

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

🎉 MILESTONE DAY — آج کا دن خاص ہے!

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

📋 Today's Learning Goals (آج کے اہداف)

By the end of today, you will:

- ☐ کیوں ضروری ہے form validation سمجھنا کہ
- ☐ چیک کرنا form inputs سے JavaScript
- ☐ دکھانا red error messages کو Users
- ☐ کرنا validate سب — Email، phone، age
- ☐ بنانا Student Registration Form (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 جو **Student Registration Form** آج ہم ایک

- لے گی Name، Email، Phone، Age، University، Program
- نہیں ہونے دے گی data submit غلط
- بتائے گی کیا غلط ہے exactly کو User
- دکھائے گی success message پر valid submission

Why This Matters for Your Career:

"Fix this form validation." اکثر یہی ہوتا ہے task وہاں پہلا — Arbisoft، Devsinc، NetSol — کرو گے apply جہاں تم company ہر کر لو۔ master یہ آج

🧠 SECTION 1: Conceptual Foundation (40%)

کیا ہے اور کیوں؟ Form Validation: سمجھنا

The Real-World Analogy — Daraz Order سوچو



نہ کرے تو Daraz validation کرتے ہو۔ سوچو اگر address form fill کرتے ہو تو order پر کچھ Daraz جب تم

- X کر دے empty address submit کوئی
- X کا لکھ دے 3 digits Phone number
- X لکھ دے "abc123" میں City field

کا نقصان۔ Company ناراض۔ Customer کھو جائے گا۔ Parcel؟ کہاں جائے گا rider پھر

Validation = Guard at the Gate (دروازے پر چوکیدار)

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

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

Why Does This Matter?

Without Validation:

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

With Validation:

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

The Mental Model — تین سوال

صرف تین سوالوں کا جواب ہے validation rule ہر

- | | |
|----------------------------|-------------------------------|
| 1. "یہ field empty کیا | → ضروری ہے field ہے؟ |
| 2. صحیح ہے؟ format کیا | → "Email میں @ ہونا چاہیے" |
| 3. میں ہے؟ value range کیا | → "Age 1 سے 120 کے درمیان ہو" |

یہ تین سوال یاد رکھو — باقی سب انہی سے بنتا ہے۔

SECTION 2: Building Blocks (30%)

ایک ایک چیز سیکھیں

Building Block #1: Form سے Value لینا

کیا ہے؟ (What is it?)

پڑھنا۔ value کی input field سے JavaScript

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 == "") { ... } // ہے string اب ✓
```


Check Your Understanding:

- ☐ کے کیا ملتا ہے؟ value. بغیر
- ☐ کیا ہوگی value لکھے تو spaces نے user اگر

Building Block #2: Empty Check (خالی چیک)

کیا ہے؟

نے کچھ لکھا یا نہیں۔ user کرنا کہ Check

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

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

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

Building Block #3: Error Message دکھانا

کیا ہے؟

میں بتانا کیا غلط ہے۔ User کو red text

HTML Structure:

```
<div>  
  <label>نام:</label>  
  <input type="text" id="studentName">  
  <!-- کے لیے ہے error div یہ خالی -->  
  <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 بٹاؤ جب valid ہو
  const errorSpan = document.getElementById(elementId);
  errorSpan.textContent = "";
}

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

```

Building Block #4: Email Validation

کیا ہے؟

صحیح ہے۔ email کا format کرنا کہ Check

Simple Logic (پہلے regex بغیر):

```

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

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


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

  return ""; //  Valid
}

// TODO: test کرو 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) {  
  // کرتا ہے: کچھ@کچھ.کچھ pattern check یہ  
  const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;  
  
  // .test() بتاتا ہے true/false  
  if (!emailPattern.test(email)) {  
    return "Email کا format ہے غلط";  
  }  
  
  return ""; //  Valid  
}
```

کو ڈرو مت! اسے ایسے پڑھو Regex:

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

Building Block #5: Pakistan Phone Validation

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

```
function validatePhone(phone) {  
  // Pakistan mobile number pattern  
  // 03 2 ہر 7 digits، dash، 7 digits سے شروع ہو، ہر 2  
  const pakistanPhone = /^(03\d{2}-\d{7}|\+92-3\d{2}-\d{7})$/;  
  
  if (!pakistanPhone.test(phone)) {  
    return "Phone format: 03XX-XXXXXXX 3-92+ یا XX-XXXXXXX";  
  }  
  
  return ""; //  Valid  
}  
  
// یہ valid ہیں:  
// "0311-1234567"   
// "+92-311-1234567"   
  
