

background: #fdc11f;

padding: 10px;

border-radius: 50%;

color: #253a49;
}

.button{

margin-top: 8px;

background-color: #70683b;

padding: 7px;

border-radius: 50%;
```

```
}

.banner{

  height: 70rem;

  display: flex;

  width: 100%;

  justify-content: center;

}

.left, .right{

  height: 100%;

  width: 50%;

  display: flex;

  justify-content: flex-end;

  align-items: flex-end;

}

.right{

  justify-content: flex-start;

  background-image: url(image/right1.png);

  background-position: center left;

  background-repeat: no-repeat;

  max-width: 100%;

}

.left_inner_container{

  height: 75%;

  display: block;

  flex-direction: column;

  justify-content: space-between;

  align-items: flex-start;

  width: 80%;

}

.left_inner_container h2{

  font-size: 37pt;

  width: 49rem;

  line-height: 42pt;

  color: #fff;
```

```
}

.left_inner_container h2 span{

  color: #fdc11f;

}

.info{

  margin-top: 20px;

  font-size: 11pt;

  width: 40rem;

  line-height: 14pt;

  word-spacing: 2px;

  color: #fff;

}

.right_inner_container{

  height: 77%;

  display: flex;

  flex-direction: column;

  justify-content: space-between;

  align-items: flex-start;

  width: 80%;

}

.box input,

.box button{

  display: block;

}

.box{

  margin-top: 20px;

}

.box input{

  width: 290px;

  height: 40px;

  border-radius: 20px;

  outline: none;

  border: none;
```



```
margin-bottom: 25px;

padding-left: 30px;
}

.box input::placeholder{

font-weight: light;

font-size: 11pt;
}

.box .button2{

width: 150px;

height: 50px;

border-radius: 56px;

outline: none;

border: none;

background: #fdc11f;

font-size: 18px;

color: #20364b;
}

.button2 i{

margin-left: 5px;
}

#intro{

width: 10%;

margin-left: 40px;

margin-top: 20px;
}

#intro1{

width: 10%;

margin-bottom: 173px;

margin-left: -73px;
}

.
```

```

bottomline{

width: 150px;

height: 5px;

background-color: #445668;

display: inline-block;

margin: auto;

position: absolute;

bottom: 90px;

left: 44%;

text-align: center;

border-radius: 10px;

}

.minline{

display: inline-block;

background: #fdc11f;

width: 40px;

height: 5px;

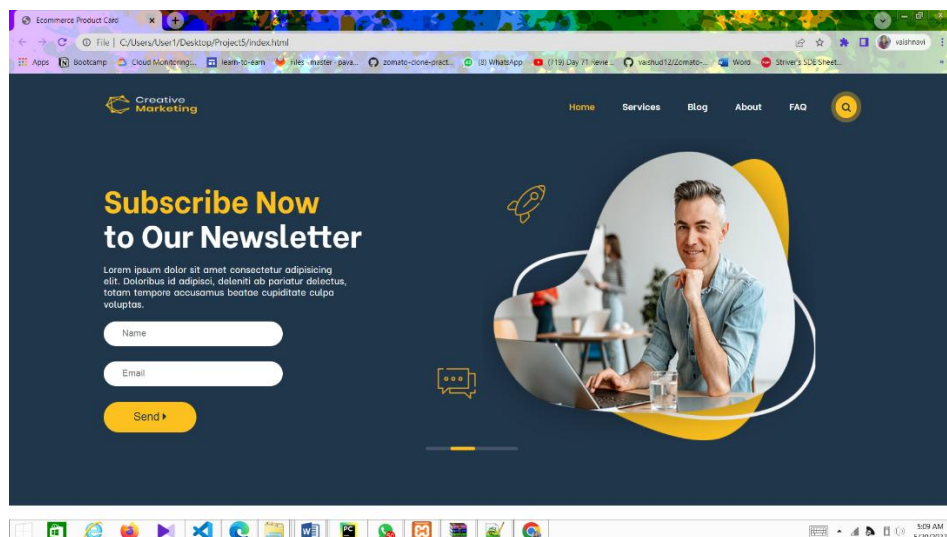
position: absolute;

border-radius: 10px;

left: 40px;

}

```



CONCLUSION: thus successfully implemented html web site using css

Experiment No. 03

AIM: Client side scripts for validating web page form controls using DHTML.

THEORY:

DHTML INTRODUCTION:

DHTML stands for Dynamic HTML, it is totally different from HTML. The browsers which support the dynamic HTML are some of the versions of Netscape Navigator and Internet Explorer of version higher than 4.0. The DHTML is based on the properties of the HTML, javascript, CSS, and DOM (Document Object Model which is used to access individual elements of a document) which helps in making dynamic content. It is the combination of HTML, CSS, JS, and DOM. The DHTML make use of Dynamic object model to make changes in settings and also in properties and methods. It also makes uses of Scripting and it is also part of earlier computing trends.

DHTML allows different scripting languages in a web page to change their variables, which enhance the effects, looks and many others functions after the whole page have been fully loaded or under a view process, or otherwise static HTML pages on the same. But in true ways, there is noting that as dynamic in DHTML, there is only the enclosing of different technologies like CSS, HTML, JS, DOM, and different sets of static languages which make it as dynamic.

DHTML is used to create interactive and animated web pages that are generated in real-time, also known as dynamic web pages so that when such a page is accessed, the code within the page is analyzed on the web server and the resulting HTML is sent to the client's web browser.

Key Features:

Following are the some major key features of DHTML:

- Tags and their properties can be changed using DHTML.
- It is used for real-time positioning.
- Dynamic fonts can be generated using DHTML.
- It is also used for data binding.
- It makes a webpage dynamic and be used to create animations, games, applications along with providing new ways of navigating through websites.
- The functionality of a webpage is enhanced due to the usage of low-bandwidth effect by DHTML.
- DHTML also facilitates the use of methods, events, properties, and codes.

Why Use DHTML?

DHTML makes a webpage dynamic but Javascript also does, the question arises that what different does DHTML do? So the answer is that DHTML has the ability to change a webpages look, content and style once the document has loaded on our demand without changing or deleting everything already existing on the browser's webpage. DHTML can change the content of a webpage on demand without the browser having to erase everything else, i.e. being able to alter changes on a webpage even after the document has completely loaded.

Advantages:

- Size of the files are compact in compared to other interractional media like Flash or Shockwave, and it downloads faster.
- It is supported by big browser manufacturers like Microsoft and Netscape.
- Highly flexible and easy to make changes.

- Viewer requires no extra plug-ins for browsing through the webpage that uses DHTML, they do not need any extra requirements or special software to view it.
- User time is saved by sending less number of requests to the server. As it is possible to modify and replace elements even after a page is loaded, it is not required to create separate pages for changing styles which in turn saves time in building pages and also reduces the number of requests that are sent to the server.
- It has more advanced functionality than a static HTML. It is capable of holding more content on the web page at the same time.

Disadvantages:

- It is not supported by all the browsers. It is supported only by recent browsers such as Netscape 6, IE 5.5, and Opera 5 like browsers.
- Learning of DHTML requires a lot of pre-requisites languages such as HTML, CSS, JS, etc should be known to the designer before starting with DHTML which is a long and time-consuming in itself.
- Implementation of different browsers are different. So if it worked in one browser, it might not necessarily work the same way in another browser.
- Even after being great with functionality, DHTML requires a few tools and utilities that are some expensive. For example, the DHTML text editor, Dreamweaver. Along with it the improvement cost of transferring from HTML to DHTML makes cost rise much higher.

Difference between HTML and DHTML:

- HTML is a markup language while DHTML is a collection of technologies.
- HTML is used to create static webpages while DHTML is capable of creating dynamic webpages.
- DHTML is used to create animations and dynamic menus but HTML not used.
- HTML sites are slow upon client-side technologies whereas DHTML sites are comparatively
-

faster.

- Web pages created using HTML are rather simple and have no styling as it uses only one language whereas DHTML uses HTML, CSS, and Javascript which results in a much better and way more presentable webpage.
- HTML cannot be used as server side code but DHTML used as server side code.
- DHTML needs database connectivity but not in case of HTML.
- Files in HTML are stored using .htm or .html extension while DHTML uses .dhtm extension.
- HTML requires no processing from the browser but DHTML does.

Procedure:

STEP 1 : Start

STEP 2 : Open Notepad.

STEP 3 : Write HTML code using different HTML tags and make use of function using

Javascript.

STEP 4 : Save the notepad document with .html extension.

STEP 5 : Open (.html) file in web browser to see web page.

STEP 6 : Stop

Code:

```
<html>

  <title>registration form</title>

  <link rel="stylesheet" type="text/css" href="style.css">

  <div class="container">

    <h1>REGISTRATION FORM</h1>

    <form name="registration" class="registartion-form" onsubmit="return
formValidation()">

      <table>

        <tr>

          <td><label for="name">Name:</label></td>

          <td><input type="text" name="name" id="name" placeholder="your name"></td>

        </tr>

        <tr>

          <td><label for="email">Email:</label></td>

          <td><input type="text" name="email" id="email" placeholder="your email"></td>

        </tr>

        <tr>

          <td><label for="password">Password:</label></td>

          <td><input type="password" name="password" id="password"></td>

        </tr>

        <tr>

          <td><label for="phoneNumber">Phone Number:</label></td>

          <td><input type="number" name="phoneNumber" id="phoneNumber"></td>

        </tr>

      </table>

    </form>

  </div>

</html>
```

```
<tr>

<td><label for="gender">Gender:</label></td>

<td>Male: <input type="radio" name="gender" value="male">

Female: <input type="radio" name="gender" value="female">

Other: <input type="radio" name="gender" value="other"></td>

</tr>
```

```
<tr>

<td><label for="Course">Course</label></td>

<td>

<select name="language" id="course">

<option value="BTech">Btech</option>

<option value="BCA">BCA</option>

<option value="Bpharma">Bpharma</option>

<option value="BSC">BSC</option>

<option value="Mtech">Mtech</option>

</select>

</td>
```

```
<tr>

<td><label for="Course Name">Department Name:</label></td>

<td><input type="text" name="Department Name" id="Department
name" placeholder="Department Name"></td>

</tr>
```

```
<tr>

<td><label for="about">About:</label></td>
```

```

        <td><textarea name="about" id="about" placeholder="Write about
yourself..."></textarea></td>

    </tr>

    <tr>

        <td colspan="2"><input type="submit" class="submit" value="Register" ></td>

    </tr>

    <script language="javascript">

        onclick="Register" (alert("sucessfully"));

    </script>

</table>

</form>

</div>

<script src="script.js"></script>

</html>

```

Javascript:

```

// Select all input elements for varification

const name = document.getElementById("name");

const email = document.getElementById("email");

const password = document.getElementById("password");


const phoneNumber = document.getElementById("phoneNumber");

const gender = document.registration;

const CourseName = document.getElementById("CourseName");

const DepartmentName= document.getElementById("DepartmentName");

const Register= document.getElementById("Register");

// function for form varification

function formValidation() {

```



```
// checking name length

if (name.value.length < 2 || name.value.length > 20) {

    alert("Name length should be more than 2 and less than 21");

    name.focus();

    return false;

}

// checking email

if (email.value == ""){

    window.alert(

        "Please enter a valid e-mail address.");

    email.focus();

    return true;

}

// checking password

if (!password.value.match(/^.{5,15}$/)) {

    alert("Password length must be between 5-15 characters!");

    password.focus();

    return false;

}

// checking phone number

if (!phoneNumber.value.match(/^[1-9][0-9]{9}$/)) {

    alert("Phone number must be 10 characters long number and first digit can't be 0!");

    phoneNumber.focus();

    return false;

}

// checking gender

if (gender.value === "") {

    alert("Please select your gender!");
```

```
        return false;

    }

    // checking CourseName

    if (CourseName.value == ""){

        alert("Please select your CourseName!");

        return false;

    }


    if (Register.value != ""){

        alert("Form Submitted Successfully");

        return true;

    }

}
```

Css:

```
* {

    margin: 0

}


.container {

    display: flex;


    justify-content: center;

    align-items: center;

    flex-direction: column;

    height: 100vh;

    background-color: #52e6a6;
```

```
}
```

```
.container h1 {  
  color: rgb(27, 13, 13);  
  font-family: sans-serif;  
  margin: 20px;  
}
```

```
.registartion-form {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  width: 600px;  
  color: rgb(255, 255, 255);  
  font-size: 18px;  
  font-family: sans-serif;  
  background-color: #9efade;  
  padding: 20px;  
}
```

```
.registartion-form input,  
.registartion-form select,  
.registartion-form textarea {  
  border: none;  
  padding: 5px;  
  margin-top: 10px;  
  font-family: sans-serif;
```

