JavaScript Form Validation

JavaScript form validation checks user input before submitting the form to ensure it's correct. It helps catch errors and improves the user experience.

What we are going to Create

In this article, we will guide you through creating a form validation application. This application will validate essential fields such as firstName, lastName, userName, email, and password. We will explore two different approaches for form validation

- **Using Conditional Logic**: This approach involves using if-else conditions to validate the input based on simple rules.
- **Using Regular Expressions (Regex)**: A more advanced approach where patterns are matched for validating inputs such as emails, passwords, and usernames.

Approach 1: Using Conditional Logic

This validates a registration form by ensuring that all required fields are filled out correctly. It checks the name, email, password, course selection, and agreement checkbox, displaying error messages for any invalid input. If all fields pass validation, the form is submitted successfully; otherwise, submission is prevented.

```
function validateForm() {
    const name = document.getElementById("name").value;
    const addr = document.getElementById("address").value;
    const email = document.getElementById("email").value;
    const pass = document.getElementById("password").value;
    const sub = document.getElementById("subject").value;
    const agree = document.getElementById("agree").checked;

    const nameErr = document.getElementById("name-error");
    const addrErr = document.getElementById("address-error");
    const emailErr = document.getElementById("email-error");
    const passErr = document.getElementById("password-error");
    const subErr = document.getElementById("subject-error");
    const agreeErr = document.getElementById("agree-error");

    nameErr.textContent = "";
    addrErr.textContent = "";
```

```
  emailErr.textContent = "";
  passErr.textContent = "";
  subErr.textContent = "";
  agreeErr.textContent = "";

  let isValid = true;

  if (name === "" || /\d/.test(name)) {
    nameErr.textContent = "Please enter your name properly.";
    isValid = false;
  }

  if (addr === "") {
    addrErr.textContent = "Please enter your address.";
    isValid = false;
  }

  if (email === "" || !email.includes("@") || !email.includes(".")) {
    emailErr.textContent = "Please enter a valid email address.";
    isValid = false;
  }

  if (pass === "" || pass.length < 6) {
    passErr.textContent = "Please enter a password with at least 6 characters.";
    isValid = false;
  }

  if (sub === "") {
    subErr.textContent = "Please select your course.";
    isValid = false;
  }
```

```
  if (!agree) {
    agreeErr.textContent = "Please agree to the above information.";
    isValid = false;
  }

  if (isValid) {
    alert("Form submitted successfully!");
    return true;
  }
  else {
    return false;
  }
}

function resetErrors() {
  document.getElementById("name-error").textContent = "";
  document.getElementById("address-error").textContent = "";
  document.getElementById("email-error").textContent = "";
  document.getElementById("password-error").textContent = "";
  document.getElementById("subject-error").textContent = "";
  document.getElementById("agree-error").textContent = "";
}
```

