

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width,initial-scale=1" />
<title>Interactive Form Validation — Single File</title>
<style>
:root{
  --bg:#0f1724; --card:#0b1220; --muted:#94a3b8; --accent:#60a5fa; --danger:#fb7185;
  --success:#34d399;
  --glass: rgba(255,255,255,0.03);
  font-family: Inter, ui-sans-serif, system-ui, -apple-system, "Segoe UI", Roboto, "Helvetica
Neue", Arial;
}
html,body{height:100%;}
body{
  margin:0;
  background: linear-gradient(180deg,#071028 0%, #071230 40%, #07162a 100%);
  color:#e6eef8;
  display:flex;
  align-items:center;
  justify-content:center;
  padding:24px;
}

.container{
  width:min(920px,96%);
  background: linear-gradient(180deg, rgba(255,255,255,0.02), rgba(255,255,255,0.01));
  border-radius:14px;
  box-shadow: 0 10px 30px rgba(2,6,23,0.6);
  padding:24px;
  backdrop-filter: blur(6px);
  overflow:hidden;
}

header{
  display:flex; align-items:center; gap:14px; margin-bottom:18px;
}

.logo{
  width:56px; height:56px; border-radius:12px;
background:linear-gradient(135deg,var(--accent),#7c3aed);
  display:flex; align-items:center; justify-content:center; font-weight:700; color:#04263b;
font-size:20px;
  box-shadow: 0 4px 18px rgba(96,165,250,0.12), inset 0 -4px 10px rgba(0,0,0,0.15);
```

```

}
h1{font-size:20px; margin:0;}
p.lead{margin:0; color:var(--muted); font-size:13px;}

form{
  display:grid;
  grid-template-columns: 1fr 320px;
  gap:20px;
  align-items:start;
  margin-top:16px;
}

.card{
  background:var(--card);
  padding:18px; border-radius:10px; box-shadow: 0 6px 24px rgba(2,6,23,0.5);
}

label{display:block; font-size:13px; font-weight:600; margin-bottom:6px;}
.field{
  margin-bottom:14px;
}
input[type="text"], input[type="email"], input[type="password"], input[type="tel"], select,
textarea{
  width:100%;
  padding:10px 12px;
  border-radius:8px;
  border:1px solid rgba(255,255,255,0.06);
  background:var(--glass);
  color:inherit;
  outline:none;
  transition: box-shadow .12s, border-color .12s;
  box-sizing:border-box;
  font-size:14px;
}
input:focus, select:focus, textarea:focus{
  box-shadow: 0 6px 18px rgba(96,165,250,0.08);
  border-color: rgba(96,165,250,0.6);
}

.row { display:flex; gap:10px; }
.hint { font-size:12px; color:var(--muted); margin-top:6px; }
.error { color:var(--danger); font-size:13px; margin-top:6px; display:flex; align-items:center; gap:8px; }

```

```
.ok { color:var(--success); font-size:13px; margin-top:6px; display:flex; align-items:center; gap:8px; }
```

```
.small { font-size:12px; color:var(--muted); }
```

```
.status-dot { width:10px; height:10px; border-radius:50%; display:inline-block; }
```

```
.status-unknown{ background:#94a3b8; }
```

```
.status-taken{ background:var(--danger); }
```

```
.status-free{ background:var(--success); }
```

```
.pw-meter {  
  height:8px; border-radius:6px; background: rgba(255,255,255,0.04); overflow:hidden;  
  margin-top:8px;  
}
```

```
.pw-meter > i{  
  display:block; height:100%; width:0%; background:  
linear-gradient(90deg,#fb7185,#f97316,#60a5fa);  
  transition: width .18s ease;  
}
```

```
.submit-col {  
  display:flex; flex-direction:column; gap:12px;  
}
```

```
.summary {  
  background:linear-gradient(180deg, rgba(255,255,255,0.01), rgba(255,255,255,0.005));  
  padding:14px; border-radius:10px;  
}
```

```
button{  
  padding:10px 14px; border-radius:10px; border:0; cursor:pointer; font-weight:700;  
  background:linear-gradient(90deg,var(--accent),#7c3aed); color:#04263b;  
  box-shadow: 0 8px 26px rgba(96,165,250,0.12);  
  font-size:14px;  
}
```

```
button[aria-disabled="true"]{ opacity:0.5; cursor:not-allowed; transform: none; }
```

