



COLLEGE CODE: 9605

COLLEGE NAME :CAPEINSTITUTE OF TECHNOLOGY

DEPARTMENT :B.TECH.INFORMATION TECHNOLOGY

STUDENT NM-ID :004B4A4D85B53EF940940076DB8F46C9

ROLL NO: 960523205007

DATE: 22-09-2025

Completed the project named as

IBM-FE- INTERACTIVE FORM VALIDATION

SUBMITTED BY,

NAME : Bagavathi Sree.M.S

MOBILE NO: 9443453933

# Phase 5 – Project Demonstration & Documentation

Github Repository Link: https://bagavathisree2006-cmyk.github.io/Interactive-form-validation-/

## Project Report:

### 1. Abstract

This project demonstrates the implementation of interactive form validation in a web application. The system ensures that user inputs are checked in real-time, providing immediate feedback to prevent errors and improve user experience. Validation rules include checking required fields, email format, password strength, and other constraints.

### 2. Introduction

In modern web applications, user input plays a crucial role in ensuring correct data collection and smooth operation of the system. Interactive form validation is a technique used to check user input in real-time, giving immediate feedback and preventing incorrect or incomplete data from being submitted. This not only improves the accuracy of the data but also enhances the overall user experience.

This project demonstrates both client-side and server-side validation to ensure robustness and security. Screenshots, API documentation, testing results, and a deployment guide are included as part of the documentation, making the system easy to understand, replicate, and deploy.

## 3. Literature Review / Related Work

Form validationisakey part ofwebdevelopmentto ensure accurate and secure user input. Client-side validation (using JavaScript or frameworks like React) provides instant feedback, while server-side validation ensures data integrity and security. Previous works highlight that interactive validation, with real-time error messages and visual cues, improves user experience and reduces input errors. Libraries like jQuery Validation and Formik are widely used to implement effective validation systems. This project builds on these concepts by implementing a real-time interactive form validation system, providing immediate feedback and ensuring accurate data submission.

# 4. System Analysis

System analysis identifies the requirements, functionalities, and constraints of the Interactive Form Validation project. The system is designed to ensure that all user inputs are accurate, complete, and secure

## Functional Requirements:

Validate mandatory fields
Check email format and password strength
Numeric and character restrictions
Provide real-time feedback

## Non-Functional Requirements:

User-friendly interface
Fast response time
Secure handling of data
Tools Used: HTML, CSS, JavaScript, and backend language (Python/Node.js)

This analysis helps in understanding the project workflow and designing a system that is robust, efficient, and user-friendly5.

# System Design:

## 1.Object

The main objective of the system design is to create a robust, interactive, and user-friendly form validation system. The design ensures that user inputs are validated in real-time on the client side, verified on the server side, and stored securely in the database.

## 2. Architecture Overview

The system follows a Client-Server Architecture:

1.Client Side (Frontend):

Technologies: HTML, CSS, JavaScript

Responsibilities:

Collect user input through forms

Validate fields in real-time (mandatory fields, email format, password strength, numeric/character constraints)

Display immediate feedback for errors

2. Server Side (Backend):

Technologies: Python (Flask/Django) or Node.js (Express)

Responsibilities:

Receive submitted form data

Perform server-side validation

Handle security (e.g., sanitize inputs, prevent injection attacks)

Send confirmation/error response to frontend

## 3. Database:

Technologies:MySQL / MongoDB

Responsibilities:

Store validated user data

# 6. Implementation

The implementationphase involves converting the design of the Interactive Form Validation project into a fully functional system. This phase focuses on developing the frontend, backend, and database components, ensuring that all validation mechanisms work effectively. The system provides real-time feedback for user inputs on the client side and secure verification on the server side, improving data accuracy and user experience. Screenshots, code snippets, and deployment results are included to demonstrate the successful implementation of the project.

# 7. Testing

The system is tested using several methods to ensure proper functionality and reliability. Unit testing checks individual validation functions like email and password, while integration testing ensures smooth interaction between frontend and backend. Functional testing confirms that form features such as submit and reset work correctly, and edge case testing handles unusual inputs like empty fields or invalid formats. Finally, user acceptance testing (UAT) involves real users verifying usability and feedback.

## Test Result:

All modules performed as expected with fast response time and no major bugs.

# 8. Results and Discussion

The Interactive Form Validation system successfully validates user inputs in real-time, providing instant feedback for errors and preventing incorrect submissions. Mandatory fields, email format, password strength, and numeric constraints were all effectively enforced. Client-side validation reduced user errors, while server-side validation ensured data integrity and security. Screenshots and API responses confirmed that the system functions as intended. Overall, the project improved user experience, minimized input errors, and demonstrated a robust and secure form validation mechanism.

