



No-Code Site Builders

A Complete 8-Chapter Web Platforms Course

Topics covered:

The no-code/low-code spectrum · Wix & Squarespace compared
Dreamweaver as a genuinely different kind of tool · where WordPress sits on the same spectrum
Five separated trade-off dimensions · a practical decision framework

A comparative survey, not a build-along tutorial

Capstone: three real scenarios, three genuinely different recommended platforms

Exercises: 24 hands-on reasoning scenarios with worked solutions

Format: A4 · Dark-theme code examples

Philip Osztromok · Generated with Claude

Table of Contents

- 01 The No-Code/Low-Code Spectrum
- 02 Wix
- 03 Squarespace
- 04 Dreamweaver — A Genuinely Different Kind of Tool
- 05 Where WordPress Sits on This Same Spectrum
- 06 The Real Trade-Offs, Taken Separately
- 07 A Practical Decision Framework
- 08 Capstone: Choosing the Right Platform for Three Real Scenarios

CHAPTER 1 OF 8

The No-Code/Low-Code Spectrum

No-Code Site Builders

Chapter 1 · The No-Code/Low-Code Spectrum

"No-code" gets used as though it names one single category of tool. It doesn't. This chapter lays out the real spectrum it actually spans — and previews exactly where each of this course's own later chapters sits along it, including *WordPress Fundamentals*, already covered in real depth elsewhere on this site.

A Spectrum, Not a Category

POSITION	WHAT IT ACTUALLY MEANS	EXAMPLE
Fully hosted, fully no-code	The platform owns the hosting, the editor, and the underlying code entirely — you never see or touch it	Wix, Squarespace
Self-hosted CMS + page builder	You own the hosting; day-to-day editing is drag-and-drop, but a real underlying codebase exists underneath	WordPress + a page builder plugin (e.g. Elementor)
Self-hosted CMS alone	Content management is no-code; extending functionality beyond it requires real code	WordPress, as covered site-owner-side in WordPress Fundamentals
WYSIWYG desktop authoring	A visual editor that still produces real HTML/CSS files, hosted independently — not a hosted platform at all	Dreamweaver
Fully custom code	Complete control, hand-written from scratch — maximum effort, maximum flexibility	Hand-coded HTML/CSS/JS, or a custom WordPress theme (WordPress Intermediate/Advanced)

"NO-CODE" IS A MARKETING TERM, NOT A TECHNICAL ONE

Treating every tool on this spectrum as interchangeably "the easy option" hides real, meaningfully different trade-offs — how much you actually own, how much you can later change, what happens when a project outgrows the platform. This course's own point is to make those differences concrete, not to rank the tools from best to worst.

This Course's Own Roadmap

WHERE EACH LATER CHAPTER SITS ON THE SPECTRUM ABOVE

- **Fully hosted, fully no-code** — Chapter 2 (Wix), Chapter 3 (Squarespace)
- **A genuinely different kind of tool** — Chapter 4 (Dreamweaver): not a hosted platform at all
- **Where WordPress itself sits** — Chapter 5, placing the site's own existing WordPress track honestly on this same continuum
- **The trade-offs, and the decision itself** — Chapters 6 and 7
- **Applying it for real** — Chapter 8's own capstone, three scenarios, three different right answers

What This Course Is — and Isn't

This is a comparative, decision-focused survey — not a build-along tutorial for any single tool. *WordPress Fundamentals* teaches you to actually run a WordPress site step by step; this course exists one level up from that, helping decide whether WordPress (or Wix, or Squarespace, or something else entirely) is even the right starting point for a given project in the first place.

Hands-On Exercises

EXERCISE 1

Explain why this chapter treats "no-code" as a spectrum rather than a single category, using at least two specific positions from this chapter's own table.

[View solution](#)

EXERCISE 2

Explain where WordPress, as covered in WordPress Fundamentals, sits on this chapter's own spectrum, and why that position is genuinely different from Wix or Squarespace's own position.

[View solution](#)

EXERCISE 3

Explain why this chapter warns against treating "no-code" tools as interchangeably "the easy option," and what real differences that framing tends to obscure.

