

Day 10: Forms & Input Validation

فارمز اور ڈیٹا کی تصدیق

Quote of the Day: "پہلے سوچو، پھر لکھو — اچھا کوڈ لکھنے سے پہلے اچھی سوچ ضروری ہے۔"

"Think first, then write — good thinking before good code."

— Every Senior Developer Ever

آج کا دن خاص ہے! MILESTONE DAY

جو بو، یا Careem، بو Daraz پر ہوتی ہے — چاہے real-world website آج تم ایک ایسی چیز سیکھو گے جو بر کام نہیں کر سکتی۔ app کے بغیر کوئی بھی Forms Validation اور

Today's Learning Goals (آج کے اہداف)

By the end of today, you will:

- کیوں ضروری ہے form validation
- JavaScript سے form inputs چیک کرنا
- Users کو red error messages دکھانا
- Email, phone, age — سب validate کرنا
- Student Registration Form بنانا (challenge project)

Time Breakdown (کل وقت: 150 منٹ)

- ⏱ 7:00–7:05 PM (5 min): Daily standup & yesterday's review
- ⏱ 7:05–8:05 PM (60 min): Concept learning
- ⏱ 8:05–8:50 PM (45 min): Hands-on practice
- ⏱ 8:50–9:25 PM (35 min): Challenge project
- ⏱ 9:25–9:30 PM (5 min): Quiz & reflection

What We're Building Today

بنائیں گے آج بم ایک **Student Registration Form** جو:

- Name, Email, Phone, Age, University, Program کے لئے گی
- نہیں ہونے دے گی data submit غلط
- User کو exactly بتائیں گی کیا غلط ہے
- Valid submission پر success message دکھائیں گی

Why This Matters for Your Career:

کرو گے task وباں پہلا — Arbisoft، Devsinc، NetSol میں تم "Fix this form validation." کر لو۔ آج یہ master کر لو۔

SECTION 1: Conceptual Foundation (40%)

کیا ہے اور کیوں؟ Form Validation سمجھنا

The Real-World Analogy — Daraz Order کا سوچو

جب تم Daraz validation کرتے ہو۔ سوچو اگر order form fill پر کچھ نہ کرے تو Daraz کو اگر کچھ کوئی empty address submit کر دے X

- Phone number 3 digits X
- City field "abc123" میں لکھ دے X

پھر rider کو جائے گا? Customer Company ناراض۔ نقصان۔

Validation = Guard at the Gate (دروازے پر چوکیدار)

چیک کرتا ہے، ویسے بی school gate پر چوکیدار بر student ID card کا JavaScript validation input ہر BEFORE data کرتی ہے آگے جائے۔

User types data → [JavaScript Validation Guard] →  Correct? Submit!
→  Wrong? Show Error!

Why Does This Matter?

Without Validation:

- Database mein garbage data store ہے جاتا ہے
- App crash ہو سکتی ہے
- Security risks پیدا ہوتے ہیں
- Users confuse ہو جاتے ہیں

With Validation:

- Clean, correct data ملتا ہے
- User clear feedback کو ملتا ہے
- App professional لگتی ہے
- Developer job safe رہتی ہے 😊

The Mental Model — تین سوال

صرف تین سوالوں کا جواب ہے validation rule پر:

1. field empty ہے کیا " " کیا field بے ضروری ہے؟
2. format صحیح ہے کیا @ میں ہونا چاہیے "Email"
3. value range میں ہے کیا 120 سے کم درمیان ہو "Age 1"

بے تین سوال یاد رکھو — باقی سب انہی سے بتا بے۔

SECTION 2: Building Blocks (30%)

ایک ایک چیز سیکھیں

Building Block #1: Form سے Value لینا

کیا ہے؟ (What is it?)

JavaScript سے input field کی value پڑھنا۔

Step by Step:

```
<!-- HTML میں input -->
<input type="text" id="studentName" placeholder="اپنا نام لکھیں">
<button onclick="checkForm()">Submit</button>
```

```
function checkForm() {
    // Step 1: Element پکڑو (Grab the element)
    const nameInput = document.getElementById('studentName');

    // Step 2: Value پڑھو (Read the value)
    const nameValue = nameInput.value;

    // Step 3: Check کرو (Check it)
    console.log(nameValue); // User نے کیا لکھا؟
}
```

Common Mistake:

```
// ❌ Wrong - element، نہیں value چاہیے
const name = document.getElementById('studentName');
if (name == "") { ... } // ~ element ہے، string نہیں!

// ✅ Right - .value لگاؤ
const name = document.getElementById('studentName').value;
if (name == "") { ... } // اب string ہے ✓
```

Check Your Understanding:

- کیا ملتا ہے؟ `.value`، بغیر
- کیا ہوگی `value` لکھے تو user اگر `_spaces` تو?