- **Form Validation** :The validateForm() function checks if the form fields (name, address, email, etc.) are filled out correctly before submission.
- **Error Display** :If any field is invalid, an error message is shown next to it, guiding the user to correct the issue.
- **Prevents Submission on Error** :If any field is invalid, the form submission is blocked, preventing incomplete data from being sent.
- **Clear Errors on Reset** :The resetErrors() function clears all error messages when the form is reset.
- **Organized Code** :The code is divided into two functions (validateForm() and resetErrors()), ensuring clarity and easy maintenance.

```
<html>
<head>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f5f5f5;
    }
    h1 {
      text-align: center;
      color: #333;
    }
    form {
      max-width: 600px;
      margin: 0 auto;
      padding: 20px;
      background-color: #fff;
      border-radius: 8px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }
    input[type="text"],
    input[type="password"],
    select,
    textarea {
      width: 100%;
      padding: 10px;
      margin: 5px 0;
      border: 1px solid #ccc;
      border-radius: 5px;
      box-sizing: border-box;
      font-size: 16px;
    }
```

```
select {
  width: 100%;
  padding: 10px;
  margin: 5px 0;
  border: 1px solid #ccc;
  border-radius: 5px;
  box-sizing: border-box;
  font-size: 16px;
  background-color: #fff;
  appearance: none;
  -webkit-appearance: none;
  -moz-appearance: none;
}
textarea {
  resize: vertical;
}
input[type="submit"],
input[type="reset"],
input[type="checkbox"] {
  background-color: #007bff;
  color: #fff;
  border: none;
  border-radius: 5px;
  padding: 10px 20px;
  cursor: pointer;
  font-size: 16px;
}
input[type="submit"]:hover,
input[type="reset"]:hover,
input[type="checkbox"]:hover {
  background-color: #0056b3;
```

```
    }
    .error-message {
      color: red;
      font-size: 14px;
      margin-top: 5px;
    }
  </style>
</head>
<body>
  <h1>REGISTRATION FORM</h1>
  <form name="RegForm" onsubmit="return validateForm()" onreset="resetErrors()">
    <p>
      <label for="name">Name:</label>
      <input type="text" id="name" name="Name" placeholder="Enter your full name" />
      <span id="name-error" class="error-message"></span>
    </p>
    <p>
      <label for="address">Address:</label>
      <input type="text" id="address" name="Address" placeholder="Enter your address"
/>
      <span id="address-error" class="error-message"></span>
    </p>
    <p>
      <label for="email">E-mail Address:</label>
      <input type="text" id="email" name="EMail" placeholder="Enter your email" />
      <span id="email-error" class="error-message"></span>
    </p>
    <p>
      <label for="password">Password:</label>
      <input type="password" id="password" name="Password" />
      <span id="password-error" class="error-message"></span>
```

```
    </p>
    <p>
      <label for="subject">Select Your Course:</label>
      <select id="subject" name="Subject">
        <option value="">Select Course</option>
        <option value="BTECH">BTECH</option>
        <option value="BBA">BBA</option>
        <option value="BCA">BCA</option>
        <option value="B.COM">B.COM</option>
      </select>
      <span id="subject-error" class="error-message"></span>
    </p>
    <p>
      <label for="comment">College Name:</label>
      <textarea id="comment" name="Comment"></textarea>
    </p>
    <p>
      <input type="checkbox" id="agree" name="Agree" />
      <label for="agree">I agree to the above information</label>
      <span id="agree-error" class="error-message"></span>
    </p>
    <p>
      <input type="submit" value="Send" name="Submit" />
      <input type="reset" value="Reset" name="Reset" />
    </p>
  </form>
  <script>
    function validateForm() {
      const name = document.getElementById("name").value;
      const addr = document.getElementById("address").value;
      const email = document.getElementById("email").value;
```

```
const pass = document.getElementById("password").value;
const sub = document.getElementById("subject").value;
const agree = document.getElementById("agree").checked;

const nameErr = document.getElementById("name-error");
const addrErr = document.getElementById("address-error");
const emailErr = document.getElementById("email-error");
const passErr = document.getElementById("password-error");
const subErr = document.getElementById("subject-error");
const agreeErr = document.getElementById("agree-error");

nameErr.textContent = "";
addrErr.textContent = "";
emailErr.textContent = "";
passErr.textContent = "";
subErr.textContent = "";
agreeErr.textContent = "";

let isValid = true;
if (name === "" || /\d/.test(name)) {
  nameErr.textContent = "Please enter your name properly.";
  isValid = false;
}
if (addr === "") {
  addrErr.textContent = "Please enter your address.";
  isValid = false;
}
if (email === "" || !email.includes("@") || !email.includes(".")) {
  emailErr.textContent = "Please enter a valid email address.";
  isValid = false;
}
```

```
        if (pass === "" || pass.length < 6) {
            passErr.textContent = "Please enter a password with at least 6 characters.";
            isValid = false;
        }
        if (sub === "") {
            subErr.textContent = "Please select your course.";
            isValid = false;
        }
        if (!agree) {
            agreeErr.textContent = "Please agree to the above information.";
            isValid = false;
        }
        if (isValid) {
            alert("Form submitted successfully!");
            return true;
        } else {
            return false;
        }
    }

    function resetErrors() {
        document.getElementById("name-error").textContent = "";
        document.getElementById("address-error").textContent = "";
        document.getElementById("email-error").textContent = "";
        document.getElementById("password-error").textContent = "";
        document.getElementById("subject-error").textContent = "";
        document.getElementById("agree-error").textContent = "";
    }
  </script>
</body>
</html>
```

- **Validation & Error Handling**: The validateForm() function checks user inputs (name, address, email, password, course selection, and agreement). If any input is invalid, an error message appears next to the field.
- **User Feedback**: Each field has an associated error message (e.g., name-error), which dynamically updates when validation fails. If everything is valid, a success alert is shown.
- **Reset Function**: The resetErrors() function clears all error messages when the form is reset, ensuring a clean slate for users.
- **User-Friendly Layout:** The form is designed with clear labels, input fields, a submit button, and a checkbox for user agreement, providing easy navigation.
- **CSS Styling**: The form elements are styled with rounded corners, padding, and hover effects. Error messages are highlighted in red for better visibility and user guidance.