[View solution](#)

CHAPTER 1 QUICK REFERENCE

- "No-code" spans a real spectrum: fully hosted platforms → CMS + page builder → CMS alone → WYSIWYG desktop tools → fully custom code
- Wix/Squarespace sit at the fully hosted end; Dreamweaver is a genuinely different kind of tool, not a hosted platform at all
- WordPress (covered elsewhere on this site) sits in the middle — no-code for content, real code for anything beyond it
- This course is comparative and decision-focused, not a build-along tutorial for any one tool
- **Next chapter:** Wix

CHAPTER 2 OF 8

Wix

No-Code Site Builders

Chapter 2 · Wix

Wix sits at the "fully hosted, fully no-code" end of Chapter 1's own spectrum — this chapter covers what that actually means in practice, both the real convenience it buys and the one real cost that comes with it.

The Drag-and-Drop Editor

Wix's own signature feature is genuine freeform positioning — elements can be placed anywhere on the canvas, pixel by pixel, rather than being constrained to a fixed template grid. This is real design freedom within the platform's own boundaries, closer to a visual design tool than a fill-in-the-blanks template system.

Wix ADI — AI-Generated Starting Points

Wix ADI (Artificial Design Intelligence) builds a starter site automatically from a short questionnaire — business type, style preferences, existing content — genuinely useful for getting from nothing to a reasonable first draft quickly, before manually refining it further in the drag-and-drop editor.

The App Marketplace

Wix's own app marketplace extends site functionality — booking systems, contact forms, online stores — in the same spirit as WordPress Fundamentals 6's own plugin ecosystem, but architecturally different: every app runs entirely within Wix's own closed platform, not as portable, independently-owned code the way a WordPress plugin's files are.

Real Strengths

STRENGTH	WHY IT'S GENUINE
No separate hosting to buy or configure	Directly contrasts with WordPress Fundamentals 2's own hosting-requirements chapter — Wix handles this entirely
Zero server maintenance	No updates, no backups, no security patching to manage yourself — all handled by Wix

STRENGTH**WHY IT'S GENUINE**

Genuinely fast to launch

A reasonable site can go live in hours, not days

The Real Limitation — Portability

THIS IS THE CONCRETE COST BEHIND CHAPTER 1'S OWN SPECTRUM POSITION

A Wix site cannot be exported to run anywhere else. Deciding to leave Wix — for any reason, cost, outgrowing its capabilities, wanting more control — means starting over from scratch on whatever platform comes next, not migrating existing work. Compare this directly to WordPress: because a WordPress site's content lives in a real, independently-owned MySQL database and its files live in real, independently-owned hosting (per WordPress Fundamentals 1's own LAMP-stack material), a WordPress site genuinely can move to different hosting without starting over. A Wix site has no equivalent path.

THIS IS EXACTLY WHAT "YOU NEVER SEE OR TOUCH IT" MEANT IN CHAPTER 1

Chapter 1's own spectrum table described the fully-hosted position as one where the platform owns the hosting, the editor, and the underlying code entirely. Lock-in isn't a separate, additional downside — it's the direct, structural consequence of that same ownership arrangement working exactly as designed.

Hands-On Exercises

EXERCISE 1

A small business owner wants their site live within a day and has no interest in managing hosting or server updates themselves. Explain why Wix's own real strengths, per this chapter, genuinely fit this specific need.

 [View solution](#)**EXERCISE 2**

Explain why a Wix site can't simply be "migrated" to different hosting the way a WordPress site can, using this chapter's own comparison.

 [View solution](#)**EXERCISE 3**

Explain why this chapter describes Wix's own lock-in as a "direct, structural consequence" of its hosting model rather than a separate flaw the platform could simply fix.

 [View solution](#)

CHAPTER 2 QUICK REFERENCE

- Wix's own freeform drag-and-drop editor allows genuine pixel-level positioning, not just template-constrained editing
- **Wix ADI** generates a starter site from a short questionnaire
- The app marketplace extends functionality within Wix's own closed platform — not portable code, unlike a WordPress plugin
- Real strengths: no hosting to buy/configure, zero server maintenance, genuinely fast to launch
- The real cost: a Wix site cannot be exported — leaving means starting over, unlike WordPress's own portable content/files
- **Next chapter:** Squarespace

CHAPTER 3 OF 8

Squarespace

No-Code Site Builders

Chapter 3 · Squarespace

Squarespace sits at the exact same "fully hosted, fully no-code" position on Chapter 1's own spectrum as Wix — but the editing philosophy underneath that shared position is genuinely different, and the lock-in it shares with Wix shows up from a second, distinct angle worth naming separately.