Building Block #2: Empty Check (خالی چیک)

کیا ہے؟

نے کچھ لکھا یا نہیں۔ user کرنا کہ

```
function validateName(name) {  
    // .trim() spaces سے بچ جائے  
    // " " (spaces) بھی empty count ہو گئی  
  
    if (name.trim() === "") {  
        return "نام ضروری ہے"; // Error message  
    }  
  
    if (name.trim().length < 3) {  
        return "نام کم از کم 3 حروف کا بونا چاہیے";  
    }  
  
    return ""; // Empty string = کوئی error نہیں = ✓  
}  
  
// TODO: اب تم try کرو اسے زیادہ تو تو سے error دو  
// اگر name 50 characters سے زیادہ ہو تو
```

 **Trick:** Empty string "" = false in JavaScript

```
if (!name.trim()) {  
    // اسے بھی سے empty check کرتا ہے  
}
```

Building Block #3: Error Message دکھانا

کیا ہے؟

User کو red text میں بتانا کیا غلط ہے۔

HTML Structure:

```
<div>  
    <label>نام:</label>  
    <input type="text" id="studentName">  
    <!-- div error ہے کے لیے -->  
    <span id="nameError" style="color: red;"></span>  
</div>
```

JavaScript:

```

function showError(elementId, message) {
    // Error span ڈھونڈو
    const errorSpan = document.getElementById(elementId);

    // Message لگاؤ
    errorSpan.textContent = message;
}

function clearError(elementId) {
    // Error ہٹاو جب valid ہو
    const errorSpan = document.getElementById(elementId);
    errorSpan.textContent = "";
}

// Use کیسے کریں:
showError('nameError', 'نام خالی نہیں چھوڑ سکتے');
clearError('nameError'); // جب صحیح لکھ دیں

```

Building Block #4: Email Validation

کیا ہے؟

Check کرنا کہ email کا format صحیح ہے۔

Simple Logic (پہلے regex بغير):

```

function validateEmail(email) {
    // Email میں @ ہونا چاہیے
    if (!email.includes('@')) {
        return "Email میں @ ہونا ضروری ہے";
    }

    // @ ہونا چاہیے کے بعد .
    const atIndex = email.indexOf('@');
    const afterAt = email.substring(atIndex);

    if (!afterAt.includes('.')) {
        return "Email میں domain صحیح نہیں (example@gmail.com)";
    }

    return ""; // ✅ Valid
}

// TODO: test ان values سے:
// validateEmail("sharjeel")      → کیا ملے گا?
// validateEmail("sharjeel@")     → کیا ملے گا?
// validateEmail("sharjeel@g.com") → کیا ملے گا?

```

Advanced (Regular Expression — regex):

```
function validateEmailRegex(email) {
    // ~ pattern check کچھ@کچھ.کچھ کرتا ہے:
    const emailPattern = /^[^\s@]+@[^\s@]+\.\.[^\s@]+$/;

    // .test() میں pattern match ہے true/false
    if (!emailPattern.test(email)) {
        return "Email کا format گلط ہے";
    }

    return ""; // ✅ Valid
}
```

کو ڈرو مت! اسے ایسے پڑھو:

/^[\^\s@]+ @ [\^\s@]+ \. [\^\s@]+\$/
| | |
کچھ حروف @ domain . extension
کوئی space | (gmail) (com/pk)
یا @ نہیں

Building Block #5: Pakistan Phone Validation

Format: 03XX-XXXXXXX 3-92+ L_uXX-XXXXXXX

```
function validatePhone(phone) {
    // Pakistan mobile number pattern
    // 03 2 بھر شروع و، 7 digits، بھر dash، 7 بھر digits
    const pakistanPhone = /^(03\d{2}-\d{7})|(+92-3\d{2}-\d{7})$/;

    if (!pakistanPhone.test(phone)) {
        return "Phone format: 03XX-XXXXXXX 3-92+ ٢٠٢-XXXXXXX";
    }

    return ""; // ✅ Valid
}

// ✅ valid بیں:
// "0311-1234567" ✅
// "+92-311-1234567" ✅

// ✗ invalid بیں:
// "03111234567" ✗ (dash نہیں)
// "1234567" ✗ (سٹ جھوٹا)
```

