

Web Design with Webflow — Workbook

This workbook turns the Web Design with Webflow course into a finished, live website: a multi-page marketing site plus a CMS-powered blog on your own domain. Each section maps to one course module and moves you from mastering the Designer and box model, through responsive flexbox and grid layout, into the CMS, and finally to interactions, a global style system, and a clean launch. Work through the items in order inside a single Webflow project and keep your class naming consistent throughout.

The Webflow Designer and How the Web Works

Get fluent in the Designer, the box model, and Webflow's class-based styling before you design anything serious.

Exercise: Build-and-Read the Navigator Tree

In a blank Webflow project, add a Section, then nest a Container, a Div Block, a Heading set to h2, and a Paragraph inside it. Open the Navigator (Z) and study the nesting tree. Practice selecting the Paragraph, then pressing Escape repeatedly to climb to each parent. Do not style anything; the goal is to feel selection and nesting.

- What HTML tag does each element you added render as, and how do you confirm it in the selection badge?
- How does pressing Escape change which element is selected versus clicking in the canvas?
- Where in the Navigator can you tell that the Div Block sits inside the Container and not the other way around?

Exercise: See Padding Versus Margin

Build a simple card: a Div Block with a visible background color, 24px of padding on all sides, a Heading (h3), a Paragraph, and a button inside. Give the card a 32px top margin. Toggle the card background on and off to watch padding (inside, colored) versus margin (outside, transparent) behave.

- Which spacing changed the size of the card box itself, and which changed the gap to the element above it?
- What happened to the background color when you increased padding versus margin?
- What padding values make a button feel comfortable rather than cramped?

Worksheet: Class Naming Convention Sheet

Decide your class naming system before you create your tenth class. Fill in each value and write three example class names that follow your rule.

Case style (lowercase / kebab-case / underscore)

Layout class prefix (e.g. section_, container_)

Component class pattern (e.g. button_primary)

Utility/text class pattern (e.g. text-size-large)

Example class 1

Example class 2

Example class 3

Where the convention is documented for collaborators

Checklist: Designer Foundations Locked

- Can open the Add panel (A), Style panel (S), and Navigator (Z) by keyboard
- Can climb the nesting hierarchy with Escape and the breadcrumb
- Can explain padding versus margin in one sentence each
- Set every text element to the correct semantic tag (one h1 per page)
- Read the Selector field before each style change to confirm blue base versus orange combo
- Chosen and written down a class naming convention

Responsive Layout with Flexbox and Grid

Build real layouts with flexbox and CSS Grid, then make them adapt cleanly across all four breakpoints.

Exercise: Flexbox Navbar and Card Row

Build a navbar as a flex parent (direction row, align items center, justify content space-between) with a logo on the left and three links on the right. Then build a three-card feature row as a flex parent with a 24px gap and align items stretch so all cards match the tallest. Use gap rather than per-child margins.

- Which justify-content value pushed the logo and links to opposite ends?
-

- How did align items stretch affect cards of different content lengths?
-

- Why is gap preferable to adding margin to each card?
-

Exercise: CSS Grid Gallery with fr Units

Create a six-cell image gallery using a Div Block set to display Grid, with a 3-column track of 1fr 1fr 1fr, two rows, and a 24px gap. Make one image span two columns to create a feature cell. Then change the grid to 2 columns at the tablet breakpoint and 1 column at mobile portrait.

- What does the fr unit do that fixed pixel columns cannot for responsive layout?
-

- How did you make a single item span two columns without affecting the others?
-

- When would you reach for Grid instead of flexbox on this layout?
-

Worksheet: Breakpoint Adaptation Plan

For each key layout block, plan the specific changes you will make at each breakpoint. Remember the cascade flows downward only. Fill one row per block.

Layout block (e.g. hero, card row, gallery)

Desktop base layout (992px+)

Tablet change (991px and below)

Mobile landscape change (767px and below)

Mobile portrait change (478px and below)

Heading size at mobile portrait (px)

Checklist: Responsive Quality Pass

- Built at least one flexbox layout and one CSS Grid layout
- Used gap for spacing inside flex and grid containers
- Stacked multi-column rows to direction column on mobile
- Reduced oversized desktop headings at the mobile breakpoints
- Confirmed no horizontal overflow at 478px
- Previewed on a real phone, not only the canvas

Dynamic Content with the Webflow CMS

Model your content as a Collection, display it with a Collection List, and generate a page per item with a template.

