

LAB 4

Web snippet

Work Proof

index.html

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Form Validation - JavaScript</title>

  <link rel="stylesheet" href="style.css">

</head>

<body>

  <div class="container">

    <div class="form-wrapper">

      <h1>Registration Form</h1>

      <p class="subtitle">Please fill in all required fields</p>

      <form id="registrationForm" novalidate>

        <!-- Full Name Field -->

        <div class="form-group">

          <label for="fullName">Full Name <span
class="required">*</span></label>
```

```
        <input type="text" id="fullName" name="fullName"
placeholder="Enter your full name">
        <span class="error-message" id="fullNameError"></span>
    </div>
```

```
<!-- Email Field -->
<div class="form-group">
    <label for="email">Email Address <span
class="required">*</span></label>
    <input type="email" id="email" name="email"
placeholder="example@email.com">
    <span class="error-message" id="emailError"></span>
</div>
```

```
<!-- Phone Number Field -->
<div class="form-group">
    <label for="phone">Phone Number <span
class="required">*</span></label>
    <input type="tel" id="phone" name="phone" placeholder="10-digit
phone number">
    <span class="error-message" id="phoneError"></span>
</div>
```

```
<!-- Password Field -->
<div class="form-group">
    <label for="password">Password <span
class="required">*</span></label>
```

```

        <input type="password" id="password" name="password"
placeholder="Create a strong password">

        <span class="error-message" id="passwordError"></span>

        <div class="password-requirements">

            <p>Password must contain:</p>

            <ul>

                <li id="length-check">At least 8 characters</li>

                <li id="uppercase-check">One uppercase letter</li>

                <li id="lowercase-check">One lowercase letter</li>

                <li id="number-check">One number</li>

                <li id="special-check">One special character (!@#$%^&*)</li>

            </ul>

        </div>

    </div>

<!-- Confirm Password Field -->

    <div class="form-group">

        <label for="confirmPassword">Confirm Password <span
class="required">*</span></label>

        <input type="password" id="confirmPassword"
name="confirmPassword" placeholder="Re-enter your password">

        <span class="error-message" id="confirmPasswordError"></span>

    </div>

<!-- Age Field -->

    <div class="form-group">

        <label for="age">Age <span class="required">*</span></label>

```

```
        <input type="number" id="age" name="age" placeholder="Must be  
18 or older">
```

```
        <span class="error-message" id="ageError"></span>
```

```
</div>
```

```
<!-- Terms and Conditions -->
```

```
<div class="form-group checkbox-group">
```

```
    <label>
```

```
        <input type="checkbox" id="terms" name="terms">
```

```
        I agree to the Terms and Conditions <span  
class="required">*</span>
```

```
    </label>
```

```
    <span class="error-message" id="termsError"></span>
```

```
</div>
```

```
<!-- Submit Button -->
```

```
<button type="submit" class="submit-btn">Register</button>
```

```
<!-- Success Message -->
```

```
<div class="success-message" id="successMessage">
```

```
    Registration successful! ✓
```

```
</div>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
<script src="validation.js"></script>
```

```
</body>
```

```
</html>
```

Style.css

```
* {
```

```
  margin: 0;
```

```
  padding: 0;
```

```
  box-sizing: border-box;
```

```
}
```

```
body {
```

```
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
```

```
  background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
```

```
  min-height: 100vh;
```

```
  display: flex;
```

```
  justify-content: center;
```

```
  align-items: center;
```

```
  padding: 20px;
```

```
}
```

```
.container {
```

```
  width: 100%;
```

```
  max-width: 600px;
```

```
}
```

```
.form-wrapper {
```

```
background: white;
border-radius: 10px;
padding: 40px;
box-shadow: 0 10px 40px rgba(0, 0, 0, 0.2);
}
```

```
h1 {
  color: #333;
  margin-bottom: 10px;
  font-size: 28px;
  text-align: center;
}
```

```
.subtitle {
  color: #666;
  text-align: center;
  margin-bottom: 30px;
  font-size: 14px;
}
```