// یہ invalid ہیں:  
// "03111234567"  (dash نہیں)  
// "1234567"  (بہت چھوٹا)
```

Building Block #6: Age Validation

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

SECTION 3: Progressive Practice (20%)

Exercise 1: Guided Practice (ہم 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: یہاں error span add کرو -->
</div>

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

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

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

    // TODO Step 2: کر کے لیے track بناؤ variable ایک (true/false)
    // error ہے؟ کوئی
    let isValid = true;

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

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

    // TODO Step 5: دیکھاؤ success ہے تو valid اگر سب
    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: بناؤ۔ mini form کھولنے کا JazzCash account

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. چابیئیں error spans ہیں؟ کتنے inputs پہلے سوچو: کتنے
2. کیا بے validation کی field پھر سوچو: ہر
3. ہوں تو کیا ہو pass آخر میں: سب

!کوشش کرو پہلے

Hints (اگر پھنس جاؤ):

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



SECTION 4: Challenge Project (10%)

چیلنج — Student Registration Form

The Problem:

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

What You're Building:

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

Success Criteria:

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

Phase 1: Planning (سوچیں پہلے)

Code لکھنے سے پہلے جواب دو:

- _____ ہیں input fields کتنے؟
- _____ کیا ہے validation rule کی field ہر؟
 - Name: _____
 - Email: _____
 - Phone: _____
 - Age: _____
 - University: _____
 - Program: _____
- _____ کہاں دکھائیں گے Error؟
- _____ کیسے دکھائیں گے Success؟

Planning Checkpoint:

- ☐ لکھے notes کے approach میں نے اپنے
 - ☐ کیسی ہوگی HTML structure مجھے پتہ ہے
 - ☐ بنائے ہیں functions مجھے پتہ ہے 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>

<!-- 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 (یہ پہلے سے بنے ہیں)
  // =====

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

  function clearError(inputId, errorId) {
    // Error message ہٹاؤ
    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 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 چاہیے after @)

        // 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 چاہیے
        // Range: 15 کے درمیان سے 50 کے لیے (student)
    }

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

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

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

    function validateAndSubmit() {
        // validation function call کرو
        // success ہوں تو true سب دکھاؤ

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

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

```

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

Phase 3: Milestones (سنگ میل)

Milestone 1: HTML Structure مکمل

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

Milestone 2: Validation Functions کام کر رہی ہیں

- ☐ 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

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

Milestone 4: Submit Flow مکمل

- ☐ دکھے success ہو تو valid سب
- ☐ Form clear ہو جائے
- ☐ Partial errors بھی block کریں
- Test: submit کر کے form آدھا

Debugging Guide (اگر پھنس جاؤ)

Problem: Error message نہیں آ رہا

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

Problem: Regex کام نہیں کر رہا

- ☐ Test regex میں browser console پہلے:

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

- ☐ **true** میں کچھ اور غلط ہے function، ٹھیک ہے regex ملے تو **true**

Problem: Submit ہو رہا ہے بغیر validation کے

- ☐ Check: **validateAndSubmit()** function میں **return false** اگر **invalid** ہے
- ☐ Check: validation functions **true/false** return کریں؟ سب

Problem: Form clear ہو رہا نہیں

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

Extension Challenges (بونس چیلنج — اگر وقت ہو)

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

Daily Quiz (منٹ کا ٹیسٹ 5)

!دیکھو جواب دو **notes** بغیر

1. کیا کرتا ہے **code** یہ؟

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

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

► (کرو try پہلے) جواب دیکھیں

Answer: B — **value** value لیتا ہے، **trim()** spaces ہٹاتا ہے، **!name** check empty کرتا ہے

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

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

▶ جواب دیکھیں

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

3. `isNaN("abc")` کیا `return` کرے گا؟

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

▶ جواب دیکھیں

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

4. `Form validate` 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 hidden element کو۔ A outdated ہے، B alert ہے UI، D صرف developer دیکھتا ہے






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

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

▶ جواب دیکھیں

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

Scoring:

- 5/5  Excellent! آج کا master!
- 4/5  Great! ایک review کرو
- 3/5  Good! Concepts دوبارہ پڑھو
-  5/5  — practice section repeat گئی سمجھ آ جائے گی

Today's Homework (گھر کا کام)

Required:

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

Optional (بونس):

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

Daily Reflection (روزانہ کی سوچ)

2 minutes لو اور جواب لکھو

آج میں نے کیا سیکھا

کیا مشکل لگا

مجھے مزید کیا سیکھنا ہے

confidence level (1-10): میرا ____

Tomorrow's Preview




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

آج رات تیاری

- ☐ مکمل ہو challenge project آج کا
- ☐ (button click, key press) ہوتا ہے؟ event سوچو: کیا
- ☐ کل بھی کام آئے گی validation knowledge آج کی

Resources (مزید پڑھنا ہو تو)

3G-Friendly Links:

-  [MDN — Form Validation](#) — Best reference
-  [MDN — Regular Expressions](#) — سیکھو Regex
-  Search YouTube: "JavaScript form validation Urdu" — video لیے

CodeSensei's Tip of the Day: 💡

Validation میں نہیں ہوتی JavaScript کبھی صرف **frontend** (JS) اور **backend** میں real apps — یہ آدھا کام ہے۔ پوری frontend کرتے ہیں۔ آج تم نے validate دونوں (Node.js) journey continue سیکھا —! کرو

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

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

Day 10 of 21 — Superior University ADPCS | CodeSensei