

# Visual Design for the Web — Workbook

This workbook turns the course principles into a build. As you progress, you will define a layout grid, a type scale, an accessible color system, and an image treatment, then assemble and critique a complete landing page. Keep your decisions documented in the worksheets and templates so your final page is consistent and defensible.

## Layout and Grid Systems for the Web

Define a reusable responsive grid and a spacing scale, and pressure-test a layout in grayscale.

### Exercise: Wireframe in Grayscale First

Pick one page you want to build (a landing page, an about page, or a product page). Sketch its layout in grayscale only, using boxes for content blocks. Do not add color, real images, or final type. Then apply the squint test and note whether structure still reads.

- What are the major content sections, top to bottom, and in what order should the eye meet them?  
\_\_\_\_\_
- Which blocks belong together, and how will proximity and common region show that grouping?  
\_\_\_\_\_
- When you squint at the sketch, can you still tell header from body and one section from the next? If not, what is mushing together?  
\_\_\_\_\_
- Which sections sit side by side on desktop and must stack on mobile?  
\_\_\_\_\_

### Worksheet: Grid and Breakpoint Specification

Fill in the exact numbers for your responsive grid at each breakpoint. Use these values when you set up your Figma or Webflow layout grid.

Content container max width (px)

\_\_\_\_\_

Number of columns — desktop

\_\_\_\_\_

Number of columns — tablet

\_\_\_\_\_

Number of columns — mobile

\_\_\_\_\_

Gutter width (px)

\_\_\_\_\_

Outer margin — desktop (px)

\_\_\_\_\_

Outer margin — mobile (px)

\_\_\_\_\_

\_\_\_\_\_

Breakpoint values (e.g., 1280 / 1024 / 768 / 480)

---

Columns used for body text (to keep ~60 char measure)

---

### Checklist: 8-Point Spacing Audit

- Adopt a single spacing scale (4, 8, 16, 24, 32, 48, 64, 80) as your only options
- Use large values (48 to 80) to separate major sections
- Use mid values (16 to 32) between related blocks
- Use small values (4 to 8) inside components such as icon-to-label gaps
- List every unique spacing value on the page and round any off-scale value (like 13 or 27) to the scale
- Save spacing values as reusable tokens or styles, not retyped numbers

## Typography for Screens

Choose fonts, build a modular type scale, and tune body text for comfortable reading.

### Exercise: Build and Assign a Modular Type Scale

Using [type-scale.com](http://type-scale.com) or a calculator, set a 16px base and choose a ratio. Generate the scale and assign each step to a role. Then view your headings at mobile width and reduce them 20 to 30 percent.

- Which base size and ratio did you choose, and why (subtle vs bold contrast)?

---

- List the resulting sizes and the role each one fills (H1, H2, H3, body, small).

---

- Beyond size, how will weight, color, and spacing reinforce each level so headings need not be huge?

---

- What are your reduced heading sizes for mobile?

---

### Worksheet: Type System Specification

Record the typographic decisions you will apply consistently across the page. Save these as text styles in your design tool.

Heading typeface and source

---

Body typeface and source

---

Pairing rationale (contrast / single-family)

---

Base body size (px)

---

Scale ratio

---

H1 / H2 / H3 sizes (px)

---

Body line height

---

Heading line height

---

Target line length (characters)

---

---

## Checklist: Readability Pass

- Body size set to 16 to 20 px
- Body line height set to 1.5 to 1.6
- Heading line height set to 1.1 to 1.25
- Line length constrained to 45 to 75 characters (about 60 ideal)
- Long-form text left-aligned, not justified
- Paragraph spacing of roughly 16 to 24 px between blocks
- All-caps reserved for short labels only

## Color and Visual Hierarchy

Assemble an accessible palette, verify contrast, and set a deliberate reading order.

### Exercise: Assemble a 60-30-10 Palette With Tonal Ramps

Using Coolers, Adobe Color, or the Tailwind palette, build a structured palette: a dominant neutral, a secondary, and a scarce accent, plus a neutral ramp and semantic colors. Create light, base, and dark variants of each main color.

