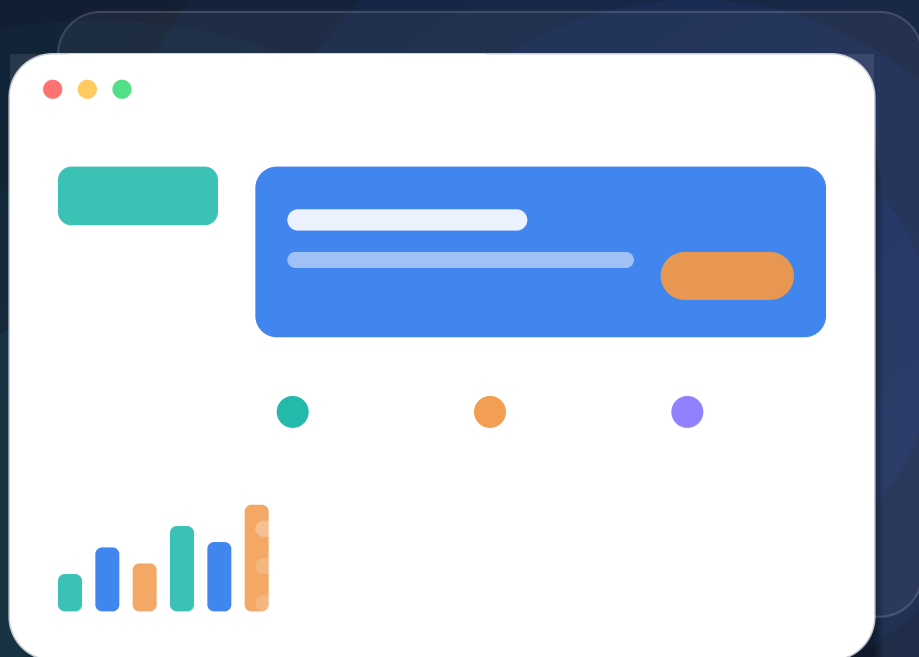


DESIGN THE WEB

A PRACTICAL DESIGN MANUAL



**A Practical Guide to
Modern Web Design**

PLAN • DESIGN • BUILD • LAUNCH

Design the Web

A Practical Guide to Modern Web Design

This guide is provided for general information. Tools, browsers and standards change over time, so check current documentation before relying on any specific detail. Only use images, fonts and code that you own or have permission to use.

Contents

- Introduction 4
- 1. How the Web Works 5
- 2. Planning Before You Design 8
- 3. Principles of Good Layout 11
- 4. Typography That Works 14
- 5. Colour, Imagery and Visual Style 17
- 6. Navigation and User Experience 20
- 7. Responsive and Mobile-First Design 23
- 8. Content, Copy and Search Visibility 26
- 9. Performance and Accessibility 29
- 10. Tools, Launch and Maintenance 32
- Appendix A: Pre-Launch Checklist 35
- Appendix B: Glossary 37

Introduction

A website is often the first thing a person sees of you, your business or your idea. Within a few seconds, visitors decide whether the site looks trustworthy, whether it is easy to use, and whether it is worth their time. Good web design is what makes those seconds count.

This book is a practical introduction to modern web design. It is written for beginners, small business owners, students and anyone who wants to understand how good websites are planned, designed and built. You do not need any prior design or coding experience.

Technical terms are explained when they first appear, and a glossary at the back covers the rest.

Web design is a blend of two skills: **deciding what a site should do and how it should feel**, and **making it work well on every screen**. This book covers both, in the order a real project unfolds.

How to use this book

- **New to web design?** Read the chapters in order. Each one builds on the one before it.
- **Working on a project right now?** Jump to the chapter that matches where you are: planning, layout, typography, colour, content, code, accessibility or launch.
- **Short on time?** Appendix A is a one-page pre-launch checklist you can print and work through.
- **Meeting a new term?** Check Appendix B, the glossary.

TIP

Design tools and browsers change quickly, but the principles in this book (clarity, hierarchy, consistency, accessibility and speed) have stayed the same for decades. Learn the principles first and the tools will be easy to pick up.

