Web Technologies Laboratory 03

Laboratory Continuous Assessment (LCA) [As per Rubrics]

Understanding of the Objective (5)	Performance (5)	Journal Submission and Ethics (Neatness, Handwriting, Timely submission) (5)	Orals (5)	Total (20)	Remarks	Instructor's Sign

Aim: Write a program to design Student registration form by using HTML, CSS and perform following validations using JavaScript:
a) All fields mandatory,
b) Phone Number, Email Address, Zip code validation etc.

Objectives:

1. To understand what is form validation.
2. To know its importance.
3. To learn how to apply various techniques to implement it.

Theory:

1. Different types of form validations.
2. Different client side validations that can be performed on the form.

FAQ:

1. What is JavaScript?
2. Enumerate the differences between Java and JavaScript?
3. What are JavaScript Data Types?
4. Write 3 reasons why Form validations are important.
5. Explain the "required" attribute in HTML5.
6. Write short note on use of regular expressions for form validation.

Output: Screenshots of the output to be attached.

JavaScript Introduction:

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

Why to Learn Javascript

Javascript is a MUST for students and working professionals to become a great Software Engineer especially when they are working in Web Development Domain. Some of the key advantages of learning Javascript:

- Javascript is the most popular **programming language** in the world. Once learnt Javascript, it helps you develop great front-end as well as back-end software using different Javascript based frameworks like jQuery, Node.JS etc.
- Javascript is everywhere, it comes installed on every modern web browser.
- Javascript helps you create really beautiful and crazy fast websites.
- JavaScript usage has now extended to mobile app development, desktop app development, and game development.
- Due to high demand, there is tons of job growth and high pay for those who know JavaScript. You can navigate over to different job sites to see what having JavaScript skills looks like in the job market.
- Great thing about Javascript is that it has tons of frameworks and Libraries already developed which can be used directly in software development to reduce your time to market.

Advantages of JavaScript

The merits of using JavaScript are −

- **Less server interaction** − One can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
- **Immediate feedback to the visitors** − Users don't have to wait for a page reload to see if they have forgotten to enter something.
- **Increased interactivity** − One can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- **Richer interfaces** − One can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

Limitations of JavaScript

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features −

- Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.

- JavaScript cannot be used for networking applications because there is no such support available.
- JavaScript doesn't have any multi-threading or multiprocessor capabilities.

Applications:

- **Client side validation** - This is really important to verify any user input before submitting it to the server and Javascript plays an important role in validating those inputs at front-end itself.
- **Manipulating HTML Pages** - Javascript helps in manipulating HTML page on the fly. This helps in adding and deleting any HTML tag very easily using javascript and modify your HTML to change its look and feel based on different devices and requirements.
- **User Notifications** - You can use Javascript to raise dynamic pop-ups on the webpages to give different types of notifications to your website visitors.
- **Back-end Data Loading** - Javascript provides Ajax library which helps in loading back-end data while you are doing some other processing. This really gives an amazing experience to your website visitors.
- **Presentations** - JavaScript also provides the facility of creating presentations which gives website look and feel. JavaScript provides RevealJS and BespokeJS libraries to build a web-based slide presentations.
- **Server Applications** - Node JS is built on Chrome's Javascript runtime for building fast and scalable network applications. This is an event based library which helps in developing very sophisticated server applications including Web Servers.

JavaScript Development Tools

Some of them are listed here −

- **Microsoft FrontPage** − Microsoft has developed a popular HTML editor called FrontPage. FrontPage also provides web developers with a number of JavaScript tools to assist in the creation of interactive websites.
- **Macromedia Dreamweaver MX** − Macromedia Dreamweaver MX is a very popular HTML and JavaScript editor in the professional web development crowd. It provides several handy prebuilt JavaScript components, integrates well with databases, and conforms to new standards such as XHTML and XML.
- **Macromedia HomeSite 5** − HomeSite 5 is a well-liked HTML and JavaScript editor from Macromedia that can be used to manage personal websites effectively.

Javascript Syntax:

JavaScript can be implemented using JavaScript statements that are placed within the **<script>... </script>** HTML tags in a web page. You can place the **<script>** tags, containing your JavaScript, anywhere within your web page, but it is normally recommended that you should keep it within the **<head>** tags.