```
.note { font-size:12px; color:var(--muted); }
```

```
/* Responsive */
```

```
@media (max-width:880px){  
  form{ grid-template-columns: 1fr; }  
}
```

```
/* accessible focus outlines */
```

```
.focus-outline:focus { outline: 3px solid rgba(96,165,250,0.18); outline-offset:3px;
border-radius:8px; }
```

```
/* tooltips small */
.tooltip {
  display:inline-block; position:relative;
}
.tooltip .tip {
  position:absolute; left:50%; transform:translateX(-50%); bottom:calc(100% + 8px);
  background:#071330; color:var(--muted); padding:8px 10px; border-radius:8px;
font-size:12px; white-space:nowrap;
  box-shadow:0 6px 20px rgba(2,6,23,0.6); display:none;
}
.tooltip:hover .tip, .tooltip:focus-within .tip { display:block; }
</style>
</head>
<body>
  <div class="container" role="main">
    <header>
      <div class="logo" aria-hidden="true">IV</div>
      <div>
        <h1>Interactive Form Validation</h1>
        <p class="lead">Real-time validation, accessible UI, password strength and username
availability (simulated).</p>
      </div>
    </header>

    <form id="regForm" novalidate>
      <!-- LEFT: Form fields -->
      <div class="card" aria-labelledby="formTitle">
        <h2 id="formTitle" style="margin-top:0;margin-bottom:8px;font-size:16px">Create
account</h2>

        <div class="field">
          <label for="fullName">Full name <span class="small" aria-hidden>·
required</span></label>
          <input id="fullName" name="fullName" type="text" required minlength="3"
placeholder="Your full name" aria-describedby="fullNameHelp" class="focus-outline" />
          <div id="fullNameHelp" class="hint">Enter at least 3 characters.</div>
          <div id="fullNameError" class="error" aria-live="polite" hidden></div>
        </div>

        <div class="field">
          <label for="username">Username
```

```

    <span class="tooltip small" tabindex="0" aria-hidden="false">
      ①
      <span class="tip">Username may contain letters, numbers, underscores. 3–16
characters.</span>
    </span>
  </label>
  <div style="display:flex;gap:10px;align-items:center;">
    <input id="username" name="username" type="text" required
pattern="^[a-zA-Z0-9_]{3,16}$" placeholder="username" aria-describedby="usernameHelp"
class="focus-outline" />
    <span id="usernameStatus" aria-hidden="true" title="username status"
style="display:inline-flex;align-items:center;gap:8px;">
      <span id="usernameDot" class="status-dot status-unknown"
aria-hidden="true"></span>
      <span id="usernameText" class="small" style="color:var(--muted)">checking...</span>
    </div>
    <div id="usernameHelp" class="hint">No spaces. 3–16 chars. We'll check
availability.</div>
    <div id="usernameError" class="error" aria-live="polite" hidden></div>
  </div>

  <div class="row">
    <div class="field" style="flex:1;">
      <label for="email">Email</label>
      <input id="email" name="email" type="email" placeholder="you@example.com" required
aria-describedby="emailHelp" class="focus-outline" />
      <div id="emailHelp" class="hint">We'll send a confirmation email.</div>
      <div id="emailError" class="error" aria-live="polite" hidden></div>
    </div>

    <div class="field" style="width:150px;">
      <label for="country">Country</label>
      <select id="country" name="country" class="focus-outline"
aria-describedby="countryHelp" required>
        <option value="">Choose</option>
        <option value="IN">India</option>
        <option value="US">United States</option>
        <option value="GB">United Kingdom</option>
        <option value="CA">Canada</option>
      </select>
      <div id="countryHelp" class="hint">Select your country.</div>
      <div id="countryError" class="error" aria-live="polite" hidden></div>
    </div>

```

</div>

```
<div class="field">
  <label for="phone">Phone</label>
  <input id="phone" name="phone" type="tel" placeholder="+91 98765 43210"
class="focus-outline" aria-describedby="phoneHelp" />
  <div id="phoneHelp" class="hint">Optional — international format preferred (e.g. +91
9876543210)</div>
  <div id="phoneError" class="error" aria-live="polite" hidden></div>
</div>
```

```
<div class="field">
  <label for="password">Password</label>
  <input id="password" name="password" type="password" placeholder="Create a strong
password" aria-describedby="passwordHelp" required class="focus-outline" />
  <div id="passwordHelp" class="hint">Min 8 chars, include uppercase, number &
symbol.</div>
```

