COLLEGE CODE : 9222

COLLEGE NAME : THENI KAMMAVAR SANGAM COLLEGEOFTECHNOLOGY

DEPARTMENT : B.tech(IT)

STUDENT NM-ID : 43CABA3EDC98SEEABE2482400E304104

ROLL NO : 23it011

DATE : 19-09-2025

Completed the project named as

Phase__1 TECHNOLOGY PROJECT

NAME : **Client side form validation**

SUBMITTED BY,

NAME : M.Gopalakrishnan

MOBILE NO : 8590886310

Problem Understanding & Requirements

1. Problem Statement

Forms are one of the most common ways to collect user information on websites. However, users often make mistakes such as leaving fields empty, entering invalid email addresses, or providing weak passwords. Without client-side validation, these errors are only caught after submission

which:

- Causes frustration for users.
- Increases server load with invalid requests.
- Leads to delays in data processing

Solution:

Implement **client-side form validation** to provide instant feedback, ensure correctness, and reduce unnecessary server calls

2. Users & Stakeholders

- **Users**
  - General website visitors filling out forms (registration, login, feedback, checkout, etc.).
- **Stakeholders**
  - **Developers** ⟶ who implement validation logic.
  - **Business owners** ⟶ who want higher conversion rates and fewer failed submissions.

- **Admins/Support teams** ⟶ who process fewer incorrect entries.

3. User Stories

- *As a user*, I want to be alerted immediately if I leave a required field empty, so I can fix it before submitting.
- *As a user*, I want to know if my email address is invalid, so I don't waste time submitting incorrect data.
- *As a user*, I want password strength indicators, so I can create a secure password.
- *As a user*, I want to see clear error messages next to the fields, so I know what needs to be corrected.
- *As a business owner*, I want fewer invalid form submissions, so my backend processes are more efficient.

4. MVP Features

- Required field validation (cannot be empty).
- Email format validation.
- Password validation (minimum length, strong password rules).
- Confirm password match check.

5. Wireframes / API Endpoint List

Wireframe (Simple Example)

Registration Form
Name: [_____] *
Email: [_____] *
Password: [_____] *
Confirm Password: [_____]
Age: [____]

[Submit]

*Inline error messages appear below fields if invalid

API Endpoint List (if backend is connected)

- `POST /api/register` → Submits validated form data.
- `POST /api/login` → Validates login form inputs.
- `GET /api/validate/username?name=abc` → (optional) Check if username exists.

6. Acceptance Criteria

- All required fields must be filled before submission.
- Invalid email addresses (e.g., `abc.com`) are rejected instantly.
- Password must meet minimum security requirements (e.g., at least 8 characters, includes number/special character).
- Confirm password must exactly match the password.
- Age must be a valid number within the allowed range (e.g., 18–60).Error messages are displayed clearly next to the corresponding input fields.
- Form submission is only possible when all validations pass.