```
.form-group {
  margin-bottom: 20px;
}
```

```
label {
  display: block;
```

```
margin-bottom: 8px;  
color: #333;  
font-weight: 500;  
font-size: 14px;  
}
```

```
.required {  
  color: #e74c3c;  
}
```

```
input[type="text"],  
input[type="email"],  
input[type="tel"],  
input[type="password"],  
input[type="number"] {  
  width: 100%;  
  padding: 12px 15px;  
  border: 2px solid #ddd;  
  border-radius: 5px;  
  font-size: 14px;  
  transition: all 0.3s ease;  
  background-color: #f9f9f9;  
}
```

```
input[type="text"]:focus,  
input[type="email"]:focus,
```

```
input[type="tel"]:focus,  
input[type="password"]:focus,  
input[type="number"]:focus {  
    outline: none;  
    border-color: #667eea;  
    background-color: white;  
}
```

```
input.error {  
    border-color: #e74c3c;  
    background-color: #ffe6e6;  
}
```

```
input.valid {  
    border-color: #2ecc71;  
    background-color: #e8f8f0;  
}
```

```
.error-message {  
    display: block;  
    color: #e74c3c;  
    font-size: 12px;  
    margin-top: 5px;  
    min-height: 18px;  
    font-weight: 500;  
}
```



```
.checkbox-group {  
    display: flex;  
    flex-direction: column;  
}
```

```
.checkbox-group label {  
    display: flex;  
    align-items: center;  
    cursor: pointer;  
}
```

```
.checkbox-group input[type="checkbox"] {  
    width: auto;  
    margin-right: 10px;  
    cursor: pointer;  
    transform: scale(1.2);  
}
```

```
.submit-btn {  
    width: 100%;  
    padding: 14px;  
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);  
    color: white;  
    border: none;  
    border-radius: 5px;
```

```
font-size: 16px;
font-weight: 600;
cursor: pointer;
transition: transform 0.2s ease, box-shadow 0.2s ease;
margin-top: 10px;
}
```

```
.submit-btn:hover {
  transform: translateY(-2px);
  box-shadow: 0 5px 15px rgba(102, 126, 234, 0.4);
}
```

```
.submit-btn:active {
  transform: translateY(0);
}
```

```
.submit-btn:disabled {
  background: #ccc;
  cursor: not-allowed;
  transform: none;
}
```

```
.success-message {
  background-color: #d4edda;
  color: #155724;
  border: 1px solid #c3e6cb;
```

```
border-radius: 5px;
padding: 12px;
margin-top: 20px;
text-align: center;
font-weight: 600;
display: none;
}
```

```
.success-message.show {
    display: block;
    animation: slideIn 0.5s ease;
}
```

```
@keyframes slideIn {
    from {
        opacity: 0;
        transform: translateY(-10px);
    }
    to {
        opacity: 1;
        transform: translateY(0);
    }
}
```

```
/* Password Requirements Styling */
.password-requirements {
```

```
background-color: #f8f9fa;  
border-left: 3px solid #667eea;  
padding: 12px 15px;  
margin-top: 10px;  
border-radius: 5px;  
font-size: 12px;  
}
```

```
.password-requirements p {  
    font-weight: 600;  
    color: #333;  
    margin-bottom: 8px;  
}
```

```
.password-requirements ul {  
    list-style: none;  
    padding-left: 0;  
}
```

```
.password-requirements li {  
    padding: 3px 0;  
    color: #666;  
    position: relative;  
    padding-left: 20px;  
}
```

```
.password-requirements li::before {
    content: "X";
    position: absolute;
    left: 0;
    color: #e74c3c;
    font-weight: bold;
}

.password-requirements li.valid::before {
    content: "✓";
    color: #2ecc71;
}

