## Experiment No. 05

**Title:** Implement Javascript validation for Website Forms.

**Batch: B2** **Roll No.: 1914078** **Experiment No.:5**

**Aim**: To Implement Javascript validation for Website Forms.

---

**Resources needed: Notepad++, Web Browser**

---

**Theory:**

JavaScript is a scripting language produced by Netscape for use within HTML Web pages. JavaScript is loosely based on Java and it is built into all the major modern browsers. JavaScript is a lightweight, interpreted programming language, Complementary to and integrated with Java , Complementary to and integrated with HTML , Open and cross-platform and is case sensitive.

Data validation is the process of ensuring that user input is clean, correct, and useful. Typical validation tasks are:

- Has the user filled in all required fields?
- Has the user entered a valid date?
- Has the user entered text in a numeric field?
- Most often, the purpose of data validation is to ensure correct user input.

Validation can be defined by many different methods, and deployed in many different ways.

- Server side validation is performed by a web server, after input has been sent to the server.
- Client side validation is performed by a web browser, before input is sent to a web server.

For example HTML form validation can be done by JavaScript. If a form field (fname) is empty, this function alerts a message, and returns false, to prevent the form from being submitted:

```
function validateForm() {
  var x = document.forms["myForm"]["fname"].value;
  if (x == "") {
    alert("Name must be filled out");
    return false;
  }
}
```

**Activity**
**Add validations for the Website Forms Such as**

1) **Name should string**
2) **Roll number should a number**
3) **Email id should have @ and . in it**
4) **Telephone number should be ten digit number.**

**Students need to add various validations to their form input as per the requirement of the user interface.**

**Results: (Program printout with output)**

**Code:**

```
<!DOCTYPE html>
<html>

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

   <title>Sign Up</title>

   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css"
rel="stylesheet"
      integrity="sha384-
giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1"
crossorigin="anonymous">
   <link rel="stylesheet" href="styles.css">

<style>
   body{
     background-color: lightblue;
   }
   .form{
        padding-left:5%;
        padding-right:5%;
        position: relative;
        background-color:#ffffff40;
        padding-bottom:20px;
        width:40%;
        margin: auto;
        padding-top:2%;
   }

   h2{
     font-family: "Comic Sans MS", "Comic Sans", cursive;
     font-size: 40px;
     color: #32b0c9;
     text-align: center;
```

```
    }

    .row input{
        background-color: white;
        color: black;
        width:90%;
        height:25px;
        border-radius:5px;
        border:none;
       }

    .row select{
        background-color: white;
        color: black;
        width:90%;
        height:25px;
        border-radius:5px;
        border:none;
       }

    .row label{
      color:black;
    }
    .button{
        margin-right:30px;
        height:25px;
        width:80px;
        color:white;
        border:none;
        background-color:#ffa30f;
        cursor:pointer;
        border-radius:5px;
       }
li a:hover {
  background-color: lightgrey;
}
</style>
</head>

<script>
// Defining a function to display error message
function printError(elemId, hintMsg) {
    document.getElementById(elemId).innerHTML = hintMsg;
}

// Defining a function to validate form
function validateForm() {
    // Retrieving the values of form elements
    var name = document.contactForm.name.value;
    var middlename=document.contactForm.middlename.value;
    var lastname = document.contactForm.lastname.value;
    var email = document.contactForm.email.value;
    var mobile = document.contactForm.mobile.value;
    var state = document.contactForm.state.value;
```

```
var password = document.contactForm.password.value;
var confirmPassword =document.contactForm.confirmPassword.value;


// Defining error variables with a default value
var nameErr =lastnameErr= emailErr = mobileErr = stateErr = passErr= true;

// Validate name
if(name == "") {
    printError("nameErr", "*Please enter name");
} else {
    var regex = /^[a-zA-Z\s]+$/;
    if(regex.test(name) === false) {
        printError("nameErr", "*Please enter a valid name");
    } else {
        printError("nameErr", "");
        nameErr = false;
    }
}

// Validate Last Name
if(lastname == "") {
    printError("lastnameErr", "*Please enter Last Name");
} else {
    var regex = /^[a-zA-Z\s]+$/;
    if(regex.test(lastname) === false) {
        printError("lastnameErr", "*Please enter a valid  Last name");
    } else {
        printError("lastnameErr", "");
        lastnameErr = false;
    }
}

// Validate email address
if(email == "") {
    printError("emailErr", "*Please enter your email address");
} else {
    // Regular expression for basic email validation
    var regex = /^\S+@\S+\.\S+$/;
    if(regex.test(email) === false) {
        printError("emailErr", "*Please enter a valid email address");
    } else{
        printError("emailErr", "");
        emailErr = false;
    }
}

// Validate mobile number
if(mobile == "") {
    printError("mobileErr", "*Please enter your mobile number");
} else {
    var regex = /^[1-9]\d{9}$/;
    if(regex.test(mobile) === false) {
```