Building Block #6: Age Validation

```
function validateAge(ageInput) {  
    // String کو number بناؤ  
    const age = parseInt(ageInput);  
  
    // اس کیا number ہے؟  
    if (isNaN(age)) {  
        return "Age چابیے صرف number بونا";  
    }  
  
    // Range check  
    if (age < 1 || age > 120) {  
        return "Age 1 اور 120 کے درمیان بونا چابیے";  
    }  
  
    return ""; // ✅ Valid  
}  
  
// TODO: ان cases کیا ہوگا؟  
// validateAge("abc") → ?  
// validateAge("0") → ?  
// validateAge("25") → ?  
// validateAge("150") → ?
```

💻 SECTION 3: Progressive Practice (20%)

Exercise 1: Guided Practice (بم ساتھ کریں)

Scenario: Careem driver registration form کا ایک حصہ بناؤ۔

Starter Code:

```
<!DOCTYPE html>  
<html>  
<head>  
    <title>Careem Driver Registration</title>  
    <style>  
        .error { color: red; font-size: 12px; }  
        input { display: block; margin: 5px 0; padding: 8px; width: 300px; }  
        button { margin-top: 10px; padding: 10px 20px; background: green;  
color: white; border: none; cursor: pointer; }  
    </style>  
</head>  
<body>  
    <h2>Careem Driver Registration</h2>
```

```

<div>
    <label>Full Name:</label>
    <input type="text" id="driverName" placeholder="Muhammad Ali">
    <!-- TODO: اپنے error span add کرو -->
</div>

<div>
    <label>Phone:</label>
    <input type="text" id="driverPhone" placeholder="03XX-XXXXXXX">
    <!-- TODO: اپنے error span add کرو -->
</div>

<button onclick="validateDriver()">Register</button>

<script>
    function validateDriver() {
        // TODO Step 1: دونوں inputs کی values کو
        const name = _____;
        const phone = _____;

        // TODO Step 2: ایک variable track کرنے کے لئے کوئی
        // کوئی کہ error ہے (true/false)
        let isValid = true;

        // TODO Step 3: Name validate کرو
        // HINT: empty تو error دکھائو، isValid = false کرو
        if (name.trim() === "") {
            showError('nameError', 'نام ضروری ہے');
            isValid = false;
        } else {
            clearError('nameError');
        }

        // TODO Step 4: Phone validate کرو (Pakistan format)
        // HINT: Building Block #5 دیکھو
    }

    // TODO Step 5: اگر سب تو success دکھائو
    if (isValid) {
        alert('✅ Driver registered successfully!');
    }
}

function showError(id, message) {
    document.getElementById(id).textContent = message;
}

function clearError(id) {
    document.getElementById(id).textContent = "";
}
</script>

```

```
</body>  
</html>
```

Test Your Code:

Expected behavior:

- Empty name → red error "نام ضروری ہے"
- Wrong phone → red error about format
- Both correct → alert "Driver registered!"

Exercise 2: Your Turn (اب آپ)

Problem: JazzCash account کھولنے کا mini form بناؤ۔

Requirements:

- Name field (3+ characters)
- CNIC field (format: XXXXX-XXXXXXX-X — 13 digits + 2 dashes)
- Phone field (Pakistan format)
- Validate on button click
- Show specific errors for each field

Thinking Framework:

1. چاہیئں error ہیں؟ کتنے inputs پر سوچو: کتنے?
2. کیا ہے field کی validation پر سوچو: بر?
3. یوں تو کیا ہو pass آخر میں: سب?

کوشش کرو پہلے! 

