

5-Day Practical Bootcamp: 50 HTML & CSS Application Problems

Day 2 — Forms, Tables, and Accessibility by Design

Learning goals: - Craft accessible forms with proper labels, grouping, and validation states (CSS only). - Design readable data tables with captions, scoping, and responsive approaches.

Very Detailed Problem Statements (11–20)

11. Course Registration Form

- Build a form for students to register for a course.
- Must include: Full Name (text input), Email (email input), Phone Number (tel input), Course Selection (select dropdown with at least 3 options), and a Submit button.
- Each field must have an associated `<label>` with `for` attribute.
- Required fields must use the `required` attribute, and invalid fields must be styled with a red border when the `:invalid` selector applies.
- Provide help text below fields using `<small id="help-text">` and reference it with `aria-describedby`.
- Group course-related inputs inside a `<fieldset>` with `<legend>`.

12. Job Application Form

- Create a multi-section form for job applications.
- Section 1: Personal Info (Name, Email, Phone).
- Section 2: Job Details (radio buttons for Position applying for, checkboxes for Skills).
- Section 3: File upload field for resume (accept only `.pdf` and `.docx`).
- Use `<fieldset>` and `<legend>` for each section.
- Custom style radio buttons and checkboxes using `appearance: none` and CSS pseudo-elements.
- Focus ring must remain visible for keyboard users.
- Submit button should be disabled until all required fields are filled (simulate by default state and instructor checking manually).

13. Event RSVP with Seating Preferences

- Build an RSVP form for an event.
- Fields: Name, Email, Number of Guests (numeric input), Seating Preference (`<select>` with `<optgroup>` for Indoor/Outdoor tables).

- Include one disabled option at the top of the dropdown as placeholder (e.g., “Select a seating preference”).
- Style the placeholder differently (light gray text).
- Add radio buttons for meal type (Veg/Non-Veg/Vegan) with custom styling.

14. **Newsletter Subscription Form**

- Inline form placed inside a hero section.
- Input fields: Email (required) + Subscribe button.
- Show two mockup states: success (green background with confirmation text) and error (red border around input, error message below field).
- Style submit button as a prominent `.btn--primary` with hover and focus states.

15. **Feedback Survey**

- Create a survey form with a Likert scale table.
- Question prompts listed in rows, response scale (Strongly Disagree → Strongly Agree) listed as columns.
- Implement using `<table>` with `<th scope="col">` for scale and `<th scope="row">` for questions.
- Inputs must be radio buttons inside table cells, grouped by question.
- Ensure focus states are visible when navigating with keyboard.

16. **Pricing Comparison Table**


- Build a pricing table with at least 3 columns (Basic, Pro, Enterprise).
- Each row represents a feature (e.g., Storage, Support, Users).
- Use checkmark (✓) and cross (✗) symbols with visually hidden text (`Included`).
- Use `<caption>` describing table purpose.
- Highlight the recommended plan column with a different background color.

17. **Product Inventory Table**

- Create a product inventory table with columns: Product Name, Category, Price, Quantity.
- Header row should be sticky using `position: sticky; top: 0;`.
- Numeric columns (Price, Quantity) must be right-aligned.
- Rows should have hover states (background color change).
- Table must be responsive: collapse into stacked cards on screens `<600px`.

18. **Restaurant Menu (Semantically Correct)**

- Build a restaurant menu using lists.
- Categories (e.g., Starters, Mains, Desserts) must be `<h2>` headings.

- Each dish should use a definition list `<d1>` with `<dt>` (dish name) and `<dd>` (description + price).
- Add allergy information using small icons (e.g., ) with visually hidden text for screen readers.
- Sections separated with horizontal rules `<hr>`.

19. Library Search Form

- Create a form to search library books.
- Fields: Keyword Search (text), Author Name (text), Category (select dropdown), Availability (checkbox: Available Only).
- Include a search button with a magnifying glass icon inside it. Use visually hidden text for accessibility.
- Ensure all form controls can be navigated by keyboard.
- Place form fields inside a `<form role="search">` element.

20. Contact Us + Map Placeholder

- Contact form with fields: Name, Email, Subject (text), Message (textarea).
- Two-column layout: form on the left, map placeholder `<div class="map-box">` on the right.
- Style map box with fixed aspect ratio (16:9) and border.
- On mobile screens (`<768px`), stack form above map.
- Add client-side required attributes for Name, Email, and Message fields.
- Provide CSS for error states with red borders and small error messages below inputs.