```
}
```

```
.registartion-form input:focus,
```

```
.registartion-form textarea:focus {
```

```
    box-shadow: 3px 3px 10px rgb(242, 239, 239), -3px -3px 10px rgb(224, 224, 224);
```

```
}
```

```
.registartion-form .submit {
```

```
    width: 100%;
```

```
    padding: 8px 0;
```

```
    font-size: 20px;
```

```
    color: rgb(44, 44, 44);
```

```
    background-color: #ba5c5c;
```

```
    border-radius: 5px;
```

```
}
```

```
.registartion-form .submit:hover {
```

```
    box-shadow: 3px 3px 6px rgb(255, 214, 176);
```

```
}
```

The screenshot shows a web browser window with a registration form titled "REGISTRATION FORM". The form is set against a green background. It contains several input fields: "Name" (filled with "vaishnavi devardekar"), "Email" (filled with "vaishu2397@gmail.com"), "Password" (empty), "Phone Number" (empty), "Gender" (radio buttons for Male, Female, and Other, with Female selected), "Course" (a dropdown menu showing "Btech"), "Department Name" (filled with "Department Name"), and "About" (a text area with the placeholder "Write about yourself..."). A red "Register" button is at the bottom. A white error message box is displayed, stating "This page says Password length must be between 5-15 characters!". The browser's address bar shows a long URL. The Windows taskbar is visible at the bottom with the date 4/5/2022 and time 11:40 PM.

Concl

usion: Thus we have validated web page form controls using DHTML.

Experiment No. 04

AIM: Write programs in java to create three-tier application using JSP and databases:

- For conducting online examination
- For displaying student mark list.

THEORY:

Java Server Pages (JSP):

Java Server Pages (JSP) is a server-side programming technology that enables the creation of dynamic, platform-independent method for building Web-based applications. JSP have access to the entire family of Java APIs, including the JDBC API to access enterprise databases.

Why Use JSP?:

Java Server Pages often serve the same purpose as programs implemented using the Common Gateway Interface (CGI). But JSP offers several advantages in comparison with the CGI.

- Performance is significantly better because JSP allows embedding Dynamic Elements in HTML Pages itself instead of having separate CGI files.

- JSP are always compiled before they are processed by the server unlike CGI/Perl which requires the server to load an interpreter and the target script each time the page is requested.
- Java Server Pages are built on top of the Java Servlets API, so like Servlets, JSP also has access to all the powerful Enterprise Java APIs, including JDBC, JNDI, EJB, JAXP, etc.
- JSP pages can be used in combination with servlets that handle the business logic, the model supported by Java servlet template engines.

Finally, JSP is an integral part of Java EE, a complete platform for enterprise class applications. This means that JSP can play a part in the simplest applications to the most complex and demanding.

Three-tier architecture:

A three-tier architecture is any system, which enforces a general separation between the following three parts:

1. Client Tier or user interface
2. Middle Tier or business logic
3. Data Storage Tier

Applied to web applications and distributed programming, the three logical tiers usually correspond to the physical separation between three types of devices or hosts:

1. Browser or GUI Application
2. Web Server or Application Server
3. Database Server (often an RDBMS or Relational Database)

However, inside of the application server, there is a further division of program code into three logical tiers. This is kind of fractal: the part (app server object design) resembles the whole (physical system architecture). In a classic JSP/Servlet system, these objects are usually implemented as:

1. JSPs or Servlets responsible for creating HTML or WML user interface pages
2. Servlets or JavaBeans responsible for business logic
3. Servlets, JavaBeans, or Java classes responsible for data access. These objects usually use JDBC to query the database.

Code:

Newjsp.jsp:

```
<%--
```

Created on : Feb 23, 2011, 7:19:15 PM

Author : A

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%> <!DOCTYPE
HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Examination Panel</title>

</head>

<body bgcolor="cyan">

<%@ page language="java" %>

<%@ page import ="java.sql.*" %>

<%

String reg= request.getParameter("txt_reg"); String name =
request.getParameter("txt_name"); out.println("<h2>Welcome" + name +
"...Your Register number is " + reg + "!!</h2>

");

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver"); String sTable = "exam";

String sSql = "SELECT * FROM " + sTable + "";

String database = "jdbc:odbc:sDBQ";

Connection cn = null;

Statement st = null;

ResultSet rs = null;

try {

cn = DriverManager.getConnection(database ,"sa",""); st =
cn.createStatement();

rs = st.executeQuery(sSql);

ResultSetMetaData rsmd = rs.getMetaData();

String s1,s2,s3,s4,

```

int i=1;

while(rs.next())

{

out.println("<form name='exam' action='newjsp2.jsp'
method='post'><b>" + i + " . " +

rs.getString(1) + "</b><br><br>");

s1 = rs.getString(2);

s2 = rs.getString(3);

s3 = rs.getString(4);

s4 = rs.getString(5);

out.println("<input type=radio name=opt"+i+" value="+s1+" CHECKED>" + s1
+" <br><br>");

out.println("<input type=radio name=opt"+i+" value="+s2+">" + s2
+"<br><br>");


out.println("<input type=radio name=opt"+i+" value="+s3+">" + s3
+"<br><br>");

out.println("<input type=radio name=opt"+i+" value="+s4+">" + s4
+"<br><br>"); i++;

}

out.println("<input name ='submit' value='Submit' type='submit'/>"); /*int n
= rsmd.getColumnCount();

out.println( "<table border=1 cellspacing=3><tr>" );

for( int i=1; i<=n; i++ ) // Achtung: erste Spalte mit 1 statt 0

out.println( "<th>" + rsmd.getColumnName( i ) + "</th>" );

while( rs.next() )

{out.println( "</tr><tr>" );

for( int i=1; i<=n; i++ ) // Achtung: erste Spalte mit 1 statt 0 o
ut.println( "<td
nowrap>" + rs.getString( i ) + "</td>" ); }

out.println( "</tr></table>" );*/

}finally {

try {

```



```
if( null != rs )  
  
rs.close();  
  
} catch( Exception ex ) {}  
  
try {  
  
if( null != st )  
  
st.close();  
  
} catch( Exception ex ) {}  
  
try {  
  
if( null != cn )  
  
cn.close();  
  
} catch( Exception ex ) {}  
  
}  
  
%>  
  
</body>
```

```
</html>
```

Newjsp1.jsp:

```
<%--
```

Document : index

Created on : Feb 9, 2011, 6:50:54 PM

Author : A

```
--%>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%> <!DOCTYPE  
HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
```

```
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
<title>Welcome to Online Examination</title>
```

```
</head>
```

```
<body bgcolor="cyan">
```

```

<form name="index" action="newjsp.jsp" method="post">
<center><h1><span><font color="red">Welcome to Online
Examination</font></span></h1>

<br>

<h2><u><span><font color="blue">Instructions to the
Candidates</font></span></u></h2>

<br><h3><ol><li>Fill the correct Registration number.</li> <br><li>Enter your
name.</li> <br><li>Read the questions carefully.</li>

<br><li>No negative marking.</li></ol></h3>

<br>

<b>Enter your Register number</b>

<input type="text" name="txt_reg">

<b>Enter your Name</b>

<input type="text" name="txt_name"><br><br>

<input name="submit" value="Submit" type="submit"/>

</center>

</form>

</body>

</html>

```

Newjsp2.jsp:

```

<%--
Document      : report
Created on : Feb 23, 2011, 9:09:37 PM
Author : A
--%>

<%@page
import="com.sun.java.swing.plaf.windows.resources.windows_es"%>

<%@page contentType="text/html" pageEncoding="UTF-8"%> <!DOCTYPE
HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

```

```

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Exam Report</title>

</head>

<body bgcolor="cyan">

<center><h1>Your Report Card</h1></center> <%@ page
language="java" %> <%@ page import ="java.sql.*" %>

<%

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

String sTable = "exam";

String sSql = "SELECT * FROM " + sTable + "";

String database = "jdbc:odbc:sDBQ";

Connection cn = null;

Statement st = null;


ResultSet rs = null;

try {

cn = DriverManager.getConnection( database ,"sa",""); st =
cn.createStatement();

rs = st.executeQuery( sSql );

ResultSetMetaData rsmd = rs.getMetaData();

String s1,s2,s3,s4;

int i=1;

int correct=0,incorrect=0,total=0;

out.println("<h2><br><br><center><table border=1 cellpadding=2
cellspacing=2><tr><th>Question</th><th>Your Answer</th><th>Correct
Answer</th><th>Status</th></tr>");

while(rs.next())

{

total++;

s1 = rs.getString(1);

```

```

s2 = request.getParameter("opt"+i);

s3 = rs.getString(6);

if(s2.equals(s3))

{ s4="Correct";

correct++;

}

else

{ s4="Incorrect";

incorrect++;

}

out.println("<tr><td>"+s1+"</td><td>"+s2+"</td><td>"+s3+"</td><td>"+s4+"
</td></tr>");

i++;

}

out.println("</table><br><br><table><b><tr><td>Correct
Answers</td><td>"+correct+"</td></tr>");

out.println("<tr><td>Incorrect Answers</td><td>"+incorrect+"</td></tr>");

out.println("<tr><td>Total
Questions</td><td>"+total+"</td></tr></table></b></center></h2>");

}

finally {

try { if( null != rs ) rs.close(); } catch( Exception ex ) {} try { if( null != st )
st.close(); } catch( Exception ex ) {} try { if( null != cn ) cn.close(); }
catch( Exception ex ) {} }

%>

</body>

</html>

```

Output:

Welcome to Online Examination

Instructions to the Candidates

1. Fill the correct Registration number.
2. Enter your name.
3. Read the questions carefully.
4. No negative marking.

Enter your Register number

40108104025

Enter your Name

hema

Submit

Welcomehema...Your Register number is 40108104025!!

1 . what is the capital of india

- ☒ a.chennai
- ☐ b.mumbai
- ☐ c.Delhi
- ☐ d.goa

2 . what is our national flower?

- ☒ a.lotus
- ☐ b.rose
- ☐ c.jasmine
- ☐ d.lily

3 . when india got world cup?

- ☒ a.1983

- ☐ b.1993
- ☐ c.2004
- ☐ d.2011
4. What is our national game?
- ☐ a.tennis
- ☐ b.hockey
- ☐ c.football
- ☐ d.volleyball
5. What is the captain of Indian cricket team?
- ☐ a.ya
- ☐ b.indian
- ☐ c.vishu
- ☐ d.dhoni
6. Which Prime Minister has served the longest?
- ☐ a.Mahatma Gandhi
- ☐ b.Jawahar Lal Nehru
- ☐ c.P. V. Nellothi Reddy
- ☐ d.Mr.Manmohan Singh
-

Your Report Card

Question	Your Answer	Correct Answer	Status
what is the capital of india	c.Delhi	c.Delhi	Correct
what is our national flower?	a.lotus	a.lotus	Correct
when india got world cup?	a.1983	a.1983	Correct
what is our national game?	b.hockey	b.hockey	Correct
What is the captain of Indian cricket team	ys	Ms.dhoni	Incorrect
Which Prime Minister has served the longest?	Mahatma	Jawahar Lal Nehru	Incorrect

Correct Answers 4
 Incorrect Answers 2
 Total Questions 6

Conclusion:-Thus in this practical, we have implemented a program in java to create three-tier application using JSP and databases.

Experiment – 5

Aim:- Write programs using XML Schema- XSLT/XLS

Theory:-

XML:

- XML stands for eXtensible Markup Language.
- XML was designed to store and transport data.
- XML was designed to be both human- and machine-readable.

XML Schema:

- An XML Schema describes the structure of an XML document.
- The XML Schema language is also referred to as XML Schema Definition (XSD).

Example:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

The purpose of an XML Schema is to define the legal building blocks of an XML document:

- the elements and attributes that can appear in a document
- the number of (and order of) child elements
- data types for elements and attributes

- default and fixed values for elements and attributes

One of the greatest strength of XML Schemas is the support for data types.

4. It is easier to describe allowable document content
5. It is easier to validate the correctness of data
6. It is easier to define data facets (restrictions on data)
7. It is easier to define data patterns (data formats)
8. It is easier to convert data between different data types

XSLT:

4. XSL (eXtensible Stylesheet Language) is a styling language for XML.
5. XSLT stands for XSL Transformations.

How XSLT Works:

An XSLT stylesheet is used to define the transformation rules to be applied on the target XML document. XSLT stylesheet is written in XML format. XSLT Processor takes the XSLT stylesheet and applies the transformation rules on the target XML document and then it generates a formatted document in the form of XML, HTML, or text format. This formatted document is then utilized by XSLT formatter to generate the actual output which is to be displayed to the end-user.

XSL:

Before learning XSLT, we should first understand XSL which stands for EXtensible Stylesheet Language. It is similar to XML as CSS is to HTML.

Need for XSL:

In case of HTML document, tags are predefined such as table, div, and span; and the browser knows how to add style to them and display those using CSS styles. But in case of XML documents, tags are not predefined. In order to understand and style an XML document, XSL is needed. An XSL document specifies how a browser should render an XML document.

Following are the main parts of XSL:

4. XSLT – used to transform XML document into various other types of document.
5. XPath – used to navigate XML document.

6. XSL-FO – used to format XML document.

ALGORITHM:

Step 1: Start the program

Step 2: Use Xml Style Sheet code to define link `<?xml-stylesheet type="text/xsl" href="yourxsl.xml"?>`

Step 3: Use the catalog tag to define CD collection details.

Step 4: Use the necessary heading for appropriate XML tag.

Step 5: Provide necessary information for CD collection details

Step 6: Stop the program

Code:-

XML File:

