**COURSE APPLICATION**

**CODE:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<title>Course Application</title>

<style>

body { font-family: system-ui, -apple-system, Segoe UI, Roboto, Arial, sans-serif; margin: 24px; }

.card { max-width: 720px; margin: 0 auto; padding: 20px; border: 1px solid #ddd; border-radius: 12px; }

.row { display: grid; grid-template-columns: 1fr 1fr; gap: 16px; }

.field { display: flex; flex-direction: column; margin-bottom: 14px; }

label { font-weight: 600; margin-bottom: 6px; }

input, select, textarea, button {

padding: 10px; border: 1px solid #ccc; border-radius: 8px; font-size: 14px;

}

textarea { resize: vertical; min-height: 80px; }

.actions { display: flex; gap: 10px; margin-top: 10px; }

.btn-primary { background: #1a73e8; color: #fff; border: none; cursor: pointer; }

.btn-secondary { background: #f1f3f4; color: #111; border: 1px solid #e0e0e0; cursor: pointer; }

.success { display: none; padding: 12px; border-radius: 10px; background: #e6f4ea; border: 1px solid #a8dab5; color: #0f5132; margin-bottom: 16px; }

.error { border-color: #e53935 !important; box-shadow: 0 0 0 3px rgba(229,57,53,0.12); }

.note { color: #666; font-size: 12px; margin-top: 4px; }

@media (max-width: 640px) { .row { grid-template-columns: 1fr; } }

</style>

</head>

<body>

<div class="card">

<h1>Course Application</h1>

<div id="ok" class="success" role="status" aria-live="polite">✅ Application submitted successfully.</div>

<form id="appForm" novalidate>

<div class="row">

<div class="field">

<label for="fullName">Full Name \*</label>

<input id="fullName" name="fullName" type="text" autocomplete="name" />

</div>

<div class="field">

<label for="email">Email \*</label>

<input id="email" name="email" type="email" autocomplete="email" />

<div class="note">Use a valid email format, e.g. name@example.com</div>

</div>

</div>

<div class="row">

<div class="field">

<label for="phone">Phone \*</label>

<input id="phone" name="phone" type="tel" inputmode="tel" placeholder="+91 98765 43210" />

<div class="note">Digits only or with country code</div>

</div>

<div class="field">

<label for="course">Select Course \*</label>

<select id="course" name="course">

<option value="">-- Choose a course --</option>

<option>Data Science Fundamentals</option>

<option>Web Development Bootcamp</option>

<option>Android App Development</option>

<option>Cybersecurity Basics</option>

<option>Machine Learning with Python</option>

</select>

</div>

</div>

<div class="row">

<div class="field">

<label for="startDate">Preferred Start Date \*</label>

<input id="startDate" name="startDate" type="date" />

</div>

<div class="field">

<label for="mode">Mode \*</label>

<select id="mode" name="mode">

<option value="">-- Select mode --</option>

<option>Online</option>

<option>On-campus</option>

<option>Hybrid</option>

</select>

</div>

</div>

<div class="field">

<label for="address">Address \*</label>

<textarea id="address" name="address" placeholder="House/Street, City, State, Postal Code"></textarea>

</div>

<div class="field">

<label for="notes">Notes (optional)</label>

<textarea id="notes" name="notes" placeholder="Any special requirements or questions?"></textarea>

</div>

<div class="actions">

<button type="submit" class="btn-primary">Submit Application</button>

<button type="reset" class="btn-secondary">Reset</button>

</div>

</form>

</div>

<script>

(function () {

const form = document.getElementById('appForm');

const ok = document.getElementById('ok');

const requiredFields = [

{ id: 'fullName', label: 'Full Name', test: v => v.trim().length > 0 },

{ id: 'email', label: 'Email', test: v => /^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(v) },

{ id: 'phone', label: 'Phone', test: v => /^[+()\-.\s]\*\d[\d()+\-. \s]{6,}$/.test(v.trim()) },

{ id: 'course', label: 'Course', test: v => v.trim().length > 0 },

{ id: 'startDate', label: 'Preferred Start Date', test: v => v.trim().length > 0 },

{ id: 'mode', label: 'Mode', test: v => v.trim().length > 0 },

{ id: 'address', label: 'Address', test: v => v.trim().length > 0 }

];

function clearErrors() {

ok.style.display = 'none';

requiredFields.forEach(f => document.getElementById(f.id).classList.remove('error'));

}

function validate() {

clearErrors();

const missing = [];

requiredFields.forEach(f => {

const el = document.getElementById(f.id);

const valid = f.test(el.value || '');

if (!valid) {

el.classList.add('error');

missing.push(f.label);

}

});

if (missing.length) {

alert("⚠️ Please correct the following:\n\n- " + missing.join("\n- "));

const firstInvalid = requiredFields.find(f => document.getElementById(f.id).classList.contains('error'));

if (firstInvalid) document.getElementById(firstInvalid.id).focus();

return false;

}

return true;

}

form.addEventListener('submit', function (e) {

e.preventDefault();

if (validate()) {

ok.style.display = 'block';

alert("✅ Application submitted successfully!");

}

});

form.querySelectorAll('input, select, textarea').forEach(el => {

el.addEventListener('input', () => el.classList.remove('error'));

});

form.addEventListener('reset', () => setTimeout(clearErrors, 0));

})();

</script>

</body>

</html>

**OUTPUT:**

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.