Web Technologies Laboratory 03

Name: Shabbar Adamjee
Roll No.: PB57
PRN: 1032221508

Aim: Client-side Form Validations using JavaScript, DOM real-time update, JQuery to develop Ajax based applications.

Aim: Write a program to perform following form validations using JavaScript:

a) All fields mandatory,
b) Phone number, Email Address, Zip code Validation etc.
Include JavaScript to access and manipulate Document Object Model (DOM) objects in an HTML web page.
Include JQuery to develop to develop your application as an Ajax based application.

Objectives:

1. To understand what form validation is.
2. To learn basic functioning of DOM objects.
3. To learn how to apply various techniques to implement it.

Theory:

1. Different types of form validations.
   Form validation is crucial in web development to ensure that the data submitted by users meets the required criteria. There are two main types of form validations:

   Client-Side Validation

- **Performed in the Browser**: This type of validation is done on the user's browser before the data is sent to the server.
- **Uses HTML5, JavaScript, or jQuery**: Common techniques include using HTML5 attributes (e.g., required, pattern), JavaScript functions, or jQuery plugins.
- **Fast Feedback**: Users get immediate feedback, allowing them to correct mistakes before submission.
- **Examples**:
  - Required fields (<input type="text" required>).
  - Pattern matching (e.g., validating an email format).
  - Length restrictions (e.g., minimum and maximum character length).

Server-Side Validation

- **Performed on the Server**: After the data is submitted, it is validated on the server side.
- **Security**: It ensures that data is validated even if client-side validation is bypassed or disabled.
- **Languages Used**: This can be implemented using server-side languages like PHP, Python, Java, etc.
- **Examples**:
  - Checking for required fields and valid data types.
  - Ensuring that the data meets business rules (e.g., username uniqueness).
  - Preventing injection attacks (e.g., SQL injection).

2. HTML Document Object Model.

The HTML Document Object Model (DOM) is a programming interface for web documents. It represents the structure of an HTML document as a tree of objects, allowing scripts to interact with the content, structure, and style of a webpage.

Key Concepts:

- **Tree Structure**: The DOM represents an HTML document as a tree of nodes, where each node corresponds to a part of the document (e.g., elements, attributes, text).
- **Manipulation**: You can use JavaScript or other scripting languages to manipulate the DOM. This allows for dynamic changes to the content, structure, and style of a webpage.
- **Accessing Elements**: Elements can be accessed and modified using methods like getElementById, getElementsByClassName, querySelector, etc.
- **Event Handling**: The DOM allows for event-driven programming. Events like clicks, keypresses, or page loads can trigger JavaScript functions.

3. What is JQuery? Write various JQuery Selectors.

jQuery is a fast, small, and feature-rich JavaScript library. It simplifies tasks like HTML document traversal and manipulation, event handling, animation, and Ajax with an easy-to-use API that works across a multitude of browsers.

- **Simplifies JavaScript**: jQuery provides easy-to-use functions that simplify common JavaScript tasks.
- **Cross-Browser Compatibility**: jQuery handles the differences between browsers, providing a consistent experience.
- **Extensible**: There are many plugins available that extend jQuery's functionality.

Various jQuery Selectors

jQuery selectors are used to "find" (or select) HTML elements based on their name, id, classes, types, attributes, values, and more.

- **Basic Selectors**:
  - $("#id"): Selects an element by its ID.
  - $(".class"): Selects all elements with a given class.
  - $("element"): Selects all elements of a specific type (e.g., $("p") selects all paragraphs).
- **Attribute Selectors**:
  - $("[attribute]"): Selects elements with a specific attribute (e.g., $("input[name='email']")).
  - $("[attribute='value']"): Selects elements with a specific attribute value.
- **Hierarchy Selectors**:
  - $("parent > child"): Selects all child elements of a specific parent.
  - $("ancestor descendant"): Selects all descendants of a specific ancestor.
- **Pseudo-Class Selectors**:
  - :first: Selects the first matched element (e.g., $("p:first")).
  - :last: Selects the last matched element (e.g., $("p:last")).
  - :even: Selects even elements (e.g., $("li:even")).
- **Form Selectors**:
  - :input: Selects all input elements (e.g., text fields, checkboxes, etc.).
  - :checked: Selects all checked elements (e.g., $("input:checked")).

