

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

Day 3 — Responsive Design, Grid Mastery, Utilities & Theming

Learning goals: - Build responsive layouts with Grid and Flexbox using auto-fit/auto-fill and minmax(). - Apply fluid typography and spacing via CSS clamp(). - Create utility classes and theme with CSS variables.

Very Detailed Problem Statements (21–30)

21. E- Commerce Product Grid

- Create a page with heading `<h1>Shop Products`.
- Use CSS Grid with `grid-template-columns: repeat(auto-fit, minmax(250px, 1fr))`.
- Each product card: `<article>` with `` (product), `<h2>` title, `<p>` price, `<p>` rating (use ★ characters), and “Add to Cart” button.
- Cards must have consistent padding and border radius.
- Prices styled with CSS variable `--price-color`.
- Typography scale must use `clamp()` for product titles.
- On hover/focus, card lifts with shadow and button changes background color.

22. Magazine- Style Home

- Layout: Complex Grid with template areas: hero, main articles, side column, footer.
- Hero spans full width: `<section>` with large `<h1>` and background image ``.
- Main articles arranged in 2- column grid, each `<article>` with image + headline.
- Side column `<aside>` includes trending list with numbered CSS counters.
- Include at least one pull- quote styled with larger font and border accent.
- Typography scale must adjust at breakpoints (desktop vs mobile).

23. SaaS Pricing Page

- Build 3–4 pricing cards inside a `<section>` with heading `<h1>` “Plans & Pricing”.
- Grid layout: 3 columns on $\geq 1024\text{px}$, 1 column on mobile.
- Each card: plan name, price/month, list of features with checkmarks, signup button.
- Highlight “Pro” card with `.featured` modifier: border, background color, subtle scale.
- Theme: Use CSS custom properties for colors; provide `.theme-dark` class on `<body>` that changes background, text, and card colors.
- Test with forced dark theme to ensure readability.

24. Photo Gallery with Masonry- like Grid

- Gallery `<section>` with heading `<h1>` “Photo Gallery”.

- Use CSS Grid with `grid-auto-rows` technique to mimic masonry. Images should vary in height.
- Each item is `<figure>` with `` and `<figcaption>`.
- On hover/focus, show semi-transparent overlay caption.
- Responsive: 4 columns desktop, 2 tablet, 1 mobile.
- Ensure images fill width with `object-fit: cover`.

25. Dashboard (Static)

- Create a layout with sidebar `<nav>` and main content `<main>`.
- Sidebar: list of links with icons (use placeholders like Unicode symbols).
- Main content: top row of KPI tiles (cards showing metrics like “Users: 1200”).
- Use CSS Grid for KPI row (4 columns desktop, 2 tablet, 1 mobile).
- Below KPIs: 2 larger cards side by side (e.g., Sales, Activity). Equal heights enforced.
- Use utility classes `.text-muted`, `.text-bold`, `.flex-center`.
- Sidebar collapses to top nav on `<768px`.

26. Recipe Collection

- Page titled “Our Recipes”.
- Grid of recipe cards: each has image, name, cooking time, tags (as pill-shaped spans).
- Tags must have consistent style and accessible color contrast.
- Add a print-friendly stylesheet: hide images, display only text.
- On hover, show subtle zoom of recipe image.
- Responsive: 3 columns desktop, 2 tablet, 1 mobile.

27. Travel Itinerary Planner (Static)

- Create itinerary with days as columns in CSS Grid (auto-fit minmax 200px).
- Each day `<section>` has heading “Day 1”, “Day 2”... and list of activities.
- Date column (leftmost) is sticky using `position: sticky`.
- Activities styled as timeline: time + description in horizontal row.
- Add highlight bar for “current day” using `.is-today` class.

28. Edu Course Catalog

- Create a searchable course catalog layout.
- Top bar: search input + filter chips (All, Undergrad, Grad, Online). Chips styled toggle buttons with `.active` state.
- Course list: grid of 3 columns desktop, 2 tablet, 1 mobile. Each card: title, code, credits, short description.
- Each card must have equal height and footer link “View Details”.
- Responsive typography using `clamp()` for course titles.