```
<div class="pw-meter" aria-hidden="true">
  <i id="pwBar"></i>
</div>
```

```
<div id="pwCriteria" class="small" style="margin-top:8px; display:grid; gap:6px;">
  <div id="cLen" class="small">• <span class="small" style="margin-left:6px">8+
characters</span></div>
  <div id="cUpper" class="small">• <span style="margin-left:6px">Uppercase
letter</span></div>
  <div id="cNumber" class="small">• <span style="margin-left:6px">Number</span></div>
  <div id="cSymbol" class="small">• <span style="margin-left:6px">Symbol (e.g.
!@#)</span></div>
</div>
```

```
<div id="passwordError" class="error" aria-live="polite" hidden></div>
</div>
```

```
<div class="field">
  <label for="confirm">Confirm password</label>
  <input id="confirm" name="confirm" type="password" placeholder="Repeat password"
required class="focus-outline" aria-describedby="confirmHelp" />
  <div id="confirmHelp" class="hint">Must match the password.</div>
  <div id="confirmError" class="error" aria-live="polite" hidden></div>
</div>
```

```
<div class="field">
```

```

        <label for="bio">Short bio <span class="small">(optional)</span></label>
        <textarea id="bio" name="bio" rows="3" placeholder="Tell us in a sentence..."
class="focus-outline"></textarea>
        <div class="hint">Max 200 characters.</div>
    </div>

<!-- RIGHT: summary and submit -->
<aside class="submit-col">
    <div class="card summary" aria-labelledby="summaryTitle">
        <h3 id="summaryTitle" style="margin:0 0 8px 0;font-size:15px">Summary</h3>
        <div class="small note">Live validation status and final submission preview.</div>

        <ul style="margin-top:12px; padding-left:18px; color:var(--muted); font-size:14px;
line-height:1.6;">
            <li id="sName">Name: —</li>
            <li id="sUser">Username: —</li>
            <li id="sEmail">Email: —</li>
            <li id="sCountry">Country: —</li>
        </ul>

        <div style="margin-top:12px;">
            <button id="submitBtn" type="submit" aria-disabled="true">Create account</button>
        </div>
        <div class="hint" style="margin-top:8px">All required fields must be valid to submit.</div>
    </div>

    <div class="card" style="text-align:left;">
        <div style="display:flex; justify-content:space-between; align-items:center;">
            <div>
                <div class="small" style="color:var(--muted)">Password strength</div>
                <div id="pwStrengthText" class="ok" style="display:inline-block;color:var(--muted);
margin-top:8px;">—</div>
            </div>
            <div style="text-align:right">
                <div class="small" style="color:var(--muted)">Required</div>
                <div style="font-weight:700; margin-top:6px">Name · Username · Email ·
Password</div>
            </div>
        </div>
    </div>

    <div class="card" id="finalCard" style="display:none;">

```

```
        <h4 style="margin:0 0 8px 0">Submission result</h4>
        <div id="finalMessage" class="small"></div>
    </div>
</aside>
</form>
</div>
```

```
<script>
```

```
/*
```

```
    Interactive form validation script
```

- Realtime checks for inputs
- Simulated username availability (with debounce)
- Password strength meter and criteria
- Inline accessible error messages
- Prevent submission while invalid

```
*/
```

```
(function(){
```

```
    // Elements
```

```
    const form = document.getElementById('regForm');
    const fullname = document.getElementById('fullname');
    const username = document.getElementById('username');
    const usernameDot = document.getElementById('usernameDot');
    const usernameText = document.getElementById('usernameText');
    const email = document.getElementById('email');
    const country = document.getElementById('country');
    const phone = document.getElementById('phone');
    const password = document.getElementById('password');
    const confirm = document.getElementById('confirm');
```

```
    const pwBar = document.getElementById('pwBar');
    const pwStrengthText = document.getElementById('pwStrengthText');
```

```
    const submitBtn = document.getElementById('submitBtn');
```

```
    // Errors
```

```
    const errs = {
        fullname: document.getElementById('fullnameError'),
        username: document.getElementById('usernameError'),
        email: document.getElementById('emailError'),
        country: document.getElementById('countryError'),
        phone: document.getElementById('phoneError'),
        password: document.getElementById('passwordError'),
        confirm: document.getElementById('confirmError')
```



```

};

// Summary fields
const sName = document.getElementById('sName');
const sUser = document.getElementById('sUser');
const sEmail = document.getElementById('sEmail');
const sCountry = document.getElementById('sCountry');

// Simulated taken usernames (in real app => server call)
const takenUsernames = new Set(['admin','user','test','john','santosh','root']);