A simple syntax:

```
<script ...>
   JavaScript code
</script>
```

The script tag takes two important attributes −

- **Language** − This attribute specifies what scripting language you are using. Typically, its value will be javascript. Although recent versions of HTML (and XHTML, its successor) have phased out the use of this attribute.
- **Type** − This attribute is now recommended to indicate the scripting language in use and its value should be set to "text/javascript".

So your JavaScript segment will look like −

```
<script language = "javascript" type = "text/javascript">
   JavaScript code
</script>
```

Hello World using Javascript

```
<html>
   <body>
      <script language = "javascript" type = "text/javascript">
         <!--
            document.write("Hello World!")
         //-->
      </script>
   </body>
</html>
```

What is Form Validation?

Go to any popular site with a registration form, and you will notice that they provide feedback when you don't enter your data in the format they are expecting. You'll get messages such as:

- "This field is required" (You can't leave this field blank.)
- "Please enter your phone number in the format xxx-xxxx" (The form enforces three numbers followed by a dash, followed by four numbers.)
- "Please enter a valid email address" (Used if your entry is not in the format of "somebody@example.com.")
- "Your password needs to be between 8 and 30 characters long and contain one uppercase letter, one symbol, and a number."

Why do we insist on validating our forms? There are three main reasons:

1. **We want to get the right data, in the right format.** Our applications won't work properly if our users' data is stored in the incorrect format, if they don't enter the correct information, or if they omit information altogether.
2. **We want to protect our users' accounts**. Forcing our users to enter secure passwords makes it easier to protect their account information.
3. **We want to protect ourselves**.

Different types of form validation:

There are two types of form validations on the web:

Client-side validation: Validation that occurs in the browser before the data has been submitted to the server. Client-side validation is more user-friendly than server-side validation because it gives an instant response. Client-side validation is further subdivided into the following categories:

- **JavaScript** validation is coded using JavaScript. This validation is completely customizable.
- **Built-in form validation** uses HTML5 form validation features. This validation generally doesn't require JavaScript. Built-in form validation has better performance than JavaScript. Highly customizable, native validation is not as customizable as JavaScript.