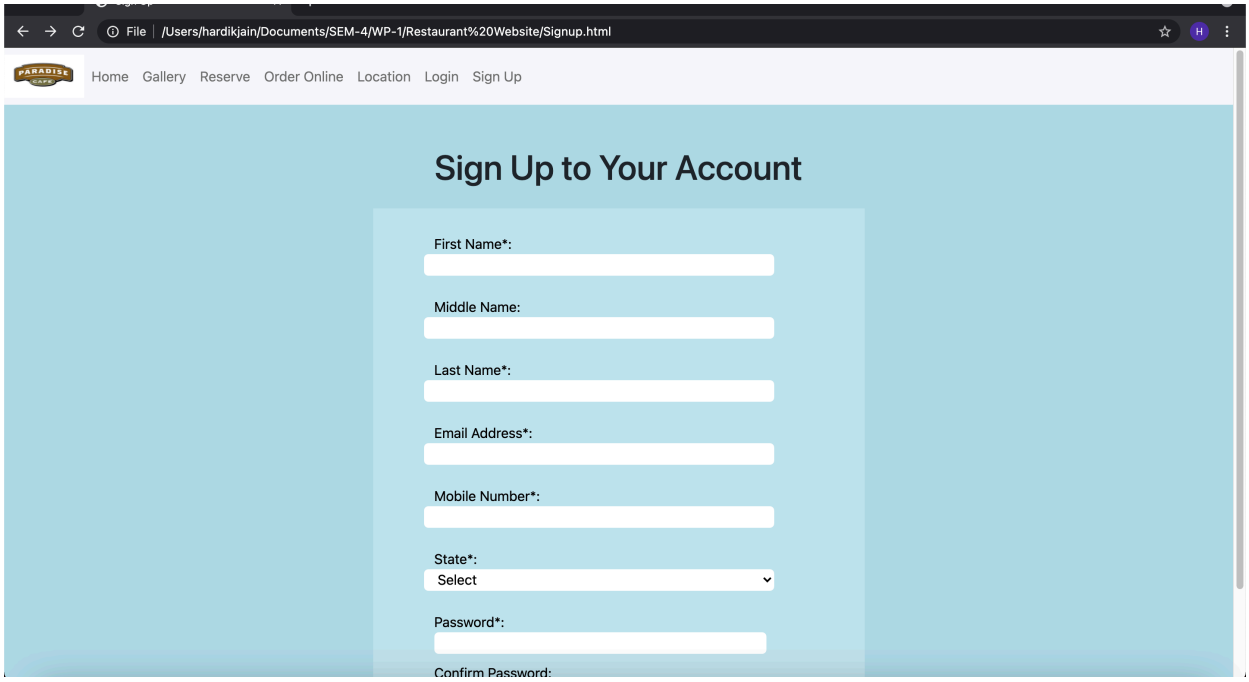
```
            printError("mobileErr", "*Please enter a valid 10 digit mobile number");
        } else{
            printError("mobileErr", "");
            mobileErr = false;
        }
    }

    if(state == "Select") {
        printError("stateErr", "*Please select a state");
    } else {
        printError("stateErr", "");
        stateErr = false;
    }

    if (password != confirmPassword) {
        printError("passErr","*Passwords do not match.");
    }else{
        printError("passErr","")
        passErr= false;
    }


    // Prevent the form from being submitted if there are any errors
    if((nameErr || emailErr || mobileErr || stateErr|| passErr) == true) {
      return false;
    } else {
        // Creating a string from input data for preview
        var dataPreview = "You've entered the following details: \n" +
                          "Full Name: " + name + "\n" +
                          "Middle Name: " + middlename + "\n" +
                          "Last Name: " + lastname+ "\n"+
                          "Email Address: " + email + "\n" +
                          "Mobile Number: " + mobile + "\n" +
                          "State: " + state + "\n";
        // Display input data in a dialog box before submitting the form
        alert(dataPreview);
    }
};
</script>

<body>
<nav class="navbar navbar-expand-lg navbar-light bg-light">
   <img align="left"
src="https://www.brandcrowd.com/gallery/brands/pictures/picture13897453145721.jpg"
width="100px" height="50px">
     <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
       <span class="navbar-toggler-icon"></span>
     </button>

     <div class="collapse navbar-collapse" id="navbarSupportedContent">
       <ul class="navbar-nav mr-auto mt-2 mt-lg-0">
         <li class="nav-item">
```