```
<?xml version="1.0" encoding="UTF-8"?>
<breakfast_menu>

<food>
<name>Belgian Waffles</name>
<price>$5.95</price>
<description>Two of our famous Belgian Waffles with plenty of real maple
syrup</description>
<calories>650</calories>
</food>

<food>
<name>Strawberry Belgian Waffles</name>
<price>$7.95</price>
<description>Light Belgian waffles covered with strawberries and whipped
cream</description>
<calories>900</calories>
</food>
```

```

<food>
<name>Berry-Berry Belgian Waffles</name>
<price>$8.95</price>
<description>Light Belgian waffles covered with an assortment of fresh
berries and whipped cream</description>
<calories>900</calories>
</food>

<food>
<name>French Toast</name>
<price>$4.50</price>
<description>Thick slices made from our homemade sourdough
bread</description>
<calories>600</calories>
</food>

<food>
<name>Homestyle Breakfast</name>
<price>$6.95</price>
<description>Two eggs, bacon or sausage, toast, and our ever-popular hash
browns</description>
<calories>950</calories>
</food>

</breakfast_menu>

```

XSLT Stylesheet:

```

<?xml version="1.0" encoding="UTF-8"?>
<html xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<body style="font-family:Arial;font-size:12pt;background-color:#EEEEEE">
<xsl:for-each select="breakfast_menu/food">
  <div style="background-color:teal;color:white;padding:4px">
    <span style="font-weight:bold"><xsl:value-of select="name"/> - </span>
    <xsl:value-of select="price"/>
  </div>

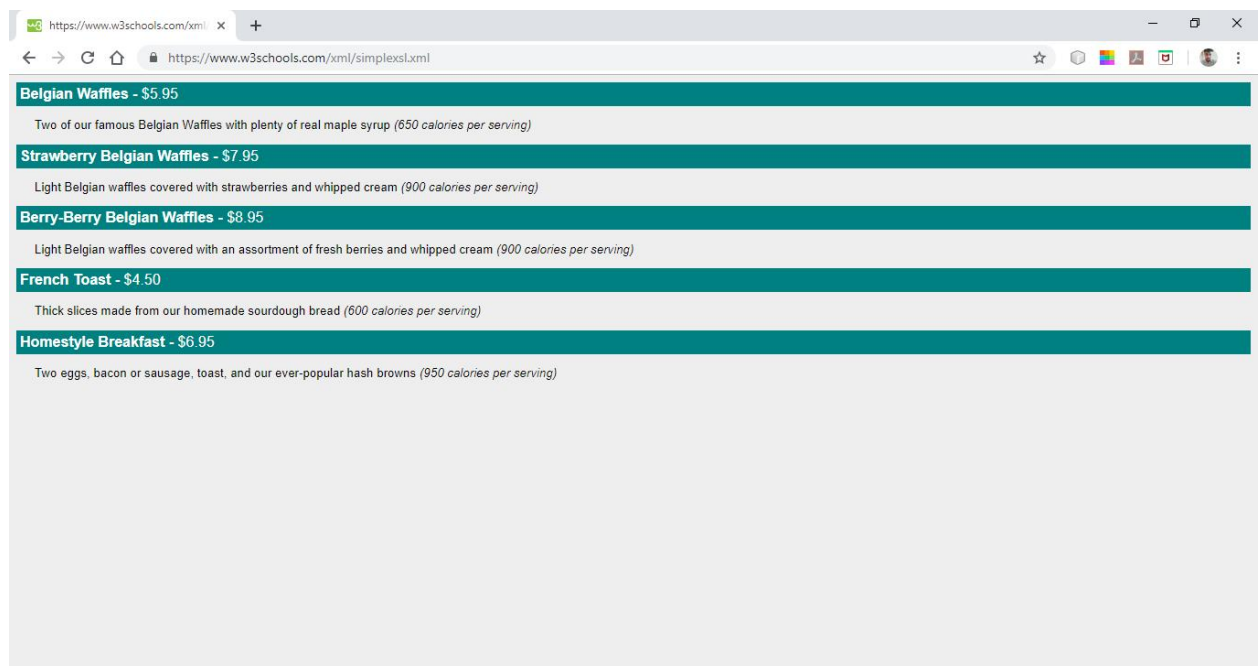
```

```

<div style="margin-left:20px;margin-bottom:1em;font-size:10pt">
  <p>
    <xsl:value-of select="description"/>
    <span style="font-style:italic"> (<xsl:value-
of select="calories"/> calories per serving)</span>
  </p>
</div>
</xsl:for-each>
</body>
</html>

```

Output:-



Conclusion:-

Thus, in this practical we have studied and created XML document using XML Schema, XSLT.

Experiment No. 06

Aim:- Write programs using AJAX

Theory:-

AJAX stands for **A**synchronous **J**avaScript and **X**ML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script.

- Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display.
- Conventional web applications transmit information to and from the sever using synchronous requests. It means you fill out a form, hit submit, and get directed to a new page with new information from the server.
- With AJAX, when you hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.
- XML is commonly used as the format for receiving server data, although any format, including plain text, can be used.
- AJAX is a web browser technology independent of web server software.
- A user can continue to use the application while the client program requests information from the server in the background.
- Intuitive and natural user interaction. Clicking is not required, mouse movement is a sufficient event trigger.
- Data-driven as opposed to page-driven.

Rich Internet Application Technology

AJAX is the most viable Rich Internet Application (RIA) technology so far. It is getting tremendous industry momentum and several tool kit and frameworks are emerging. But at the same time, AJAX has browser incompatibility and it is supported by JavaScript, which is hard to maintain and debug.

AJAX is Based on Open Standards

AJAX is based on the following open standards –

- Browser-based presentation using HTML and Cascading Style Sheets (CSS).
- Data is stored in XML format and fetched from the server.
- Behind-the-scenes data fetches using XMLHttpRequest objects in the browser.
- JavaScript to make everything happen.

AJAX cannot work independently. It is used in combination with other technologies to create interactive webpages.

JavaScript

7. Loosely typed scripting language.
8. JavaScript function is called when an event occurs in a page.
9. Glue for the whole AJAX operation.

DOM

9. API for accessing and manipulating structured documents.
10. Represents the structure of XML and HTML documents.

CSS

10. Allows for a clear separation of the presentation style from the content and may be changed programmatically by JavaScript

XMLHttpRequest

- JavaScript object that performs asynchronous interaction with the server.

Program:

Ajax-data.txt

```
<html>
```

```
<body>
```

```
<font face="pristina" size="20px">Let's see the image has been disappeared !!</font>
```

```
</body>
```

```
</html>
```

Index.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Ajax program</title>

<script>

functionloadXMLDoc()

{

varxmlhttp;

if (window.XMLHttpRequest)

{

xmlhttp = new XMLHttpRequest();

}

else

{

xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange = function()

{

if (xmlhttp.readyState == 4 &&xmlhttp.status == 200)

{

document.getElementById("myImage").innerHTML = xmlhttp.responseText;

}

}

xmlhttp.open("GET","ajax-data.txt",true);

xmlhttp.send();

}

</script>

</head>
```

```
<body style="text-align: center;">
```

```
<!--An image goes here to test the Ajax program.-->
```

```
<div id="myImage">
```

```
<h2>Click the button below to make the image disappear.</h2>
```

```
<imgsrc="image8.jpg" width="500px" height="300px" title="Yellow Flower" alt="an image of a yellow flower"/>
```

```
</div>
```

```
<br/>
```

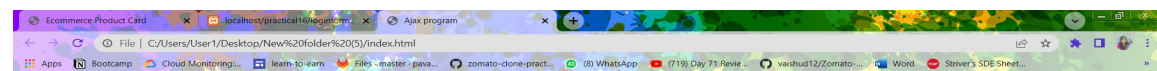
```
<!--A button goes here-->
```

```
<button type="button" onclick="loadXMLDoc()">Click here to make the picture disappear!</button>
```

```
</body>
```

```
</html>
```

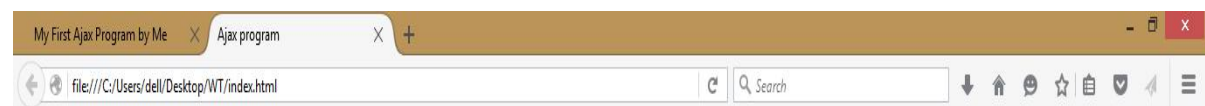
Output :-



Click the button below to make the image disappear.



Click here to make the picture disappear!



Let's see the image has been disappeared !!

Click here to make the picture disappear!

Conclusion :-

Thus, in this practical, we have studied about AJAX and also created a program using AJAX.

Experiment No. 7

Aim:- Implementing python's Flask frame work to host a site on internet.

Theory:-

Flask

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks.

Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more frequently than the core Flask program.

Applications that use the Flask framework include Pinterest and LinkedIn

Features:

- Development server and debugger
- Integrated support for unit testing
- RESTful request dispatching
- Uses Jinja templating
- Support for secure cookies (client side sessions)
- 100% WSGI 1.0 compliant
- Unicode-based
- Extensive documentation
- Google App Engine compatibility
- Extensions available to enhance features desired

Example:

The following code shows a simple web application that displays "Hello World!" when visited:

```
from flask import Flask
```

```
app = Flask(__name__)
```

```
@app.route("/")
```

```
def hello():
```

```
    return "Hello World!"
```

```
if __name__ == "__main__":
```

```
    app.run()
```

Implementing the flask server:

1: First import the library.

```
from flask import Flask
```

2: Then create a Flask object

```
app = Flask(__name__)
```

3: Specify the function which will execute when a certain URL is accessed. Specify it using `@app.route("/")`, where `<site_name> '/'` will execute the following function.

```
def hello():
```

```
    return "Hello World!"
```

4: Run the server from the object created earlier.

```
app.run()
```

HTML CODE: (which should be located in a folder called *'templates'*)

Uses Jinja2 template, to place the processed data onto the webpage.

Base.html

```
<!doctype html>
```

```
<html>
```

```
    <head>
```

```
        <title>{{ the_title }}</title>
```

```
        <link rel="stylesheet" href="static/hf.css" />
```

```
    </head>
```

```
    <body>
```

```
        {% block body %}
```

```
        {% endblock %}
```

```
    </body>
```

```
</html>
```

Entry.html

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

```
{% block body %}
```

```
    <h2>{{ the_title }}</h2>
```

```

<form method='POST' action='/results'>

<table>

<p>Fill the details:</p>

<tr><td>Name:</td><td><input name='name' type='TEXT' width='60'></td></tr>

<tr><td>Age:</td><td><input name='age' type='TEXT' ></td></tr>

<tr><td>Gender:</td><td><input name='gender' type='TEXT'</td></tr>

</table>

<p>When you're ready, click this button:</p>

<p><input value='See The Results' type='SUBMIT'></p>

</form>

```

```
{% endblock %}
```

Results.html

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

```
{% block body %}
```

```
<h2>{{ the_title }}</h2>
```

```
<p><b>{{ the_name }}<b> Here Are Your Results</p>
```

```
<p> You will get married at the age of </p>
```

```
<h3>{{ the_results }}</h3>
```

```
{% endblock %}
```

CSS CODE: (which should be located in a folder named 'static' with the name 'hf.css')

```
body {
```

```
font-family: Verdana, Geneva, Arial, sans-serif;
```

```
font-size:    medium;

background-color: tan;

margin-top:    5%;

margin-bottom: 5%;

margin-left:   10%;

margin-right:  10%;

border:        1px dotted gray;

padding:       10px 10px 10px 10px;
}

a {

    text-decoration: none;

    font-weight:    600;

}

a:hover {

    text-decoration: underline;

}

a img {

    border:        0;

}

h2 {

    font-size:     150%;

}

table {

    margin-left:    20px;

    margin-right:   20px;

    caption-side:   bottom;

    border-collapse: collapse;

}

td, th {

    padding:        5px;
```

```
    text-align:    left;
}
```

```
.copyright {
    font-size:     75%;
    font-style:    italic;
}
```

```
.slogan {
    font-size:     75%;
    font-style:    italic;
}
```

```
.confirmentry {
    font-weight:   600;
}
```

```
/* ** Tables ** */
```

```
table {
    font-size:     1em;
    background-color: #fafcff;
    border:        1px solid #909090;
    color:         #2a2a2a;
    padding:       5px 5px 2px;
    border-collapse: collapse;
}
```

```
td, th {
    border:        thin dotted gray;
}
```

```
/* ** Inputs ** */
```

```
input[type=text] {
```

```

font-size:    115%;
width:        30em;
}
input[type=submit] {
font-size:    125%;
}
select {
font-size:    125%;
}

```

BACKEND CODE:

Marriage.py

```
from random import randint
```

```
def marriage(age: int) -> list:
```

```
    x= int(randint(0,50)) + int(age)
```

```
    if x in [0,1,2,3,4,5,6,7,8,9,10]:
```

```
        r = 'You seem to be getting married soon!!!'
```

```
    if x in [11,12,13,14,15,16,17,18,19,20]:
```

```
        r = 'Teenage Turtle!!!'
```

```
    if x in [21,22,23,24,25,26,27,28,29,30]:
```

```
        r = 'Youth getting FRUIT!!!'
```

```
    if x in [31,32,33,34,35,36,37,38,39,40]:
```

```
        r = "What you were doing in your 20's ? "
```

```
    if x in [41,42,43,44,45,46,47,48,49,50]:
```

```
        r = 'I think you are well settled at this age, eh?'
```

```
    if x in [51,52,53,54,55,56,57,58,59,60]:
```

```
        r = 'Looks like you are getting married after your RETIREMENT'
```

```
    if x in [61,62,63,64,65,66,67,68,69,70]:
```

```
        r = 'How are you doing OLDY???'