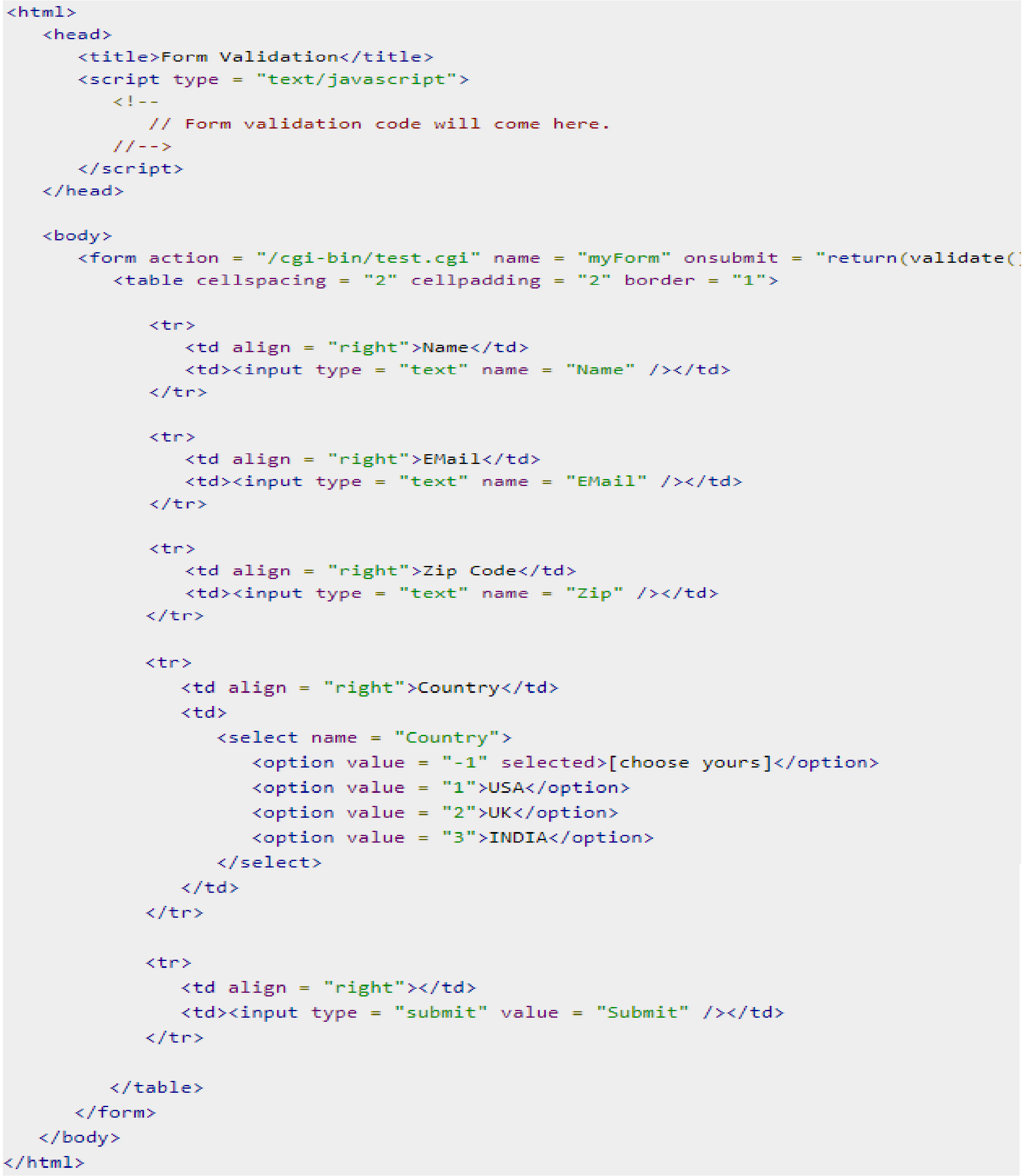
Server-side validation: Validation that occurs on the server after the data has been submitted. Server-side code is used to validate the data before the data is saved in the database or otherwise used by the application. If the data fails validation, a response is sent back to the client with corrections that the user needs to make. Server-side validation is your application's last line of defense against incorrect or malicious data.

JavaScript provides a way to validate form's data on the client's computer before sending it to the web server. Form validation generally performs two functions.

- **Basic Validation** − First of all, the form must be checked to make sure all the mandatory fields are filled in. It would require just a loop through each field in the form and check for data.
- **Data Format Validation** − Secondly, the data that is entered must be checked for correct form and value. Your code must include appropriate logic to test correctness of data.

