

Your website. Your next steps.

The complete launch and ownership handbook

By [Maverick](#) / [Interstate Rankers](#)

A practical reference for business owners who received a Next.js website and want to understand, host, update and look after it. No technical background assumed.

YOUR FILES	The website source, images and content
GITHUB	A private copy with a history of changes
VERCEL	The service that builds and hosts the site
YOUR DOMAIN	The web address customers visit
RESEND	Delivery of contact-form notifications

Includes copyable commands, launch checks, troubleshooting and official references. Examples use generic names and sample values. Keep passwords and real API keys out of public files.

Edition 1.0 / 26 October 2026

A NOTE FROM MAVERICK / INTERSTATE RANKERS

Your files. Your choice.

I wrote this guide because receiving your website files should give you a clear next step. At Interstate Rankers, we provide those files when you want to host independently, move on from working with us, or simply keep your own copy while we continue working together.

What we hand over

Our handover ZIP contains your website's source files and a README. It gives you the files to run the site on your own hosting setup; it does not transfer our service accounts.

Our paid accounts and your independent setup

While we manage your site, we use our own paid service accounts and cover those subscriptions without billing clients separately for them. If you choose to host independently, you will set up your own accounts and take responsibility for any plans, usage charges and renewals you choose.

Guidance does not end with the ZIP

Handing over the files does not end our guidance. We continue helping you understand the setup and work through questions. On your independent setup, we guide you through the changes rather than directly adding to or editing the website.

Built around the business

We build each website around the business's niche. The first few seconds should make it clear what the business does and which services it offers. That business-specific approach is what we aim to preserve when we hand the site over.

A free conversation about your next step

Whether you need a new website, help launching files you already have, ongoing maintenance, or advice about a problem with your current site, we offer a free consultation to discuss your situation and the next steps.

[Talk to Interstate Rankers - free consultation](#)

Choose your route

Read this page first. A website that is already live does not necessarily need a new deployment.

If Interstate Rankers currently manages your site

Our managed setup uses our own paid accounts. Those accounts stay with Interstate Rankers; the handover is your source files and README. If you only want a backup while we keep working together, save your ZIP and no hosting change is needed. For independent hosting, prepare accounts you control, confirm the domain and DNS arrangements with us, and coordinate the cutover only after the new deployment is tested. We continue providing guidance through your questions.

If you only received a ZIP

Save the original unchanged. Extract it into a working folder. Follow chapters 1-8 in order, then test the complete customer journey before switching a domain away from an existing site.

How to use the commands

Run one line at a time in the project folder. Replace sample paths, account names and email addresses before running a command. Lines beginning with # are comments. A command printed here is an instruction for you, not evidence that it has already run. Never paste command output containing secrets into public support posts.

Inside this handbook

- 01 Understand your files and ownership
- 02 Install the tools and open a terminal
- 03 Preview your website locally
- 04 First Git upload to GitHub
- 05 Import into Vercel and deploy
- 06 Configure contact-form email
- 07 Connect a domain without breaking email
- 08 Launch checks and search visibility
- 09 Update content with Git and previews
- 10 Useful commands and optional CLI deployment
- 11 Recover from a bad change
- 12 Troubleshoot Git and local setup
- 13 Troubleshoot hosting, DNS and enquiries
- 14 Keep the site healthy and hand it over
- 15 Official references

WEBSITE OWNERSHIP / 01

Understand what you received

This guide matches a Next.js delivery with Markdown blog articles and a server-side contact form. Other platforms require different instructions.

The folder map

your-website/	
app/	Pages and server routes
components/	Navigation and page sections
public/	Images, video, icons
content/blog/	Article text files
lib/	Business data and helpers
package.json	Software and script definitions
package-lock.json	Exact dependency versions
.gitignore	Files Git should leave out
.env.example	Template for private settings
README.md	Project-specific technical notes

Do not double-click a source file and expect a complete site. Next.js must install its dependencies and build or run the application. The contact endpoint needs a compatible server runtime; a static file host alone is not enough.