```

```
    if x in [71,72,73,74,75,76,77,78,79,80]:

```

```
r = "What you've been doing in your YOUNG age?"
```

```
if x > 80:
```

```
r = "I don't think so you'll be alive till then!!!"
```

```
found = [x,r]
```

```
return (found)
```

FLASK SERVER CODE:

```
Marriageweb.py
```

```
from flask import Flask
```

```
from flask import render_template
```

```
from flask import request
```

```
from marriage import marriage
```

```
app = Flask(__name__)
```

```
@app.route('/')
```

```
@app.route('/entry')
```

```
def entry_page() -> 'html':
```

```
    return render_template('entry.html', the_title = 'Know When You Are Getting Married!!!')
```

```
@app.route('/results', methods = ['POST'])
```

```
def result_page() -> 'html':
```

```
    name = request.form['name']
```

```
    age = request.form['age']
```

```
    gender = request.form['gender']
```

```
    result = marriage(age)
```

```
    if gender == 'M' or gender == 'Male':
```

```
        return render_template('results.html', the_title = 'HEY BOY!!! Your Result Looks  
Amazing : ', the_name = name, the_results = result,)
```

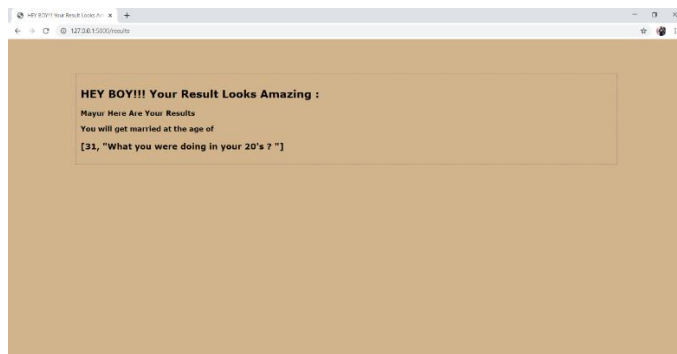
OUTPUT:

server script:

The image displays two sequential screenshots of a web browser window. The browser's address bar shows the URL "127.0.0.1:5000". The webpage has a light brown background and features a form titled "Know When You Are Getting Married!!!".

In the first screenshot, the form contains the text "Fill the details:" followed by three input fields labeled "Name:", "Age:", and "Gender:". These fields are currently empty. Below the form is a button labeled "See The Results".

In the second screenshot, the same form is shown, but the input fields are now populated with the text "Mayur", "20", and "Male" respectively. The "See The Results" button remains visible below the form.



Conclusion: So, the flask server has been implemented successfully and the site has been hosted on the local host without any error.

EXPERIMENT NO.8

Design the following static web pages required for an online book store web site.

- 1) HOME PAGE:** The static home page must contain three **frames**.
- 2) LOGIN PAGE**
- 3) CATALOGUE PAGE:** The catalogue page should contain the details of all the books available in the web site in a table.
- 4) REGISTRATION PAGE**

Aim: Design the following static web pages required for online book store.

- 1. Home page:-** the static home page must contains three pages
- 2. Top frame:-** logo and college name and links to homepage, login page, registration page and catalogue page
- 3. Left frame:-** at least four links for navigation which will display the catalogue of Respective links
- 4. Right frame:-** the pages to links in the left frame must be loaded here initially it Contains the description of the website.

DESCRIPTION: In this program the entire web paged are created by using basic HTML tags. Home page is divided into 3 frames by using <frameset> and <frame> tags. A frame is used to display a web page within a web page.

<frameset>:

- The <frameset> tag defines a frameset.
- The <frameset> element holds one or more <frame> elements.
- Each <frame> element can hold a separate document.

- The <frameset> element specifies HOW MANY columns or rows there will be in the frameset, and HOW MUCH percentage/pixels of space will occupy each of them.

<frame>:

- The <frame> tag defines one particular window (frame) within a <frameset>.
- Each <frame> in a <frameset> can have different attributes, such as border, scrolling, the ability to resize, etc.

PROGRAM:

Home.html

```
<frameset rows="20%,*">
```

```
<frame src="top.html" noresize scrolling="NO" name="topframe"> <frameset cols="10%,*">
```

```
<frame src="left.html" noresize scrolling="NO" name="leftframe"> <frame src="right.html"
noresize name="rightframe" scrolling="auto">
```

```
</frameset>
```

```
</frameset>
```

```
<frameset rows="30%,*">
```

```
<frame src="top.html" noresize scrolling="NO" name="topframe"> <frameset cols="10%,*">
```

```
<frame src="left.html" noresize scrolling="NO" name="leftframe"> <frame src="right.html" noresize
name="rightframe" scrolling="auto">
```

```
</frameset>
```

```
</frameset>
```

Catalogue.html

```
<html>
```

```
<head>
```

```
<title> Catalogue </title>
```

```
</head>
```

```
<body bgcolor="pink">
```

```
<option value="Java">Java
```

```
<option value="Oracle">Oracle
```

```
<option value="Ms SQL Server">Ms SQL Server <option value="MySql">MySql
```

```
</select>
```

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

```
<tr>
```

```
<td>Quantity</td>
```

```
<td><input type="text" id="q"></td>
```

```
</tr>
```

```
<tr>
```

```
<td></td>
```

```
<td>
```

```
<form method=post action="order.html">
```

```
<input type="submit" value=ok />
```

```
</form>
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
<center>
```

```
<pre> Cost of one book is"500" + shipping "100" </pre> </center>
```

```
</body>
```

```
</html>
```

Ese.html

```
<html>
```

```
<body bgcolor="Plum">
```

```
<h1><font color="blue">Electronics and Communication Engineering</font></h1>
```

```
<h2>
```

```
<ul>
```

```
<li>Digital Circuits</li> <li>Signals and Systems</li> <li>Digital Communication</li> </ul>
```

```
</h2>
```

</body>

</html>

Eee.html

<html>

<body bgcolor="Plum">

<h1>Electrical and Electronics Engineering</h1> <h2>

<ul type="square">

Concepts in Electric Circuits Introduction to Electronic Engineering Electrical Power

</h2>

</body>

</html>

Left.html

<html>

<body align="center" bgcolor="bisque">

CSE

 ECE

 EEE

 MECH

</body>

</html>

Login.html

<html>

<body bg color="powderblue">

<basefont face="Cambria" size="4">

<center>

Enter Login Details:


```

</center>

<form name="f1" method="post" action="right.html">

<table align="center" width="100" height="150" cellspacing="15"> <tr><td><b>Login ID:</b></td>

<td><input type="text" name="t1"></td>

</tr>

<tr>

<td><b>Password:</b></td>

<td><input type="password" name="t2"></td> </tr>

<tr align="center">

<td><input type="submit" name="b1" value="Submit"></td> <td><input type="reset" name="b2"
value="Reset"></td>

</tr>

</table>    </form>    </basefont>    </body></html>

```

Mech.html

```

<html>

<body bgcolor="Plum">

<h1><font color="blue">Electronics and Communication Engineering</font></h1> <h2>

<ol type="I">

<li>Theory of Machines</li>

<li>Automation and Robotics</li>

<li>Engineering Fluid Mechanics</li>

</ol>

</h2>

</body>

</html>

```

Order.html

```

<html>

<head><title>order conformation</title></head> <body bgcolor="cyan">

<center>

<pre><strong>

<b>Your order Is Conformed

</strong></pre>

```



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

Top.html

```
<html>
<head>
<title>Top Frame</title>
</head>
<body bgcolor="yellow">


 <center>
<marquee bgcolor="YellowGreen" width="300" behavior="alternate">
<font face="calibri" size="6" color="Red"><b>Online Book Store</b> </font>
</marquee> <br>
<font face="Brush Script" size="5" color="black"><b>Created & Maintained By
MRCET</b></font>

</center>
<br>

<table width="50%" height="50%" cellpadding=6> <tr align="center">

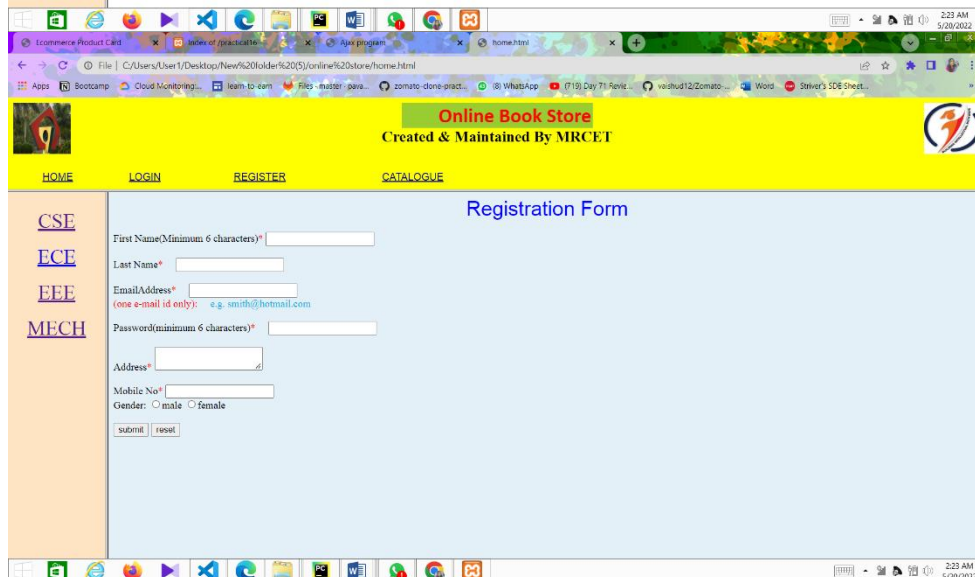
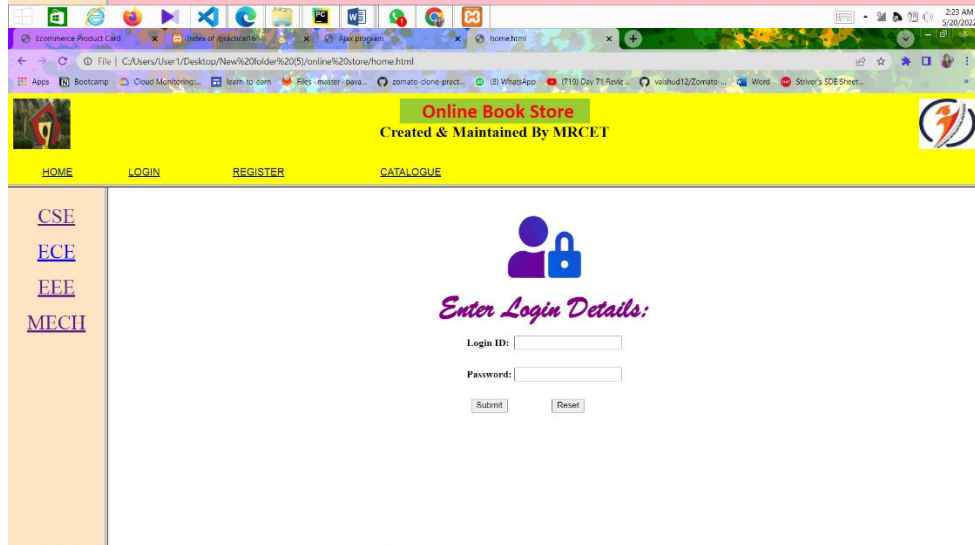
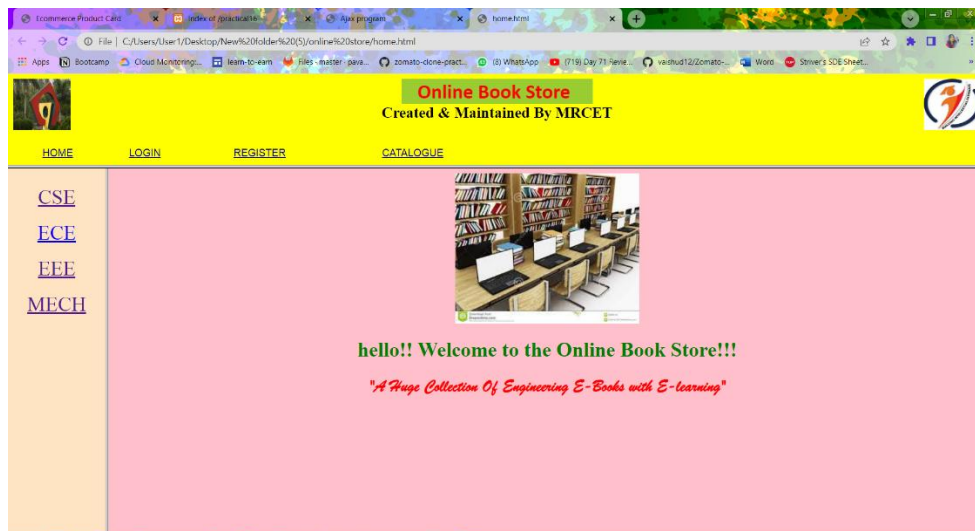
<td> <a href="Home.html" target="_parent"><font face="arial" size="3"
color="navy">HOME </a> </td>

<td> <a href="login.html" target="rightframe"><font face="arial" size="3"
color="navy">LOGIN</a> </td>

<td> <a href="registration.html" target="rightframe"><font face="arial"
size="3" color="navy">REGISTER</a> </td>

<td> <a href="catalogue.html" target="rightframe"> <font face="arial" size="3"
color="navy">CATALOGUE</a> </td>

</tr>
</table>
</body>
</html>
```




