

Form Validation

Sujay Sharma 2447155

Q - Implement form validation using JavaScript, focusing on delivering a user-friendly experience by appropriately handling events and providing clear feedback.

- Create the HTML structure for the registration form, including input fields for each of the above requirements, labels, and a submit button.
- Implement validation that checks each field upon form submission. If any field fails validation, display an error message next to it and prevent form submission until all fields are valid.
- As the user interacts with the form, provide real-time feedback. For example, show a green checkmark for valid inputs and a red "x" for invalid ones. This should update dynamically as the user types.
- Continuously check if the "Password" and "Confirm Password" fields match, and display a message if they don't.
- Calculate the user's age based on the "Date of Birth" field, and disable the submit button if the user is under 18.
- Attach event listeners to the form elements to trigger validation and provide immediate feedback as the user interacts with the form.
- Design the form to visually represent validation states. Use styles such as red borders for invalid inputs, green borders for valid inputs, and clear error messages to guide the user.
- You may use plain JavaScript or leverage libraries/frameworks of your choice to accomplish these tasks.
- Document your code and explain the rationale behind your design and implementation decisions.

Below is the HTML code –

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sign Up</title>
    <script src="https://cdn.tailwindcss.com"></script>
    <script src="https://kit.fontawesome.com/3386fd30d1.js"
crossorigin="anonymous"></script>
    <script src="validation.js" defer></script>
</head>

<body class="min-h-screen bg-gradient-to-r from-purple-400 via-pink-500 to-red-500 flex items-center justify-center">
    <div class="max-w-md w-full bg-white shadow-lg rounded-lg p-8">
        <h1 class="text-3xl font-bold text-center mb-6">UniQuest</h1>
        <h2 class="text-3xl font-bold text-center mb-6">Create Your Account</h2>
        <form id="registrationForm">

            <div class="mb-5">
                <label for="email" class="block text-sm font-medium text-gray-700 col-form-label">Email</label>
                <span></span>
                <input type="email" id="email" name="email" class="block w-full px-4 py-2 border border-gray-300 rounded-lg" value="user@example.com">
            </div>
        </form>
    </div>
</body>
```

```
        class="form-control mt-1 block w-full px-3 py-2 border border-gray-300 rounded-md shadow-sm focus:outline-none focus:ring-indigo-500 focus:border-indigo-500 sm:text-sm">
            placeholder="you@example.com">
        </div>
        <div class="mb-5">
            <label for="username" class="block text-sm font-medium text-gray-700 col-form-label">Username</label>
            <span></span>
            <input type="text" id="username" name="username"
                class="form-control mt-1 block w-full px-3 py-2 border border-gray-300 rounded-md shadow-sm focus:outline-none focus:ring-indigo-500 focus:border-indigo-500 sm:text-sm"
                placeholder="Your username">
        </div>
        <div class="mb-5">
            <label for="birthday" class="block text-sm font-medium text-gray-700 col-form-label">Date of Birth</label>
            <span></span>
            <input type="date" id="birthday" name="birthday"
                class="form-control mt-1 block w-full px-3 py-2 border border-gray-300 rounded-md shadow-sm focus:outline-none focus:ring-indigo-500 focus:border-indigo-500 sm:text-sm">
        </div>

        <div class="mb-5">
            <label for="password" class="block text-sm font-medium text-gray-700 col-form-label">Password</label>
            <span></span>
            <input type="password" id="password" name="password"
                class="form-control mt-1 block w-full px-3 py-2 border border-gray-300 rounded-md shadow-sm focus:outline-none focus:ring-indigo-500 focus:border-indigo-500 sm:text-sm"
                placeholder="Create a password">
        </div>

        <div class="mb-5">
            <label for="rePassword" class="block text-sm font-medium text-gray-700 col-form-label">Confirm Password</label>
            <span></span>
            <input type="password" id="rePassword" name="rePassword"
                class="form-control mt-1 block w-full px-3 py-2 border border-gray-300 rounded-md shadow-sm focus:outline-none focus:ring-indigo-500 focus:border-indigo-500 sm:text-sm"
                placeholder="Re-enter your password">
        </div>

