



**COLLEGE CODE : 9222**

**COLLEGE NAME : THENI KAMMAVAR SANGAM  
COLLEGE OF TECHNOLOGY**

**DEPARTMENT : B.tech(IT)**

**STUDENT NM-ID : ADC7870C46CFF6DA10D421BCC4347BFB**

**ROLL NO : 23it049**

**DATE : 17-10-2025**

**Completed the project named as**

**Phase\_\_5 TECHNOLOGY PROJECT**

**NAME : Client side form validation**

**SUBMITTED BY,**

**NAME : Sundaramahalingam M**

**MOBILE NO : 9360128907**

# Final Demo Walkthrough

The client-side form validation implemented in this project ensures that user inputs are checked instantly in the browser before submission. It uses a combination of HTML5 validation attributes and custom JavaScript to verify required fields, proper formats (like email), and complex rules such as password strength and confirmation.

Users receive immediate, clear error messages when inputs are invalid, and the form prevents submission until all fields meet the criteria. This improves user experience by guiding corrections in real-time and reduces unnecessary server requests with invalid data.

The validation complements backend checks for overall security and data integrity. Challenges like browser compatibility and balancing strictness with usability were addressed by combining native and custom validation techniques.

The project repository contains the full source code and setup instructions, while the deployed app demonstrates the validation features live.

# PROJECT REPORT

```
<!DOCTYPE html>

<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title>Mithran Design - Client Registration</title>
  <style>
    body {
      margin: 0;
      font-family: 'Poppins', sans-serif;
      background: linear-gradient(135deg, #1f1c2c, #928dab);
      color: #fff;
      display: flex;
      justify-content: center;
      align-items: center;
      min-height: 100vh;
    }

    .form-wrapper {
      background-color: rgba(255, 255, 255, 0.1);
      padding: 40px;
      border-radius: 20px;
      width: 450px;
      box-shadow: 0 0 20px rgba(0, 0, 0, 0.4);
    }

    .form-wrapper h1 {
      text-align: center;
      margin-bottom: 10px;
      font-size: 28px;
      color: #f9f9f9;
    }

    .form-wrapper p {
      text-align: center;
      font-size: 14px;
      color: #ddd;
      margin-bottom: 30px;
    }

    label {
      display: block;
```

```
margin-top: 15px;
font-weight: 500;
}

input, select, textarea {
  width: 100%;
  padding: 10px;
  margin-top: 5px;
  border: none;
  border-radius: 8px;
  background-color: #fff;
  color: #333;
  font-size: 1em;
}

textarea {
  resize: vertical;
}

.error {
  color: #ffb3b3;
  font-size: 0.85em;
}

button {
  width: 100%;
  padding: 12px;
  margin-top: 25px;
  background-color: #ff6f61;
  border: none;
  border-radius: 8px;
  color: white;
  font-size: 1em;
  cursor: pointer;
  transition: background-color 0.3s ease;
}

button:hover {
  background-color: #e94e3c;
}

.logo {
  text-align: center;
  margin-bottom: 20px;
}
```

```
.logo img {
  width: 80px;
  height: auto;
}
</style>
</head>
<body>
  <div class="form-wrapper">
    <div class="logo">
      <!-- Replace with your actual logo -->
      
    </div>
    <h1>Client Registration</h1>
    <p>Welcome to Mithran Design. Let's get started!</p>
    <form id="clientForm">
      <label for="clientName">Full Name:</label>
      <input type="text" id="clientName" name="clientName" required />
      <div class="error" id="clientNameError"></div>

      <label for="company">Company Name:</label>
      <input type="text" id="company" name="company" />

      <label for="email">Email Address:</label>
      <input type="email" id="email" name="email" required />
      <div class="error" id="emailError"></div>

      <label for="phone">Phone Number:</label>
      <input type="tel" id="phone" name="phone" pattern="[0-9]{10}" required />
      <div class="error" id="phoneError"></div>

      <label for="projectType">Project Type:</label>
      <select id="projectType" name="projectType" required>
        <option value="">Select</option>
        <option value="Logo Design">Logo Design</option>
        <option value="Website Design">Website Design</option>
        <option value="Branding Package">Branding Package</option>
        <option value="UI/UX Design">UI/UX Design</option>
      </select>
      <div class="error" id="projectTypeError"></div>

      <label for="budget">Estimated Budget (USD):</label>
      <input type="number" id="budget" name="budget" min="100" required />
      <div class="error" id="budgetError"></div>
    </form>
  </div>
</body>
</html>
```

```

    <label for="details">Project Details:</label>
    <textarea id="details" name="details" rows="4" placeholder="Tell us more
about your project..." required></textarea>
    <div class="error" id="detailsError"></div>

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

<script>
  const form = document.getElementById('clientForm');

  form.addEventListener('submit', function(event) {
    let isValid = true;
    document.querySelectorAll('.error').forEach(el => el.textContent = '');

    const name = document.getElementById('clientName').value.trim();
    if (name.length < 2) {
      document.getElementById('clientNameError').textContent = 'Please enter
your full name.';
      isValid = false;
    }

    const email = document.getElementById('email').value.trim();
    if (!email.includes('@')) {
      document.getElementById('emailError').textContent = 'Enter a valid email
address.';
      isValid = false;
    }

    const phone = document.getElementById('phone').value;
    if (!/^d{10}$/.test(phone)) {
      document.getElementById('phoneError').textContent = 'Enter a valid 10-
digit phone number.';
      isValid = false;
    }

    const projectType = document.getElementById('projectType').value;
    if (!projectType) {
      document.getElementById('projectTypeError').textContent = 'Please select
a project type.';
      isValid = false;
    }
  });