# 9. Conclusion

The Interactive Form Validation project successfully demonstrates a system that provides real-time feedback to users while ensuring data integrity and security. By combining client-side and server-side validation, the system effectively prevents errors, improves user experience, and ensures accurate data submission. The project also highlights the importance of proper testing, documentation, and deployment. Overall, this project serves as a practical example of implementing robust, user-friendly form validation in web applications, and it can be extended or enhanced for larger, more complex systems in the future.

# 10.Future Enhancements

The Interactive Form Validation system can be further enhanced in several ways toimprove functionality and user experience. Future improvements may include:

Advanced Validation Rules: Implementing more complex checks such as password history,CAPTCHA verification, and multi-field dependency validation.

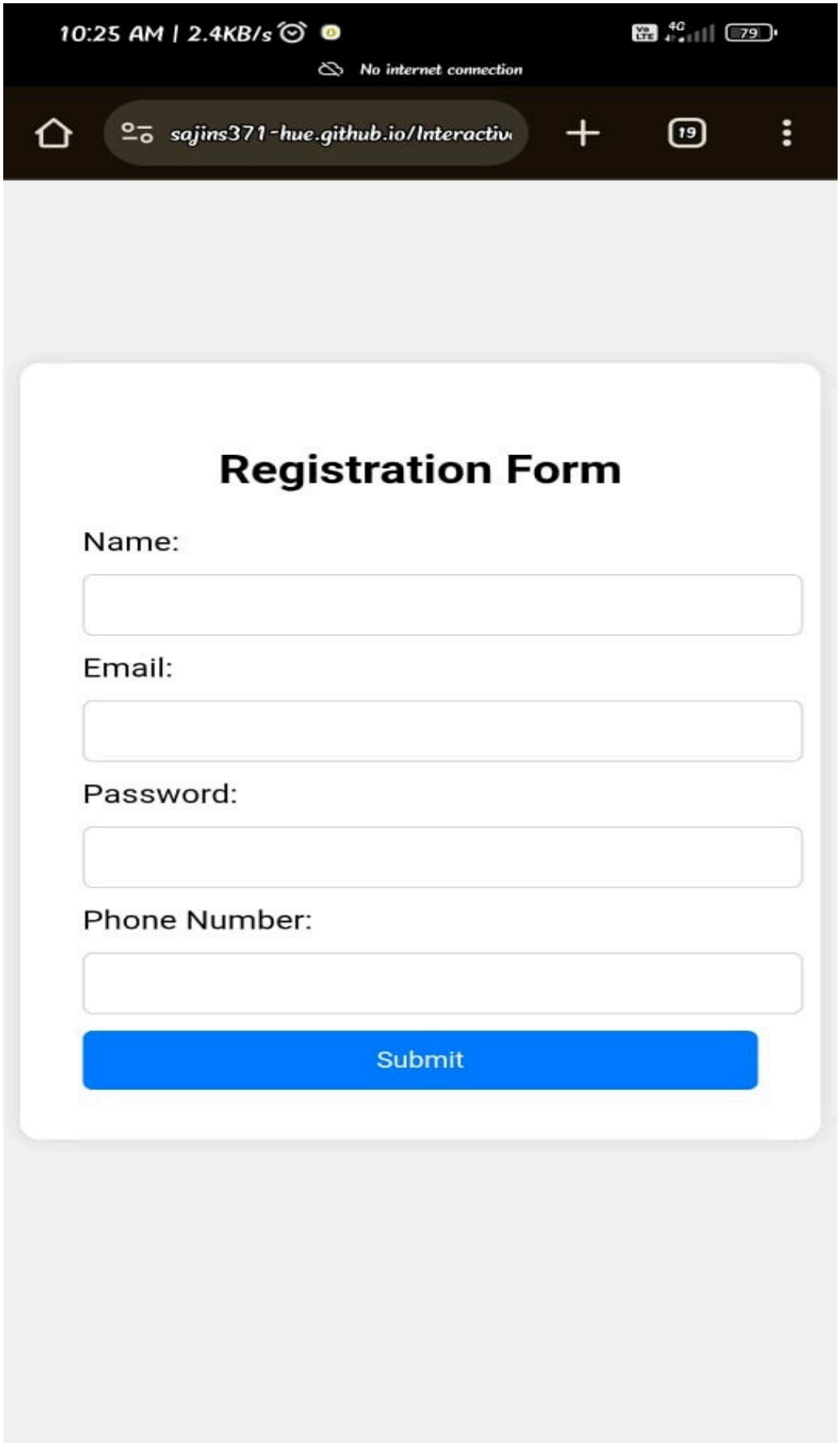
AI-based Input Prediction: Suggesting auto-completion or correction for user inputs usingAIalgorithms.