FAQ:

1. Write 3 reasons why Form validations are important.

Data Integrity:

- Form validations ensure that the data submitted by users is accurate, complete, and meets the required criteria. This prevents incorrect or incomplete data from entering the database, maintaining the overall integrity of the system.

User Experience:

- Validations provide immediate feedback to users, helping them correct mistakes before submitting the form. This enhances the user experience by making it clear what is expected and reducing frustration.

Security:

- Validations, especially server-side, protect against malicious inputs, such as SQL injections or script injections. They act as a first line of defense, ensuring that the data is safe before it reaches the server for processing.

2. Give an example of how to modify an attribute value using DOM.

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

<img id="myImage" src="old-image.jpg" alt="Old Image">

<button onclick="changeImage()">Change Image</button>

<script>
function changeImage() {
  var img = document.getElementById("myImage");

  img.src = "new-image.jpg";
}
</script>

</body>
</html>
```

3. What is jQuery Ajax?

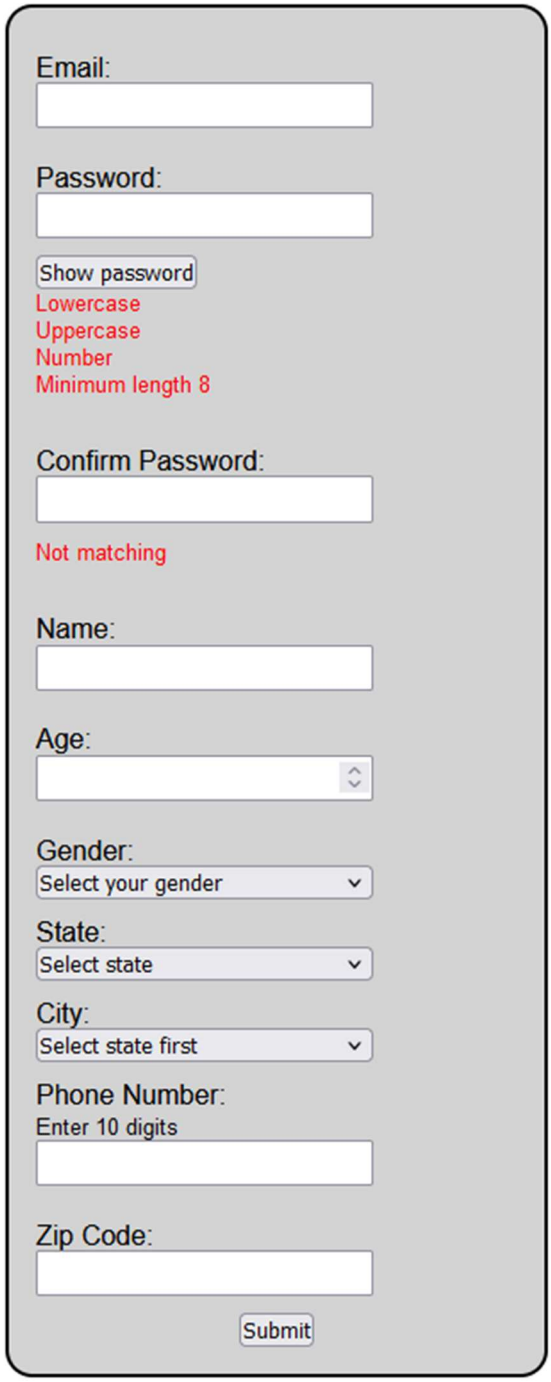
jQuery Ajax is a powerful feature of the jQuery library that allows you to perform asynchronous HTTP requests (Ajax requests) without reloading the entire webpage. Ajax stands for Asynchronous JavaScript and XML, and it enables web pages to update content dynamically by exchanging data with the server in the background.

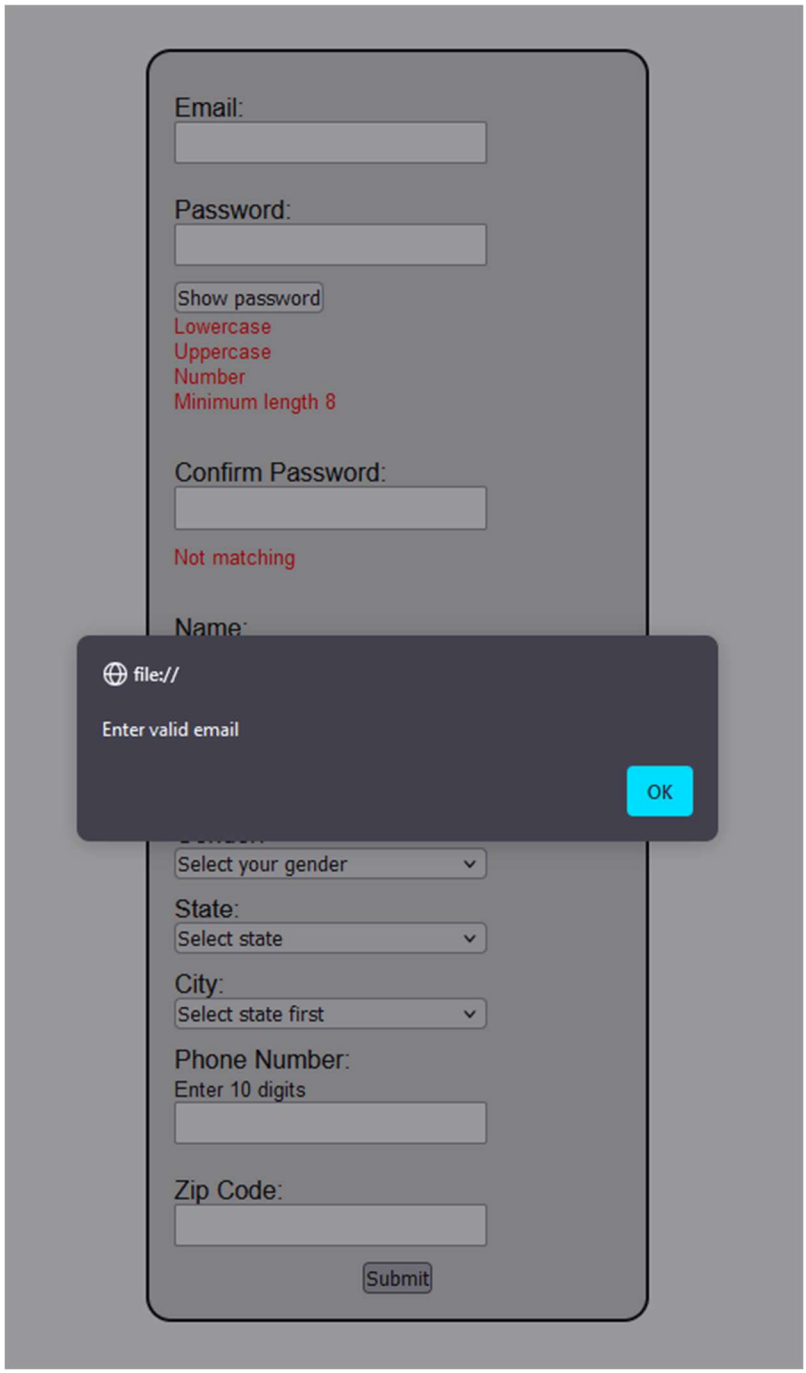
Key Features:

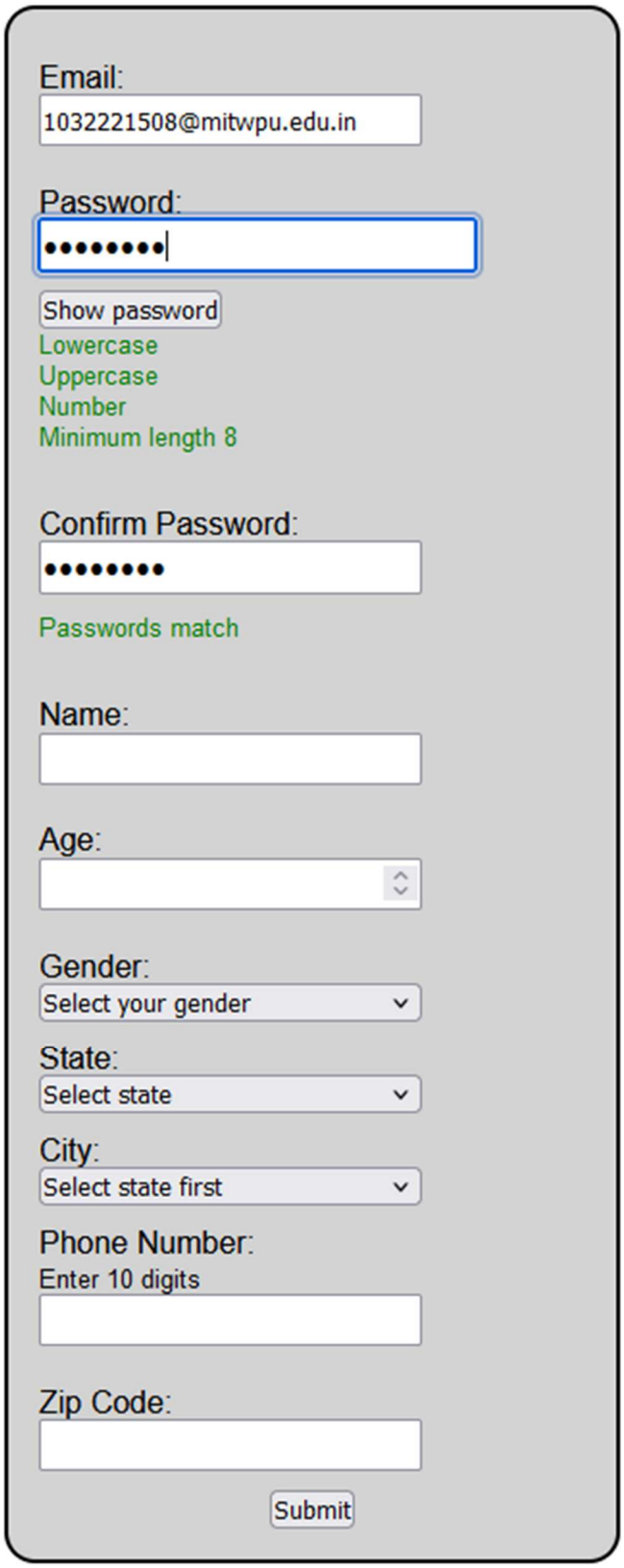
- **Asynchronous Requests**: jQuery Ajax allows you to send and receive data from the server asynchronously, meaning the rest of the webpage can continue to function while the request is being processed.
- **Cross-Browser Compatibility**: jQuery handles the differences between browsers, ensuring that Ajax requests work consistently across different platforms.

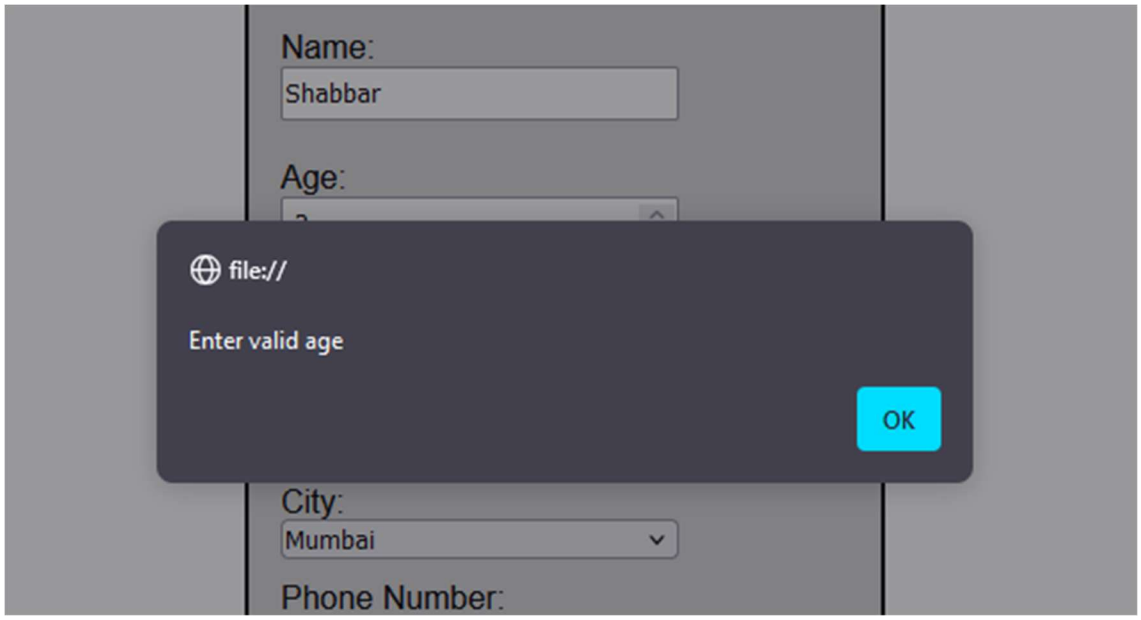
- **Data Handling**: You can send and receive data in various formats such as JSON, XML, HTML, or plain text.