Hints (اگر بہنس جاؤ):

- CNIC format hint: `/^\d{5}-\d{7}-\d$/`
- \d کا مطلب ہے: کوئی بھی digit (0-9)
- {5} کا مطلب ہے: exactly 5 بار



SECTION 4: Challenge Project (10%)

اج کا چیلنچ — Student Registration Form

The Problem:

Superior University کو ایک online student registration form چاہیے جو غلط data accept نہ کرے۔

What You're Building:

کرے۔ ایک complete HTML form with JavaScript validation جو students details collect validate کرے۔

Success Criteria:

- 6 fields validate ہوں
 - field کا اپنا specific error message ہو
 - valid submission پر success message آئے
 - Form submit کے بعد clear ہو جائے
 - No console errors
-

Phase 1: Planning (سوچیں پہلے)

Code دو جواب سے پہلے لکھنے سے:

1. کتنے input fields ہیں? _____
2. field کی validation rule ہے کیا؟
 - Name: _____
 - Email: _____
 - Phone: _____
 - Age: _____
 - University: _____
 - Program: _____
3. Error دکھائیں گے کہاں؟ _____
4. Success دکھائیں گے کیسے؟ _____

Planning Checkpoint:

- approach میں نے اپنے notes لکھے
 - HTML structure مجھے پتھے کیسی ہو گی
 - functions بنانے ہیں مجھے پتھے کتنے
-

Phase 2: Foundation (بنیاد)

Starter Code:

```
<!DOCTYPE html>
<html lang="ur">
<head>
    <meta charset="UTF-8">
    <title>Superior University - Student Registration</title>
    <style>
        * { box-sizing: border-box; margin: 0; padding: 0; }
        body {
            font-family: Arial, sans-serif;
            background: #f0f4f8;
            display: flex;
            justify-content: center;
            padding: 40px 20px;
        }
        .form-container {
```

```
background: white;
padding: 30px;
border-radius: 10px;
box-shadow: 0 4px 20px rgba(0,0,0,0.1);
width: 100%;
max-width: 500px;
}
h2 {
    text-align: center;
    color: #2c3e50;
    margin-bottom: 25px;
}
.form-group {
    margin-bottom: 20px;
}
label {
    display: block;
    margin-bottom: 5px;
    font-weight: bold;
    color: #34495e;
}
input, select {
    width: 100%;
    padding: 10px;
    border: 2px solid #ddd;
    border-radius: 5px;
    font-size: 14px;
}
input:focus, select:focus {
    outline: none;
    border-color: #3498db;
}
.error-msg {
    color: #e74c3c;
    font-size: 12px;
    margin-top: 4px;
    display: block;
}
/* TODO: Add .input-error class for red border on invalid inputs */
/* TODO: Add .input-success class for green border on valid inputs */

button {
    width: 100%;
    padding: 12px;
    background: #2980b9;
    color: white;
    border: none;
    border-radius: 5px;
    font-size: 16px;
    cursor: pointer;
}
button:hover { background: #1a5276; }
```

```
#successMessage {
    display: none;
    background: #2ecc71;
    color: white;
    padding: 15px;
    border-radius: 5px;
    text-align: center;
    margin-top: 15px;
    font-size: 16px;
}
</style>
</head>
<body>
<div class="form-container">
    <h2>🎓 Student Registration</h2>

    <!-- Field 1: Name -->
    <div class="form-group">
        <label for="studentName">Full Name *</label>
        <input type="text" id="studentName" placeholder="Muhammad Ali Khan">
        <!-- TODO: Add error span with id="nameError" and class="error-msg" -->
    </div>

    <!-- Field 2: Email -->
    <div class="form-group">
        <label for="studentEmail">Email Address *</label>
        <input type="text" id="studentEmail" placeholder="student@superior.edu.pk">
        <!-- TODO: Add error span for email -->
    </div>

    <!-- Field 3: Phone -->
    <div class="form-group">
        <label for="studentPhone">Phone Number *</label>
        <input type="text" id="studentPhone" placeholder="03XX-XXXXXXX">
        <!-- TODO: Add error span for phone -->
    </div>

    <!-- Field 4: Age -->
    <div class="form-group">
        <label for="studentAge">Age *</label>
        <input type="text" id="studentAge" placeholder="19">
        <!-- TODO: Add error span for age -->
    </div>

    <!-- Field 5: University -->
    <div class="form-group">
        <label for="studentUniversity">University Name *</label>
        <input type="text" id="studentUniversity" placeholder="Superior University">
        <!-- TODO: Add error span for university -->
    </div>
</div>
```

```

</div>

<!-- Field 6: Program -->
<div class="form-group">
    <label for="studentProgram">Program *</label>
    <!-- TODO: Change this to a <select> with options:
        - Select Program (default, disabled)
        - BS Computer Science
        - BS Software Engineering
        - BS Information Technology
        - ADP Computer Science
    -->
    <input type="text" id="studentProgram" placeholder="BS Computer
Science">
    <!-- TODO: Add error span for program -->
</div>

<button onclick="validateAndSubmit()">Register Student</button>

<!-- Success Message (hidden by default) -->
<div id="successMessage">
     Registration Successful! Welcome to Superior University!
</div>
</div>

<script>
// =====
// HELPER FUNCTIONS (ہم سے بننے والے اسے کیا جائے)
// =====

function showError(inputId, errorId, message) {
    // Error message کیجاوے
    document.getElementById(errorId).textContent = message;
    // TODO: Input border red کرو (add class 'input-error')
}

function clearError(inputId, errorId) {
    // Error message بے
    document.getElementById(errorId).textContent = "";
    // TODO: Input border green کرو (add class 'input-success')
}

// =====
// VALIDATION FUNCTIONS (تم بناؤ)
// =====

function validateName() {
    const name = document.getElementById('studentName').value;

    // TODO: Check 1 - empty ہے؟
    // TODO: Check 2 - 3 کم از کم 3 characters?
}

