

Tops Technology

Module 16)

Python DB and Framework

Presented By : Nandni Vala

Form Validation using JavaScript

1. Using JavaScript for front-end form validation.

➤ **Create an HTML Form**

- Define the form structure with input fields, validation rules, and error messages.

- `<form id="registrationForm">`
 - `<label for="username">Username:</label>`
 - `<input type="text" id="username" name="username" required>`
 - `<small id="usernameError" class="error-message"></small>`
 - `<label for="email">Email:</label>`
 - `<input type="email" id="email" name="email" required>`
 - `<small id="emailError" class="error-message"></small>`
 - `<label for="password">Password:</label>`
 - `<input type="password" id="password" name="password" minlength="6" required>`
 - `<small id="passwordError" class="error-message"></small>`
-

- `<button type="submit">Register</button>`
 - `</form>`
 - **Add JavaScript for Validation**
 - Attach event listeners and define validation logic.
 - `document.getElementById('registrationForm').addEventListener('submit', function (e) {`
 - `e.preventDefault(); // Prevent form submission for validation`
 - `let isValid = true;`
 - `// Validate Username`
 - `const username =`
 `document.getElementById('username').value.trim();`
 - `if (username === "") {`
 - `showError('usernameError', 'Username is required.');`
 - `isValid = false;`
 - `} else {`
 - `showError('usernameError', "");`
 - `}`
-

```
➤ // Validate Email
➤ const email = document.getElementById('email').value.trim();
➤ const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
➤ if (!emailRegex.test(email)) {
➤     showError('emailError', 'Enter a valid email address.');
```

➤ isValid = false;

➤ } else {

➤ showError('emailError', "");

➤ }const password = document.getElementById('password').value.trim();

➤ if (password.length < 6) {

➤ showError('passwordError', 'Password must be at least 6 characters.');

➤ isValid = false;

➤ } else {

➤ showError('passwordError', "");

➤ }

➤ // Submit the form if all fields are valid

➤ if (isValid) {

➤ alert('Form submitted successfully!');

➤ // e.target.submit(); // Uncomment to submit the form

➤ }

➤ });

- `function showError(elementId, message) {`
- `document.getElementById(elementId).innerText = message;`
- `}`

➤ 3. CSS for Error Styling

- Style the error messages for better visibility.

- `.error-message {`

- `color: red;`

- `font-size: 12px;`

- `margin-top: 5px;`

- `}`

➤ **Advantages:**

- Real-time feedback.

- Reduces invalid data submission.

- Enhances user experience.

