

# Form Validation Tutorial

Let's break this assignment into three key parts:

1. **Understanding Form Validation Requirements**
  2. **Step-by-Step Implementation**
  3. **Full Code for HTML, CSS, and JavaScript**
- 

## Understanding Form Validation Requirements

A form typically requires user input for **Name, Email, and Password**. Before submitting, we need to check:

- ✓ Name is not empty and contains only alphabets.
- ✓ Email follows the standard email pattern (e.g., `name@example.com`).
- ✓ Password is strong (at least 8 characters, including a letter, number, and special character).

We'll handle these using **JavaScript** with **Event Listeners** and **Regular Expressions (Regex)**.

---

## Step-by-Step Implementation

### Creating the HTML Form

- We'll use a `<form>` element containing:
  - A name `<input>` field.
  - An email `<input>` field.
  - A password `<input>` field.
  - A submit `<button>`.
- Each field will have a `<span>` for displaying error messages.

### Styling the Form (CSS)

- We'll style the form to look neat.
- Error messages will be highlighted in **red**.

### Adding JavaScript for Validation

- Attach an `onsubmit` event to the form.
- Use `event.preventDefault()` to stop submission if validation fails.
- Validate each field using JavaScript:
  - **Name:** Only letters (using regex  `/^[a-zA-Z ]+$/` ).
  - **Email:** Matches standard email format (using regex  `/^[^\s@]+@[^\s@]+\.[^\s@]+$/` ).

- **Password:** At least 8 characters, including letters, numbers, and special characters (using regex `^(?=.*[A-Za-z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}$`).

## Full Code Implementation

### ✓ 1. HTML (Form Structure)

```
html

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Form Validation</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

  <div class="container">
    <h2>Register</h2>
    <form id="registrationForm">
      <label for="name">Name:</label>
      <input type="text" id="name" placeholder="Enter your name">
      <span class="error" id="nameError"></span>

      <label for="email">Email:</label>
      <input type="email" id="email" placeholder="Enter your email">
      <span class="error" id="emailError"></span>

      <label for="password">Password:</label>
      <input type="password" id="password" placeholder="Enter your password">
      <span class="error" id="passwordError"></span>

      <button type="submit">Submit</button>
    </form>
  </div>

  <script src="script.js"></script>
</body>
</html>
```

### ✓ 2. CSS (Styling the Form)

```
css

/* styles.css */
body {
  font-family: Arial, sans-serif;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  background-color: #f4f4f4;
}

.container {
  background: white;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  width: 300px;
}
```

```
h2 {
  text-align: center;
}

label {
  display: block;
  font-weight: bold;
  margin-top: 10px;
}

input {
  width: 100%;
  padding: 8px;
  margin-top: 5px;
  border: 1px solid #ccc;
  border-radius: 4px;
}

button {
  width: 100%;
  padding: 10px;
  margin-top: 15px;
  background-color: #28a745;
  color: white;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

button:hover {
  background-color: #218838;
}

.error {
  color: red;
  font-size: 12px;
  margin-top: 5px;
  display: block;
}
```

### ✓ 3. JavaScript (Form Validation Logic)

```
js

// script.js
document.getElementById("registrationForm").addEventListener("submit", function(event) {
  event.preventDefault(); // Prevents form submission if validation fails

  // Get input values
  let name = document.getElementById("name").value.trim();
  let email = document.getElementById("email").value.trim();
  let password = document.getElementById("password").value.trim();

  // Error fields
  let nameError = document.getElementById("nameError");
  let emailError = document.getElementById("emailError");
  let passwordError = document.getElementById("passwordError");

  // Clear previous errors
  nameError.innerText = "";
  emailError.innerText = "";
  passwordError.innerText = "";

  let isValid = true;

  // ✓ Name Validation (Only alphabets allowed)
  let namePattern = /^[a-zA-Z ]+$/;
  if (name === "") {
    nameError.innerText = "Name is required";
    isValid = false;
  }
```

```

} else if (!namePattern.test(name)) {
    nameError.innerText = "Name should contain only letters";
    isValid = false;
}

// ✅ Email Validation (Standard Email Format)
let emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
if (email === "") {
    emailError.innerText = "Email is required";
    isValid = false;
} else if (!emailPattern.test(email)) {
    emailError.innerText = "Invalid email format";
    isValid = false;
}

// ✅ Password Validation (At least 8 characters, one letter, one number, one special character)
let passwordPattern = /^(?=.*[A-Za-z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}$/;
if (password === "") {
    passwordError.innerText = "Password is required";
    isValid = false;
} else if (!passwordPattern.test(password)) {
    passwordError.innerText = "Password must be at least 8 characters long and include a letter";
    isValid = false;
}

// ✅ If all fields are valid, submit the form (for demonstration, we alert success)
if (isValid) {
    alert("Form submitted successfully!");
    document.getElementById("registrationForm").reset();
}
});

```

## What You'll Learn from This Assignment

- ✅ **HTML:** Form structure, input elements, labels, buttons.
- ✅ **CSS:** Styling forms, error messages, button hover effects.
- ✅ **JavaScript:**

- Handling form submission using `addEventListener("submit")``.
- Validating inputs with **regular expressions (regex)**.
- Displaying dynamic error messages using **DOM manipulation**.
- Preventing form submission using `event.preventDefault()``.

## Next Steps (Bonus Enhancements)

- ♦ Add **real-time validation** (validate while typing).
- ♦ Show/hide password with a toggle button.
- ♦ Store form data locally using **LocalStorage**.

This assignment will give you a **strong understanding of form validation** and **handling user input correctly** in JavaScript! 🚀

ChatGPT can make mistakes. Check important info.