



COLLEGE CODE: 9111

COLLEGE NAME : SRM MADURAI COLLEGE FOR
ENGINEERING AND TECHNOLOGY

DEPARTMENT: BTECH INFORMATION TECHNOLOGY

STUDENT NM ID: 6AF6A507C81A5A1B0A842198852F12D6

ROLL NO : 23IT50

REGISTER NO : 911123205052

DATE: 22 -09-2025

COMPLETED THE PROJECTED NAMED AS :

Phase 3 Front end technologies : MVP Implementation

PROJECT NAME: Interactive Form Validation

Submitted by :

K SRIKA

6380419626

1. Project Setup :

Project Name: Form Validation System

Objective: To create a user-friendly form validation system that ensures data integrity before submission.

Technology Stack:

- **Frontend:** HTML, CSS, JavaScript
- **Backend (optional):** Node.js/Express or Python Flask (if server-side validation is implemented)
- **Version Control:** Git/GitHub

Setup Steps:

Create project folder structure:

- 1) /form-validation
- 2) /css
- 3) /js
- 4) index.html
- 5) Link external CSS and JavaScript files to HTML.
- 6) Initialize Git repository and connect to GitHub.

2. Core Features / Implementation :

Core Features:

1. **Field Validation:**
 - Required fields cannot be empty.
 - Email must be in valid format.
 - Password must meet criteria (length, special character, etc.).
2. **Real-time Validation:**
 - Validation messages appear as the user types.
3. **Submission Control:**
 - Form cannot be submitted if validation fails.
4. **Error Highlighting:**
 - Highlight invalid fields with red border or error message.

Implementation Example (JavaScript snippet) :

```
function validateForm() {  
    const email = document.getElementById("email").value;  
    const password = document.getElementById("password").value;  
    let valid = true;  
  
    if(!email.includes("@")) {  
        document.getElementById("emailError").innerText = "Invalid  
email";  
        valid = false;  
    } else {  
        document.getElementById("emailError").innerText = "";  
    }  
  
    if(password.length < 8) {  
        document.getElementById("passwordError").innerText = "Password  
too short";  
        valid = false;  
    } else {  
        document.getElementById("passwordError").innerText = "";  
    }  
  
    return valid;  
}
```

3. Data Storage (Local State / Database) :

- **Local State (Browser-based):**

- Store temporary data in localStorage or sessionStorage.
- Example: localStorage.setItem("userEmail", email);

- **Optional Database (for server-side projects):**

- MySQL, MongoDB, or Firebase can be used to store submitted form data securely.

4. Testing Core Features :

- **Field Validation Testing:**

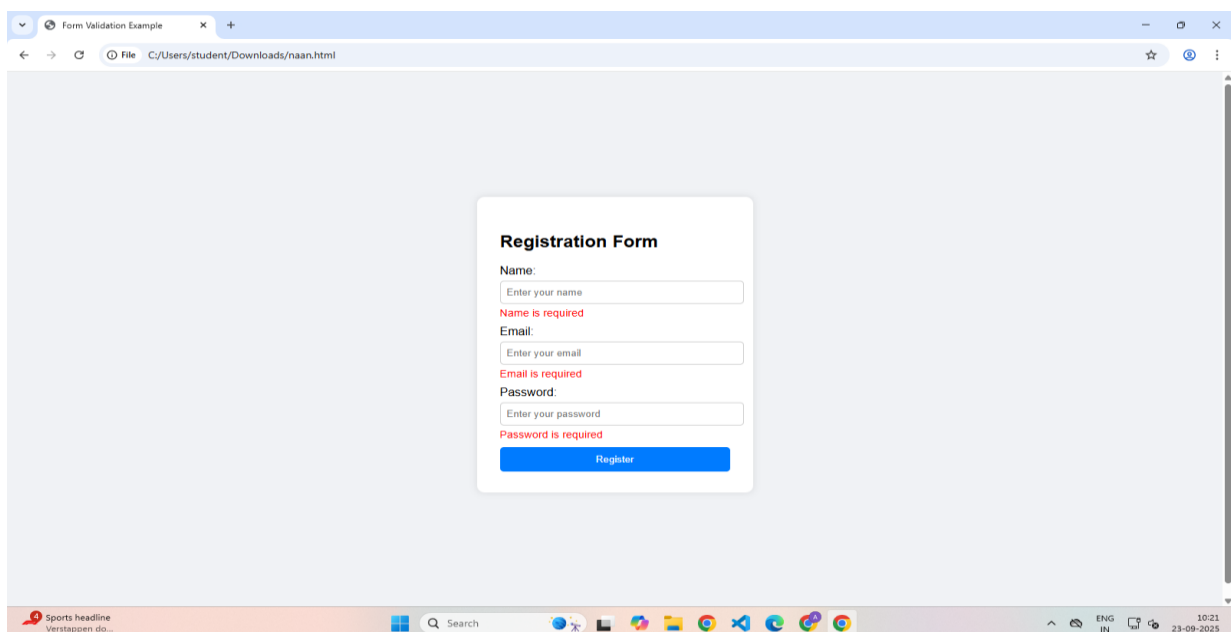
- Test empty fields → should show error messages.
- Test invalid email format → should show error.

- Test weak password → should prevent submission.
 - **Real-time Validation Testing:**
 - Enter data in each field and verify messages appear instantly.
 - **Submission Testing:**
 - Attempt submission with invalid data → should block.
 - Submit valid data → should succeed.
-

5. Testing Control (GitHub) :

- **Version Control:**
 1. Commit changes after implementing each feature.
 2. git add .
 3. git commit -m "Implemented email validation"
 4. Push updates to GitHub repository.
 5. git push origin main
- **Issue Tracking / Pull Requests:**
 - Create issues for bugs or improvements.
 - Track feature testing and fixes using GitHub issues.
- **Collaboration:**
 - Team members can clone the repo, test, and submit pull requests.

Sample input and output:



The screenshot shows a web browser window with the title "Form Validation Example". The address bar shows the file path "C:/Users/student/Downloads/naan.html". The main content area displays a "Registration Form" with the following fields and validation messages:

- Name:** A text input field with the placeholder "Enter your name". Below it, a red error message reads "Name is required".
- Email:** A text input field with the placeholder "Enter your email". Below it, a red error message reads "Email is required".
- Password:** A text input field with the placeholder "Enter your password". Below it, a red error message reads "Password is required".

At the bottom of the form is a blue button labeled "Register". The browser's taskbar at the bottom shows various application icons and the system clock indicating 10:21 on 23-09-2025.