/* Responsive Design */
@media (max-width: 600px) {
    .form-wrapper {
        padding: 25px;
    }

    h1 {
        font-size: 24px;
    }
}
```

Validation.js

```
// Get form and input elements
const form = document.getElementById('registrationForm');
const fullName = document.getElementById('fullName');
const email = document.getElementById('email');
const phone = document.getElementById('phone');
const password = document.getElementById('password');
const confirmPassword = document.getElementById('confirmPassword');
const age = document.getElementById('age');
const terms = document.getElementById('terms');
const successMessage = document.getElementById('successMessage');
```

```
// Validation patterns
```

```
const patterns = {
  email: /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6}$/ ,
  phone: /^[0-9]{10}$/ ,
  name: /^[a-zA-Z\s]{2,50}$/
};
```

```
// Password validation requirements
```

```
const passwordRequirements = {
  minLength: /.{8,}/ ,
  uppercase: /[A-Z]/ ,
  lowercase: /[a-z]/ ,
  number: /[0-9]/ ,
  special: /[!@#$%^&*() , . ? " : { } | < > ]/
};
```

```
// Real-time validation on input

fullName.addEventListener('input', () => validateField(fullName,
validateName));

email.addEventListener('input', () => validateField(email, validateEmail));
phone.addEventListener('input', () => validateField(phone, validatePhone));
password.addEventListener('input', () => {
    validateField(password, validatePassword);
    updatePasswordRequirements();
    if (confirmPassword.value) {
        validateField(confirmPassword, validateConfirmPassword);
    }
});

confirmPassword.addEventListener('input', () =>
validateField(confirmPassword, validateConfirmPassword));
age.addEventListener('input', () => validateField(age, validateAge));
terms.addEventListener('change', () => validateField(terms, validateTerms));


// Prevent form submission and validate all fields
form.addEventListener('submit', function(e) {
    e.preventDefault();

    // Validate all fields
    const isNameValid = validateField(fullName, validateName);
    const isEmailValid = validateField(email, validateEmail);
    const isPhoneValid = validateField(phone, validatePhone);
    const isPasswordValid = validateField(password, validatePassword);
```

```

    const isConfirmPasswordValid = validateField(confirmPassword,
validateConfirmPassword);

    const isAgeValid = validateField(age, validateAge);

    const isTermsValid = validateField(terms, validateTerms);


// Check if all fields are valid
if (isNameValid && isEmailValid && isPhoneValid && isPasswordValid &&
    isConfirmPasswordValid && isAgeValid && isTermsValid) {
    // Show success message
    successMessage.classList.add('show');


// Reset form after 2 seconds
setTimeout(() => {
    form.reset();
    successMessage.classList.remove('show');
    clearAllValidationStates();
}, 3000);
} else {
    // Scroll to first error
    const firstError = document.querySelector('.error');
    if (firstError) {
        firstError.scrollIntoView({ behavior: 'smooth', block: 'center' });
    }
}
});


// Generic field validation function

```



```
function validateField(field, validationFunction) {  
    const isValid = validationFunction(field);  
    updateFieldUI(field, isValid);  
    return isValid;  
}
```

// Update field UI based on validation

```
function updateFieldUI(field, isValid) {  
    if (field.type === 'checkbox') return;  
  
    if (isValid) {  
        field.classList.remove('error');  
        field.classList.add('valid');  
    } else {  
        field.classList.remove('valid');  
        field.classList.add('error');  
    }  
}
```

// Validation functions

```
function validateName(field) {  
    const errorElement = document.getElementById('fullNameError');  
    const value = field.value.trim();  
  
    if (value === '') {  
        errorElement.textContent = 'Full name is required';  
    }  
}
```

```
        return false;
    } else if (!patterns.name.test(value)) {
        errorElement.textContent = 'Please enter a valid name (2-50 characters, letters only)';
        return false;
    } else {
        errorElement.textContent = "";
        return true;
    }
}
```

```
function validateEmail(field) {
    const errorElement = document.getElementById('emailError');
    const value = field.value.trim();

    if (value === "") {
        errorElement.textContent = 'Email address is required';
        return false;
    } else if (!patterns.email.test(value)) {
        errorElement.textContent = 'Please enter a valid email address';
        return false;
    } else {
        errorElement.textContent = "";
        return true;
    }
}
```