Online Book Store
 Created & Maintained By MRCET




[HOME](#)
[LOGIN](#)
[REGISTER](#)
[CATALOGUE](#)

[CSE](#)
[ECE](#)
[EEE](#)
[MECH](#)

	Book: Web Technologies Author: NP Gopalan Publication: Oxford University Press	531	Add to cart
	Book: Python Head first Author: Paul Barry Publication: PEARSON	898	Add to cart



Online Book Store
 Created & Maintained By MRCET




[HOME](#)
[LOGIN](#)
[REGISTER](#)
[CATALOGUE](#)


[CSE](#)
[ECE](#)
[EEE](#)
[MECH](#)

Electrical and Electronics Engineering

- Concepts in Electric Circuits
- Introduction to Electronic Engineering
- Electrical Power



Online Book Store
 Created & Maintained By MRCET



[HOME](#)
[LOGIN](#)
[REGISTER](#)
[CATALOGUE](#)

[CSE](#)
[ECE](#)
[EEE](#)
[MECH](#)

Electronics and Communication Engineering

- I. Theory of Machines
- II. Automation and Robotics
- III. Engineering Fluid Mechanics



Conclusion:

Thus successfully implemented html websites

EXPERIMENT NO:9

Write *JavaScript* to validate the following fields of the Registration page.

1. **First Name** (Name should contains alphabets and the length should not be less than 6 characters).
2. **Password** (Password should not be less than 6 characters length).
3. **E-mail id** (should not contain any invalid and must follow the standard pattern name@domain.com)
4. **Mobile Number** (Phone number should contain 10 digits only).
5. **Last Name and Address** (should not be Empty)

AIM: To validate the fields of registration page using JavaScript

DESCRIPTION: In order to validate the fields of login and registration pages JavaScript is used. JavaScript is programming code that can be inserted into HTML pages. JavaScript inserted into HTML pages, can be executed by all modern web browsers. JavaScript is mainly used for validating the elements in a form submitted by the user. This JavaScript code can react to user events.

PROGRAM: After clicking OK button the page is redirected to success.html <html>

```
<head><title>Registration Form
Validation</title></head> <body bgcolor="#E4F0F8">

<script type='text/javascript'>

    function formValidator()

    {

        // Make quick references to our fields

        var firstname = document.getElementById('firstname');
        var lastname = document.getElementById('lastname');
        var email = document.getElementById('email'); var pass
        = document.getElementById('pass');

        var addr = document.getElementById('addr');

        var mobileno = document.getElementById('mobileno');

        / Check each input in the order that it appears in the form!
        if(notEmpty(firstname, "can not be null")){

            if(isAlphabet(firstname, "Please enter only letters for your
            Firstname")){ if(lengthRestriction(firstname, 6)){

                if(isAlphabet(lastname, "Please enter only letters for your
                Lastname")){ if(emailValidator(email, "Please enter a valid email
                address")){

                    if(lengthRestriction(pass, 6)){

                        if(isAlphanumeric(pass, "please enter Numbers and Letters Only for
                        password")){ if(notEmpty(addr, "please enter the address")){

                            if(isNumeric(mobileno, "Please enter a valid mobileno")){

                                if(lengthRestriction1(mobileno, 10 , 10)){
```

```

        return true;
    }
}

}

}

} }

return false;
}

function notEmpty(elem, helperMsg){

    if(elem.value.length == 0){

        alert(helperMsg);

        elem.focus(); // set the focus to this input

        return false;

    }

    return true;

}

function isNumeric(elem, helperMsg){

    var numericExpression = /^[0-9]+$/;

    if(elem.value.match(numericExpression)){

        return true;

    }else{

        alert(helperMsg);

        elem.focus();

        return false;

    }

}

```

```
function isAlphabet(elem, helperMsg){  
    var alphaExp = /^[a-zA-Z]+$/;  
    if(elem.value.match(alphaExp)){  
        return true;  
    }else{  
        alert(helperMsg);  
        elem.focus();  
        return false;  
    }  
}
```

```
function isAlphanumeric(elem, helperMsg){  
    var alphaExp = /^[0-9a-zA-Z]+$/;  
    if(elem.value.match(alphaExp)){  
        return true;  
    }else{  
        alert(helperMsg);  
        elem.focus();  
        return false;  
    }  
}
```

```
function lengthRestriction(elem, min){  
  
    var uInput = elem.value;  
    if(uInput.length >= min){  
        return true;  
    }else{  
        alert("Please enter minimum " +min+ " characters");  
        elem.focus();  
    }  
}
```

```

        return false;

    }

}

function emailValidator(elem, helperMsg)

{

var emailExp = /^[\\w\\-\\.\\+]+\\@[a-zA-Z0-9\\.\\-]+\\. [a-zA-z0-9]{2,4}$/;

    if(elem.value.match(emailExp))

    {

        return true;

    }

    else{

        alert(helperMsg);

        elem.focus();

        return false;

    }

}

function lengthRestriction1(elem, min, max)

{

    var uInput = elem.value;

    if(uInput.length >= min && uInput.length <= max)

    {

        return true;

    }

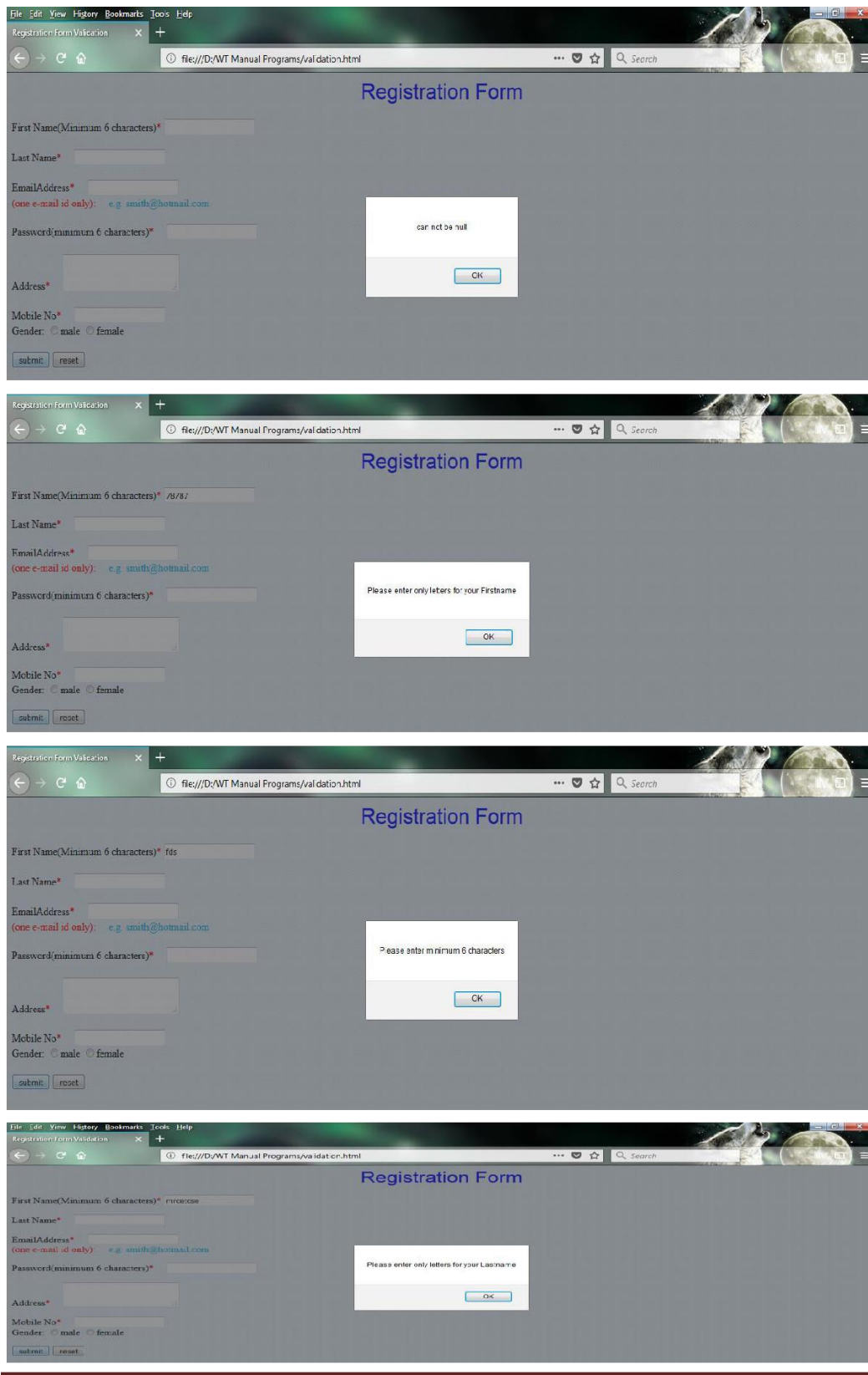
    else {

        alert("Please enter 10 numbers only");

        elem.focus();

```

$\}} \quad$



CONCLUSION: thus successfully completed javascript validation

EXPERIMENT NO - 10:

Develop and demonstrate the usage of inline, internal and external style sheet using CSS.

Aim: Design a web page using CSS which includes the following:

- 1) Use different font styles
- 2) Control the repetition of image with background-repeat and no-repeat property
- 3) Define style for links as a: link, a: active, a: hover, a: visited
- 4) Add customized cursors for links.

Cascading Style Sheets (CSS) is used to format the layout of a webpage.

With CSS, you can control the color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes, and much more!

Tip: The word **cascading** means that a style applied to a parent element will also apply to all children elements within the parent. So, if you set the color of the body text to "blue", all headings, paragraphs, and other text elements within the body will also get the same color (unless you specify something else)!

Using CSS

CSS can be added to HTML documents in 3 ways:

- **Inline** - by using the style attribute inside HTML elements
- **Internal** - by using a <style> element in the <head> section
- **External** - by using a <link> element to link to an external CSS file

The most common way to add CSS, is to keep the styles in external CSS files. However, in this tutorial we will use inline and internal styles, because this is easier to demonstrate, and easier for you to try it yourself. **ggv**

Inline CSS

An inline CSS is used to apply a unique style to a single HTML element.

An inline CSS uses the **style** attribute of an HTML element.