What you own versus what you subscribe to

Source files, photographs and content are separate from the hosting subscription and domain registration. Your contract determines rights to licensed images, fonts and other third-party assets. Keep copies of licence information supplied by your developer.

Ask for a record of the repository URL, Vercel project, production branch, email recipient, sending domain, DNS manager, analytics account and renewal responsibilities. Keep credentials in a password manager, not a public README.

CHECKPOINT

You can identify package.json in the project root, have a backup of the original delivery and know who owns the accounts.

WEBSITE OWNERSHIP / 02

Install tools. Open the right folder.

You need Git, Node.js with npm, a text editor and accounts for GitHub and Vercel. Resend is needed for this example contact form.

Install from the official sources

Get Git from git-scm.com, Node.js from nodejs.org, and an editor such as Visual Studio Code from code.visualstudio.com. Choose a currently supported Node.js LTS release compatible with the supplied Next.js version. Next.js 16 requires at least Node.js 20.9; the minimum is not a recommendation to use an unsupported release.

Open a terminal in the project folder

On Windows, open the extracted folder in File Explorer, right-click the background and choose Open in Terminal if available. In VS Code, open that folder and choose Terminal > New Terminal. On macOS, use Terminal and `cd` into the folder. Paths with spaces need quotation marks.

```
git --version
node --version
npm --version
cd "C:\path\to\your-website"
git status
```

Version numbers confirm the tools are installed. For a fresh ZIP, 'not a git repository' is expected until you initialise Git. If it already is a repository, inspect its branch and remote instead of running the new-project setup blindly.

A few words translated

- Terminal: a window where you type instructions.
- Repository: a project and its change history.
- Commit: a saved checkpoint in that history.
- Push: upload your commits to the remote repository.
- Deployment: an online built version of your website.
- Environment variable: a setting supplied to the running application.

CHECKPOINT

The version commands work after reopening the terminal. You are inside the folder containing `package.json`.

WEBSITE OWNERSHIP / 03

Preview your website locally

Local preview lets you check edits without changing what customers see. Installing dependencies is different from publishing a site.

Install the delivered dependency versions

```
npm ci
```

npm ci uses the supplied lockfile and expects it to match package.json. It replaces the dependency installation in node_modules; do not store personal files there. If it fails because the lockfile is missing or mismatched, ask the developer to supply a consistent package rather than immediately changing package versions.

Create local email settings, if needed

Only when .env.local does not already exist, copy the template. For Windows PowerShell:

```
Copy-Item .env.example .env.local
```

For macOS/Linux: cp .env.example .env.local. Open .env.local in your editor and fill in the three email values described in chapter 6. Preserve an existing settings file; do not overwrite it with sample values. Restart the development server after changing local environment settings.

Start the site

```
npm run dev
```

Open the Local URL printed in the terminal, commonly http://localhost:3000. If this project defines a different port, use that printed URL. Keep the terminal open. Press Ctrl+C to stop the server. The address only works while the server is running; localhost means your own computer.

Check the production build separately

```
npm run build  
npm run start
```

These commands create and serve a production build locally. Stop a server already using the same port first. If the project forces HTTPS in production, ask the developer for a supported local production preview instead of bypassing browser warnings.

CHECKPOINT

You can navigate the site locally. This has not uploaded your files or changed your public domain.

WEBSITE OWNERSHIP / 04

Upload a fresh ZIP with Git

Use this route only for a new local folder without existing Git history. If your developer supplied a repository, inspect that existing setup first.

On GitHub, create a new private repository. Leave README, licence and .gitignore initialisation unchecked so the remote is empty. Keep the .gitignore from your website delivery. Replace all sample identity and repository values below; user.name and user.email here are scoped to this repository.

```
git init -b main
git config user.name "Your Name"
git config user.email "your-github-email@example.com"
git remote add origin https://github.com/YOUR-ACCOUNT/YOUR-REPOSITORY.git
git status
git add .
git diff --cached --stat
git diff --cached
git commit -m "Add website files"
git push -u origin main
```

git add stages changes; it does not publish them. git diff --cached lets you inspect exactly what will be committed. Confirm there are no real API keys, .env.local, .env, node_modules or .next files. If anything private appears, stop before committing and fix the ignore rules or ask for help.