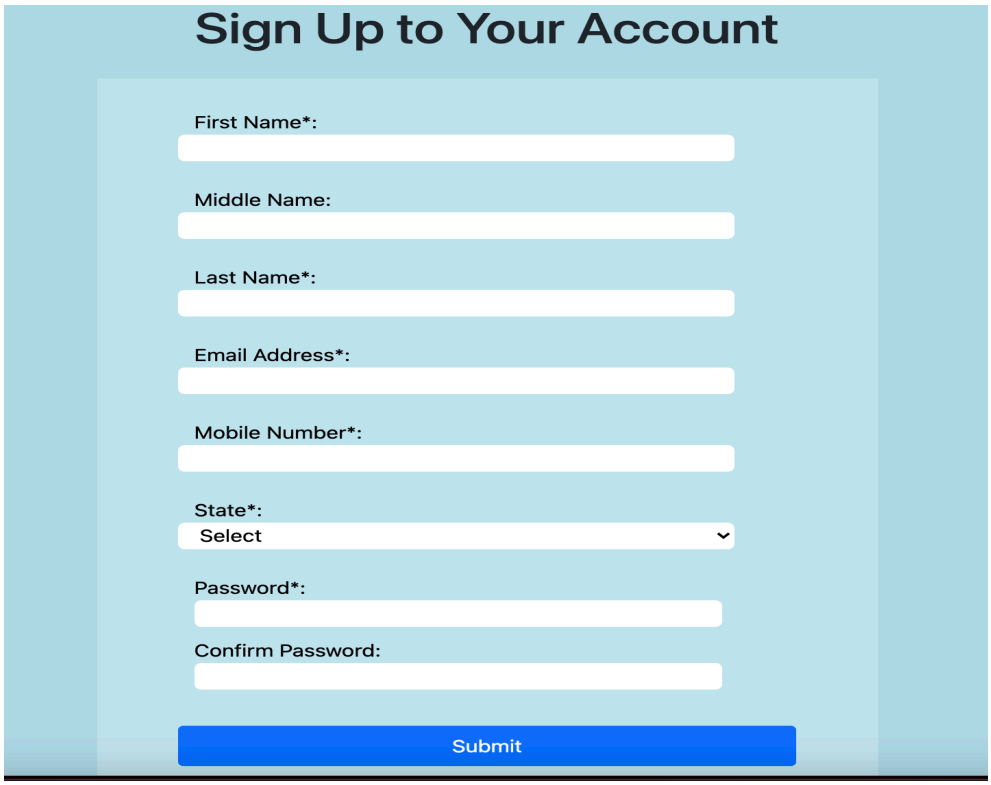
```
          <a class="nav-link" href="Restaurant_html.html">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="Gallery.html">Gallery</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="Reserve.html">Reserve</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="Order_Online.html">Order Online</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Location</a>
        </li>
        <li class="nav-item">
          <a class="nav-link"  href="#about">Login</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="Signup.html">Sign Up</a>
        </li>

      </ul>

    </div>
  </nav>
<br><br>

      <h1 class="pb-3" style="text-align: center;">Sign Up to Your Account</h1>


          <form name="contactForm" onsubmit="return validateForm()"
action="file:///Users/hardikjain/Documents/SEM-4/WP-
1/TUT/TUT%206/formvalidation.html" method="post" class="form">

            <div class="row">
              <label>First Name*:</label><br>
              <input type="text" name="name">
              <div class="error" id="nameErr"></div>
            </div><br>
            <div class="row">
              <label>Middle Name:</label><br>
              <input type="text" name="middlename">
            </div><br>
            <div class="row">
              <label>Last Name*:</label><br>
              <input type="text" name="lastname">
              <div class="error" id="lastnameErr"></div>
            </div><br>
            <div class="row">
              <label>Email Address*:</label><br>
              <input type="text" name="email">
              <div class="error" id="emailErr"></div>
            </div><br>
            <div class="row">
```

```
            <label>Mobile Number*:</label><br>
            <input type="text" name="mobile" maxlength="10">
            <div class="error" id="mobileErr"></div>
          </div><br>
          <div class="row">
            <label>State*:</label><br>
            <select name="state">
              <option>Select</option>
              <option>Goa</option>
              <option>Andaman Nicobar</option>
              <option>Maharashta</option>
              <option>Gujarat</option>
            </select>
            <div class="error" id="stateErr"></div>
          </div><br>
          <div class="row">
           <div>
            <div style="padding-bottom: 10px;">
            <label>Password*:</label><br>
            <input type="password" name="password"/><br>
             </div>
             <div style="padding-bottom: 10px;">
            <label>Confirm Password:</label><br>
            <input type="password" name="confirmPassword"/>
            </div>
           </div>
            <div class="error" id="passErr"></div>
          </div><br>
          <div class="row">
            <button href="Restaurant_html.html" type="submit" value="Submit" class="btn
btn-primary">Submit</button><br>
          </div>

        </form>
       </div>

       </div>
    </div>

</body>

</html>
```