The following example sets the text color of the <h1> element to blue, and the text color of the <p> element to red:

PROGRAM:

```
<html>
```

```
<head>
```

```
<style type="text/css">
```

```
body
```

```

{

background-image:url('img1.jpg');

background-repeat:repeat-x;
/*non-repeat*/

background-position:center center;

background-attachment:fixed;

background-color:pink;

}

a:link { text-decoration:none;color:orange; }
a:visited { text-decoration:none;color:red; }
a:hover { text-decoration:underline;color:blue; }
a:active { text-decoration:underline;color:purple; }

h3 { color:green; }

.c1{cursor:crosshair}

.c2{cursor:pointer}

.c3{cursor:move}

.c4{cursor:text}

.c5{cursor:wait}

.c6{cursor:help}

</style>

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

<body bgcolor="cyan">

<h1 style="color:blue;text-align:center;"> CSS (Inline, Internal and External)
</h1>

<p>This Paragraph is a Not Styled</p>

```

```

<p class="left">This Paragraph is Styled by class "Left"</p>

<p class="center">This Paragraph is Styled by class "Center"</p> <p
class="right">This Paragraph is Styled by class "Right"</p>

<b>This is normal Bold</b> <br>

<b id="headline">This Bold Text is Styled </b>

<h2><b><a href=" " >This is a link</a></b></h2>

<h3 class="c1">The cursor over this element is plus sign</h3>

<h3 class="c2">The cursor over this element is a pointing hand</h3> <h3
class="c3">The cursor over this element is a grasping hand</h3> <h3
class="c4">The cursor over this element is a I bar</h3> <h3 class="c5">The
cursor over this element is a wait</h3>

<h3 class="c6">The cursor over this element is a question mark</h3> </html>

```

CSS

p.left

```

{
text-align:left;

color:blue;

font-family:Cambria;

font-size:large;

text-indent:20px;
}

```

p.center

```

{
text-align:center;

text-decoration:underline;

text-transform:uppercase;
}

```

```
letter-spacing:-3px;

word-spacing:20px;

font-size:larger;

}

p.right

{
text-align:right;
color:red;
font-family:Tahoma;
font-size:15pt;
text-decoration:overline;
font-style:italic;
}

b#headline

{
color:orange;
font-size:22px;
font-family:arial;
text-decoration:underline;
}
```



CONCLUSION: thus successfully implemented css style sheet program

EXPERIMENT NO- 11:

Develop and demonstrate JavaScript with POP-UP boxes and functions for the following problems:

- a) Input: Click on Display Date button using onclick() function
Output: Display **date** in the textbox
- b) Input: A number n obtained using **prompt**
Output: **Factorial** of n number using **alert**
- c) Input: A number n obtained using **prompt**
Output: A **multiplication table** of numbers from 1 to 10 of n using **alert**
- d) Input: A number n obtained using **prompt** and add another number using **confirm**
Output: **Sum** of the entire n numbers using **alert**

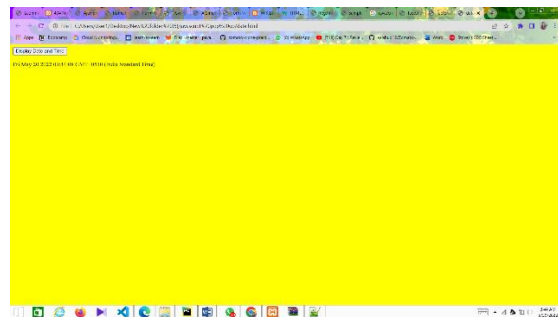
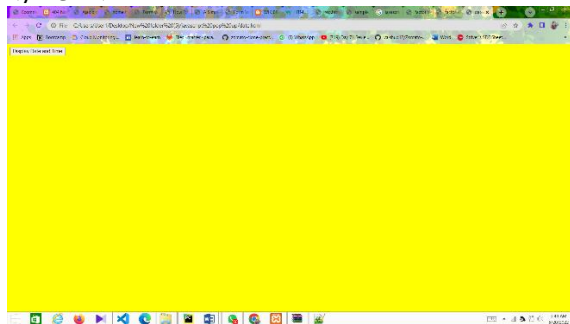
PROGRAM:

- a. Input: Click on Display Date button using onclick() function
Output: Display **date** in the textbox

```
<!DOCTYPE html>
<html>
<body bgcolor="yellow">
<button type="button"
onclick="document.getElementById('date_time_button').innerHTML = Date()">
Display Date and Time</button>

<p id="date_time_button"></p>

</body>
</html>
```



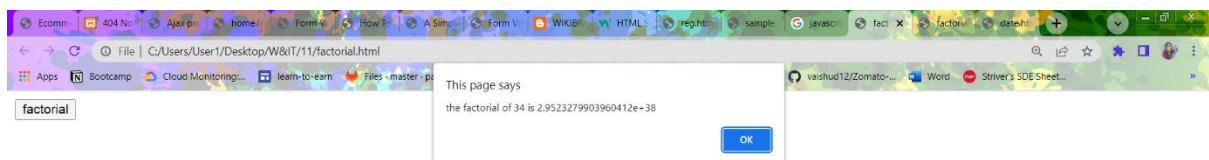
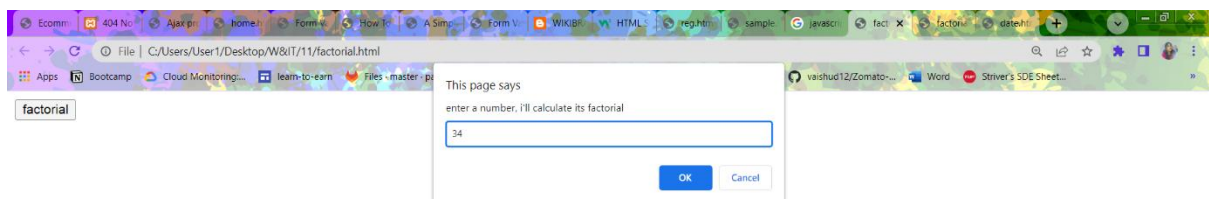
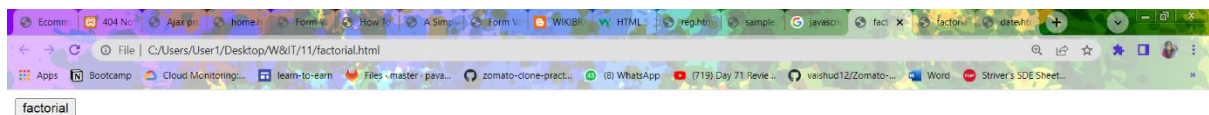
- c) Input: A number n obtained using **prompt**
Output: **Factorial** of n number using **alert**

```
<html>
<head>
<title>factorial</title>
```

```

<script language='javascript'>
function factorialcalc()
{
number = prompt("enter a number, i'll calculate its factorial", "whole numbers
bigger than zero, please")
factorial = 1
for (i=1; i <= number; i++)
{
factorial = factorial * i
}
alert("the factorial of " + number + " is " + factorial)
}
</script>
</head>
<body><form name=frm>
<input type=button value='factorial' onclick="factorialcalc();">
</form>
</body>
</html>

```



- e) Input: A number n obtained using **prompt**
 Output: A **multiplication table** of numbers from 1 to 10 of n using **alert**


```

<html>

<head><title> Multiplication Table </title></head>
<body bgcolor="pink">

<script type="text/javascript">

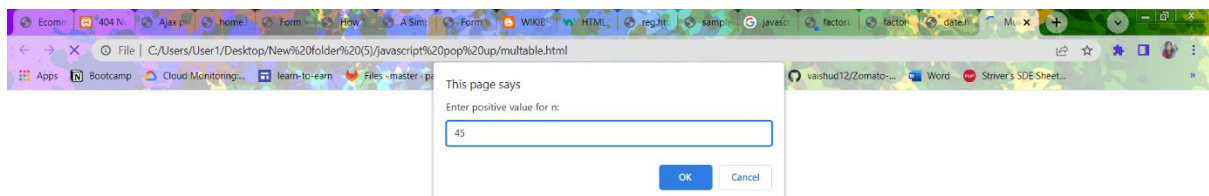
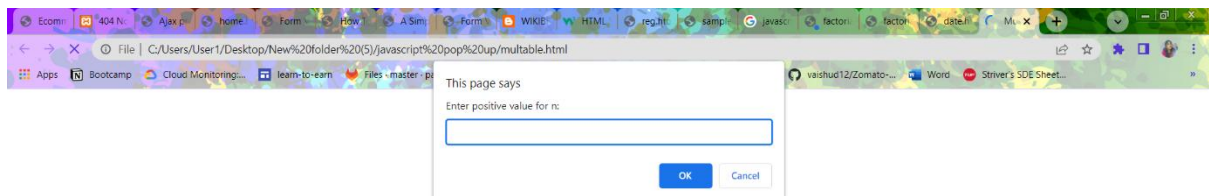
var n=prompt("Enter positive value for n: "," "); if(!isNaN(n))
{
    var table="";
    var number="";
    for(i=1;i<=10;i++) {
        number = n * i;
        table +=  n + " * " + i + " = " + number + "\n";
    }

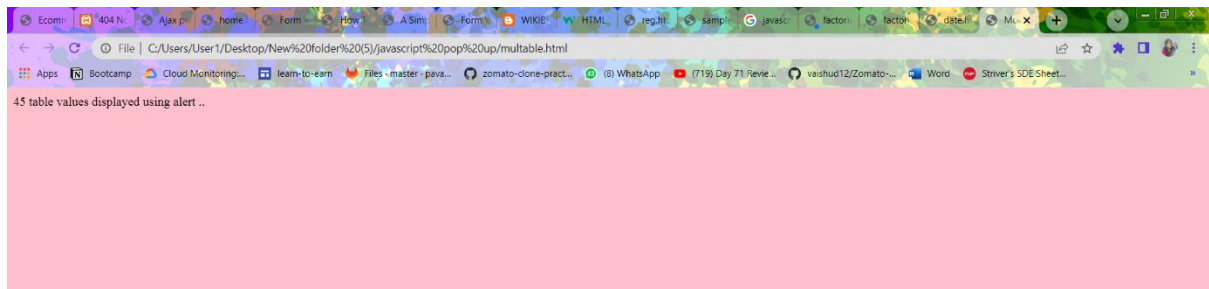
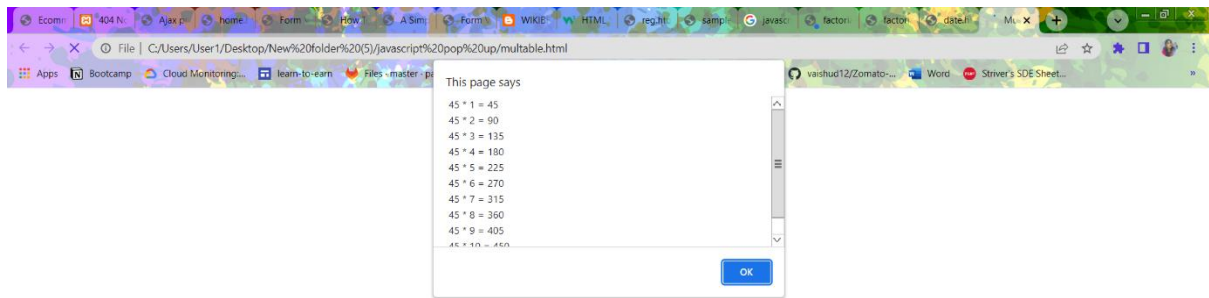
    alert(table);
}

else {
    alert("Enter positive value");
    n=prompt("Enter positive value for n: "," ");
}
document.write(n+" table values displayed using alert ..<br />");

</script> </body> </html>