// Utility: debounce
function debounce(fn, wait){ let t; return (...args)=>{ clearTimeout(t);
t=setTimeout(()=>fn(...args), wait); }; }

// Basic validators
function showError(el, message){
  errs[el].textContent = message;
  errs[el].hidden = false;
  errs[el].setAttribute('aria-hidden', 'false');
}
function clearError(el){
  errs[el].textContent = "";
  errs[el].hidden = true;
  errs[el].setAttribute('aria-hidden', 'true');
}

// Fullname live check
fullname.addEventListener('input', () => {
  const v = fullname.value.trim();
  sName.textContent = 'Name: ' + (v || '—');
  if (!v) { showError('fullname', 'Full name is required.');
```

```

}

const checkUsernameAvailability = debounce(async ()=>{
  const raw = username.value.trim();
  sUser.textContent = 'Username: ' + (raw || '—');

  if (!raw){
    setUsernameStatus('unknown', 'enter a username');
    showError('username', 'Username is required. ');
    touchFormState();
    return;
  }

  const pattern = /^[a-zA-Z0-9_]{3,16}$/;
  if (!pattern.test(raw)){
    setUsernameStatus('unknown', 'invalid format');
    showError('username', 'Use letters, numbers, underscores. 3–16 chars. ');
    touchFormState();
    return;
  }

  // show checking
  setUsernameStatus('unknown', 'checking...');
  clearError('username');

  // Simulated network delay
  await new Promise(r => setTimeout(r, 700));

  // Simulated server check
  if (takenUsernames.has(raw.toLowerCase())){
    setUsernameStatus('taken', 'taken');
    showError('username', 'This username is already taken. ');
  } else {
    setUsernameStatus('free', 'available');
    clearError('username');
  }
  touchFormState();
}, 450);

username.addEventListener('input', () => {
  setUsernameStatus('unknown', 'checking...');
  checkUsernameAvailability();
});

```

```

// Email validation
email.addEventListener('input', () => {
  const v = email.value.trim();
  sEmail.textContent = 'Email: ' + (v || '—');
  if (!v){ showError('email','Email is required.');
```

```

// Set strength text
const map = ['Too weak', 'Weak', 'Fair', 'Good', 'Strong'];
pwStrengthText.textContent = map[score];
// criteria marks
document.getElementById('cLen').style.color = pw.length >= 8 ? 'var(--success)' :
'var(--muted)';
document.getElementById('cUpper').style.color = /[A-Z]/.test(pw) ? 'var(--success)' :
'var(--muted)';
document.getElementById('cNumber').style.color = /[0-9]/.test(pw) ? 'var(--success)' :
'var(--muted)';
document.getElementById('cSymbol').style.color = /^[A-Za-z0-9]/.test(pw) ? 'var(--success)' :
'var(--muted)';
}

password.addEventListener('input', ()=>{
  updatePwUI();
  validatePassword();
  validateConfirm();
  touchFormState();
});

confirm.addEventListener('input', ()=>{
  validateConfirm();
  touchFormState();
});

function validatePassword(){
  const v = password.value;
  if (!v){ showError('password', 'Password is required. '); return false; }
  const reasons = [];
  if (v.length < 8) reasons.push('At least 8 characters');
  if (!/[A-Z]/.test(v)) reasons.push('Include an uppercase letter');
  if (!/[0-9]/.test(v)) reasons.push('Include a number');
  if (!/^[A-Za-z0-9]/.test(v)) reasons.push('Include a symbol (e.g. !@#)');
  if (reasons.length){
    showError('password', 'Weak password — ' + reasons.join(', ') + '. ');
    return false;
  }
  clearError('password');
  return true;
}

function validateConfirm(){
  if (!confirm.value){ showError('confirm', 'Please confirm your password. '); return false; }

```

```
    if (confirm.value !== password.value){ showError('confirm','Passwords do not match.');
```

false; }

```
    clearError('confirm');
    return true;
}
```

```
// Initial UI update
updatePwUI();
setUsernameStatus('unknown','enter a username');
sName.textContent = 'Name: —';
sUser.textContent = 'Username: —';
sEmail.textContent = 'Email: —';
sCountry.textContent = 'Country: —';
```

```
// Form validity aggregator
function isFormValid(){
    // run key validators
    const checks = [];
    checks.push(fullname.value.trim().length >= 3 && errs.fullname.hidden);
    checks.push(username.value.trim().length >= 3 && errs.username.hidden &&
usernameText.textContent === 'available');
    checks.push(email.checkValidity() && errs.email.hidden);
    checks.push(country.value !== "" && errs.country.hidden);
    const pwOk = validatePassword();
    const confOk = validateConfirm();
    checks.push(pwOk && confOk);
    // phone optional: if present must be valid
    if (phone.value.trim()) checks.push(errs.phone.hidden);
    return checks.every(Boolean);
}
```