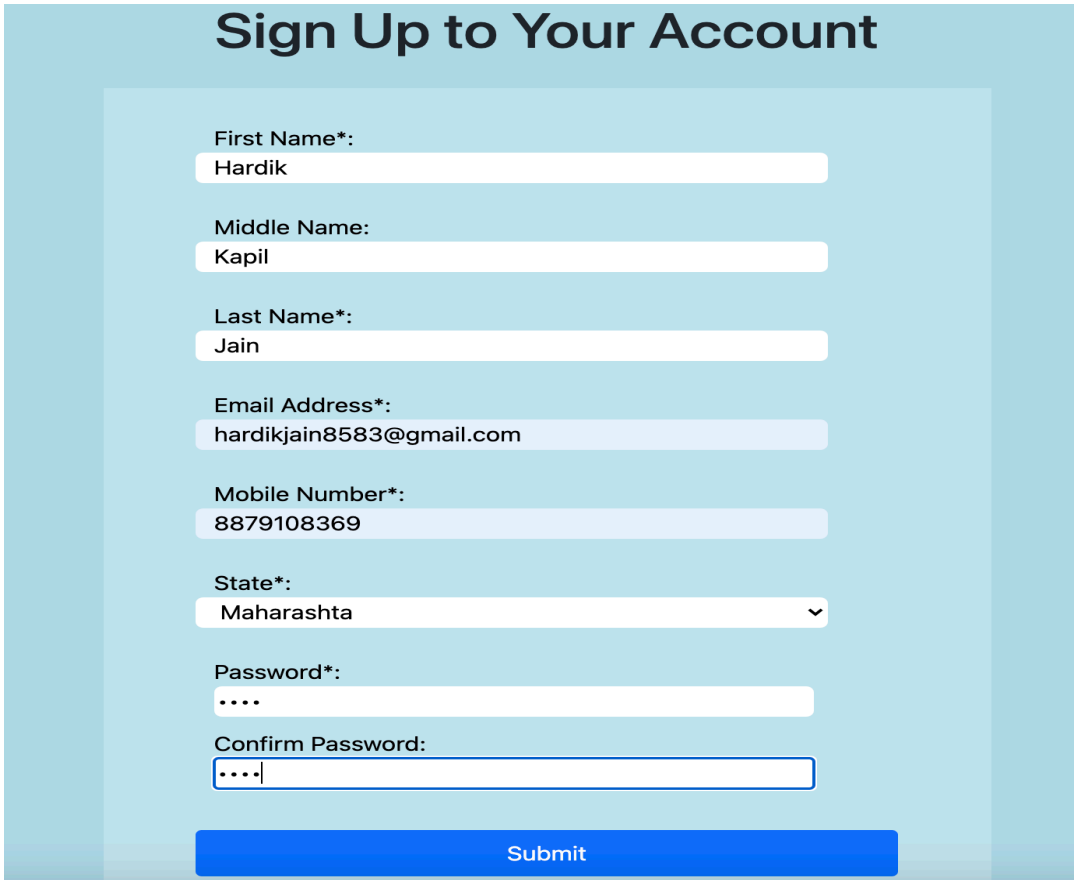
**Output:**

**We implement this experiment on the 'Signup' Page of our website and the form looks like this :**