How the Web Works

You do not need to be an engineer to design websites, but you do need a clear picture of what a website actually is. This chapter gives you that picture in plain language.

Browsers, servers and addresses

When you type an address such as **example.com** into a browser, the browser asks a computer called a **server** for the files that make up that page. The server sends them back, and the browser assembles them into the page you see. The address is called a **URL**, and the name part of it is the **domain name**. A system called **DNS** works like a phone book, turning the domain name into the numeric address of the server.

The three building blocks

Almost every web page is built from three technologies, each with its own job:

- **HTML** (HyperText Markup Language) gives the page its **structure and content**: headings, paragraphs, images, links and buttons.
- **CSS** (Cascading Style Sheets) controls the **appearance**: colours, fonts, spacing and layout.
- **JavaScript** adds **behaviour**: menus that open, forms that check themselves, content that updates without reloading.

A helpful comparison is a house. HTML is the walls and rooms, CSS is the paint and furniture, and JavaScript is the electricity and plumbing.

A first look at HTML

HTML uses **tags** to mark up content. Here is a tiny but complete page:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport"
    content="width=device-width, initial-scale=1">
  <title>My First Page</title>
</head>
<body>
  <h1>Hello, web</h1>
  <p>This is my first page.</p>
</body>
</html>
```

Everything inside `<head>` is information about the page. Everything inside `<body>` is what visitors see.

Front end, back end, and hosting

- The **front end** is everything the visitor sees and uses in the browser. This is the focus of web design.
- The **back end** is the software on the server that stores data, handles logins, processes orders and so on.
- **Hosting** is the service that keeps your site's files on a server connected to the internet, so people can visit at any time.

Static sites, CMS sites and site builders

There are three common ways to make a site, and the right one depends on your needs and skills:

- **Hand-coded static sites** are plain HTML, CSS and JavaScript files. They are fast, cheap to host and give total control, but changing content means editing files.
- **Content management systems (CMS)** such as WordPress let you edit content through a friendly dashboard, and are suited to blogs, news sites and larger business sites.

- **Website builders** such as Wix, Squarespace and Webflow offer drag-and-drop design and hosting in one package, and are ideal for getting a good-looking site online quickly.

TIP

Whichever route you take, the design principles in the rest of this book apply equally. A well-designed site on a builder beats a badly designed hand-coded one every time.

Planning Before You Design

The biggest web design mistakes are made before anyone opens a design tool. A site that starts with a clear plan is easier to design, easier to build and far more likely to succeed. Spend your first hours here.

Step 1: Define the purpose

Write one sentence that completes this: *"This website exists to help [who] do [what]."* For example: "This website exists to help local families book a table at our restaurant." Every decision that follows should be tested against that sentence.

Step 2: Know your audience

Different visitors want different things. Describe two or three typical visitors in a few lines each, called **personas**. Include their goals, their level of technical comfort, and the device they are most likely to use. A site for busy parents on phones looks very different from a site for engineers on large monitors.

- What does this person want to achieve in a visit?
- What questions do they have before they are ready to act?
- What might make them hesitate or leave?

Step 3: Set measurable goals

Decide what success looks like and how you will know. Goals might be the number of enquiries received, newsletter sign-ups, products sold, or bookings made. A clear goal turns design from a matter of opinion into a matter of evidence.

Step 4: Look at your competitors and your favourites

Visit five sites in your field and five sites you simply admire. Note what works (clear navigation, strong photography, simple checkout) and what frustrates you. The aim is to learn, not to copy.

Step 5: Gather and organise your content

A site is only as good as what is on it. List every page you need, and for each one, collect the text, images and any documents. Many projects stall because the content arrives late, so start gathering it early.

Step 6: Map the site structure

A **sitemap** is a simple diagram of your pages and how they relate. A small business site often looks like this:

- **Home**
- **About** (story, team)
- **Services** or **Products** (with a page for each)
- **Contact** (form, address, map)
- **Blog** or **News** (optional)

Keep the structure as shallow as possible. Visitors should reach anything important within about three clicks.

Step 7: Sketch wireframes

A **wireframe** is a rough, black-and-white layout of a page that shows where things go without any colour, fonts or photos. It can be drawn on paper in minutes. Wireframes let you test the layout and the flow of the page before investing time in visual design.

TIP

Sketch the mobile version of each page first. The small screen forces you to decide what really matters, and the larger layouts become much easier.

Step 8: Set your budget and timeline