Template-Driven, Not Freeform

Where Wix's own editor allows placing elements anywhere on a blank canvas, Squarespace starts from professionally-designed templates and edits within defined sections and blocks rather than free positioning. This constraint is deliberate, not a missing feature — it exists specifically to guide design decisions the user might otherwise get wrong.

WIX (CHAPTER 2)	SQUARESPACE
Freeform canvas, place elements anywhere	Structured sections and blocks within a chosen template
More flexibility, higher risk of an inconsistent result without design experience	Less flexibility, but a professional-looking result that's genuinely harder to get wrong
Freedom as the selling point	Guardrails as the selling point

NEITHER APPROACH IS OBJECTIVELY BETTER

This is a real trade-off, not a case of one platform simply being more advanced than the other — Chapter 1's own spectrum framing applies here too: the right amount of guidance versus freedom depends entirely on what a specific user and project actually need.

Design-Forward Strengths

A GENUINELY EARNED REPUTATION

Squarespace has a real, specific reputation for portfolio, photography, and other visually-driven creative sites — its templates lean toward strong typography defaults and large, image-forward layouts, a genuine strength for exactly this category of site rather than a general-purpose claim.

E-Commerce Features

Squarespace includes built-in online store functionality — inventory management, payment processing — comparable in spirit to Wix's own store apps, but more tightly integrated into the core platform by default rather than added on separately.

The Same Lock-In, From a Different Angle

Squarespace shares Wix's own structural platform lock-in exactly, per Chapter 2's own reasoning — a Squarespace site can't be exported to run elsewhere any more than a Wix site can.

A SECOND, DISTINCT KIND OF LOCK-IN WORTH NAMING SEPARATELY

Because Squarespace content is organized around each template's own specific section structure, switching templates *within* Squarespace itself can require real rework — content built for one template's own sections doesn't always map cleanly onto a different template's own structure. This is a genuinely different concern from Chapter 2's own platform-level lock-in: it's a real cost to changing your mind *within* the platform, not just a cost to leaving it entirely.

Hands-On Exercises

EXERCISE 1

A freelance photographer with no design background wants a professional-looking portfolio site quickly. Explain why this chapter's own comparison of Wix and Squarespace favors Squarespace here specifically.

 [View solution](#)

EXERCISE 2

Explain the difference between Squarespace's own platform-level lock-in and its template-switching cost, using this chapter's own material to distinguish the two.

 [View solution](#)

EXERCISE 3

Explain why this chapter says neither Wix's freeform editor nor Squarespace's template-driven editor is "objectively better," connecting your answer back to Chapter 1's own spectrum framing.

 [View solution](#)

CHAPTER 3 QUICK REFERENCE

- Squarespace edits within structured template sections, not a freeform canvas like Wix — a deliberate design constraint, not a missing feature
- The trade-off: less flexibility, but a professional-looking result that's harder to get wrong without design experience
- A genuinely earned reputation for portfolio/photography/creative sites specifically
- Built-in e-commerce, more tightly integrated by default than Wix's own store apps
- Shares Wix's exact platform-level lock-in, plus a second, distinct template-switching cost within the platform itself
- **Next chapter:** Dreamweaver — A Genuinely Different Kind of Tool

CHAPTER 4 OF 8

Dreamweaver — A Genuinely Different Kind of Tool

No-Code Site Builders

Chapter 4 · Dreamweaver — A Genuinely Different Kind of Tool

Wix and Squarespace share one deep architectural trait: both are hosted platforms managing everything end to end. Dreamweaver shares almost nothing with either — it isn't a competitor occupying a nearby spot on Chapter 1's own spectrum, it's a fundamentally different category of tool entirely.

What Dreamweaver Actually Is

Dreamweaver is a desktop application (part of Adobe Creative Cloud) combining a visual "Design" view with a real "Code" view, often shown side by side — you can drag elements visually or write HTML/CSS/JavaScript directly, watching both stay in sync. Critically, Dreamweaver produces real, standalone HTML/CSS/JS files — it doesn't bundle hosting the way Wix or Squarespace do. Those files still need to be uploaded to independently-purchased hosting, historically via FTP, before anyone else can see them.