```
// Keep submit button enabled/disabled based on form
function touchFormState(){
    const ok = isFormValid();
    submitBtn.setAttribute('aria-disabled', ok ? 'false' : 'true');
    submitBtn.disabled = !ok;
}
```

```
// run once after a small delay in case async username check is pending
setInterval(()=>{ touchFormState(); }, 800);
```

```
// Prevent native submission and show final result
form.addEventListener('submit', (ev) => {
    ev.preventDefault();
```

```

// final re-check
if (!isFormValid()){
  // focus first invalid
  const firstErr = Object.keys(errs).find(k => !errs[k].hidden);
  if (firstErr) {
    const el = document.getElementById(firstErr);
    if (el) el.focus();
  }
  return;
}

// show final summary
const finalCard = document.getElementById('finalCard');
const finalMessage = document.getElementById('finalMessage');
finalCard.style.display = 'block';
const payload = {
  name: fullname.value.trim(),
  username: username.value.trim(),
  email: email.value.trim(),
  country: country.value,
  phone: phone.value.trim() || '—',
  bio: document.getElementById('bio').value.trim() || '—'
};

finalMessage.innerHTML = `
<strong style="color:var(--muted)">Account created (simulated)</strong>
<div style="margin-top:8px">
  <div><strong>Name:</strong> ${escapeHtml(payload.name)}</div>
  <div><strong>Username:</strong> ${escapeHtml(payload.username)}</div>
  <div><strong>Email:</strong> ${escapeHtml(payload.email)}</div>
  <div><strong>Country:</strong> ${escapeHtml(payload.country)}</div>
  <div><strong>Phone:</strong> ${escapeHtml(payload.phone)}</div>
  <div style="margin-top:8px" class="hint">This is a demo — in a real application the form
would be submitted to the server.</div>
</div>
`;

// show success visuals
submitBtn.textContent = 'Done';
submitBtn.disabled = true;
submitBtn.setAttribute('aria-disabled', 'true');

// Mask password fields (security)
password.value = "";
confirm.value = "";

```

```

    updatePwUI();
    clearError('password'); clearError('confirm');
  });

  // On page load, set submit disabled
  touchFormState();

  // Utility: simple HTML escape
  function escapeHtml(s){
    return String(s).replace(/&<>"/g, function(m){ return
    ({'&':'&amp;','<':'&lt;','>':'&gt;','"':'&quot;','"':'&#39;'}[m]); });
  }

  // Accessibility: show inline errors when inputs are blurred if invalid
  [
    {el:fullname, fn: ()=> {
      if (!fullname.value.trim()) showError('fullname','Full name is required.');
```

else if (fullname.value.trim().length < 3) showError('fullname','Please enter at least 3 characters.');

else clearError('fullname');

```
    }},
    {el:username, fn: ()=> {
      if (!username.value.trim()) showError('username','Username is required.');
```

else if (!/^[a-zA-Z0-9\_]{3,16}\$/.test(username.value.trim())) showError('username','Use letters, numbers, underscores. 3–16 chars.');

else if (usernameText.textContent === 'taken') showError('username','This username is already taken.');

else clearError('username');

```
    }},
    {el:email, fn: ()=> {
      if (!email.value.trim()) showError('email','Email is required.');
```

else if (!email.checkValidity()) showError('email','Please enter a valid email address.');

else clearError('email');

```
    }},
    {el:country, fn: ()=> {
      if (!country.value) showError('country','Select a country.');
```

else clearError('country');

```
    }},
    {el:password, fn: ()=> validatePassword()},
    {el:confirm, fn: ()=> validateConfirm()}
  ].forEach(item => item.el.addEventListener('blur', item.fn));

  // allow submit by pressing Enter in any input (respecting validation)
  form.addEventListener('keydown', (e) => {
```

```
if (e.key === 'Enter'){
  // if focused on textarea => do nothing
  if (document.activeElement && document.activeElement.tagName === 'TEXTAREA') return;
  // otherwise try to submit
  e.preventDefault();
  // try submit programmatically (will run form submit handler)
  form.requestSubmit();
}
});

// Small UX: show/hide password with long-press (optional)
// (Not implemented here to keep single-file simple — could be added.)

})();
</script>
</body>
</html>
```