Be realistic about the cost of a domain name, hosting, premium themes or tools, photography, copywriting and ongoing maintenance. A site is not a one-off purchase; it needs care after launch.

Principles of Good Layout

Layout is how a page is organised: where things sit, how much space surrounds them, and in what order the eye travels. A good layout feels effortless. A poor one feels confusing, even when visitors cannot say why.

Visual hierarchy

Hierarchy tells visitors what is most important and guides them through the page. You create it with size, weight, colour, contrast and position. The biggest, boldest, highest-contrast element is seen first. A strong page has one clear starting point, then a clear second and third.

Try the **squint test**: squint at your page until the details blur. The elements you can still make out are your hierarchy. If the wrong things stand out, adjust.

White space

White space (also called negative space) is the empty area around and between elements. It is not wasted space. It gives content room to breathe, groups related items, and makes a page look calm and professional. Crowded pages feel cheap and tiring. When in doubt, add more space.

Alignment and consistency

Elements that line up look intentional. Choose a few alignment lines (for example a left edge for text and images) and stick to them. Use the same spacing, button style and heading style on every page, so that visitors learn how your site works once and can then rely on it.

Proximity and grouping

Things that belong together should sit close together; things that are unrelated should be further apart. A heading should be nearer to the paragraph it introduces than to the one above it. Grouping reduces the effort needed to understand a page.

Grids

A **grid** is an invisible set of columns that organises the page. Most websites use a 12-column grid because 12 divides neatly into halves, thirds, quarters and sixths. Grids keep layouts tidy and make it easy to rearrange content for different screen sizes.

Reading patterns

Studies of how people scan web pages show common patterns. On text-heavy pages, readers often scan in an **F shape**: across the top, then down the left side. On simpler pages with less text, they follow a **Z shape**. Put the most important things (your headline, main message and main button) where these paths pass.

Above the fold

The area visible without scrolling is often called **above the fold**. It should quickly answer three questions: *Where am I? What can I do here? Why should I care?* Do not cram everything in, but do not hide your main message below it either.

Common page sections

Most home pages are assembled from a familiar set of blocks:

- A **header** with the logo and main navigation.
- A **hero** section: a headline, a short supporting line, an image and one main call to action.
- A **benefits or services** section, usually three to six short items.
- **Social proof**: testimonials, client logos, ratings or case studies.
- A closing **call to action**.

- A **footer** with contact details, secondary links and legal pages.

TIP

Give each page one main job and one main button. A page that tries to do five things usually does none of them well.

Typography That Works

Roughly 95 percent of the web is text. Choosing and setting type well is therefore one of the most powerful things you can do to make a site look professional and be pleasant to read.

Choosing typefaces

A **typeface** (or font family) is a design such as Georgia or Helvetica. Keep your choices simple:

- Use **no more than two typefaces**: one for headings and one for body text, or just one family in several weights.
- Pair contrasting but compatible styles, for example a **serif** heading (with small strokes at the letter ends) with a **sans-serif** body, or the reverse.
- Pick fonts that suit the personality of the brand: friendly, formal, playful or technical.
- Choose fonts with several weights (regular, medium, bold) so you can create hierarchy without adding new families.

Free libraries such as **Google Fonts** offer hundreds of high-quality typefaces. Load only the weights you actually use, since every extra font file slows the page.

Size and readability

- Set body text at **16 pixels or larger**. Smaller text is hard to read, especially on phones.
- Use a **line height** of about 1.5 times the font size for body text, and slightly tighter for large headings.
- Keep lines to roughly **45 to 75 characters**. Lines that are too long tire the eye; too short feel choppy.

- Left-align body text. Avoid justified text on the web, which creates uneven gaps.
- Avoid long passages in capitals or in italics.

Building a type scale

A **type scale** is a set of sizes that follow a consistent ratio, so headings feel related. A simple scale for the web might be:

- Body text: **16 px**
- Small text and captions: **14 px**
- Subheading (h3): **20 to 24 px**
- Section heading (h2): **28 to 36 px**
- Page title (h1): **40 to 56 px**

Define the scale once and reuse it everywhere. Consistency matters far more than the exact numbers.

Contrast and colour of text

Text must stand out clearly from its background. Dark grey (such as #1B1F27) on white is easier on the eyes than pure black on pure white. Never place text over a busy photo without a dark overlay behind it. Chapter 5 explains how to measure contrast.

System fonts for speed

If speed matters most, use the fonts already installed on the visitor's device. A **system font stack** loads instantly and often looks very good:

```
font-family: system-ui, -apple-system, "Segoe UI",  
    Roboto, Helvetica, Arial, sans-serif;
```

TIP

If your text looks "off" and you cannot say why, check three things: the size is large enough, the line height is generous enough, and the lines are not too long.

Colour, Imagery and Visual Style

Colour sets the mood of a site, directs attention and communicates meaning. Used with restraint, it makes a design memorable. Used carelessly, it makes a page chaotic.

The basics of colour

- **Hue** is the colour itself (blue, red, green).
- **Saturation** is how intense or muted it is.
- **Lightness** is how light or dark it is.

On screens, colours are written as **hex codes** such as #1F6FEB, or as RGB or HSL values. A good tool for exploring combinations is a colour picker or palette generator, many of which are free online.

Build a simple palette

Most successful sites use a small, disciplined palette. A reliable recipe:

1. **One primary colour** that represents the brand, used for buttons, links and key highlights.
2. **One secondary or accent colour** used sparingly for emphasis.
3. **Neutrals**: a dark grey for text, a mid grey for borders and secondary text, and a very light grey or white for backgrounds.
4. **Status colours** for feedback: green for success, amber for warnings, red for errors.

A well-known guideline is the **60-30-10 rule**: about 60 percent dominant neutral, 30 percent secondary colour, and 10 percent accent.

The meaning of colour

Colours carry associations, though they vary between cultures. In many Western contexts, blue suggests trust and calm, green suggests growth and health, red suggests urgency or passion, and yellow suggests optimism. Use these as a guide, but test them with your real audience.

Contrast and accessibility

Roughly one in twelve men and one in two hundred women have some form of colour vision deficiency, and many more people have low vision or use screens in bright sunlight. To include everyone:

- Make sure text meets the WCAG minimum **contrast ratio: 4.5 to 1** for normal text and **3 to 1** for large text.
- Never use colour as the only way to convey information. Add an icon, label or underline as well (for example, show error messages in text, not just red borders).
- Check your pairs with a free contrast checker before you commit to them.

Dark mode

Many people prefer dark interfaces, and browsers can tell your site which one the visitor uses. If you support dark mode, define your colours as variables and swap them. Do not simply invert everything: use dark grey rather than pure black, and soften bright colours.

Photography and images

- Use **real, high-quality photographs** where you can. Generic stock images of smiling strangers weaken trust.
- Keep a consistent style across the site: similar lighting, colour treatment and framing.
- Always include **alt text**, a short description of each image for screen readers and for when an image fails to load.

- Crop images for the layout rather than stretching them.
- Only use images you own or have permission or a licence to use.

Icons, illustration and graphics

Use one icon set throughout so the line weights and styles match.

SVG graphics stay sharp at any size and are tiny in file size, which makes them ideal for logos, icons and simple illustrations.

TIP

A reliable way to look more professional: use fewer colours, more white space, and one consistent image style.

Navigation and User Experience

User experience (UX) is how easy and pleasant a site is to use. A beautiful site that people cannot navigate is a failure. This chapter covers the habits that make sites feel effortless.