Where This Actually Sits on Chapter 1's Own Spectrum

TOOL	HOSTING	WHAT IT PRODUCES
Wix / Squarespace	Bundled, fully managed	No accessible underlying code at all
Dreamweaver	Not included — bring your own	Real, standalone HTML/CSS/JS files
Fully hand-coded (no tool)	Not included — bring your own	The same kind of real files, written directly rather than assisted visually

NEITHER "FULLY HOSTED" NOR QUITE "FULLY CUSTOM"

Dreamweaver doesn't map cleanly onto either end of Chapter 1's spectrum. It shares fully-custom-code's complete independence from any hosting platform, while still offering WYSIWYG assistance fully hand-coding doesn't provide — genuinely its own position, not a midpoint average of the other two.

Historical Context, Honestly

Dreamweaver (originally Macromedia's, acquired by Adobe in 2005) was the dominant professional web design tool through the 1990s and much of the 2000s — the default choice before hosted platforms and modern frameworks existed in anything like their current form. Wix (2006) and Squarespace (2004) hadn't yet captured the "no-code, no hosting hassle" audience Dreamweaver was never built to serve in the first place.

WHY ITS DOMINANCE GENUINELY DECLINED — SEVERAL REAL REASONS AT ONCE

- WYSIWYG-generated HTML/CSS was widely, and often fairly, criticized as bloated and non-semantic compared to hand-written code, a real technical complaint from developers of that era
- The rise of dedicated code editors developers increasingly preferred for hand-coding, alongside modern build tools and frameworks that don't fit a WYSIWYG-first workflow well
- Hosted platforms captured users who wanted "no code, no separate hosting" — a need Dreamweaver structurally never addressed
- Content management systems captured users who wanted ongoing editing without reopening a desktop application for every change

Why It's Still Genuinely Encountered Today

This isn't purely historical — real, current reasons keep Dreamweaver relevant in specific contexts: legacy sites originally built in it still exist and occasionally need maintenance; some creative-agency workflows still favor designers comfortable with a visual tool over a pure code editor; some small studios and individuals who learned it years ago have simply never had a reason to switch.

A GENUINELY USEFUL SKILL: RECOGNIZING A LEGACY DREAMWEAVER SITE

Inheriting an older site built this way often means encountering non-semantic markup, absolute-positioning quirks, or even table-based layouts predating modern CSS layout entirely — real, practical patterns worth recognizing before attempting to maintain or modernize such a site.

Hands-On Exercises

EXERCISE 1

Explain why this chapter says Dreamweaver "doesn't map cleanly onto either end" of Chapter 1's own spectrum, rather than sitting somewhere between Wix and fully custom code.

 [View solution](#)

EXERCISE 2

Explain why this chapter lists several separate, distinct reasons for Dreamweaver's decline rather than pointing to a single cause.

 [View solution](#)

EXERCISE 3

A developer inherits a client's old site and notices it uses HTML tables to lay out the page rather than CSS. Explain what this suggests about the site's likely origin, using this chapter's own material.

 [View solution](#)

CHAPTER 4 QUICK REFERENCE

- Dreamweaver is a desktop WYSIWYG-plus-code editor producing real, standalone HTML/CSS/JS files — no hosting bundled, unlike Wix/Squarespace
- It sits in its own position on Chapter 1's spectrum — not fully hosted, not quite the same as fully hand-coded either
- Once the dominant professional web design tool (1990s-2000s), before hosted platforms and modern frameworks matured
- Decline had several real, distinct causes: WYSIWYG code-quality criticism, the rise of dedicated code editors/frameworks, hosted platforms, and CMS platforms
- Still genuinely encountered via legacy sites and some creative-agency workflows — recognizing its own era's markup patterns (tables for layout, non-semantic HTML) is a real, practical skill
- **Next chapter:** Where WordPress Sits on This Same Spectrum

CHAPTER 5 OF 8

Where WordPress Sits on This Same Spectrum

No-Code Site Builders

Chapter 5 · Where WordPress Sits on This Same Spectrum

Chapter 1's own spectrum table already separated WordPress into two distinct positions — "self-hosted CMS alone" and "self-hosted CMS + page builder." This chapter goes deep on both, and places them honestly against Wix, Squarespace, and Dreamweaver rather than treating WordPress as automatically superior just because this site already covers it in depth.

WordPress Without a Page Builder

WordPress Fundamentals covers exactly this position — content-level editing via the Block Editor (its own Chapter 4) is already genuinely no-code, letting a site owner build real, flexible page layouts using blocks, patterns, and reusable blocks with no PHP at all. Full site-wide design control — headers, footers, complex custom templates — traditionally required either a classic PHP theme (real code, covered in *WordPress Intermediate/Advanced*) or, on modern block themes, Full Site Editing, itself also block-based and no-code.