Would you like me to now expand **Day 3 (Responsive Design & Grid Mastery)** with the same very detailed specifications for each exercise?

Day 2 — Forms, Tables, and Accessibility by Design

Learning goals: - Craft accessible forms (labels, field groups, help/error text, keyboard order). - Use native controls correctly (type, required, pattern, min/max, placeholders sparingly). - Build readable, responsive tables with captions, scoping, sticky headers, and stacking strategies.

Mini- demos (15 min): Label/for + id → fieldset/legend → help text via aria-describedby → custom focus states → table scoping + captions → mobile stacking patterns.

Very Detailed Problem Statements (11–20)

11. Course Registration Form

- Files: index.html, styles.css.

- Structure: `<main>` contains one `<form>` wrapped by `<section>` with `<h1>` “Course Registration”.
- Required fields (with required): Full Name (`<input type="text">`), Email (`<input type="email">`), Phone (`<input type="tel" pattern="[0-9]{10}">`), Course Selection (`<select>` with at least 4 options), Preferred Start Date (`<input type="date">`).
- Group schedule preferences in a `<fieldset>` with `<legend>` “Mode of Study”; provide radios: Online, On- campus, Hybrid (exactly one must be selectable).
- Add a comments `<textarea>` (optional, `maxlength="300"`).
- Every control must have a `<label for>` with matching id.
- Provide per- field help text (`<small id="...-help">...</small>`) and associate via `aria-describedby`.
- Error/success styles: form supports state classes on the `<form>`: `.is-error` (show inline `.error` messages under the field) and `.is-success` (show a success banner above the form). Implement CSS selectors; logic not required.
- Submit button: `.btn .btn--primary`. Disabled look when `[disabled]` present.
- Layout: 2- column form on $\geq 1024\text{px}$ using CSS Grid (label+control stack inside each cell), 1 column below 1024px .
- Keyboard: visible `:focus-visible` outline on all focusable elements.

12. Job Application Form

- Header `<h1>` “Job Application – Frontend Developer”.
- Sections: Applicant Info, Experience, Links, Consent.
- Applicant Info: First/Last name, Email (`type=email`), Portfolio URL (`type=url`), Location (free text), Relocation (radio Yes/No).
- Experience: Years of Experience (`<input type="number" min="0" max="40">`), Skills (checkboxes: HTML, CSS, JS, Accessibility, Performance).
- Links: GitHub URL, LinkedIn URL (both `type=url`, optional). Use helper text to suggest formats.
- Consent: one required checkbox “I agree to the privacy policy”.
- Use `<fieldset>` for each section with descriptive `<legend>`.
- Attach a resume placeholder using `<input type="file" accept=".pdf, .doc, .docx">` (optional). Add note on max 5MB (text only).
- Error state mock: add `.field--error` class on wrapper to style border and show `<p class="error" role="alert">` with message.
- Submit + Reset buttons; Reset has `.btn--ghost` style.
- Responsive: 2- col grid for Applicant Info on $\geq 768\text{px}$; others full width.

13. Event RSVP with Seating Preferences

- Title `<h1>` “RSVP – Annual Gala”.
- Party size (`<input type="number" min="1" max="8" required>`), Meal preference (select: Veg, Non- veg, Vegan), Seating zone (select with `<optgroup>`: “Front”, “Middle”, “Back”; at least 6 concrete options total).

- Accessibility: associate help text for party size (e.g., “Max 8 per RSVP”).
- Special needs textarea (optional).
- Contact fields: Email (required), Phone (optional pattern="[0-9]{10}").
- Confirm attendance radio (Yes/No) required.
- Include a non- functional map placeholder `<div class="map" aria-hidden="true">` styled to 16:9.
- Layout: card- style form with max- width 720px centered; spacing scale uses CSS variables.
- Buttons: “Save RSVP” primary, “Clear” secondary.