Worksheet: Collection Field Plan

Plan your Blog Posts collection before creating it. List every field, its type, and whether it is required. Add a row for each field, and note any supporting collections (Categories, Authors) you must build first. Field name (e.g. Name, Main Image, Post Body)

Field type (Plain Text / Rich Text / Image / Reference / Switch / Date)

Required? (yes/no)

Notes (e.g. Slug auto-generates from Name)

Supporting collection needed (Categories / Authors / none)

Exercise: Build a Dynamic Blog Index

Create the Blog Posts collection with at least six fields plus a referenced Categories collection, and add three or four real sample items. On the blog index page, add a Collection List, design one card, and bind the heading to Name, the image to Main Image, the summary to Summary, and the card link to the item. Sort newest first and style the Empty State.

- Which elements did you bind, and which field did each connect to?
 - What sort and filter settings did you apply, and what does the result show?
 - What message did you put in the Empty State so the page never looks broken?
-

Exercise: Design the Collection Page Template

Open the Blog Posts Collection Page template. Bind the title to Name, the hero image to Main Image, and the article body to the Rich Text Post Body field. Constrain the body to about 720px reading width. Bind the page Title Tag to Name and Meta Description to Summary. Add a related-posts Collection List filtered to the same category.

- How does one template generate a unique URL and page for every item?

- What reading width did you set, and how does it improve legibility?

- Which dynamic SEO fields did you bind so each post has unique meta data?

Checklist: CMS Build Complete

- Collection created with correctly typed fields and required flags set
- Categories (and any Authors) collection built and referenced
- Collection List on the index bound, sorted, and with a styled empty state
- Homepage teaser list limited to the 3 most recent posts
- Collection Page template binds title, image, and rich text body
- Per-item Title Tag, Meta Description, and Open Graph image bound to fields

Interactions, Polish, and Going Live

Add purposeful motion, lock in a global style system, run pre-launch QA, and publish to a custom domain.

Exercise: Scroll-Fade and Hover-Lift Interactions

Add a While Scrolling Into View interaction to your sections: initial state at 0 percent opacity and moved down 40px, visible state at 100 percent opacity and 0px. Add a Mouse Hover interaction to your cards: lift up 4px and increase the box shadow on hover, returning on hover-out, with a 200 to 300 millisecond ease-out duration.

- What duration and easing felt polished rather than jittery or sluggish?

- Which elements benefited from the scroll-fade, and where did it feel like too much?

- How will you respect users who prefer reduced motion?

Worksheet: Global Style System Sheet

Define your site-wide system so consistency happens by default. Fill in your variables and base type settings, then confirm your shared components.

Brand primary color (variable)

Brand secondary / accent color (variable)

Type scale ratio (e.g. 1.25) and base size (e.g. 16px)

Resulting heading sizes (h1..h3 in px)

Body line-height (e.g. 1.5)

Heading typeface / Body typeface

Components built (navbar, footer, CTA)

Worksheet: Custom Domain DNS Worksheet

Record the exact DNS values Webflow provides so you can connect your domain correctly. Connect the domain at least a day before any launch deadline to allow propagation.

Domain name to connect

Registrar (e.g. GoDaddy, Namecheap, Cloudflare)

Root A record value(s) from Webflow

www CNAME value from Webflow

Default chosen (root or www)

SSL enabled (yes/no)

Date connected (for propagation tracking)

Checklist: Pre-Launch QA and SEO

- Checked every page at all four breakpoints and on a real phone
- Clicked every link and button, including CMS links, with no broken or # targets
- Replaced all Lorem Ipsum and placeholder images; added alt text everywhere
- Submitted every form and confirmed receipt of the submission
- Designed a custom 404 page
- Compressed images so no single asset is multi-megabyte
- Set a unique Title Tag and Meta Description on every page
- Connected the custom domain with SSL and enabled the sitemap

Your Action Plan

1. Create one Webflow project and set your class naming convention before building
2. Master the box model by building a card and toggling padding versus margin
3. Build the desktop base layout with flexbox for rows and CSS Grid for galleries
4. Make every layout responsive across all four breakpoints, stacking columns on mobile
5. Plan and create the Blog Posts collection with typed fields and a Categories reference
6. Build the blog index with a bound, sorted Collection List and a styled empty state
7. Design the Collection Page template with bound rich text, reading width, and dynamic SEO
8. Add purposeful scroll-fade and hover-lift interactions with 200 to 300ms ease-out timing
9. Define brand color and type-scale variables and turn the navbar and footer into components
10. Run the full pre-launch QA and SEO checklist, then connect a custom domain with SSL and publish