```

```

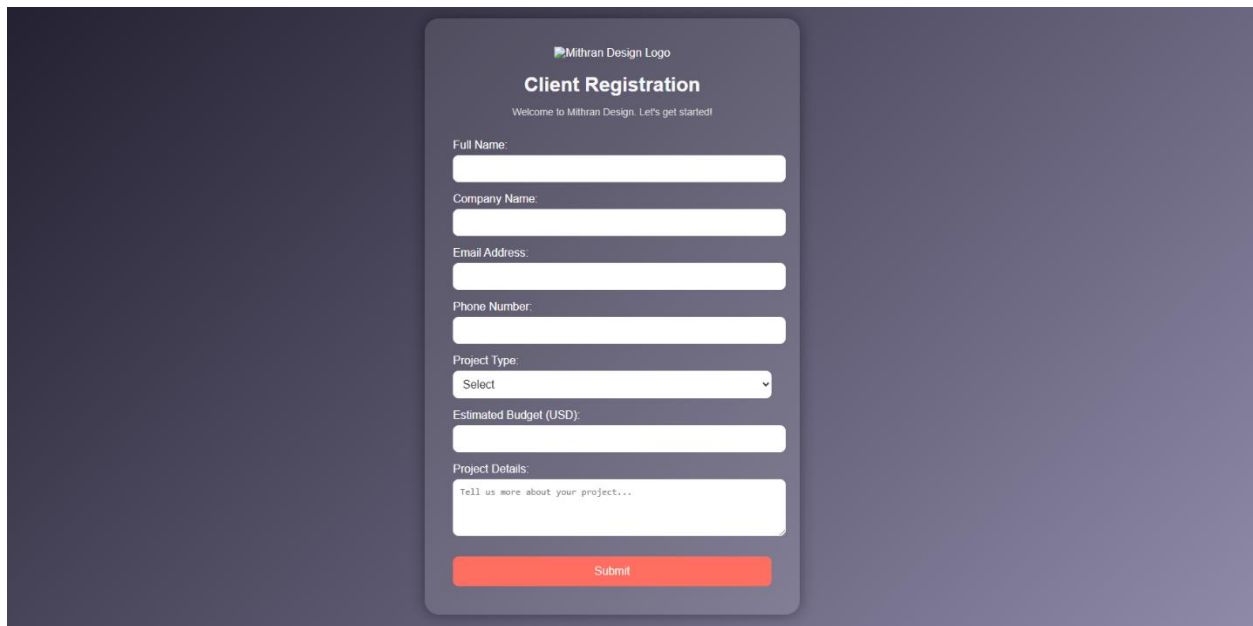
const budget = document.getElementById('budget').value;
if (!budget || budget < 100) {
  document.getElementById('budgetError').textContent = 'Minimum budget is
$100.';
  isValid = false;
}

const details = document.getElementById('details').value.trim();
if (details.length < 10) {
  document.getElementById('detailsError').textContent = 'Please provide
more details about your project.';
  isValid = false;
}

if (!isValid) {
  event.preventDefault();
}
});
</script>
</body>
</html>

```

## SCREENSHOT & API DOCUMENTATION



Mithran Design Logo

### Client Registration

Welcome to Mithran Design. Let's get started!

Full Name:

Company Name:

Email Address:

Phone Number:

Project Type:

Estimated Budget (USD):

Project Details:  
 Tell us more about your project...

Submit

## CHALLENGES & SOLUTIONS

### Challenge 1: Browser Compatibility

Different browsers handle HTML5 validation attributes inconsistently, causing unpredictable behavior.

- **Solution:** Combined native HTML5 validation with custom JavaScript functions as fallbacks to ensure consistent validation and error messaging across all major browsers.

### Challenge 2: Complex Validation Rules

Some validation requirements, like password strength or confirming matching passwords, cannot be handled by simple HTML attributes.

- **Solution:** Implemented custom JavaScript validation using regular expressions and event listeners to enforce complex rules and provide dynamic feedback.

### Challenge 3: User Experience vs. Strict Validation

Overly strict validation or unclear error messages can frustrate users and lead to abandonment.

- **Solution:** Designed clear, concise, and friendly error messages displayed inline near the relevant fields, allowing users to understand and fix errors quickly without confusion.

### Challenge 4: Preventing Form Submission on Invalid Input

Ensuring the form cannot be submitted until all validations pass is essential but sometimes tricky to implement cleanly.



- **Solution:** Disabled the submit button or prevented default form submission via JavaScript until all validations are satisfied, ensuring only valid data is sent to the server.

### **Challenge 5: Handling Edge Cases and Unexpected Input**

Users may enter unexpected or malicious inputs that bypass simple checks.

- **Solution:** Added comprehensive input sanitization in the client code and relied on server-side validation as a secure backup to catch anything missed on the client.

## **GitHub README& setup guide**

### **Overview:**

A JavaScript library/project that helps validate HTML form inputs on the client side, ensuring data correctness before submitting to the server.

### **Key Features:**

- Validates required fields, email formats, password lengths, patterns, etc.
- Supports custom validation rules (e.g., matching passwords).
- Displays user-friendly error messages near inputs.
- Works in real-time as users type or on form submission.
- Allows disabling native browser validation with `novalidate`.
- Accessible error messaging with `aria-live`.

**URL :** <https://github.com/sundar006pro/project.git>

**Final Submission(Repo + Deployed Link)**