WORDPRESS CORE IS ALREADY PARTLY NO-CODE, ON ITS OWN

It's a common mistake to assume WordPress requires a separate page-builder plugin to be no-code at all. It doesn't — the Block Editor alone already delivers real no-code content editing; a page builder adds something more specific on top of that, covered next.

Page Builders — A Further No-Code Layer

Plugins like Elementor add a more Wix-like freeform, drag-and-drop visual builder on top of WordPress — letting a site owner construct full page layouts (in some cases including headers and footers) visually, without touching PHP templates or the classic theme system covered in *WordPress Intermediate/Advanced* at all.

SETUP	DESIGN FREEDOM	OWNERSHIP/PORTABILITY
Wix	Freeform, no code	None — locked to Wix, per Chapter 2
WordPress + page builder	Freeform, no code — genuinely close to Wix's own experience	Full — real hosting, real database, per <i>WordPress Fundamentals</i> 1

SETUP	DESIGN FREEDOM	OWNERSHIP/PORTABILITY
WordPress alone (Block Editor)	Real, but more content-focused than full freeform layout	Full

THE GENUINE "BEST OF BOTH WORLDS" CASE

WordPress plus a page builder gets close to Wix's own drag-and-drop convenience while keeping WordPress's own core structural advantage: the site's content and files remain genuinely, independently owned, hosted on infrastructure the site owner actually controls — none of Chapter 2's own platform lock-in applies here.

An Honest Caveat — Page Builders Aren't Free of Trade-Offs Either

REAL PERFORMANCE COST

Page builders commonly generate more bloated HTML/CSS than a hand-coded theme or the Block Editor's own native output — directly relevant to *WordPress Intermediate/Advanced 9's* own performance material and *Performance & Core Web Vitals's* LCP concerns specifically.

A GENUINELY NEW FORM OF LOCK-IN, WORTH NAMING HONESTLY

Each page builder stores its own layout data in its own proprietary format inside the WordPress database. Switching from one page builder to a different one can require significant rework, for essentially the same underlying reason Chapter 3 named for Squarespace's own template-switching cost — content structured around one system's own specific format doesn't map cleanly onto a different system's format, even while staying on the exact same underlying WordPress installation the whole time.

Hands-On Exercises

EXERCISE 1

Explain why WordPress with a page builder is described as "genuine best of both worlds" in one specific dimension, without claiming it has no downsides at all.

 [View solution](#)

EXERCISE 2

Explain why this chapter's own page-builder lock-in warning is compared directly to Chapter 3's own Squarespace template-switching cost, rather than to Chapter 2's own Wix platform lock-in.

[View solution](#)

EXERCISE 3

A learner assumes WordPress only becomes "no-code" once a page builder plugin is installed. Explain what's wrong with this assumption, using this chapter's own material.

[View solution](#)

CHAPTER 5 QUICK REFERENCE

- WordPress alone (via the Block Editor) is already genuinely no-code at the content level — a page builder isn't required for that baseline
- Page builders (e.g. Elementor) add Wix-like freeform visual layout on top of WordPress, without touching PHP
- WordPress + a page builder gets close to Wix's own design freedom while keeping WordPress's own full ownership/portability — no platform lock-in
- Honest trade-offs remain: real performance cost from bloated output, and a genuine, proprietary-format lock-in when switching between page builders
- Page-builder lock-in echoes Chapter 3's own template-switching cost, not Chapter 2's own platform-level lock-in
- **Next chapter:** The Real Trade-Offs, Taken Separately

CHAPTER 6 OF 8

The Real Trade-Offs, Taken Separately

No-Code Site Builders

Chapter 6 · The Real Trade-Offs, Taken Separately

Chapters 2 through 5 surfaced real trade-offs one platform at a time. This chapter pulls five of them out as their own named dimensions, deliberately kept separate rather than folded into one vague "it depends" — because a platform can genuinely win on one dimension and lose on another, and collapsing them together hides exactly that.

1. Portability & Vendor Lock-In

PLATFORM	WHAT YOU CAN TAKE WITH YOU
Wix / Squarespace	Nothing — no export path exists (Chapters 2, 3)
WordPress + page builder	The WordPress install and content fully; the specific page-builder layout data, less cleanly (Chapter 5)
WordPress alone / Dreamweaver / hand-coded	Everything — real files and/or a real database, independently owned

2. Cost Structure — Not Just "Which Is Cheaper"

This is about the *shape* of the cost, not simply which option is cheapest overall — that depends heavily on how long a site runs and how it's used.