Approach 2: Using Regular Expressions

Form validation using regular expressions (regex) ensures user input matches specific patterns, like email formats or password rules. It checks inputs upon submission, showing error messages if they don't match the defined regex patterns, offering an efficient way to validate data with minimal code.

```
const form = document.getElementById("myForm");


form.addEventListener("submit", function (e) {
  e.preventDefault();


  const fName = document.getElementById("firstName").value;
  const lName = document.getElementById("lastName").value;
  const user = document.getElementById("username").value;
  const email = document.getElementById("email").value;
  const pass = document.getElementById("password").value;
  const phone = document.getElementById("phone").value;
  const dob = document.getElementById("dob").value;


  let isValid = true;


  document.getElementById("firstName-error").textContent = "";
  document.getElementById("lastName-error").textContent = "";
```

```
document.getElementById("username-error").textContent = "";
document.getElementById("email-error").textContent = "";
document.getElementById("password-error").textContent = "";
document.getElementById("phone-error").textContent = "";
document.getElementById("dob-error").textContent = "";

const nameRegex = /^[a-zA-Z]{1,30}$/;
if (!nameRegex.test(fName)) {
  document.getElementById("firstName-error").textContent =
    "First name must be 1-30 letters.";
  isValid = false;
}
if (!nameRegex.test(lName)) {
  document.getElementById("lastName-error").textContent =
    "Last name must be 1-30 letters.";
  isValid = false;
}
const userRegex = /^[a-zA-Z0-9_]{3,15}$/;
if (!userRegex.test(user)) {
  document.getElementById("username-error").textContent =
    "Username must be 3-15 chars (letters, numbers, underscores).";
  isValid = false;
}
const emailRegex = /^([a-zA-Z0-9._%+-]+)@([a-zA-Z0-9.-]+)\.([a-zA-Z]{2,})$/;
if (!emailRegex.test(email)) {
  document.getElementById("email-error").textContent =
    "Invalid email address.";
  isValid = false;
}
const passRegex = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%?&])[A-Za-z\d@$!%?&]{8,}$/;
if (!passRegex.test(pass)) {
```

```
    document.getElementById("password-error").textContent =
      "Password must have 8+ chars, 1 uppercase, 1 number, 1 special char.";
    isValid = false;
  }
  const phoneRegex = /^[7-9][0-9]{9}$/;
  if (!phoneRegex.test(phone)) {
    document.getElementById("phone-error").textContent =
      "Phone must be 10 digits, starting with 7-9.";
    isValid = false;
  }
  const today = new Date();
  const bDate = new Date(dob);
  const age = today.getFullYear() - bDate.getFullYear();
  const month = today.getMonth() - bDate.getMonth();
  if (age < 18 || (age === 18 && month < 0)) {
    document.getElementById("dob-error").textContent =
      "You must be at least 18 years old.";
    isValid = false;
  }
  if (isValid) {
    alert("Form submitted successfully!");
  }
});
```

- **Form Submission**: The script intercepts the form submission, preventing the default action and validating the form data.
- **Clear Error Messages**: Before validating, it clears any previous error messages to ensure a fresh validation.
- **Field Validation**: Each field (first name, last name, username, email, password, phone, and DOB) is validated using regular expressions to ensure correct formatting and requirements.
- **Display Errors**: If validation fails, specific error messages are displayed next to the corresponding fields.