- What are your dominant (60%), secondary (30%), and accent (10%) colors?

---

- List your neutral ramp from background to darkest text (5 to 9 steps).

---

- What are your success, warning, error, and info colors?

---

- Where on the page will the accent appear, and how will you keep it scarce?

---

### Worksheet: WCAG Contrast Log

List every text-and-background pair and check each in WebAIM or a Figma contrast plugin. Record the ratio and pass/fail. Fix any failures by adjusting color, not by enlarging text alone.

Text color (hex)

---

Background color (hex)

---

Text role (body / large heading / UI element)

---

Measured contrast ratio

---

Required minimum (4.5:1 body, 3:1 large/UI)

---

Pass or fail

---

Fix applied (darken text / lighten background)

---

---

### Checklist: Hierarchy and Color Meaning Check

- Each screen has exactly one clear focal point
- Primary action is high-contrast and solid; secondary actions are quieter
- Most important element sits where the F- or Z-pattern scan begins
- Hierarchy holds when the page is viewed in grayscale

- No meaning is carried by color alone (paired with icon, text, or pattern)
- Links in body text are distinguishable by more than hue (e.g., underline)
- Light-gray-on-white text checked and confirmed to pass 4.5:1

## Imagery, Components, and Finishing a Page

Prepare imagery, standardize components, then run a structured critique before shipping.

### Exercise: Curate and Crop a Coherent Image Set

Select the images your page needs from Unsplash, Pexels, or original photography. Crop each using the rule of thirds and to the aspect ratio your layout requires. Plan where text overlays an image and how you will keep it readable.

- What shared style, lighting, and color mood ties your images together?

---

- What aspect ratios does your layout need (hero, card, mobile), and where is each image's focal point under responsive cropping?

---

- Which images carry overlaid text, and what overlay or scrim keeps contrast at WCAG AA?

---

- What alt text will you write for each meaningful image?

---

### Worksheet: Component and Button State Specification

Define your reusable components and every interactive state so the page stays consistent and accessible. Build these as components or styles you reuse everywhere.  
Corner radius value used everywhere (px)

---

Shadow / elevation style

---

---

Primary button: fill, text color, padding

---

---

Secondary button: style (outline / muted)

---

---

Button states defined (default / hover / pressed / focus / disabled)

---

---

Minimum touch target size (px)

---

---

Card internal padding (from spacing scale)

---

---

Card internal order (image / heading / text / action)

---

### Checklist: Pre-Ship Visual QA

- Grayscale test passed: hierarchy holds without color
- Alignment audited with the grid on; off-edge elements fixed
- Spacing audited: every value on the 8-point scale
- Contrast re-checked: every text pair meets WCAG AA
- Viewed at 375 px mobile, tablet, and full desktop
- Hover and focus states work on every interactive element
- Images optimized (WebP/AVIF/SVG), compressed, and have alt text
- Same radius, shadow, button, and type styles used throughout

[ ] Five-second test run on someone else; key message and action landed

## Your Action Plan

1. Choose one real page to build and wireframe it in grayscale, ordering sections by importance.
2. Set up a 12-column responsive grid template with your chosen container, gutters, margins, and breakpoints.
3. Lock an 8-point spacing scale and use it for every gap, margin, and padding.
4. Select one or two web fonts and generate a modular type scale, saved as text styles.
5. Tune body text to 16 to 20 px, 1.5 to 1.6 line height, and a 45 to 75 character measure.
6. Build a 60-30-10 palette with neutral and semantic ramps, saved as color tokens.
7. Log and pass WCAG AA contrast for every text and UI pairing.
8. Curate and crop a coherent image set, optimize formats, and plan text overlays.
9. Design reusable buttons and cards with full states and a single radius and shadow style.
10. Run the full pre-ship visual QA checklist and a five-second test, then finalize the page.