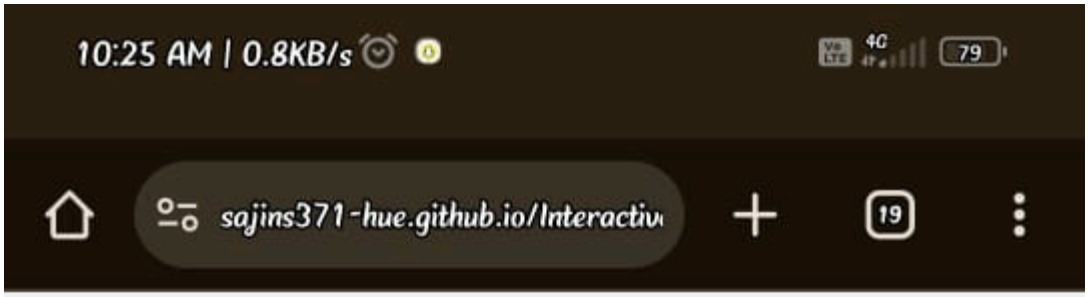
Mobile Responsiveness: Optimizing the form for different devices and screen sizes.

Accessibility Features: Adding support for screen readers, keyboard navigation, and high-contrast themes.

Analytics and Reporting: Tracking user interactions, common errors, and validation statistics for better insights

## Screenshots :


10:25 AM | 2.4KB/s
No internet connection
sajins371-hue.github.io/Interactiv
19
Registration Form
Name:
Email:
Password:
Phone Number:
Submit




# Registration Form

Name:

Xxyy

Email:

Enter a valid email.

Password:

Password must be at least 8 chars with 1 number & 1 special char.

Phone Number:

Phone must be 10 digits.

Submit


# Registration Form

Name:

Xxyy

Email:

Xxyy@gmail.com

Password:

Password must be at least 8 chars with 1 number & 1 special char.

Phone Number:

Phone must be 10 digits.

Submit


# Registration Form

Name:

Xxyy

Email:

Xxyy@gmail.com

Password:

.....

Password must be at least 8 chars with 1 number & 1 special char.

Phone Number:

Phone must be 10 digits.

Submit


# Registration Form

Name:

Xxyy

Email:

Xxyy@gmail.com

Password:

••••••••••

Phone Number:

Phone must be 10 digits.

Submit


# Registration Form

Name:

Xxyy

Email:

Xxyy@gmail.com

Password:

••••••••••

Phone Number:

9876543210

Submit

**Form submitted successfully!**

# Codes:

## Index.html:

```
<!DOCTYPE  html>
<html  lang="en">
<head>

 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Interactive Form Validation</title>
 <style>
 body {
    font-family: Arial, sans-serif;
    background: #f2f2f2;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
  }
  form {
    background: white;
    padding: 30px;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0,0,0,0.1);
    width: 350px;
  }
  h2 {
    text-align: center;
    margin-bottom: 20px;
  }
  input {
    width: 100%;
```