14. **Newsletter Subscription (Inline Hero Form)**

- Build a hero section with heading `<h1>`, subheading `<p>`, and inline form (email + subscribe button) placed on one row on ≥ 640 px; stacked on smaller.
- Email field must be `type=email` required with placeholder “you@example.com”.
- Provide success and error banners (non- functional): `.banner--success` and `.banner--error` above the hero, hidden by default.
- Add a privacy note link under the form using `<small>`.
- Ensure the form is announced correctly to screen readers: `<form aria-labelledby="hero-title">` where `id="hero-title"` is on the `<h1>`.
- Implement a subtle background image using `` in markup within the hero (not CSS background) and hide in print.

15. **Feedback Survey with Likert Scale**

- Build a survey with 4 statements (rows) and a 5- point Likert scale (columns: Strongly Disagree → Strongly Agree).
- Use an HTML `<table>` with `<caption>Course Feedback</caption>`.
- Top header row uses `<th scope="col">`; leftmost statement cells use `<th scope="row">`.
- Each cell contains one radio input; radios for a given row share the same name attribute.
- Ensure labels are clickable: wrap radio + label text in `<label>` or use `for`.
- Provide an open- ended comments textarea below the table.
- Submit button centered below; table horizontally scrolls on small screens (wrap in `.table-responsive` with `overflow-x:auto`).

16. **Pricing Comparison Table**

- Build a comparison between 3 plans: Basic, Pro, Enterprise.
- Use `<table>` with `<caption>Pricing Plans</caption>`.
- The first column is feature names (`<th scope="row">`), the header row has plan names (`<th scope="col">`).
- Include at least 8 features (e.g., Storage, Support SLA, Users, Custom Domain...).
- Use `✓/—` indicators with visually hidden text (e.g., `Included`).

- Highlight “Pro” column with `.is-featured` style and add `aria-label="Recommended plan"` on that header cell.
- Zebra striping rows; sticky header on vertical scroll.
- On <640px, implement a stacked layout: hide the table with `.visually-hidden` for mobile users and show three `.plan-card` blocks rendering the same info (duplicate in markup allowed for this exercise).

17. Product Inventory Table

- Build a data table of at least 10 products with columns: SKU, Name, Category, Qty, Price, Status.
- Table must have `<caption>Inventory - September</caption>`.
- Numeric columns (Qty, Price) right-aligned; use monospace for Price.
- Add row hover highlight and focus- within highlight when a row contains a focused link.
- Sticky header row and sticky first column (SKU) using `position: sticky`.
- Implement a simple filter row above the table (non- functional) with input/select controls.
- On narrow screens, stack each row into a definition- list style block (each cell preceded by its header label using `data-label` CSS technique).

18. Restaurant Menu (Semantic Lists)

- Build a semantic menu with categories (Starters, Mains, Desserts, Drinks).
- Use `<section>` per category with `<h2>` and a `<dl>` for items: `<dt>` item name + `<dd>` description/price.
- Add dietary icons (V, GF, DF) next to items using `<abbr>` with `title` attributes; include visually hidden text for screen readers.
- Include an allergy advisory blockquote or callout `<aside>`.
- Layout: two columns on ≥992px, one column below.
- Add a small sticky category nav on the left (anchors to sections) with `aria-label="Menu categories"` and a visible focus state.

19. Library Search Form

- Provide a search interface with: Query (`type=search` required), Media Type (select: Book, Journal, eBook), Year Range (two `type=number` fields with `min=1900` `max=2100`), Availability (checkbox: Available Now).
- Place a visually hidden `<h1>` for screen readers and a visible `<legend>` inside `<fieldset>` grouping filters.
- Add a clear button (resets inputs) and a submit button.
- Include an icon button for “Advanced Tips” (non- functional) that has assistive text via `aria-label`.
- Responsive: filters in a 3- column grid on desktop, stacking to 1 column on mobile.
- Provide helper text under Year Range explaining acceptable values, linked via `aria-describedby`.

20. Contact Us + Map Placeholder

- Two- column layout on $\geq 768\text{px}$: left column is the contact form; right column is a map placeholder.
 - Form fields: Name (required), Email (required), Subject (select with 4 topics), Message (required textarea with `minlength=30`), Consent checkbox (required) for data processing notice.
 - Each field wrapped in `.field` container with label, control, and `<small class="help">` block (connected via `aria-describedby`).
 - Show a non- functional success message panel `.notice.notice--success` above the form (hidden by default).
 - Map placeholder is a `<figure>` with a grey box maintaining 16:9 via aspect- ratio, with `<figcaption>` describing the location.
 - Print styles: hide the map column and keep only form content.
-