- **Error Handling**: Provides mechanisms to handle errors, timeouts, and other issues that may arise during the request.

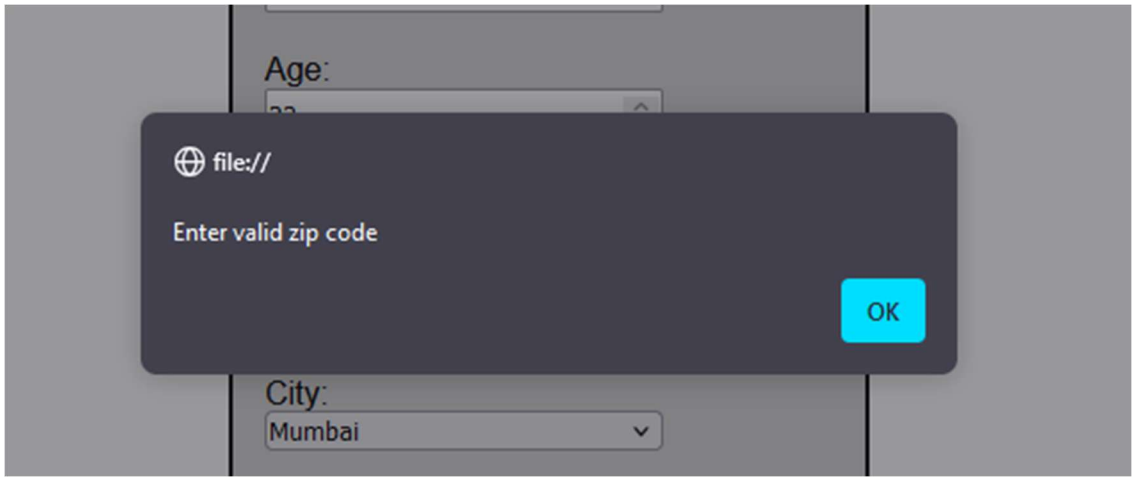
Output: Screenshots of the output to be attached.






Email:
1032221508@mitwpu.edu.in
Password:
Show password
Lowercase
Uppercase
Number
Minimum length 8
Confirm Password:
Passwords match
Name:
Age:
Gender:
Select your gender
State:
Select state
City:
Select state first
Phone Number:
Enter 10 digits
Zip Code:
Submit





Problem Statement:

Write a program to design Student registration form by using HTML, CSS having following fields: Username, Email, Phone number, Password, Confirm Password and write external javascript code to achieve following validations

- Fields should not be empty. If spaces are entered those should be considered empty
- Phone number must accept only numeric values and it should be 10 digits
- Password length must be at least 7 and it should contain at least one capital letter, one digit and one special character from the set (&,$,#@)
- Value entered in password field and confirm password fields must match

Email address must contain @ sign and a ., there should be few letters before the @ sign, there should be three letters between @ sign and a . There must be 3 or 2 letters after the . (hint: Use regular expression)

Write a client-side script with JavaScript to access and manipulate Document Object Model (DOM) objects in an HTML web page. Develop a dynamic web page using javascript and DOM. Make use of the following for accessing elements

- getElementById, getElementsByTagName,getElementsByClassName
- Change the text using innerHTML property
- Change the CSS properties like color, position of a particular element on the page
- Change the image source after clicking on a button
- Add a text node and attach it to a parent node
- Delete a node

Include jQuery to perform following operations:

- Change button text using jQuery.
- Set background-image using jQuery CSS property.
- Access HTML form data using jQuery.
- Add attribute using jQuery

Use this reference link for jQuery : https://www.w3resource.com/jquery-exercises/part1/index.php