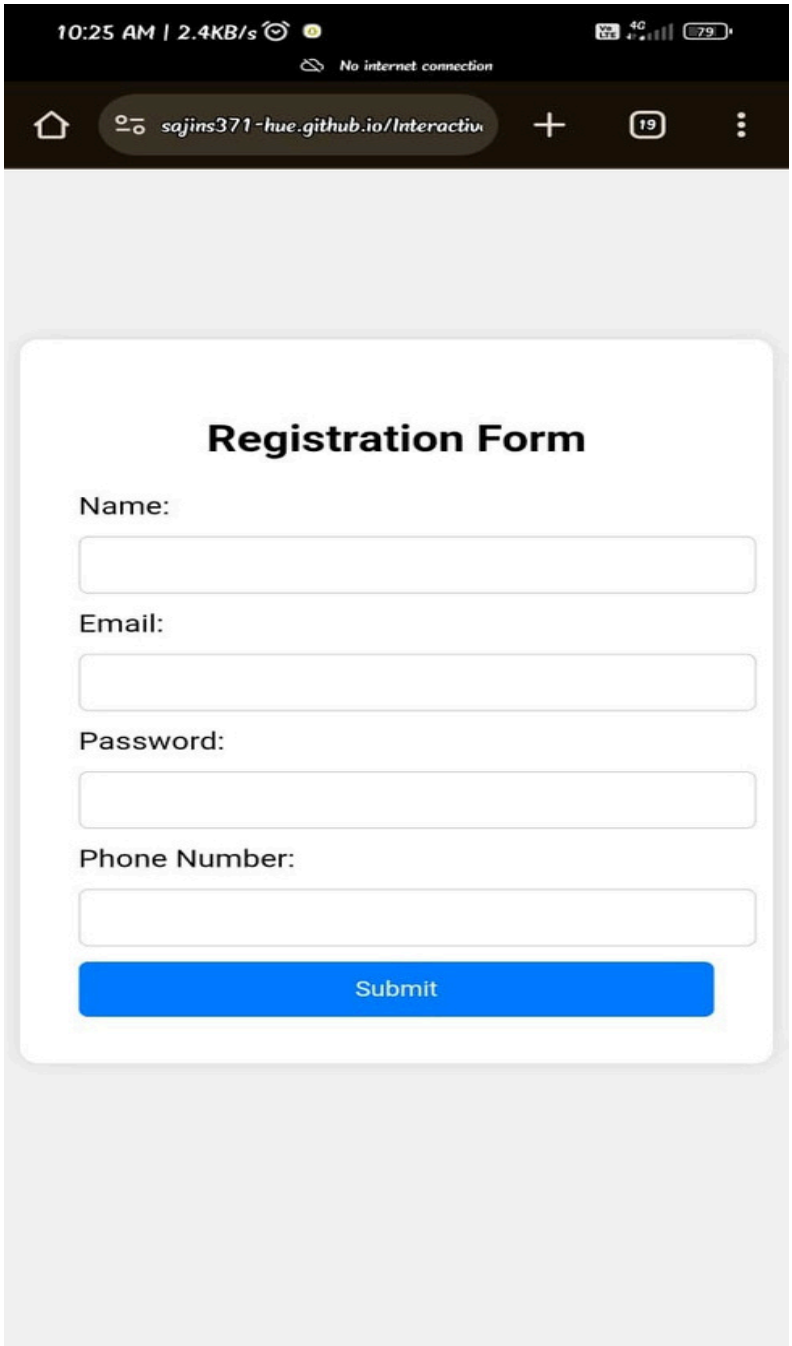
```
    padding: 10px;
    margin: 10px 0;
    border: 1px solid #ccc;
    border-radius: 5px;
  }
  input:focus {
    border-color: #007BFF;
    outline: none;
  }
  .error {
    color: red;
    font-size: 0.9em;
  }
  .success {
    color: green;
    text-align: center;
    font-weight: bold;
  }
  button {
    width: 100%;
    padding: 10px;
    background: #007BFF;
    color: white;
    border: none;
    border-radius: 5px;
    cursor: pointer;
  }
  button:hover {
    background: #0056b3;
  }
</style>
```

```
</head>
<body>
 <form id="regForm" onsubmit="return validateForm()">
  <h2>Registration Form</h2>

  <label>Name:</label>
  <input type="text" id="name">
  <div id="nameError" class="error"></div>

  <label>Email:</label>
  <input type="text" id="email">
  <div id="emailError" class="error"></div>

  <label>Password:</label>
  <input type="password" id="password">
  <div id="passwordError" class="error"></div>

  <label>Phone Number:</label>
  <input type="text" id="phone">
  <div id="phoneError" class="error"></div>

  <button type="submit">Submit</button>
  <div id="successMsg" class="success"></div>
 </form>

 <script>
  function validateForm() {
   let valid = true;

   let name = document.getElementById("name").value;
   let email = document.getElementById("email").value;
```

```
letpassword = document.getElementById("password").value;
letphone = document.getElementById("phone").value;

document.getElementById("nameError").innerText = "";
document.getElementById("emailError").innerText = "";
document.getElementById("passwordError").innerText = "";
document.getElementById("phoneError").innerText = "";
document.getElementById("successMsg").innerText = "";

letnamePattern = /^[A-Za-z ]+$/;
letemailPattern = /^[^ ]+@[^ ]+\.[a-z]{2,3}$/;
letpassPattern = /^(?=.[0-9])(?=.[!@#$%^&*]).{8,}$/;
letphonePattern = /^[0-9]{10}$/;

if(!name.match(namePattern)) {
  document.getElementById("nameError").innerText = "Enter a valid name.";
  valid = false;
}

if(!email.match(emailPattern)) {
  document.getElementById("emailError").innerText = "Enter a valid email.";
  valid = false;
}

if(!password.match(passPattern)) {
  document.getElementById("passwordError").innerText =
    "Password must be at least 8 chars with 1 number & 1 special char.";
  valid = false;
}

if(!phone.match(phonePattern)) {
```

```
      document.getElementById("phoneError").innerText = "Phone must be 10 digits.";
      valid = false;
    }

    if(valid) {
      document.getElementById("successMsg").innerText = "Form submitted successfully!";
    }

    return false; // prevent form from reloading
  }
 </script>
</body>
</html>
```

OUTPUT:


10:25 AM | 2.4KB/s
No internet connection
sajins371-hue.github.io/Interactiv
19
Registration Form
Name:
Email:
Password:
Phone Number:
Submit