```
function validatePhone(field) {  
    const errorElement = document.getElementById('phoneError');  
    const value = field.value.trim();  
  
    if (value === '') {  
        errorElement.textContent = 'Phone number is required';  
        return false;  
    } else if (!patterns.phone.test(value)) {  
        errorElement.textContent = 'Please enter a valid 10-digit phone number';  
        return false;  
    } else {  
        errorElement.textContent = '';  
        return true;  
    }  
}
```

```
function validatePassword(field) {  
    const errorElement = document.getElementById('passwordError');  
    const value = field.value;  
  
    if (value === '') {  
        errorElement.textContent = 'Password is required';  
        return false;  
    } else if (!passwordRequirements.minLength.test(value)) {  
        errorElement.textContent = 'Password must be at least 8 characters';  
        return false;  
    }  
}
```

```

    } else if (!passwordRequirements.uppercase.test(value)) {
        errorElement.textContent = 'Password must contain at least one uppercase
letter';
        return false;
    } else if (!passwordRequirements.lowercase.test(value)) {
        errorElement.textContent = 'Password must contain at least one lowercase
letter';
        return false;
    } else if (!passwordRequirements.number.test(value)) {
        errorElement.textContent = 'Password must contain at least one number';
        return false;
    } else if (!passwordRequirements.special.test(value)) {
        errorElement.textContent = 'Password must contain at least one special
character';
        return false;
    } else {
        errorElement.textContent = '';
        return true;
    }
}

```

```

function validateConfirmPassword(field) {
    const errorElement = document.getElementById('confirmPasswordError');
    const value = field.value;

    if (value === '') {
        errorElement.textContent = 'Please confirm your password';
    }
}

```

```
        return false;
    } else if (value !== password.value) {
        errorElement.textContent = 'Passwords do not match';
        return false;
    } else {
        errorElement.textContent = "";
        return true;
    }
}
```

```
function validateAge(field) {
    const errorElement = document.getElementById('ageError');
    const value = field.value;

    if (value === "") {
        errorElement.textContent = 'Age is required';
        return false;
    } else if (parseInt(value) < 18) {
        errorElement.textContent = 'You must be at least 18 years old';
        return false;
    } else if (parseInt(value) > 120) {
        errorElement.textContent = 'Please enter a valid age';
        return false;
    } else {
        errorElement.textContent = "";
        return true;
    }
}
```

```
}  
}
```

```
function validateTerms(field) {  
  const errorElement = document.getElementById('termsError');  
  
  if (!field.checked) {  
    errorElement.textContent = 'You must agree to the terms and conditions';  
    return false;  
  } else {  
    errorElement.textContent = '';  
    return true;  
  }  
}
```

```
// Update password requirements checklist  
function updatePasswordRequirements() {  
  const value = password.value;  
  
  updateRequirement('length-check',  
passwordRequirements.minLength.test(value));  
  
  updateRequirement('uppercase-check',  
passwordRequirements.uppercase.test(value));  
  
  updateRequirement('lowercase-check',  
passwordRequirements.lowercase.test(value));  
  
  updateRequirement('number-check',  
passwordRequirements.number.test(value));  
}
```

```
    updateRequirement('special-check',  
passwordRequirements.special.test(value));  
}
```

```
function updateRequirement(elementId, isValid) {  
    const element = document.getElementById(elementId);  
    if (isValid) {  
        element.classList.add('valid');  
    } else {  
        element.classList.remove('valid');  
    }  
}
```

// Clear all validation states

```
function clearAllValidationStates() {  
    const inputs = form.querySelectorAll('input');  
    inputs.forEach(input => {  
        input.classList.remove('error', 'valid');  
    });  
};
```

```
const errorMessages = form.querySelectorAll('.error-message');  
errorMessages.forEach(msg => {  
    msg.textContent = "";  
});
```

// Reset password requirements

```
const requirements = document.querySelectorAll('.password-requirements
li');
requirements.forEach(req => {
  req.classList.remove('valid');
});
}
```