Example

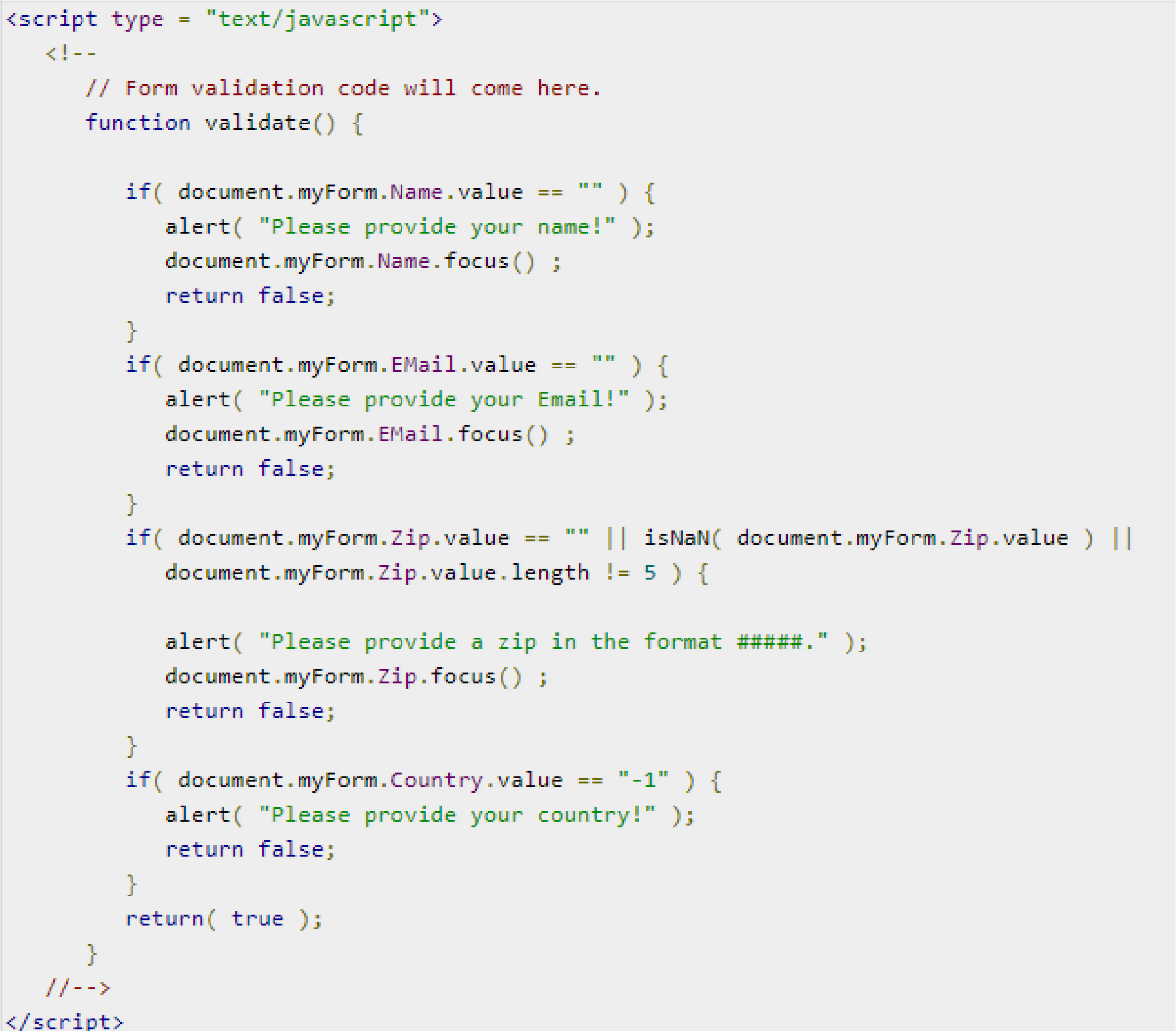
We will take an example to understand the process of validation. Here is a simple form in html format.



```
<html>
   <head>
      <title>Form Validation</title>
      <script type = "text/javascript">
         <!--
            // Form validation code will come here.
         //-->
      </script>
   </head>

   <body>
      <form action = "/cgi-bin/test.cgi" name = "myForm" onsubmit = "return(validate()
         <table cellspacing = "2" cellpadding = "2" border = "1">

            <tr>
               <td align = "right">Name</td>
               <td><input type = "text" name = "Name" /></td>
            </tr>

            <tr>
               <td align = "right">EMail</td>
               <td><input type = "text" name = "EMail" /></td>
            </tr>

            <tr>
               <td align = "right">Zip Code</td>
               <td><input type = "text" name = "Zip" /></td>
            </tr>

            <tr>
               <td align = "right">Country</td>
               <td>
                  <select name = "Country">
                     <option value = "-1" selected>[choose yours]</option>
                     <option value = "1">USA</option>
                     <option value = "2">UK</option>
                     <option value = "3">INDIA</option>
                  </select>
               </td>
            </tr>

            <tr>
               <td align = "right"></td>
               <td><input type = "submit" value = "Submit" /></td>
            </tr>

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

Basic Form Validation

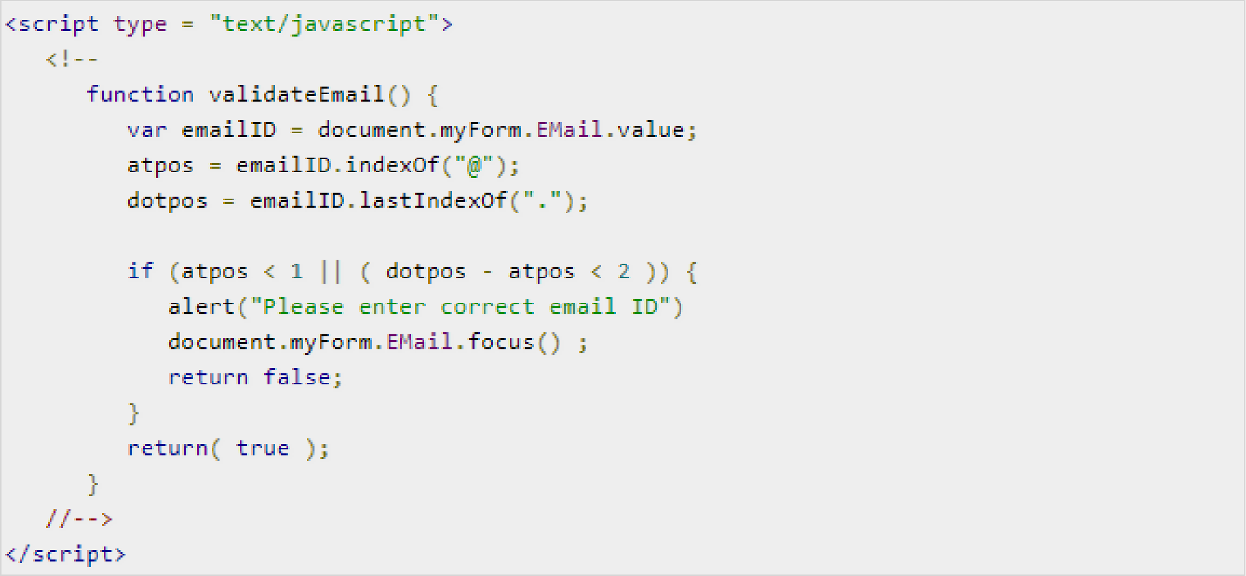
In the above form, we are calling **validate()** to validate data when **onsubmit** event is occurring. The following code shows the implementation of this validate() function.