PLATFORM	COST SHAPE
Wix / Squarespace	One recurring subscription, hosting included
WordPress	Free core software, separate hosting cost, optional premium plugins/themes
Dreamweaver	A software cost for the application itself, plus entirely separate hosting — no ongoing platform fee once files exist

3. The Real Ceiling

What happens when a project needs something the platform's own built-in tools genuinely can't do?

A GENUINELY DIFFERENT CEILING FOR EACH

Wix/Squarespace have a real, structural ceiling — app marketplaces stretch it somewhat, but you can never reach below the platform's own abstraction layer. WordPress's ceiling is much higher, since real PHP development (*WordPress Intermediate/Advanced*) is always available as a genuine escape hatch. Dreamweaver and fully hand-coded sites have no ceiling at all — but the trade-off is starting further from "done" on any given feature, since nothing is pre-built for you.

4. SEO Control

This deserves an honest, non-oversimplified treatment rather than a blanket verdict either way.

HISTORICALLY REAL, MEANINGFULLY IMPROVED, STILL NOT EQUAL

Hosted platforms have historically had real, documented limitations on technical SEO control — URL structure customization, canonical tag handling — genuine concerns *SEO Fundamentals* covers in depth. Modern Wix and Squarespace have meaningfully improved this over the years, and the older blanket claim that "no-code platforms are bad for SEO" is dated and oversimplified today. That said, WordPress combined with dedicated SEO tooling still generally offers more granular control over the same technical levers — a real, if narrower, difference than it once was.

5. Ongoing Maintenance Burden

PLATFORM	WHO HANDLES UPDATES/SECURITY/BACKUPS
Wix / Squarespace	The platform, entirely — a genuine strength, per Chapter 2
WordPress	You (or your host) — a real, ongoing responsibility, per WordPress Fundamentals 10's own maintenance material
Dreamweaver / hand-coded	Entirely you — arguably more burden still, since there's no CMS at all to help manage ongoing edits without reopening the desktop app

Hands-On Exercises

EXERCISE 1

Explain why this chapter treats "cost structure" as a separate question from "which platform is cheapest," using this chapter's own three cost-shape examples.

 [View solution](#)

EXERCISE 2

A learner concludes "no-code platforms are just bad for SEO" after reading older material online. Explain what this chapter says is wrong with that conclusion today, and what a more accurate version of the claim would be.

[View solution](#)**EXERCISE 3**

Explain why this chapter says a platform can "genuinely win on one dimension and lose on another," using two specific dimensions from this chapter to illustrate a real case where that happens.

[View solution](#)**CHAPTER 6 QUICK REFERENCE**

- Five separate dimensions: portability/lock-in, cost structure, the real ceiling, SEO control, ongoing maintenance burden
- Cost structure is about the shape of the cost, not simply which platform is cheapest overall
- Wix/Squarespace have a real, structural ceiling; WordPress's is far higher via real PHP development; hand-coding has none, at the cost of starting further from "done"
- SEO control on hosted platforms was historically limited, has meaningfully improved, but WordPress + dedicated tooling still generally offers more granular control
- Maintenance burden runs from zero (Wix/Squarespace) to real and ongoing (WordPress) to heaviest (fully hand-coded, no CMS at all)
- **Next chapter:** A Practical Decision Framework

CHAPTER 7 OF 8

A Practical Decision Framework

No-Code Site Builders

Chapter 7 · A Practical Decision Framework

Chapter 6 named five real trade-off dimensions, deliberately kept separate. This chapter turns each one into a concrete question you can actually ask about a real project — this course's own central chapter, and the one every scenario in Chapter 8's own capstone will run through directly.

Five Questions, One Per Dimension

DIMENSION (CHAPTER 6)	THE QUESTION TO ACTUALLY ASK
Portability / lock-in	How much would it actually cost me — in redone work, not just feelings — if I could never move this site anywhere else?
Cost structure	Would I rather pay one predictable bundled fee, or take on more complexity in exchange for more control over where money goes?
The real ceiling	Is there any real chance this project eventually needs something a page builder or app marketplace genuinely can't do?
SEO control	Is search ranking a primary driver of this project's success, or a secondary nice-to-have?
Maintenance burden	Do I have the time, interest, or budget to handle ongoing updates and security — or do I want zero technical responsibility at all?