        <div>
            <button id="signin"
                class="w-full bg-indigo-600 hover:bg-indigo-700 text-white font-bold py-2 px-4 rounded-md transition duration-300">
                Sign Up
            </button>
        </div>
    </form>
</div>
</body>
```

```
</html>
```

Below is the JavaScript validation code –

```
let email = document.getElementById("email");
let username = document.getElementById("username");
let birthday = document.getElementById("birthday");
let password = document.getElementById("password");
let rePassword = document.getElementById("rePassword");
let signinButton = document.getElementById("signin");
const nameRegex = /^[a-zA-Z\s'-]+$/;
const emailRegex = /^[a-zA-Z0-9.]+@[a-zA-Z0-9.]+\.[a-zA-Z]{2,}$/;
const usernameRegex = /^[a-zA-Z0-9._-]+$/;
const passwordRegex = /^(?=.*[A-Z])(?=.*\d).+$/;
var birthdayRegex = /^\d{4}-(\d{2}|\d{1})-(\d{2}|\d{1})$/;
// console.log(name.previousElementSibling.innerText);
email.addEventListener('input', validateEmail);
username.addEventListener('input', validateUsername);
birthday.addEventListener('input', validateBirthday);
password.addEventListener('input', validatePassword);
rePassword.addEventListener('input', validateRepassword);
signinButton.addEventListener('click', checkLabelsBeforeSubmit);

function validateEmail(){
    if (emailRegex.test(email.value)){
        email.previousElementSibling.innerHTML = "<i class=\"fa-solid fa-check\"></i>";
        email.previousElementSibling.style.color = "green";
        email.previousElementSibling.previousElementSibling.style.color = "green";
        email.setAttribute("class", "form-control border border-success border-2");
    }
    else{
        email.previousElementSibling.innerHTML = "<i class=\"fa-solid fa-xmark\"></i> invalid email";
        email.previousElementSibling.style.color = "red";
        email.previousElementSibling.previousElementSibling.style.color = "red";
        email.setAttribute("class", "form-control border border-warning border-2");
    }
}

function validateUsername(){
    if (usernameRegex.test(username.value)){
        username.previousElementSibling.innerHTML = "<i class=\"fa-solid fa-check\"></i>";
        username.previousElementSibling.style.color = "green";
        username.previousElementSibling.previousElementSibling.style.color = "green";
        username.setAttribute("class", "form-control border border-success border-2");
    }
    else{
        username.previousElementSibling.innerHTML = "<i class=\"fa-solid fa-xmark\"></i> invalid username";
        username.previousElementSibling.style.color = "red";
        username.previousElementSibling.previousElementSibling.style.color = "red";
        username.setAttribute("class", "form-control border border-warning border-2");
    }
}

function validateBirthday() {
```

```

// var birthday = document.getElementById("birthday").value; // Assuming the input
element has an id="birthday"
// var birthdayRegex = /\^\\d{4}-\\d{2}-\\d{2}\$/;

// Validate format using regex
if (!birthdayRegex.test(birthday.value)) {
    birthday.previousElementSibling.innerHTML = "<i class=\"fa-solid fa-xmark\"></i>
Invalid date format";
    birthday.previousElementSibling.style.color = "red";
    birthday.previousElementSibling.previousElementSibling.style.color = "red";
    birthday.setAttribute("class", "form-control border border-warning border-2");
} else {
    var date = birthday.value.split("-");
    var year = parseInt(date[0]);
    var month = parseInt(date[1]) - 1; // JavaScript months are 0-indexed
    var day = parseInt(date[2]);

    var birthDate = new Date(year, month, day);