```
<script type = "text/javascript">
   <!--
      // Form validation code will come here.
      function validate() {

         if( document.myForm.Name.value == "" ) {
            alert( "Please provide your name!" );
            document.myForm.Name.focus() ;
            return false;
         }
         if( document.myForm.EMail.value == "" ) {
            alert( "Please provide your Email!" );
            document.myForm.EMail.focus() ;
            return false;
         }
         if( document.myForm.Zip.value == "" || isNaN( document.myForm.Zip.value ) ||
            document.myForm.Zip.value.length != 5 ) {

            alert( "Please provide a zip in the format #####." );
            document.myForm.Zip.focus() ;
            return false;
         }
         if( document.myForm.Country.value == "-1" ) {
            alert( "Please provide your country!" );
            return false;
         }
         return( true );
      }
   //-->
</script>
```

Data Format Validation

The following example shows how to validate an entered email address. An email address must contain at least a '@' sign and a dot (.). Also, the '@' must not be the first character of the email address, and the last dot must at least be one character after the '@' sign.



```
<script type = "text/javascript">
   <!--
      function validateEmail() {
         var emailID = document.myForm.EMail.value;
         atpos = emailID.indexOf("@");
         dotpos = emailID.lastIndexOf(".");

         if (atpos < 1 || ( dotpos - atpos < 2 )) {
            alert("Please enter correct email ID")
            document.myForm.EMail.focus() ;
            return false;
         }
         return( true );
      }
   //-->
</script>
```

Alphanumeric validation using Regular Expressions

Regular Expressions can be used for powerful pattern matching. These are very useful for the form validation examples such as email address, phone number patterns etc.

Javascript function to check if a field input contains letters and numbers only:

```
function alphanumeric(inputtxt)
{
 var letterNumber = /^[0-9a-zA-Z]+$/;
 if((inputtxt.value.match(letterNumber))
 {
   return true;
 }
else
 {
   alert("message");
   return false;
 }
}
```

Conclusion:

Written and successfully executed Client Side validations on a web page using JavaScript.