These Answers Don't Have to Agree — That's the Point

A REAL PROJECT'S ANSWERS ARE USUALLY MIXED

A project might genuinely want zero maintenance burden (favoring Wix) while also having a real chance of needing custom functionality later (favoring WordPress). That tension is real, not a sign the framework is broken — the honest job here is surfacing it clearly, not forcing five separate questions to collapse into one artificially tidy answer.

A ROUGH HEURISTIC, NOT A RIGID FORMULA

Counting how many of the five questions lean toward "no-code" versus "more custom control" is a reasonable starting point for judgment — but it's not a strict scoring algorithm. One dimension mattering enormously for a

specific project (the real ceiling question, for a business planning real growth) can reasonably outweigh three other questions leaning the other way.

A Light Worked Example

A PERSONAL HOBBY BLOG

Portability: barely matters — low personal stakes if it were ever lost entirely. Cost structure: a single predictable fee is genuinely appealing for a hobby project. Ceiling: essentially no realistic chance of needing custom functionality. SEO: a secondary nice-to-have, not the point of the project. Maintenance: zero interest in managing updates for a hobby. Four or five of five questions lean cleanly toward a hosted, no-code platform here — a genuinely easy case, precisely because the answers aren't in tension with each other at all.

Chapter 8's own capstone applies this same framework to three genuinely harder cases, where the five answers pull in real, different directions.

Hands-On Exercises

EXERCISE 1

Explain why this chapter says a mixed set of answers across the five questions is "not a sign the framework is broken," using this chapter's own reasoning.

[!\[\]\(b18c17fc657b587c69a5722b7427ea01_img.jpg\) View solution](#)

EXERCISE 2

Explain why the hobby-blog example in this chapter is described as "a genuinely easy case," connecting your answer to why the five questions didn't conflict with each other there.

[!\[\]\(cebdf28220713f21f1ebb8bbb41971b3_img.jpg\) View solution](#)

EXERCISE 3

Explain why this chapter warns against treating the five-question count as "a strict scoring algorithm," and describe a realistic situation where one question could outweigh the other four.

[!\[\]\(8b01870f65b6c3ffa52a01ab5a9ceebc_img.jpg\) View solution](#)

CHAPTER 7 QUICK REFERENCE

- Five questions, one per Chapter 6 dimension: portability cost, payment shape preference, real ceiling risk, SEO priority, maintenance appetite
- A mixed set of answers reflects a genuine real-world tension, not a framework failure
- Counting leaning answers is a rough heuristic, not a rigid formula — one high-stakes question can reasonably outweigh several lower-stakes ones
- An easy case (like a hobby blog) is easy specifically because the five answers don't conflict — not because the framework itself is simpler there
- **Next chapter:** Capstone — Choosing the Right Platform for Three Real Scenarios

CHAPTER 8 OF 8

Capstone: Choosing the Right Platform for Three Real Scenarios

No-Code Site Builders

Chapter 8 · Capstone — Choosing the Right Platform for Three Real Scenarios

Seven chapters have built the pieces: a real spectrum, four platforms covered honestly, five separated trade-off dimensions, and a five-question framework. This capstone runs three realistic projects through that exact framework — deliberately landing on three different recommended platforms, since a single universal answer would contradict everything this course has argued.

Scenario 1 — A Local Bakery

QUESTION	ANSWER FOR THIS PROJECT
Portability cost	Low — no realistic reason to ever need to leave
Cost structure preference	One predictable bundled fee genuinely preferred — simple budgeting matters
Real ceiling risk	Minimal — a menu, hours, location, and basic online ordering, all within a hosted platform's own app marketplace
SEO priority	Secondary — local "bakery near me" visibility, which modern hosted platforms handle adequately
Maintenance appetite	None — the owner is running a bakery, not a website

RECOMMENDATION: WIX

All five questions lean cleanly toward a hosted, no-code platform. Between Wix and Squarespace specifically, Wix's own app marketplace (Chapter 2) — genuine built-in ordering and booking apps — edges out Squarespace's own design-forward strength here, since a bakery's functional needs (online ordering) matter more than the more visually-driven advantage Squarespace offers.