Clear navigation

- Limit the main menu to **five to seven items**.
- Use **plain, familiar labels** such as "Services", "About" and "Contact" rather than clever ones.
- Put the logo at the top left and make it link to the home page.
- Show **where the visitor is** by highlighting the current page in the menu.
- Place the same navigation in the same place on every page.
- Consider **breadcrumbs** (Home > Shoes > Running) on sites with many levels.
- Add a **search box** if the site has a lot of content.

Mobile navigation

On small screens, menus usually collapse behind a button, often called a "hamburger" icon. Label it clearly, make the tap target large, and ensure the menu is easy to close. Keep the most important action, such as "Call" or "Book", visible rather than buried.

Calls to action

A **call to action** (CTA) is a button or link that tells visitors what to do next. Good ones are specific and visible:

- Use **action words**: "Get a quote", "Book a table", "Download the guide", instead of "Submit" or "Click here".
- Make the main button the **most visually prominent** item on the page.
- Offer only **one primary action** at a time; secondary actions should look quieter.
- Repeat the CTA at natural decision points down a long page.

Forms that people finish

Forms are where visitors become customers, so remove every obstacle:

- Ask only for what you truly need. Every extra field reduces completion.
- Put labels **above** the fields and keep them visible, not just as placeholder text that vanishes.
- Use the right input types (email, phone, date) so phones show the correct keyboard.
- Show errors **next to the field** in plain words, and explain how to fix them.
- Make the submit button's label say what will happen.
- Confirm success clearly after sending.

Feedback and states

Interactive elements should respond. Buttons and links need distinct **hover**, **focus**, **active** and **disabled** states, and slow actions should show a loading indicator. Silence leaves users wondering whether anything happened.

Building trust

- Show real contact details, including an address or phone number where appropriate.

- Include testimonials, reviews, client logos or case studies.
- Publish clear privacy, returns and terms pages.
- Use HTTPS (the padlock in the address bar) on every page.
- Keep content up to date, since an outdated date or broken page signals neglect.

Testing with real people

The quickest way to find problems is to watch someone use your site. Ask three to five people to complete a task (such as "find the opening hours" or "buy a gift card") and stay silent while they try. Note where they hesitate or get lost. You will learn more in half an hour than from weeks of debate.

TIP

Do not make visitors think. If a label, button or layout requires explanation, simplify it.

Responsive and Mobile-First Design

Visitors reach your site on phones, tablets, laptops and giant monitors, and in most markets phones account for more than half of all visits. A **responsive** site adapts its layout to fit any screen, so one site works everywhere.

Mobile-first thinking

Mobile-first means designing the small-screen version first and then enhancing it for larger screens. This forces focus on the essentials and tends to produce faster, simpler sites. It is much easier to add content for a big screen than to squeeze a desktop design onto a phone.

The viewport tag

Every responsive page needs this line in its head, which tells phones to use the real screen width:

```
<meta name="viewport"
      content="width=device-width, initial-scale=1">
```

Fluid layouts and flexible images

- Use **relative units** such as percentages, rem and fr instead of fixed pixel widths for layout.
- Make images flexible with `max-width: 100%; height: auto;` so they never overflow their container.
- Let text wrap naturally; avoid fixed-height boxes that clip content.

Media queries and breakpoints

A **media query** applies CSS only when a condition is true, usually when the screen is wider than a certain size. The sizes where the layout changes are called **breakpoints**. Common starting points are about 600, 900 and 1200 pixels, but let your content decide: widen the browser and add a breakpoint wherever the layout starts to look awkward.

```
/* Mobile first: one column by default */
.cards { display: grid; gap: 16px; }

/* Three columns on wider screens */
@media (min-width: 900px) {
  .cards { grid-template-columns: repeat(3, 1fr); }
}
```

Flexbox and Grid

Two modern CSS layout systems do most of the work:

- **Flexbox** arranges items in a single row or column, and is perfect for menus, toolbars and aligning things.
- **CSS Grid** arranges items in rows and columns together, and is ideal for page layouts and card galleries.

Designing for touch

- Make tap targets at least **44 by 44 pixels**, with space between them.
- Place key actions within comfortable thumb reach.
- Do not rely on hover effects, which do not exist on touch screens.
- Use input types that bring up the right keyboard.

Testing across devices

Use your browser's built-in **device mode** in the developer tools (usually opened with F12) to preview many screen sizes. Then test on at least one real phone, because emulators cannot reproduce everything, such as touch feel and real network speed.

TIP

If you can use your site comfortably with one thumb on a phone on a slow connection, you have probably done most things right.

Content, Copy and Search Visibility

Design and content cannot be separated. A beautiful layout filled with vague, wordy text will not persuade anyone. Likewise, even a great site achieves nothing if nobody can find it.

Writing for the web

People scan rather than read. Write for scanners:

- Lead with the **most important point** first.
- Use **short sentences and short paragraphs**, ideally one to three sentences.
- Break text up with **clear headings**, bullet lists and bold key phrases.
- Use plain language and speak directly to the reader ("you").
- Cut every word that does not earn its place.

Headlines that work

The main headline should state clearly what you offer and for whom, in plain words. Compare "Welcome to our website" with "Fresh bread, baked daily in Johannesburg". The second one tells the visitor exactly what they have found.

Structure with real headings

Use heading levels (h1 to h6) in order, as an outline of the page: one **h1** for the page title, **h2** for main sections, and **h3** for subsections. Do not choose headings for their size; use CSS to style them. Proper headings help screen-reader users and search engines.

Search engine optimisation (SEO) basics

SEO is the practice of helping search engines understand your pages so the right people find them. The foundations are simple:

- **Useful content.** Pages that answer real questions, written for people, rank best over time.
- **A clear page title.** The `<title>` tag appears in search results. Keep it under about 60 characters and include the page's main topic.
- **A meta description.** A one- or two-sentence summary (about 150 characters) that appears under the title in results. It does not boost rankings directly, but it influences clicks.
- **Descriptive URLs** such as `/services/web-design` rather than `/page?id=273`.
- **Alt text** on images.
- **Fast, mobile-friendly pages** (Chapters 7 and 9).
- **Internal links** between related pages, with descriptive link text instead of "click here".
- **Links from other reputable sites**, which are earned by creating content worth sharing.

Local visibility

If you serve a local area, claim and complete your free business profile on the major search and map services, keep your name, address and phone number identical everywhere, and encourage satisfied customers to leave reviews.

Social sharing

Add **Open Graph** tags so that links to your pages show a proper title, description and image when shared on social platforms and messaging apps.

Keeping content fresh

Review your pages at least twice a year. Update prices, hours and team details, remove outdated news, and fix broken links. Fresh content builds trust with visitors and with search engines.

TIP

Write your content before, or at least alongside, the design. Designing around real words produces far better results than placeholder text.

Performance and Accessibility

Two qualities separate professional websites from amateur ones, and neither is about looks. A professional site is **fast** and it is **usable by everyone**.

Why speed matters

Visitors are impatient. A page that takes more than a few seconds to appear loses many of them, and search engines favour fast sites. Speed is a feature, especially for people on slow connections or older phones.

Making pages faster

- **Optimise images.** Images are usually the heaviest part of a page. Resize them to the size they are displayed, compress them, and use modern formats such as **WebP** or **AVIF**.
- **Lazy-load images** below the fold, so they only download as the visitor scrolls toward them (`loading="lazy"`).
