

Training Day 14

Day 14– 10th July 2025

Forms and Form Validation using JavaScript

Detailed Description:

On Day 14, we explored **HTML forms** and how **JavaScript enhances their functionality** by validating input before submission.

The instructor emphasized that **form validation** improves user experience, prevents errors, and ensures **reliable data collection**.

◆ 1. Introduction to HTML Forms

Forms are essential for **collecting user input** such as text, email, passwords, and selections. We learned about basic form elements:

- `<input type="text">` – Single-line text
- `<input type="email">` – Email input
- `<input type="password">` – Password input
- `<textarea>` – Multi-line text
- `<select>` – Dropdown selection
- `<button>` or `<input type="submit">` – Submission

Example practiced:

```
<form id="myForm">
```

By : Malika

URN : 2302600

CRN :2315145

Name: <input type="text" id="name">

Email: <input type="email" id="email">

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

</form>

◆ 2. Importance of Form Validation

Form validation ensures that users enter **correct and complete information**.

Benefits include:

- Preventing empty or invalid input.
- Reducing server errors and improving data integrity.
- Providing instant feedback to users.

◆ 3. JavaScript Validation Techniques

We practiced using **JavaScript to validate forms** before submission:

Basic validation example:

```
document.getElementById("myForm").onsubmit = function() {
```

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

```
    if(name == "") {
```

```
        alert("Name cannot be empty!");
```

```
    return false; // Prevent form submission

}

}
```

- We learned to check for empty fields, proper email format, password strength, and dropdown selection.

Email validation example:

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

let regex = /^S+@S+\.S+$/;

if(!regex.test(email)) {

    alert("Enter a valid email address!");

    return false;

}
```

◆ 4. Providing User Feedback

We added **real-time feedback** to users by changing **styles or displaying messages**:

```
if(name == "") {

    document.getElementById("name").style.border = "2px solid red";

    alert("Please enter your name");

}
```

This improved **user experience** and visually guided the user to correct mistakes.

◆ 5. Practical Exercises

- Created a **contact form** with name, email, message, and validation checks.
 - Validated input for **required fields, email format, and password length**.
 - Prevented form submission until all conditions were met.
 - Practiced giving **instant feedback** through alerts and dynamic style changes.
-

Learning Outcomes:

- Learned how to **create and structure HTML forms**.
- Understood the importance of **form validation** for usability and data integrity.
- Practiced using **JavaScript to validate user input**.
- Learned to provide **instant feedback** to users through alerts and styling.
- Gained skills to build **interactive, error-free forms** for websites.