```

```

// اگر سب ٹھیک:
// clearError('studentName', 'nameError');
// return true;
}

function validateEmail() {
    const email = document.getElementById('studentEmail').value;

    // TODO: Empty check

    // TODO: @ contains check

    // TODO: Domain check (. بونا چابے after @)

    // ل regex use کرو: /^[^@\s]+@[^\s]+\.[^\s]+$/
}

function validatePhone() {
    // TODO: Complete this function
    // Pakistan format: 03XX-XXXXXXX
    // Regex: ^03\d{2}-\d{7}$
}

function validateAge() {
    // TODO: Complete this function
    // Number بونا چابے
    // Range: 15 کے درمیان 50 تک (student کے لئے)
}

function validateUniversity() {
    // TODO: Complete this function
    // Empty تو نہ
}

function validateProgram() {
    // TODO: Complete this function
    // Select بونا ضروری
}

// =====
// MAIN SUBMIT FUNCTION
// =====

function validateAndSubmit() {
    // بر validation function call کرو
    // دکھاؤ بون تو success سب

    // TODO: تمام validations call کرو
    // HINT: const nameValid = validateName();

    // TODO: بیں تو اگر سب valid:
    // 1. Success message دکھاؤ
    // 2. Form clear کرو (بر input کی value = "")
}

```

```

        // 3. 3 seconds بعد success message
    }
</script>
</body>
</html>

```

Phase 3: Milestones (سنگ میل)

Milestone 1: HTML Structure مکمل

- میں 6 fields HTML بین تمام
- field کا error span بے
- Program field ایک `<select>` بے
- Test: Browser سب دکھو لو — سب دکھ رہا ہے؟

Milestone 2: Validation Functions کام کر رہی بین

- Name validation کام کرتی ہے
- Email validation کام کرتی ہے
- Phone Pakistan format چیک ہوتا ہے
- Test: Console میں `validateName()` call کرو

Milestone 3: Error Messages صحیح جگہ

- اپنے نیچے آتا ہے field کا error
- Valid input پر error clear ہوتی ہے
- Red/green border کام کرتی ہے
- Test: ایک field empty چھوڑ کر submit کرو

Milestone 4: Submit Flow مکمل

- success ہے تو سب دکھ
- Form clear ہو جائے
- Partial errors ہمیں block کریں
- Test: آدھا form fill کرو کر کے submit کرو

Debugging Guide (اگر پہنس جاؤ)

Problem: Error message نہیں آ رہا

- Check: Error span کا id صحیح ہے document.getElementById میں؟
- Check: `class="error-msg"` لگا ہے span پر؟
- Check: F12 → Console میں کوئی red error ہے؟

Problem: Regex کام نہیں کر رہا

- Test regex تسلیم browser console میں:

```
// کیا ملتا ہے؟  
/^03\d{2}-\d{7}$/.test("0311-1234567")
```

- میں کچھ اور غلط ہے، ٹھیک ہے regex ملے تو

Problem: Submit ہو رہا ہے بغیر validation کے

- Check: validateAndSubmit() function میں return false لگا ہے اگر invalid?
- Check: validation functions true/false return کر رہی ہیں؟

Problem: Form clear ہو رہا ہے

```
// نہیں کرو:  
document.getElementById('studentName').value = "";  
// سب ایک ساتھ:  
document.getElementById('registrationForm').reset(); // اگر form کو id دو
```

Extension Challenges — اگر وقت ہو (بونس چیلنچ)

- **Real-time validation:** Submit سے پہلے بھی، جیسے error (use oninput event)
- **Password field add:** strength indicator کے ساتھ (Weak/Medium/Strong)
- **Confirm Password matching check:** کرو
- **CNIC field add:** format XXXXX-XXXXXXX-X

Daily Quiz (5 منٹ کا ٹیسٹ)

دیکھے جواب دو بغیر notes!