- **Limit fonts.** Use few families and weights, or use system fonts.
- **Cut unnecessary scripts.** Every plugin, tracker and widget adds weight. Remove what you do not need.
- **Minify** CSS and JavaScript, which strips out unneeded characters to shrink files.
- **Use caching and a CDN.** A **content delivery network** keeps copies of your files on servers around the world so they load from somewhere close to the visitor.
- **Choose good hosting.** Cheap, overcrowded hosting can make even a well-built site slow.

Measuring performance

Free tools such as **PageSpeed Insights** and the Lighthouse panel built into Chrome test your pages and suggest fixes. Pay attention to the **Core Web Vitals**, which measure how quickly the main content appears, how soon the page responds to input, and how stable the layout is while loading.

What accessibility means

Accessibility means designing so that people with disabilities can perceive, understand and use your site. That includes people who are blind, have low vision, are colour blind, are deaf or hard of hearing, have motor impairments, or have cognitive differences. It also helps everyone: people on phones in sunlight, with a broken trackpad, or with a temporary injury.

In many countries accessibility is also a legal expectation. The international standard is the **Web Content Accessibility Guidelines (WCAG)**, which are organised around four principles: content must be **perceivable, operable, understandable and robust**.

A practical accessibility checklist

- Give every meaningful image **alt text**; use empty alt text for purely decorative ones.
- Ensure sufficient **colour contrast** (see Chapter 5).
- Make everything usable with the **keyboard alone**: press Tab to move through the page and check that the focus is always visible and the order is logical.
- Use **semantic HTML**: real buttons for actions, real links for navigation, real headings for structure, and lists for lists.
- Label every **form field** and associate the label with it.
- Do not rely on colour alone to convey meaning.
- Provide **captions or transcripts** for video and audio.

- Avoid content that flashes, and let users pause moving or auto-playing content.
- Allow text to be enlarged without breaking the layout.
- Write meaningful link text, such as "Read the pricing guide" rather than "Click here".

Testing accessibility

Run an automated checker such as Lighthouse or WAVE, but remember that automated tools find only a portion of the issues. Also try the keyboard-only test, zoom the page to 200 percent, and, if possible, try a screen reader such as NVDA (Windows), VoiceOver (Mac and iPhone) or TalkBack (Android).

TIP

Accessibility is much cheaper to build in from the start than to repair later. Treat it as part of design quality, not an extra.

Tools, Launch and Maintenance

With a plan, a design and content in hand, it is time to build, test and publish. This final chapter covers the tools that make the work easier and the steps that take a site from a draft to a living website.

Design tools

- **Figma** is a free-to-start, browser-based tool widely used for wireframes, visual design and prototypes, and for collaborating with others.
- **Canva** is a simple option for graphics and social images.
- **Pen and paper** remain unbeatable for the first rough sketches.

Build tools

- A **code editor** such as Visual Studio Code is the standard for writing HTML, CSS and JavaScript.
- **Browser developer tools** (press F12) let you inspect any page, try CSS changes live and debug problems.
- **Version control** with Git records every change and lets you undo mistakes. Services such as GitHub store your code safely online.
- **Frameworks and libraries** such as Bootstrap or Tailwind CSS speed up styling; JavaScript frameworks such as React are used for complex, app-like sites. Beginners do not need them.

Choosing a domain name

- Keep it **short, memorable and easy to spell and say aloud**.
- Prefer **.com** or the appropriate country extension for your market (such as **.co.za** in South Africa).

- Avoid hyphens and numbers, which are easily confused.
- Check that the name is not already used or trademarked by someone else.
- Register it through a reputable registrar and turn on auto-renew so you do not lose it.

Pre-launch testing

Before you publish, work through the checklist in Appendix A. In particular, test every link and form, view each page on a phone and a desktop, check the site in at least two browsers, proofread all text out loud, and confirm that contact forms actually reach you.

Going live

1. **Choose hosting.** Pick a reputable host that suits your site type, with backups and support.
2. **Point your domain** at the host by updating its DNS settings, which can take a few hours to take effect.
3. **Upload or publish** the site.
4. **Turn on HTTPS** with an SSL certificate, which most hosts provide free.
5. **Set up analytics** to learn how visitors use the site.
6. **Submit your sitemap** to the major search engines.
7. **Test again on the live site**, since problems sometimes appear only after publishing.

Security basics

- Keep your CMS, themes and plugins **updated**.
- Use **strong, unique passwords** and two-factor authentication on hosting, domain and admin accounts.
- Limit the number of plugins and remove unused ones.
- Take **regular backups**, and test that you can restore them.

Learning from analytics

Analytics show how many people visit, where they come from, which pages they view, and where they leave. Return to the goals you set in Chapter 2 and check them regularly. If many visitors leave from one page, that page needs attention.