```





- f) Input: A number n obtained using **prompt** and add another number using **confirm**
Output: **Sum** of the entire n numbers using **alert**

```
<html>
```

```
<head><title>sum of n numbers using popup  
boxes</title> <script language='javascript'>
```

```
function addsum()
```

```
{
```

```
    alert("you're going to give me a list of numbers. i'm going to add them together for you");  
    var keepgoing = true
```

```
    var sumofnums = 0
```

```
    while (keepgoing) {
```

```
        sumofnums = sumofnums + parseInt(prompt("what's the next number to add?", ""))
```

```
        keepgoing = confirm("add another number?")
```

```
    }
```

```
    alert("the sum of all your numbers is " + sumofnums)
```

```

    }

</script>

</head>

<body>

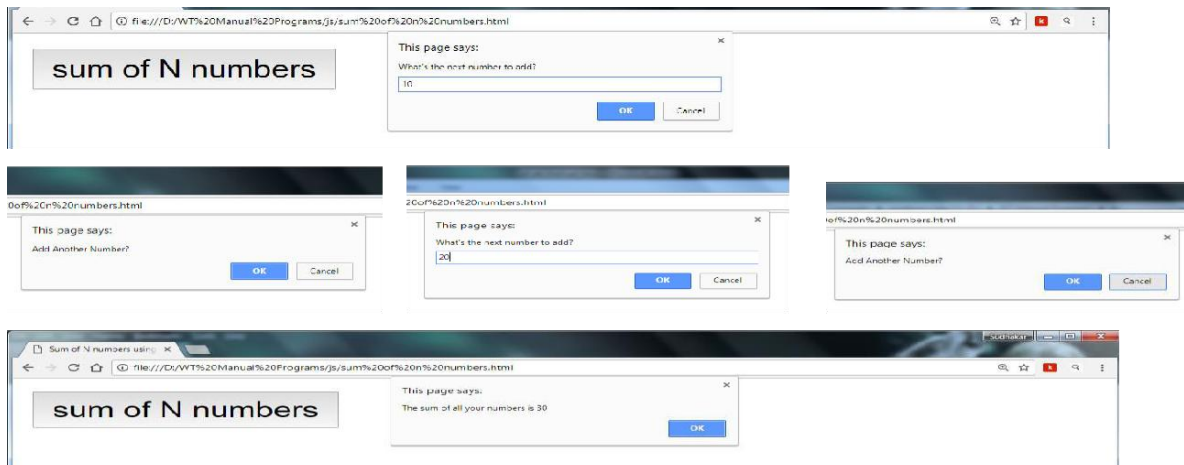
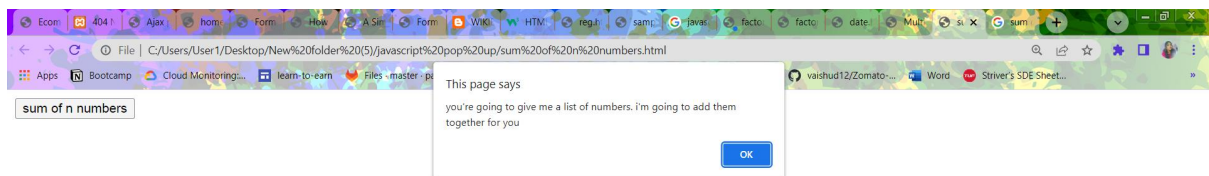
    <form name=frm>

        <input type=button value='sum of n numbers'
        onclick="addsum();"> </form>

</body>

</html>

```



CONCLUSION:

Thus successfully completed and implemented javascript pop ups

EXPERIMENT NO - 12:

Write an HTML page that contains a selection box with a list of 5 countries. When the user selects a country, its capital should be printed next in the list. Add CSS to customize the properties of the font of the capital (color, bold and font size)

Programs

```
<html>
<head>
<body background

<title>Html web</title> </head>

<style>

body {
    background-image: url('R.jpg');
}

h1
{
    color: red;
    text-align: center;
}

.textbox1
{
    color: blue;
    font-size: 30px;
    font-weight: bold;
}
</style>
<body>
<center>

<h1> Select the country name to find its capital</h1> <form name="myform">

Select Country <select name="country" id="sbox1" onClick="myFunction()"
required> <option value=""></option>

<option value="NEW DELHI">INDIA</option> <option
value="CANBERRA">AUSTRALIA</option> <option value="WASHINGTON
D.C">AMERICA</option> <option value="LONDON">UNITEDKINGDOM</option> <option
value="BERLIN">GERMANY</option> </select><br><br>

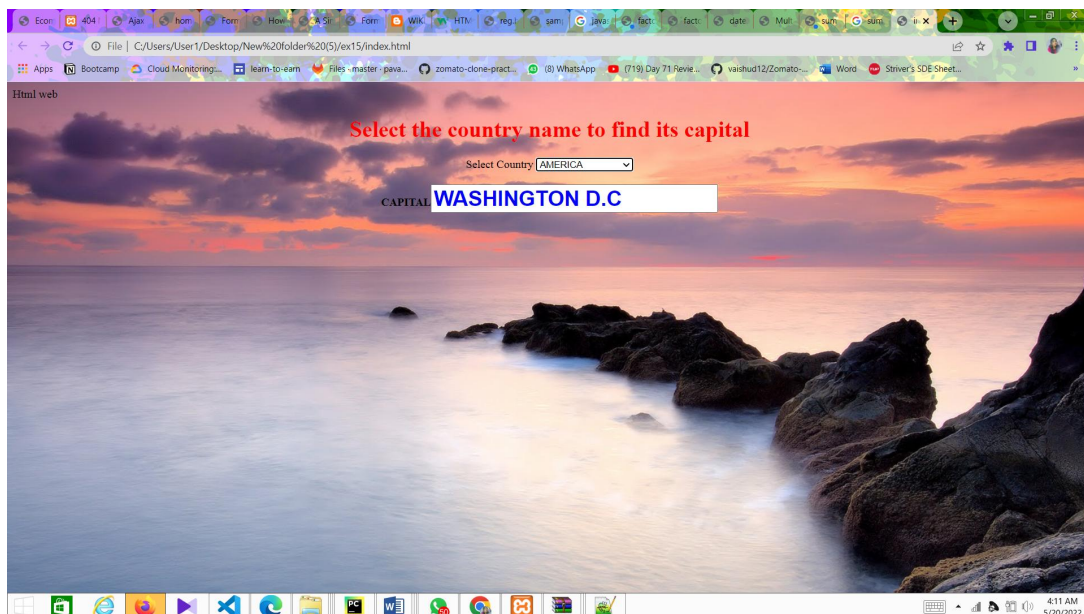
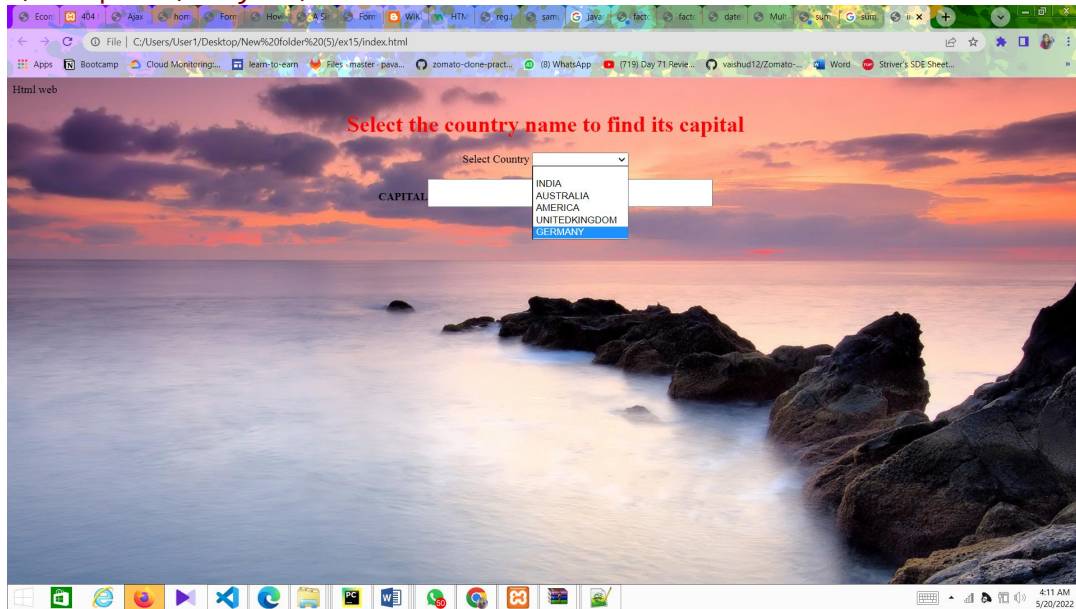
<B>CAPITAL</B><input type="text" class="textbox1" id="sbox2"> </form>
```

```

</center>
<script>
function myFunction()
{var a=document.getElementById("sbox1").value;
document.getElementById("sbox2").value=a;}

</script> </body> </html>

```



Conclusion:

Thus successfully implemented html page for capital...

EXPERIMENT NO- 13:

Write an HTML page including any required JavaScript that takes a number from text field in the range of 0 to 999 and shows it in words. It should not accept four and above digits, alphabets and special characters.

Programs:

```
<html>
<head>
<body bgcolor="pink">

<title>HTML - Convert numbers to words using JavaScript</title> <SCRIPT
language=Javascript>
<script>
function NumToWord(inputNumber, outputControl)
{
var str = new String(inputNumber)
var splt = str.split("");
var rev = splt.reverse();

var once = ['Zero', ' One', ' Two', ' Three', ' Four', ' Five', ' Six', '
Seven', ' Eight', ' Nine'];
var twos = ['Ten', ' Eleven', ' Twelve', ' Thirteen', ' Fourteen', ' Fifteen',
' Sixteen', 'Seventeen', ' Eighteen', ' Nineteen'];

var tens = ['', 'Ten', ' Twenty', ' Thirty', ' Forty', ' Fifty', ' Sixty', '
Seventy', ' Eighty', ' Ninety'];

numLength = rev.length;
var word = new Array();
var j = 0;
for (i = 0; i < numLength; i++) {
switch (i) {

case 0:
if ((rev[i] == 0) || (rev[i + 1] == 1)) {
word[j] = '';
}
else {
word[j] = once[rev[i]];
}
word[j] = word[j];
break;
case 1:
aboveTens();

break;
case 2:
```

```

    if (rev[i] == 0) {
    word[j] = '';
    }
    else if ((rev[i - 1] == 0) || (rev[i - 2] == 0)) {
    word[j] = once[rev[i]] + " Hundred ";
    }
    else {
    word[j] = once[rev[i]] + " Hundred and";
    }
    break;
    default: break;
    }

    j++;
    }
    function aboveTens() {
    if (rev[i] == 0) { word[j] = ''; }

    else if (rev[i] == 1) { word[j] = twos[rev[i - 1]]; } else { word[j] =
    tens[rev[i]]; }

    }
    word.reverse();
    var finalOutput = '';
    for (i = 0; i < numLength; i++) {
    finalOutput = finalOutput + word[i];
    }
    document.getElementById(outputControl).innerHTML = finalOutput;

    }
</script>
</head>
<body>
<h1>HTML - Convert numbers to words using JavaScript</h1>

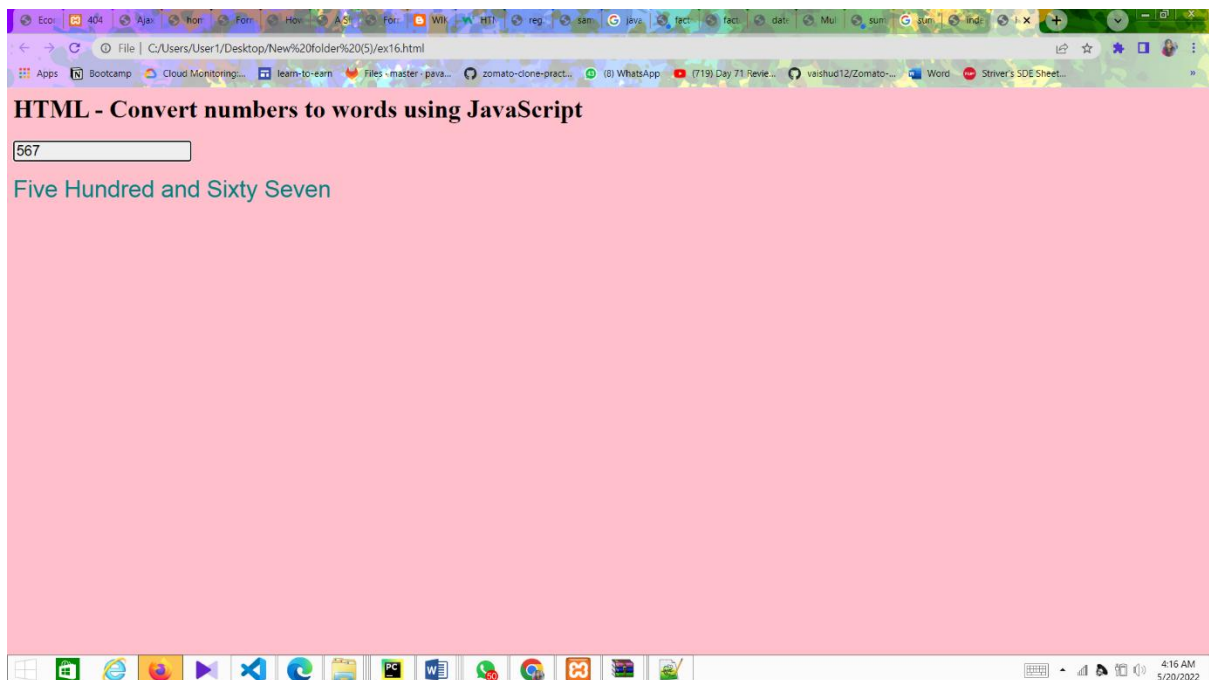
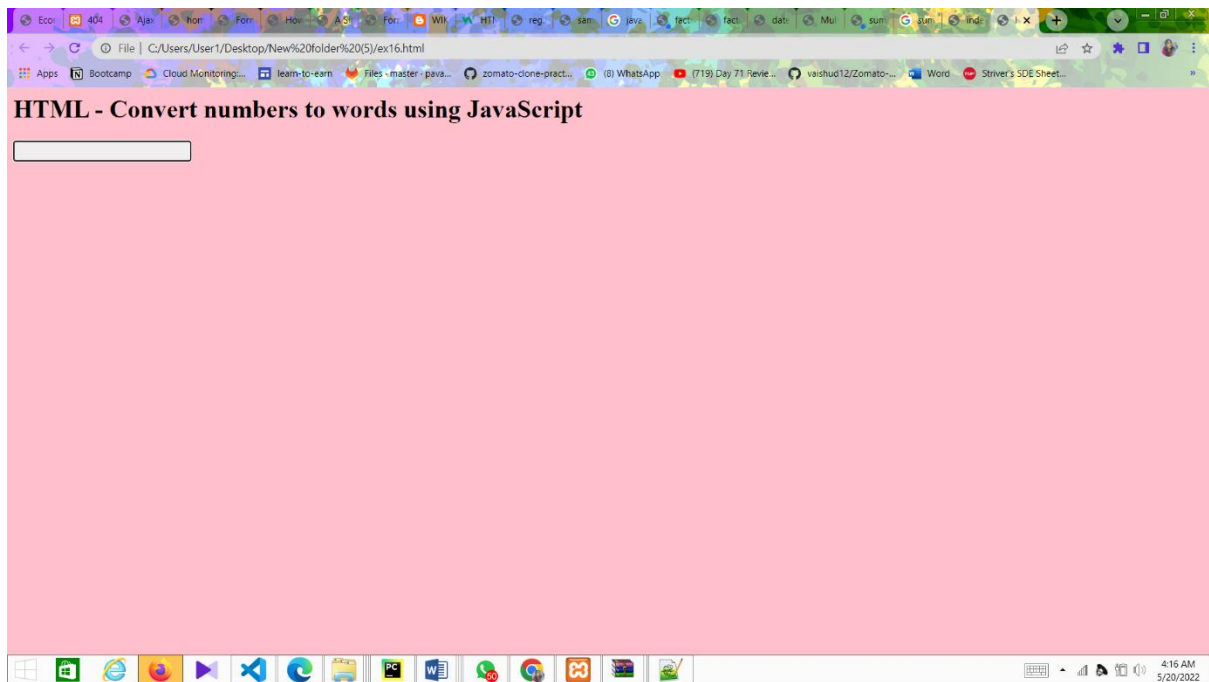
<input id="Text1" type="text" onkeypress="return isNumberKey(event)"
onkeyup="NumToWord(this.value,'divDisplayWords');" maxlength="3"
style="background-color: #efefef; border: 2px solid #CCCCC; font-size: large"
/>

<br /> <br />

<div id="divDisplayWords" style="font-size: 30; color: Teal; font-family:
Arial;"> </div>

</body>
</html>