Complete the sign-in prompt from Git Credential Manager if offered. A GitHub password is not accepted as a Git-over-HTTPS password. Depending on your setup, use credential-manager browser authentication, an appropriately scoped token or an SSH key. Never paste a token into the repository URL or guide.

If there is already a remote

```
git remote -v
git branch --show-current
git status
```

If origin is correct, skip the remote-add command. If it is wrong, confirm the intended destination before changing it with git remote set-url origin URL. Do not delete the .git folder or force-push as a shortcut.

CHECKPOINT

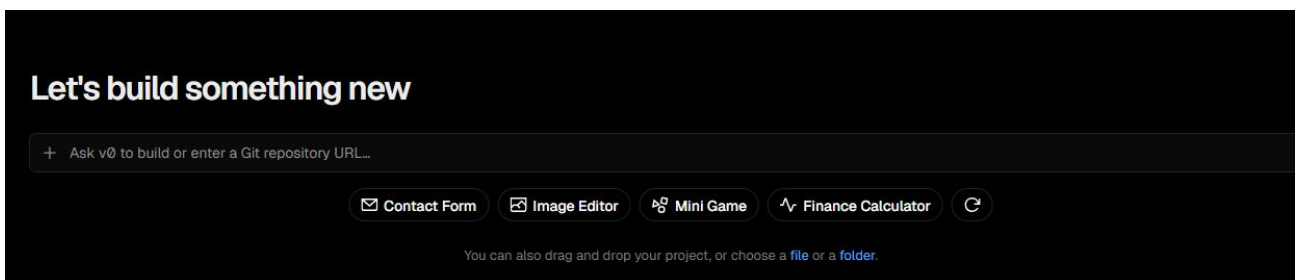
The GitHub repository is Private and shows your website folders and package.json on the intended branch.

WEBSITE OWNERSHIP / 05

Import into Vercel

Use your business-owned Vercel account or team. Check current plan terms: Hobby is limited to personal, non-commercial use.

- Choose Add New > Project and connect GitHub if needed. Authorise access to the intended repository.
- Under Import Git Repository, find the private repository and choose Import. If missing, check which GitHub account and repositories the integration can access.
- Confirm Next.js as the framework. Set Root Directory to the folder containing package.json. A nested folder from an extracted ZIP is a common source of mistakes.
- Keep the automatically detected install, build and output settings unless the supplied README specifies an override.
- Add the three email environment variables from chapter 6 before deploying. Include Production in their target selection.
- Choose Deploy. Wait until the build completes, then open the generated .vercel.app address.



Real Vercel entry screen, captured October 26, 2026. The Git repository import area appears below this part of the page. Other configuration examples in this handbook are teaching instructions, not live dashboard captures.

What a successful first deployment proves

It proves that Vercel can build and serve this version. It does not prove that your custom domain, enquiry email, analytics or account ownership is configured correctly. Test those independently. A preview URL is not the same thing as the final domain.

CHECKPOINT

The temporary Vercel URL loads your homepage, contact page, navigation, images and blog articles.

WEBSITE OWNERSHIP / 06

Make contact-form email work

This project sends contact notifications through Resend. It does not create a business mailbox or store enquiries in a database.

Verify the sender before using it

In Resend, add a domain you control. Add the exact DNS records Resend provides in the active DNS manager and wait for verification. A sending subdomain can keep website notifications separate from your main mailbox configuration. Preserve the existing records used by your normal business email.

Three settings, three jobs

```
RESEND_API_KEY=re_YOUR_PRIVATE_KEY  
CONTACT_TO_EMAIL=leads@yourbusiness.com  
CONTACT_FROM_EMAIL=Business <website@yourbusiness.com>
```

The first value is a private key with permission to send email. The second is an existing receiving inbox; this example supports multiple recipients separated by commas. The third is the sender and must use your verified sending domain. All shown values are examples.