- **Form Success**: If all fields are valid, an alert confirms successful submission; otherwise, the form submission is prevented.

```
<html>
<head>
  <style>
    body {
      font-family: Arial, sans-serif;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 300vh;
      background-color: #f4f4f9;
      margin: 0;
    }
    .form-container {
      background-color: #fff;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
      width: 300px;
    }
    h2 {
      text-align: center;
      color: #333;
    }
    input {
      width: 100%;
      padding: 10px;
      margin: 10px 0;
      border: 1px solid #ccc;
```

```
      border-radius: 4px;
    }
    button {
      width: 100%;
      padding: 10px;
      background-color: #4CAF50;
      color: white;
      border: none;
      border-radius: 4px;
      cursor: pointer;
    }
    button:hover {
      background-color: #45a049;
    }
    .alert {
      color: red;
      font-size: 12px;
      margin-top: -5px;
    }
  </style>
</head>
<body>
  <div class="form-container">
    <h2>Sign Up</h2>
    <form id="myForm">
      <input type="text" id="firstName" placeholder="First Name" required>
      <div id="firstName-error" class="alert"></div>
      <input type="text" id="lastName" placeholder="Last Name" required>
      <div id="lastName-error" class="alert"></div>
      <input type="text" id="username" placeholder="Username" required>
      <div id="username-error" class="alert"></div>
```

```
    <input type="email" id="email" placeholder="Email" required>
    <div id="email-error" class="alert"></div>
    <input type="password" id="password" placeholder="Password" required>
    <div id="password-error" class="alert"></div>
    <input type="tel" id="phone" placeholder="Phone Number" required>
    <div id="phone-error" class="alert"></div>
    <input type="date" id="dob" placeholder="Date of Birth" required>
    <div id="dob-error" class="alert"></div>
    <button type="submit">Submit</button>
  </form>
</div>
<script>
  const form = document.getElementById("myForm");
  form.addEventListener("submit", function (e) {
    e.preventDefault();
    const fName = document.getElementById("firstName").value;
    const lName = document.getElementById("lastName").value;
    const user = document.getElementById("username").value;
    const email = document.getElementById("email").value;
    const pass = document.getElementById("password").value;
    const phone = document.getElementById("phone").value;
    const dob = document.getElementById("dob").value;

    let isValid = true;
    document.getElementById("firstName-error").textContent = "";
    document.getElementById("lastName-error").textContent = "";
    document.getElementById("username-error").textContent = "";
    document.getElementById("email-error").textContent = "";
    document.getElementById("password-error").textContent = "";
    document.getElementById("phone-error").textContent = "";
    document.getElementById("dob-error").textContent = "";
```

```
const nameRegex = /^[a-zA-Z]{1,30}$/;
if (!nameRegex.test(fName)) {
   document.getElementById("firstName-error").textContent =
      "First name must be 1-30 letters.";
   isValid = false;
}
if (!nameRegex.test(lName)) {
   document.getElementById("lastName-error").textContent =
      "Last name must be 1-30 letters.";
   isValid = false;
}
const userRegex = /^[a-zA-Z0-9_]{3,15}$/;
if (!userRegex.test(user)) {
   document.getElementById("username-error").textContent =
      "Username must be 3-15 chars (letters, numbers, underscores).";
   isValid = false;
}
const emailRegex = /^([a-zA-Z0-9._%+-]+)@([a-zA-Z0-9.-]+)\.([a-zA-Z]{2,})$/;
if (!emailRegex.test(email)) {
   document.getElementById("email-error").textContent =
      "Invalid email address.";
   isValid = false;
}
const passRegex = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%?&])[A-Za-z\d@$!%?&]{8,}$/;
if (!passRegex.test(pass)) {
   document.getElementById("password-error").textContent =
      "Password must have 8+ chars, 1 uppercase, 1 number, 1 special char.";
   isValid = false;
}
const phoneRegex = /^[7-9][0-9]{9}$/;
```

```
    if (!phoneRegex.test(phone)) {
      document.getElementById("phone-error").textContent =
        "Phone must be 10 digits, starting with 7-9.";
      isValid = false;
    }
    const today = new Date();
    const bDate = new Date(dob);
    const age = today.getFullYear() - bDate.getFullYear();
    const month = today.getMonth() - bDate.getMonth();
    if (age < 18 || (age === 18 && month < 0)) {
      document.getElementById("dob-error").textContent =
        "You must be at least 18 years old.";
      isValid = false;
    }
    if (isValid) {
      alert("Form submitted successfully!");
    }
  });
 </script>
</body>
</html>
```

- **HTML Structure**: The HTML creates a simple form with input fields for name, username, email, password, phone, and DOB, and includes placeholders and error message containers for validation feedback.
- **CSS Styling**: The CSS centers the form on the page and applies styling such as padding, borders, shadows, and hover effects to create a clean and user-friendly interface.
- **JavaScript Validation**: The JS validates user input on form submission using regular expressions (regex) for fields like email, username, password, and phone, and calculates age for the DOB field to ensure the user meets the age requirement.
- **Error Handling**: For each invalid input, the JS displays error messages dynamically below the relevant input field, guiding the user to correct their data before submission.
- **Form Interactivity**: The form includes a submit button that triggers the validation process, preventing form submission if any input is invalid, ensuring data integrity before sending it to the server.

To perfect these skills and more, our **JavaScript Course** dives deep into form validation, covering best p