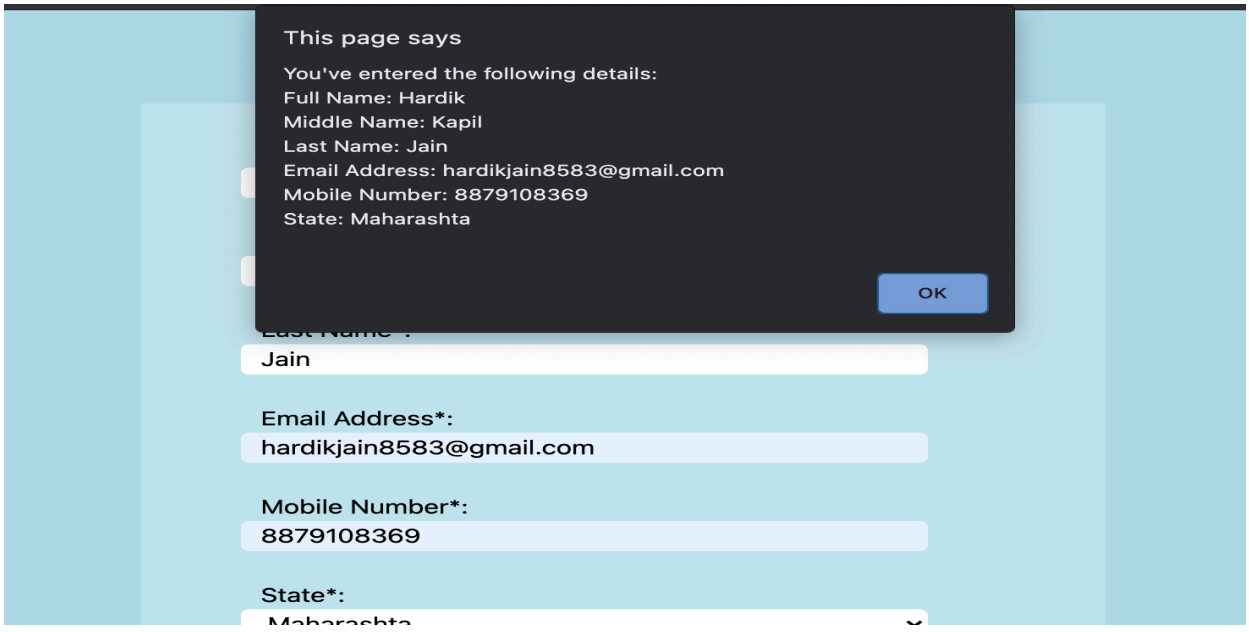


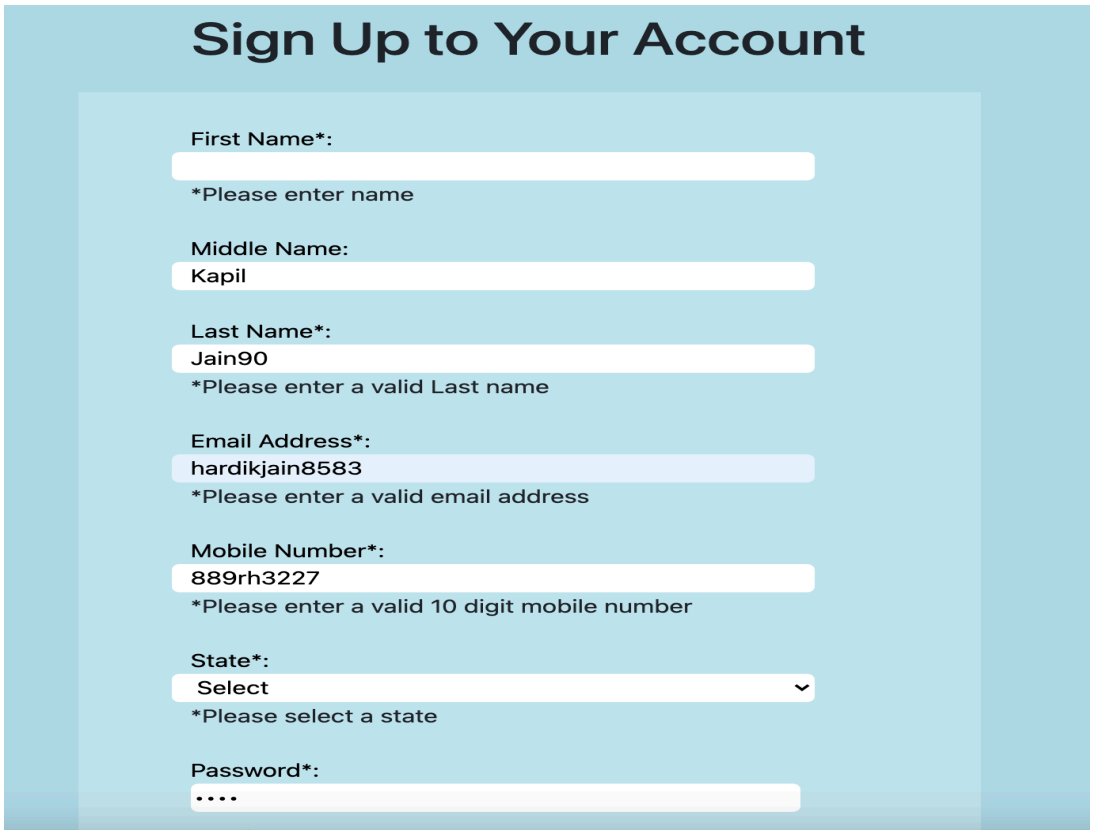
- **Here, all the correct information is filled in each input type including the password.**