Scenario 2 — A Freelance Photographer's Portfolio

QUESTION	ANSWER FOR THIS PROJECT
Portability cost	Low-to-moderate — personal branding, but not extreme stakes
Cost structure preference	Indifferent — either shape is acceptable
Real ceiling risk	Minimal — showcasing images well is the core, achievable need
SEO priority	Secondary — visual first impression matters more than search ranking for winning client trust
Maintenance appetite	None — wants to focus on photography, not technology

RECOMMENDATION: SQUARESPACE

This lands on the exact same conclusion Chapter 3's own exercise reached for a nearly identical scenario — but now arrived at through the full five-question framework rather than the editor-philosophy comparison alone, confirming the same answer two different ways. Squarespace's own design-forward templates (Chapter 3) directly serve this project's real priority: a professional-looking result without requiring design expertise.

Scenario 3 — A Growing SaaS Startup's Marketing Site

QUESTION	ANSWER FOR THIS PROJECT
Portability cost	High — a scaling company can't risk being permanently locked into a platform with no escape hatch
Cost structure preference	Flexibility over predictability — subscription tiers that scale with usage become expensive at real growth
Real ceiling risk	High — real chance of needing custom integrations (analytics, CRM, A/B testing) no page builder supports
SEO priority	Primary — organic search is often a core growth channel for SaaS marketing specifically
Maintenance appetite	Manageable — a growing company has, or will hire, real technical resources

RECOMMENDATION: WORDPRESS — LIKELY A CUSTOM THEME, NOT OFF-THE-SHELF

All five questions lean toward the self-hosted end of Chapter 1's spectrum. Given genuine growth plans and the real chance of needing custom functionality (Chapter 6's own ceiling material), this is a strong candidate for real custom theme and plugin development, exactly the territory *WordPress Intermediate/Advanced* covers — not just plain WordPress Fundamentals-level site ownership.

Why Dreamweaver Doesn't Win Any of These Three

AN HONEST, DELIBERATE OBSERVATION

None of these three scenarios recommends Dreamweaver — and that's consistent with Chapter 4's own honest framing, not an oversight. Dreamweaver's genuine relevance today is largely in maintaining existing legacy sites, not starting new ones; none of these three scenarios involves inheriting an existing Dreamweaver-built site, so the question never actually arises for any of them.

Chapter Attribution

PIECE	SOURCE CHAPTER
The overall spectrum framing	Chapter 1
Wix's own app marketplace (Scenario 1)	Chapter 2
Squarespace's own design-forward strength (Scenario 2)	Chapter 3
Why Dreamweaver doesn't apply here	Chapter 4
WordPress + custom development as a real escape hatch (Scenario 3)	Chapter 5
The five separated trade-off dimensions	Chapter 6
The five-question framework itself	Chapter 7

Honest Scope Note

WHAT THIS COURSE DELIBERATELY DOESN'T COVER

- No hands-on build-along for any specific tool — this course is comparative, not a tutorial, per Chapter 1's own stated scope
- No dedicated e-commerce deep dive for any platform — a genuinely large enough topic to warrant its own separate treatment if pursued later
- No specific pricing figures treated as durable facts — platform pricing changes too often for any number stated here to remain accurate for long

Hands-On Exercises

EXERCISE 1

Explain why the bakery scenario recommends Wix specifically over Squarespace, even though both are "fully hosted, fully no-code" platforms per Chapter 1's own table.

 [View solution](#)

EXERCISE 2

Explain why the SaaS startup scenario recommends WordPress Intermediate/Advanced-level custom development rather than plain WordPress Fundamentals-level site ownership.

 [View solution](#)**EXERCISE 3**

Explain why this chapter treats Dreamweaver's absence from all three recommendations as consistent with Chapter 4's own honest framing, rather than as evidence Dreamweaver is simply a worse tool overall.

 [View solution](#)**CHAPTER 8 QUICK REFERENCE — COURSE COMPLETE**

- Bakery → Wix (five questions lean cleanly hosted; app marketplace edges out Squarespace)
- Photographer's portfolio → Squarespace (design-forward strength directly serves the stated priority; confirms Chapter 3's own earlier conclusion via the full framework)
- Growing SaaS startup → WordPress, likely custom-developed (all five questions lean self-hosted; real ceiling risk is decisive)
- Dreamweaver's absence from all three is consistent with its real, legacy-maintenance-focused relevance today, not a ranking judgment
- This completes No-Code Site Builders — closing the Web Platforms subject's current roadmap alongside the completed WordPress track