    // Check if the constructed date is valid
    if (birthDate.getFullYear() !== year || birthDate.getMonth() !== month ||
    birthDate.getDate() !== day) {
        birthday.previousElementSibling.innerHTML = "<i class=\"fa-solid fa-xmark\"></i>
Invalid date";
        birthday.previousElementSibling.style.color = "red";
        birthday.previousElementSibling.previousElementSibling.style.color = "red";
        birthday.setAttribute("class", "form-control border border-warning border-2");
        console.log(birthDate.getFullYear());
        console.log(birthDate.getMonth());
        console.log(birthDate.getDate());
    }
    // Check if age is more than 18 years
    else {
        var today = new Date();
        var age = today.getFullYear() - birthDate.getFullYear();
        var monthDiff = today.getMonth() - birthDate.getMonth();

        if (monthDiff < 0 || (monthDiff === 0 && today.getDate() < birthDate.getDate()))
{
            age--;
        }

        if (age < 18) {
            birthday.previousElementSibling.innerHTML = "<i class=\"fa-solid fa-
xmark\"></i> Age less than 18";
            birthday.previousElementSibling.style.color = "red";
            birthday.previousElementSibling.previousElementSibling.style.color = "red";
            birthday.setAttribute("class", "form-control border border-warning border-
2");
            // loginButton.style.background = "grey";
            loginButton.setAttribute('disabled', true);
            loginButton.style.background="grey";
            loginButton.children[0].removeAttribute("href")
            signinButton.setAttribute('disabled', true);
            signinButton.style.background="grey";
            signinButton.children[0].removeAttribute("href")
        } else {
            // If valid, clear any previous error messages
        }
    }
}

```

```

        birthday.previousElementSibling.innerHTML = "<i class=\"fa-solid fa-
check\"></i>";
        birthday.previousElementSibling.style.color = "green";
        birthday.previousElementSibling.previousElementSibling.style.color = "green";
        birthday.setAttribute("class", "form-control border border-success border-
2");
        loginButton.setAttribute('disabled', false);
        loginButton.style.background="#fc4c4c";
        loginButton.children[0].setAttribute("href", "home.html")
        signinButton.setAttribute('disabled', false);
        signinButton.style.background="#fc4c4c";
        signinButton.children[0].setAttribute("href", "home.html")
    }
}
}

function validatePassword(){
    if (passwordRegex.test(password.value) ){
        password.previousElementSibling.innerHTML = "<i class=\"fa-solid fa-check\"></i>";
        password.previousElementSibling.style.color = "green";
        password.previousElementSibling.previousElementSibling.style.color = "green";
        password.setAttribute("class", "form-control border border-success border-2");
    }
    else{
        password.previousElementSibling.innerHTML = "<i class=\"fa-solid fa-xmark\"></i>
weak password";
        password.previousElementSibling.style.color = "red";
        password.previousElementSibling.previousElementSibling.style.color = "red";
        password.setAttribute("class", "form-control border border-warning border-2");
    }
}
function validateRepassword(){
    if (rePassword.value.match(password.value)){
        rePassword.previousElementSibling.innerHTML = "<i class=\"fa-solid fa-
check\"></i>";
        rePassword.previousElementSibling.style.color = "green";
        rePassword.previousElementSibling.previousElementSibling.style.color = "green";
        rePassword.setAttribute("class", "form-control border border-success border-2");
    }
    else{
        rePassword.previousElementSibling.innerHTML = "<i class=\"fa-solid fa-xmark\"></i>
password doesn't match";
        rePassword.previousElementSibling.style.color = "red";
        rePassword.previousElementSibling.previousElementSibling.style.color = "red";
        rePassword.setAttribute("class", "form-control border border-warning border-2");
    }
}
function checkLabelsBeforeSubmit(event) {
    // Select all the labels with the class 'col-form-label'
    let labels = document.querySelectorAll(".col-form-label");

    // Iterate through each label
    for (let label of labels) {
        // Check if the label or its associated icon span is not green
        let iconSpan = label.nextElementSibling;

```

```
        if (label.style.color !== "green" || (iconSpan && iconSpan.style.color !== "green")) {
            event.preventDefault();
            alert("Please enter valid values in the form correctly to proceed.");
            return;
        }
    }
}
```

Design and Implementation of code –

- The Header section of the HTML file contains the script of Tailwind CSS and the Script for the font awesome icons.
- In the Validation File firstly all the variables are declared and get by the document ID.
- For the comparison purpose the predefined syntax is given and checked with the user input
- All the variables which are declared are further have logic and converted into functions which implements the logics (i.e.) checks the user's input to given syntax
- For the Tick and cross Mark the font awesome icons are used.