29. Team Directory

- Grid of team members with avatars (circular images).
- Each card: ``, `<h3>` name, `<p>` job title.

- Hover effect reveals additional text (e.g., email, phone) with transition.
- Use CSS focus-visible to ensure keyboard users see outline.
- Responsive: 4 columns desktop, 2 tablet, 1 mobile.
- Provide alt text for all images.

30. Landing Page with Split- Screen

- Layout: hero section split 50/50. Left side text (headline, subheading, CTA), right side image.
- Implement using CSS Grid with two equal columns.
- On screens <768px, stack vertically with image first.
- Ensure CTA button has accessible color contrast.
- Overlap headline slightly on image with safe padding.
- Provide utility class `.btn--outline` for secondary CTA.

Day 3 — Responsive Design, Grid Mastery, Utilities & Theming

Learning goals: - Build fluid, breakpoint- based layouts using Grid/Flex + `minmax()`, `auto-fit/auto-fill`, `clamp()`. - Establish a tokenized design system with CSS variables and small utility classes. - Deliver robust responsive behavior (mobile- first) with accessible, keyboard- friendly patterns.

Mini- demos (15 min): Container queries vs media queries → Grid template areas → `clamp()` for type scale → Utility classes (`.stack-*`, `.grid-*`, `.sr-only`) → Theming via `:root` and `[data-theme]`.

Very Detailed Problem Statements (21–30)

21.E- Commerce Product Grid

- Files: `index.html`, `styles.css`.
- Design tokens: define in `:root` → `--space-1..4`, `--radius`, `--brand`, `--text`, `--muted`.
- Header: brand logotype (text) at left, simple nav at right (Home, Shop, Cart). Ensure nav collapses into a single row; no hamburger required.
- Product grid: use `<section aria-label="Products">` containing `<ul class="product-grid">` of 12 `` cards.
- Grid: `grid-template-columns: repeat(auto-fit, minmax(220px, 1fr));` with `gap: var(--space-3)`.
- Card content order: image (16:9, `object-fit: cover`), product title (`<h3>`), price, rating (Unicode ★), and an “Add” button.
- Typography: use `clamp()` for card titles (e.g., `clamp(1rem, 1vw + .9rem, 1.25rem)`).
- Accessibility: each image has alt text; button has `aria-label="Add {ProductName} to cart (demo)"`.

- Focus: `.card:focus-within` elevates with shadow; `:focus-visible` outline on buttons/links.
- Responsive: at $\leq 480\text{px}$, reduce gap and ensure two cards per row minimum; single column permitted when viewport $< 260\text{px}$.

22. Magazine- Style Home (Grid Areas)

- Create a layout with named grid areas: hero, lead, secondary, aside, footer.
- Desktop ($\geq 1024\text{px}$):
 - hero full- width top row.
 - lead (main article) large left column; secondary (two stacked stories) to right.
 - aside rightmost narrow column (pull- quotes, tags).
- Tablet ($640\text{--}1023\text{px}$): hero spans full; lead above secondary; aside below as full- width strip.
- Mobile ($< 640\text{px}$): single column stacking in order: hero \rightarrow lead \rightarrow secondary \rightarrow aside \rightarrow footer.
- Implement via `grid-template-areas` with corresponding children using `grid-area`.
- Images use `<figure>+<figcaption>`; pull- quote uses `<blockquote>` with `cite` attribute.
- Use a max content width `.container` (1200px) and center it.

23. SaaS Pricing Page with Theme Toggle (No JS)

- Tokens: define a base theme in `:root` and an alternate high- contrast theme under `[data-theme="contrast"]` changing `--bg`, `--text`, `--brand`.
- Place a theme toggle as a stylesheet- only control: a checkbox in the header; when `#contrast:checked` on a parent `.page`, apply `[data-theme="contrast"]` using the `:has()` selector **or** using an additional `.contrast` class toggled by the checkbox via the sibling/descendant selector trick (CSS- only visual demo; functionality may be simulated by author moving class in HTML).
- Pricing cards (Basic, Pro, Enterprise) in a grid `repeat(auto-fit, minmax(260px, 1fr))`.
- Each card: plan name, price per month, feature list (ul with 6+ items), CTA button.
- Emphasize Pro card: transform lift on hover/focus; add badge “Recommended” using `::before`.
- Respect `prefers-reduced-motion`: disable lift animation when set.
- Accessibility: ensure contrast $\geq 4.5:1$ in both themes; add visually hidden text for price period “per month”.