کیا کرتا ہے؟ 1.

```
const name = document.getElementById('myInput').value.trim();  
if (!name) { ... }
```

- A) Element ڈھونڈتا ہے
- B) Value کرتا ہے کہ لیتا ہے اور empty تو نہیں
- C) Input delete کرتا ہے
- D) Error دکھاتا ہے

► (کرو پہلے) جواب دیکھیں

Answer: B — .value value spaces، !name check empty لیتا ہے، بٹاٹا ہے

2. Pakistan phone 0311-1234567 validate کرنے کے لیے کون سا regex صحیح ہے؟

- A) /03\d{2}\d{7}/
- B) /^03\d{2}-\d{7}\$/
- C) /03XX-XXXXXXX/
- D) /phone+pakistan/

► جواب دیکھیں

Answer: B — ^ \$ پر ختم dash سے شروع، \d{7} exactly 7 digits

3. isNaN("abc") کیا کرے گا؟

- A) false
- B) true
- C) "abc"
- D) 0

► جواب دیکھیں

Answer: B (true) — isNaN = "is Not a Number" — "abc" number نہیں ہے، اس لیے true

4. Form validate کرنے کے بعد success message کیسے دکھائیں؟

- A) document.write("success")
- B) alert("success") صرف
- C) document.getElementById('msg').style.display = 'block'
- D) console.log("success")

► جواب دیکھیں

Answer: C — display:none سے block کریں کو۔ A outdated ہے، B alert ہے UI نہیں دیکھتا ہے

5. اگر validateEmail() false return کرے تو کیا ہونا چاہیے؟

- A) Form submit ہو جائے
- B) Form submit ہو، نہ error دکھائے
- C) Page refresh ہو
- D) کچھ نہیں

► جواب دیکھیں

Answer: B — Validation کا مقصد ہے data submit ہے، نہ ہو غلط ہے

Scoring:

- 5/5 Excellent! آج کا master!
- 4/5 Great! ایک review کرو
- 3/5 Good! Concepts دوبارہ پڑھو
- ❤️/5 کرو، سمجھو آجائے گی practice section repeat — کوئی بات نہیں

گھر کا کام (Today's Homework)

Required:

- مکمل کرو اگر نہیں بوا Challenge project
- GitHub پر commit کرو: git commit -m "Day 10: Student Registration Form with validation"
- مشکل لگا، وہ دوبارہ پڑھو جو concept

Optional (Bonus):

- Real-time validation add کرو (oninput event)
 - Password strength indicator بناؤ
 - کیا ہے کیا explain کرو کسی Urdu میں validation دوست کو
-

روزانہ کی سوچ (Daily Reflection)

لو اور جواب لکھو:

آج میں نے کیا سیکھا:

کیا مشکل لگا:

مجھے مزید کیا سیکھنا ہے:

میرا confidence level (1-10): _____

Tomorrow's Preview

کل ہم سیکھیں گے **Events & DOM Manipulation** — جہاں تم بناؤ گے ایک interactive Daraz-style product page!

آج رات تیاری:

- مکمل بوا آج کا challenge project
 - ہوتا ہے؟ event (button click, key press) سیکھو: کیا سوچو?
 - کل بھی کام آئے گے آج کی validation knowledge
-

مزید پڑھنا ہو تو (Resources)

3G-Friendly Links:

- MDN — Form Validation — Best reference
- MDN — Regular Expressions — Regex سیکھو
- Search YouTube: "JavaScript form validation Urdu" — video سیکھنے والوں کے لیے

CodeSensei's Tip of the Day:

Validation میں نہیں ہوتی کہی صرف JavaScript — real apps اور (JS) backend (Node.js) دونوں journey continue کرو!

"کوڈ سیکھنا ایک سفر ہے، منزل نہیں۔ بہ دن ایک قدم آگے۔"

"Learning to code is a journey, not a destination. One step forward every day."