Tutorial No. 6

Title: Form Validation with JavaScript

KJSCE/IT/SYBTech/SEMIV/WP-I (Tut)/ 2021-22

Batch:A2

Roll No.:16010421063

Tutorial No.: 6

Aim: Design a web page containing a form and perform appropriate client side validation on that form

---

Theory

What is form validation:- Form validation helps us to ensure that users fill out forms in the correct format, making sure that submitted data will work successfully with our applications. 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" (it enforces three numbers followed by a dash, followed by four numbers)
- "Please enter a valid e-mail address" (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"

Client-side validation is validation that occurs in the browser before the data has been submitted to the server. This is more user-friendly than server-side validation as it gives an instant response. This can be further subdivided:

- JavaScript validation is coded using JavaScript. It is completely customizable. • Built-in form validation using HTML5 form validation features. This generally does not require JavaScript. Built-in form validation has better performance, but it is not as customizable as JavaScript**.**

Activity

Apply validation in form using Regex/Built in validation/Event handling. Choose appropriate techniques

---

Results: (Program printout with output)

```
<!DOCTYPE html>

<html lang="en">

   <head>

       <meta charset="UTF-8" />

       <meta http-equiv="X-UA-Compatible" content="IE=edge" />

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

       <title>Document</title>

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

   </head>

   <body>

       <form onsubmit="myFunction(event)" action="#" class="form">

           <h1>Register</h1>

           <div>

               <div>

                   <label for="name"> Enter Username </label>

                   <input type="text" name="username" id="username"
required />

               </div>

               <div>

                   <label for="name"> Enter Full Name </label>

                   <input type="text" name="name" id="name" required
```

```
/>

                </div>

                <div>

                    <label for="email"> Enter Email </label>

                    <input type="email" name="email" id="email"
required />

                </div>

                <div>

                    <label for="MNumber"> Enter Mobile number </label>

                    <input

                        type="tel"

                        name="MNumber"

                        id="MNumber"

                        pattern="[0-9]{10}"

                        required

                    />

                </div>

                <div>

                    <label for="name"> Enter Password </label>

                    <input type="password" name="password" id="pass"
required />

                </div>
```

```
                <div>
                    <label for="name"> Reenter password </label>
                    <input type="password" name="password" id="repass"
required />
                </div>

                <input type="submit" value="Submit" />
            </div>
            <div class="register">
                <a href="/index.html">Login</a>
            </div>
        </form>

        <script>
            const Human = {
                username: "",
                name: "",
                pass: "",
                email: "",
                mobile: "",
                age: "",
                checkAge: function () {
```

```
                    if (this.age < 18) {
                        alert(`Access this website in ${18 - this.age}
years`);
                    } else {
                        alert(`You have registered ${this.username}`);
                    }
                },
                getAge: function () {
                    this.age = prompt("Enter your age");
                },
            };


            function myFunction(event) {
                event.preventDefault();
                Human.username = event.target["username"].value;
                Human.name = event.target["name"].value;
                Human.email = event.target["email"].value;
                Human.mobile = event.target["MNumber"].value;
                if (event.target["repass"].value !=
event.target["pass"].value) {
                    alert("Please enter same password");
                } else {
```

```
                    Human.pass = event.target["pass"].value;

                    Human.getAge();

                    Human.checkAge();

                }

            }

        </script>

    </body>

</html>
```

Style.css

```
.register {

    margin: 2vh 0;

}



.form > div {

    display: flex;

    flex-direction: column;

}



* {

    margin: 0;
```

```
    padding: 0;
    box-sizing: border-box;
    color: #d7e2ff;
}

body {
    background-color: #16171a;
}

.form {
    position: fixed;
    top: 50%;
    left: 50%;
    /* width: 30vw; */
    height: fit-content;
    width: fit-content;
    /* padding: 0 5vw; */
    padding: 2vh 1vw;
    transform: translate(-50%, -50%);
    text-align: center;
    background-color: #202226;
```

```
   border-radius: 16px;
}

.form > div > div {
   margin: 2vh 0;
   display: flex;
   justify-content: space-between;
   font-size: 1.4rem;
}
.form > div > div > label {
   margin: 0 1vw;
}
.form > div > div > input {
   color: #16171a;
}
.form > div > input {
   /* color: #16; */
   padding: 2%;
   background-color: #16171a;
}
```

3. State What is difference between Let and Const keywords

let	**const**
The scope of a *let* variable is block scope.	The scope of a *const* variable is block scope.
It can be updated but cannot be re-declared into the scope.	It cannot be updated or re-declared into the scope.
It can be declared without initialization.	It cannot be declared without initialization.
It cannot be accessed without initialization otherwise it will give ‘referenceError’.	It cannot be accessed without initialization, as it cannot be declared without initialization.
Hoisting is done, but not initialized (this is the reason for the error when we access the let variable before	Hoisting is done, but not initialized (this is the reason for error when we access the const variable before

declaration/initialization	declaration/initialization

6. Explore what is the use of DOM Manipulation.

Let's say you have a button on your web page and you want to change the text inside a paragraph element when the button is clicked. You can use JavaScript to do this.

```
<button id="change-text-btn">Change Text</button>

<p id="text-to-change">Original Text</p>
```

Then you can click the button and Use DOM to change value of text in p tag

```
const changeTextBtn = document.getElementById("change-text-btn");

const textToChange = document.getElementById("text-to-change");

changeTextBtn.addEventListener("click", () => {

  textToChange.textContent = "New Text";

});
```

Outcomes:

CO 3 Apply JavaScript and JSON for Web Application development

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

Successfully implemented the requirements and used regex for validation

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