24. Photo Gallery with Masonry- like Grid

- Use Grid with `grid-auto-rows` technique: compute row height (e.g., 8px) and span images via a utility class `.span-2`, `.span-3` to simulate masonry.
- Provide at least 15 images with varied heights; wrap each in `<figure>` with `<figcaption>`.

- Images must use `loading="lazy"` and `decoding="async"`.
- Add a filter bar UI (non- functional) for tags; ensure keyboard tab order is sensible.
- Breakpoints: 5 columns ($\geq 1280\text{px}$), 3 columns ($\geq 768\text{px}$), 1–2 columns on mobile.

25. **Dashboard (Static, Cards & KPIs)**

- Shell: sidebar `<aside>` (nav list) + main content using CSS Grid (two columns) with sticky sidebar.
- KPI strip at top: 4 cards (Total Sales, Orders, Users, Conversion) in a responsive grid using `auto-fit/minmax(160px, 1fr)`.
- Content area: two panels → “Recent Orders” (table) and “Traffic by Channel” (static list with percentage bars styled via CSS only).
- Require equal card heights within each row using `align-self: stretch` and a `.card` structure.
- Focus management: tabbing through sidebar links shows visible outline; active link uses `[aria-current="page"]` style.
- Reduce motion per preference media query.

26. **Recipe Collection (Cards + Print Styles)**

- Grid of recipe cards (at least 9). Each card: image, name, tags (chips), 2- line description clamped with `line-clamp` or `max-height+overflow`.
- Filters toolbar (non- functional) with select inputs for cuisine and difficulty.
- Print stylesheet must reflow to a single column, hide navigation, and show full recipe descriptions (no clamping) with black text on white background.
- Use `@page` to set margins for print.

27. **Travel Itinerary Planner (Static)**

- Represent a 3- day plan using a responsive grid: day columns across, time- based items stacked within each day.
- Left sticky column shows Day 1/2/3 labels; content area scrolls.
- Use a semantic list for items; each item shows time range, activity title, and location icon (Unicode).
- Mobile: stack days vertically; ensure headings maintain hierarchy.
- Implement timeline guide lines using CSS gradients.

28. **Edu Course Catalog (Filterable UI – Visual Only)**

- Catalog grid of at least 16 courses; each card: course code, title, short blurb, duration, badge for “Beginner/Intermediate/Advanced”.
- Top filter bar: search field (`type=search`), topic select, level chips (buttons with `aria-pressed`).
- Layout uses `repeat(auto-fill, minmax(240px, 1fr))`.
- Cards equal height via grid/flex; CTA “View Syllabus” styled as link- button.
- Accessibility: provide a skip link and ensure labels for all controls.

29. **Team Directory (Avatars + Focus States)**

- Grid of people cards (≥ 20). Card: avatar (1:1, object-fit: cover with rounded corners), name, role, contact icon links (mailto/tel placeholders).
- Hover reveals role using opacity transition; keyboard focus must also reveal (use :focus-within).
- Add a “Departments” filter (non- functional) as a horizontal list with chips.
- Breakpoints: 6/4/2/1 columns from desktop \rightarrow mobile.
- Provide a high- contrast theme variant triggered by adding .theme-contrast to <body> (demonstrate token overrides).

30. Landing Page with Split- Screen

- Above- the- fold split layout: left side text (headline, subhead, two CTAs), right side image or illustration.
 - Implement with Grid: two equal columns; on ≤ 768 px, stack with image first.
 - Content must remain inside a .container with max- width 1200px while background spans full width.
 - Provide safe overlap: a small card overlaps the split (e.g., testimonial) using translate and z-index without breaking flow.
 - Ensure keyboard navigation order remains logical; add skip link to main content.
-