```

Conclusion:

Thus successfully implemented javascript converting numbers

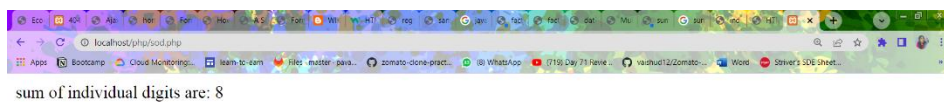
EXPERIMENT NO - 14:

Develop and demonstrate PHP Script for the following problems:

- a) Write a PHP Script to find out the Sum of the Individual Digits.
- b) Write a PHP Script to check whether the given number is Palindrome or not

a) Find out Sum of the individual Digits

```
<?php
$n=323;
$sum=0;
while($n>0)
{
    $r=$n%10;
    $sum+=$r;
    $n=$n/10;
}
echo "sum of individual digits are: $sum";
?>
```



b) Check whether the given number is Palindrome or not

```
<?php
```

```
$n=323;
```

```
$t=$n;
```

```
$rev=0;
```

```
while($n>0)
```

```
{
```

```
$r=$n%10;
```

```
$rev=$rev*10+$r;
```

```
$n=(int)($n/10);
```

```
}
```

```
echo "Reverse digits are: $rev <br>";
```

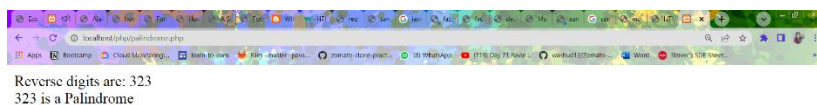
```
if($t==$rev)
```

```
echo "$rev is a Palindrome";
```

```
else
```

```
echo "$rev is not a Palindrome";
```

```
?>
```



Conclusion: thus successfully implemented php programs

EXPERIMENT NO - 15:

Create an XML document that contains 10 users information. Write a Java Program, which takes User Id as input and returns the user details by taking the user information from XML document using DOM parser or SAX parser.

AIM: Takes User Id as input and returns the user details using XML with DOM

PROGRAM:

users.xml

```
<usersinformation>

  <user>

    <rollno>501</rollno>

    <name>aaa</name>

    <branch>cse</branch>

    <college>mrcet</college>

  </user>

  <user>

    <rollno>502</rollno>

    <name>bbb</name>

    <branch>cse</branch>

    <college>mrcet</college>

  </user>

  <user>

    <rollno>503</rollno>

    <name>ccc</name>

    <branch>cse</branch>

    <college>mrcet</college>

  </user>

  <user>
```

```
<rollno>504</rollno>

<name>ddd</name>

<branch>cse</branch>

<college>mrcet</college>

</user>

<user>

    <rollno>505</rollno>

    <name>eee</name>

    <branch>cse</branch>

    <college>mrcet</college>

</user>

<user>

    <rollno>506</rollno>

    <name>fff</name>

    <branch>cse</branch>

    <college>mrcet</college>

</user>

<user>

    <rollno>507</rollno>

    <name>ggg</name>

    <branch>cse</branch>

    <college>mrcet</college>

</user>

<user>

    <rollno>508</rollno>

    <name>hhh</name>

    <branch>cse</branch>

    <college>mrcet</college>

</user>

<user>
```

```

        <rollno>509</rollno>

        <name>iii</name>

        <branch>cse</branch>

        <college>mrcet</college>

    </user>

    <user>

        <rollno>510</rollno>

        <name>jjj</name>

        <branch>cse</branch>

        <college>mrcet</college>

    </user>

</usersinformation>

```

UserDom.java

```

import java.io.File;

import javax.xml.parsers.*;

import org.w3c.dom.*;

import java.util.Scanner;

public class UserDom

{

    public static void main(String args[]) throws Exception

    {

        DocumentBuilderFactory fac=DocumentBuilderFactory.newInstance(); DocumentBuilder
        b=fac.newDocumentBuilder();

        Document doc=b.parse(new File("users.xml"));
        doc.getDocumentElement().normalize(); Element
        root=doc.getDocumentElement(); Scanner in=new
        Scanner(System.in); System.out.println("Enter User ID:");
        int n=in.nextInt();

        int flag=0;
    }
}

```

```

NodeList nl=doc.getElementsByTagName("user"); for(int
i=0;i<nl.getLength();i++) {

Node node=nl.item(i);

if(node.getNodeType()==Node.ELEMENT_NODE)

{

Element e=(Element)node;

int

x=Integer.parseInt(e.getElementsByTagName("rollno").item(0).getTextContent());

if(x==n)

{

System.out.println(root.getNodeName());
System.out.println("-----");

                System.out.println("rollno:\t"+e.getElementsByTagName("rollno").item(0).getTextContent());
                System.out.println("name:\t"+e.getElementsByTagName("name").item(0).getTextContent());
System.out.println("branch:\t"+e.getElementsByTagName("branch").item(0).getTextContent());
System.out.println("college:"+e.getElementsByTagName("college").item(0).getTextContent());

flag=1;

break;

}

else

{

flag=0;

}

}

}

if(flag==0)

System.out.println("User not available");

}

```

```
Microsoft Windows [Version 10.0.19043.1586]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Wilesh>cd Desktop

C:\Users\Wilesh\Desktop>cd 15pra

C:\Users\Wilesh\Desktop\15pra>javac UserDom.java

C:\Users\Wilesh\Desktop\15pra>java UserDom
Enter User ID:
502
-----
Users Information
-----
RollNo: 502
Name: bbb
Branch: cse
College: mrcet

C:\Users\Wilesh\Desktop\15pra>
```

Conclusion:

Thus successfully implemented java xml dom program

EXPERIMENT NO- 16:

Modify the above PHP program to use an xml instead of database

AIM: To design an application that verifies user details from an xml using PHP.

PROGRAM:

db.php

Userlogin.xml:

```
<Authentication>
  <user>
    <username>cse</username>
    <password>5</password>
  </user>
  <user>
    <username>mrcet</username>
    <password>mlrd</password>
  </user>
</Authentication>
```

Loginform.php:

```
<html>
<head>
  <title> Registration page </title>
</head>
<body bgcolor="pink">
```



```

<?php

$xml=simplexml_load_file("Userlogin.xml");

$username=$_POST['uname'];

$password=$_POST['upwd'];

$xmlusername="";

$xmlpassword="";

for($i=0;$i<count($xml);$i++)

{

$xmlusername=$xml->user[$i]->username;

$xmlpassword=$xml->user[$i]->password;

if($xmlusername==$username && $xmlpassword==$password)

{

echo "welcome $username";

die();

}

}

echo "Invalid username or password";

?>

```

</body>

</html>

Loginform1.html:

<html>

<head> <title> Login Page </title> </head>

<body bgcolor="bisque">

<center>

<h1> Login Page </h1>

```

<form action="Loginform.php" method="post">
<table>

<tr> <td> <label> Name: </label> </td>

<td> <input type="text" name="uname" /> </td>
</tr>

<tr> <td> <label> Password: </label> </td>

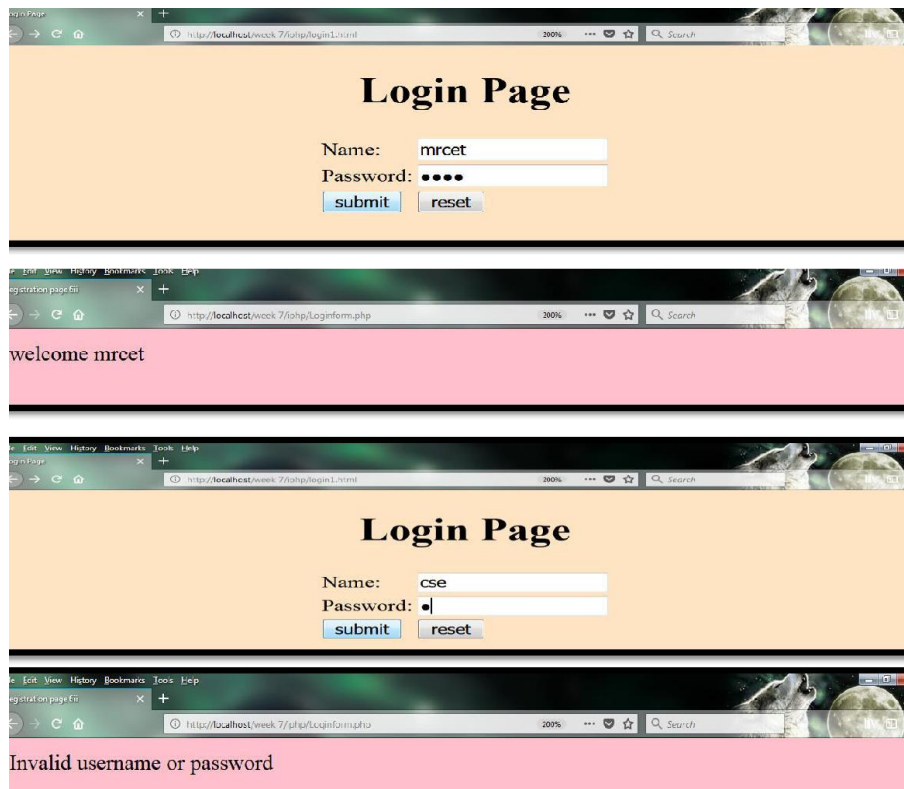
<td> <input type="password" name="upwd" /> </td>
</tr>

<tr> <td> <input type="submit" value="submit" />

<td> <input type="reset" value="reset" /> </td>
</tr>

</table> </form> </center> </body> </html>

```



Conclusion: thus successfully implemented user authentication php