Maintenance and improvement

A website is never really finished. Plan a routine:

- **Monthly:** install updates, check backups, scan for broken links, review analytics.
- **Quarterly:** review content accuracy, test forms, and check speed and accessibility scores.
- **Yearly:** renew your domain and hosting, and reconsider the design and goals.

Keep learning

Web design rewards curiosity. Study sites you admire, read the free documentation on MDN Web Docs for HTML, CSS and JavaScript, practise by rebuilding pages you like, and keep a small portfolio of your own projects. Skill grows fastest when you build real things.

TIP

Launch early, then improve. A good, simple site that is live today is worth more than a perfect one that is still being polished next year.

Appendix A: Pre-Launch Checklist

Print this page and tick off each item before your site goes live.

1. **Purpose is clear.** Each page has one main job and one main call to action.
2. **Content is final.** All text is proofread and all images are chosen, with permission to use them.
3. **Navigation works.** Menu labels are clear and every link goes to the right place.
4. **Forms work.** Test every form and confirm that submissions arrive and thank-you messages show.
5. **Mobile layout is checked** on at least one real phone and one tablet.
6. **Browsers are checked** in at least two major browsers.
7. **Text is readable.** Body text is at least 16 px with comfortable line height.
8. **Contrast passes.** Text meets the 4.5 to 1 ratio.
9. **Images are optimised** in size and format, and have alt text.
10. **Keyboard test passed.** Everything can be reached and used with Tab and Enter, and focus is visible.
11. **Page titles and meta descriptions** are written for every page.
12. **Page speed is acceptable** on a mobile connection.
13. **Favicon and social sharing image** are set.
14. **HTTPS is on** and there are no mixed-content warnings.
15. **Legal pages are present:** privacy policy, terms and cookie notice where required.
16. **Analytics are installed** and goals defined.

- 17. **Backups are scheduled** and a custom 404 page exists.
- 18. **Contact details are correct** and consistent across the site.
- 19. **Sitemap is submitted** to search engines.
- 20. **Maintenance plan is set** and passwords are stored safely.

Appendix B: Glossary

Accessibility. Designing sites so that people with disabilities can use them.

Alt text. A short written description of an image, read by screen readers and shown if the image fails to load.

Above the fold. The part of a page visible without scrolling.

Breakpoint. A screen width at which the layout changes.

Breadcrumbs. A trail of links showing where a page sits in the site's structure.

CDN. Content delivery network: servers around the world that deliver your files from a location near the visitor.

CMS. Content management system: software such as WordPress for editing a site through a dashboard.

Call to action (CTA). A button or link that prompts the visitor to take a specific step.

CSS. Cascading Style Sheets: the language that controls how a page looks.

Domain name. The human-readable address of a website, such as example.com.

DNS. The system that turns domain names into the numeric addresses of servers.

Favicon. The small icon shown in a browser tab.

Flexbox. A CSS layout method for arranging items in a row or column.

Front end / back end. The visible part of a site in the browser, versus the server software behind it.

Grid. A set of columns and rows that organises a layout; also the name of a CSS layout system.

Hero. The large opening section at the top of a page, with a headline, image and main button.

Hex code. A six-character code, such as #1F6FEB, that defines a colour.

Hosting. A service that stores your website's files on a server so people can visit.

HTML. HyperText Markup Language: the language that gives a page its structure and content.

HTTPS. The secure version of the web protocol, shown by a padlock in the browser.

JavaScript. The programming language that adds interactivity to web pages.

Lazy loading. Delaying the download of images until the visitor scrolls near them.

Media query. CSS that applies only under certain conditions, such as a minimum screen width.

Meta description. A short summary of a page that may appear under its title in search results.

Minify. Removing unnecessary characters from code to make files smaller.

Mobile-first. Designing the small-screen version first and enhancing it for larger screens.

Responsive design. Design that adapts its layout to fit any screen size.

SEO. Search engine optimisation: helping search engines understand and rank your pages.

Semantic HTML. Using HTML elements for their proper meaning, such as real headings, lists and buttons.

Sitemap. A diagram or file listing a site's pages and how they connect.

SVG. Scalable Vector Graphics: sharp, lightweight images made from shapes rather than pixels.

Typeface / font. A design of lettering, such as Georgia; fonts are the individual styles and weights.

URL. The full address of a web page.

UX. User experience: how easy and pleasant a site is to use.

Viewport. The visible area of a web page in the browser window.

White space. The empty space around and between elements in a design.

Wireframe. A rough, plain layout sketch of a page showing structure without visual styling.

WCAG. Web Content Accessibility Guidelines: the international standard for accessible web content.