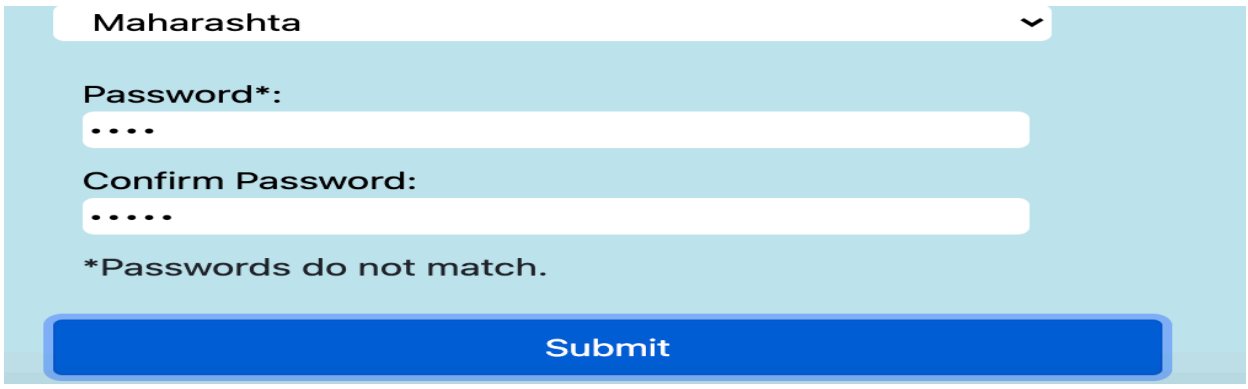

- **After clicking on the submit button we get this output which confirms the details we have entered in the form of alert-**



- **Now, we deliberately add information in the wrong form as input and click on the submit button to see all the errors.**
- **We see that on submitting we get the errors because we have added validations for the Website Forms Such as**
  - **Name should string**
  - **State must be selected.**
  - **Email id should have @ and . in it**
  - **Telephone number should be ten digit number.**




- **Here we see, even if the passwords do not match, we get an error on clicking submit button and the form does not get submitted.**



**Questions:**

**Q1) Why to carry out Validation at client side using scripting language?**

**Answer:**

- When accepting any form of user input it's necessary to validate the submitted information in order to ensure its accuracy and validity.
- Validating input is obviously a necessity, so the question is really about how and where it should occur.
- The first step in validation takes place on the client-side. Using a combination of Javascript and HTML it's possible to provide fairly robust validation rules. The main advantage of client-side validation is that it requires no HTTP request/response cycle since it all takes place on the client's machine, and thus near instantaneous feedback may be provided. This allows for a very responsive and clear user experience.
- HTML 5 provides some very handle validations right out of the box. Notice in the code above the type and the required properties of the input fields. If a user skips a required field, enters an invalid email address into the email field, or enters anything other than a number into the age field, then they'll receive a very informative visual notification of their error and the form will not submit.
- While client-side validation can provide a responsive, intuitive and smooth user experience, it's necessary to remember that it's not reliable and adds no security as it can be easily turned off by the user if they simply disable JavaScript or remove the HTML validations using standard developer tools.

- **Q2) What is the difference between client side validation and server side validation?**

**Answer:**

| Client-Side Validation | Server-Side Validation |
|---|---|
| In client-side validation method, all the input validation and error recovery process are carried out on the client side. | In server-side validation method, all the input validations and error recovery process are carried out on the server side. |
| This sort of approval is finished utilizing JavaScript. | This sort of approval is done at the server side where application dwells. |
| This kind of approval is quick and simpler for client. | This kind of approval is similar slower for client. |
| This sort of approval is for the most part shaky as the end client has simple access to the code of the page. | This sort of approval is more secure as the end client does not have simple access to the code. |
| This sort of approval is for the most part done first. | This kind of approval is for the most part done after customer side approval. |

| | |
|---|---|
| This sort of approval is done to lessen the blunders which can come amid server side preparing. | This kind of approval is done to safeguard any sort of harm which can come amid customer side approval. |
| It can be done using JavaScript, AJAX, HTML5 etc. | It can be done using programming languages like C#.NET, VB.NET, Java and JSP, Python, Ruby on Rails etc. |

**Outcomes:**

**CO3:** Apply Javascript for Web Application development.

**Conclusion: (Conclusion to be based on the outcomes achieved):**

Through this experiment we learnt To Implement Javascript validation for Website Forms. We added various types of JS Validations on our own Website's Sign Up page and successfully implemented the code.

**Grade: AA / AB / BB / BC / CC / CD /DD**

Signature of faculty in-charge with date

**References:**

**Books/ Journals/ Websites:**


- "Web technologies: Black Book", Dreamtech Publications
- http://www.w3schools.com