In a local `.env.local` file, quote the full sender value, for example `CONTACT_FROM_EMAIL="Business <website@yourbusiness.com>"`. In the Vercel dashboard value field, enter the sender without those surrounding file-format quotes. Do not rename these variables or add `NEXT_PUBLIC_` to the private key.

Production, Preview and Development are separate

Set the variables for the environments that need them. Production needs real delivery settings. Preview can use a test recipient so test submissions do not confuse the sales team. Redeploy after dashboard changes; restart `npm run dev` after local changes. Existing deployed code does not automatically receive new environment values.

Verify delivery, not just the success message

Submit one clearly labelled test. Confirm the notification arrives in the correct inbox, inspect spam and check Reply addresses the test sender. Check Resend delivery status if it is missing. The supplied endpoint rate-limits repeated submissions, so pause testing if you hit its limit.

CHECKPOINT

A test enquiry reaches the intended inbox and your team knows where to respond.

WEBSITE OWNERSHIP / 07

Connect a domain safely

A domain is an address. DNS is the set of records that tells browsers and email services where that address leads.

- Open the correct Vercel project > Settings > Domains. Add your root domain, such as yourbusiness.com.
- Add www.yourbusiness.com if desired. Choose one main address and configure the other to redirect using the options Vercel offers.
- Find the active DNS manager. This may be your registrar, Cloudflare or another provider; it is not always where you purchased the domain.
- Copy the Type, Name/Host and Value/Target from your own Vercel project. Do not copy a fixed IP or target from an unrelated tutorial.
- Record the previous website-related values before changing them. Preserve email MX, SPF, DKIM and other service records.
- Wait for Vercel verification and HTTPS provisioning, then test root and www on another network as well.

Match fields rather than guessing

A, CNAME and TXT are different record types. The root host may be shown as @, blank or the domain itself depending on your provider. Some providers append your domain automatically. Inspect the resulting full name to avoid entering the domain twice.

Why a change is not instant everywhere

Resolvers can cache earlier DNS answers until their TTL expires. The time depends on the previous TTL, provider and caches. Use the status shown by Vercel and your DNS manager rather than repeatedly making different changes. Nameserver changes affect all DNS services, so they need a complete migration plan.

Before replacing an existing live site

Check the temporary deployment first. Confirm the enquiry recipient, important URLs and redirects. Keep the old hosting available during the transition until the new site and email work. A change of domain may also require canonical URLs, sitemap URLs, schema, tracking and Resend configuration to be updated.

CHECKPOINT

The main domain opens securely, its alternate redirects correctly, and normal business email still works.

Launch checks and discoverability

Test as a customer would. A green deployment badge is not a complete launch check.

The customer journey

- Open the homepage, service pages, about page, contact page and at least one blog post. Test navigation and calls to action.
- Check a phone-sized screen. Look for clipped text, overlapping buttons, missing media and difficult form fields.
- Tap the phone number and email links. Check that they point to the business details you intended.
- Submit a test enquiry and verify receipt and reply-to behaviour.
- Check both root and www domain addresses. Important old URLs should redirect to their matching new pages rather than all going to the homepage.

Search setup after the correct domain is live

Open `/robots.txt` and `/sitemap.xml` on your domain. Confirm intended pages are accessible, not accidentally noindexed, and use the correct canonical domain. Verify domain ownership in Google Search Console and submit the live sitemap. A sitemap helps discovery; Google does not guarantee indexing or rankings.

If the domain changed, ask the developer to review canonical URLs, structured data, sitemap links and business references across the project. Updating DNS alone does not update hardcoded URLs inside source files.

Analytics and consent

Confirm the analytics property and any advertising pixels belong to the correct business. Avoid leaving a previous business or test tracker in place. Your developer should confirm the tracking and consent configuration matches the requirements for your audience and tools.

Keep a launch record

Record the production deployment URL, launch date, repository branch, account owners, recipient inbox and renewal responsibilities. Do not put API keys in that public record. Check service usage and billing alerts in the relevant dashboards.

CHECKPOINT

A real visitor can find the site, navigate it, contact you and receive a response.

Publish updates with confidence

For a connected GitHub project, a branch gives you a separate place to make changes before they reach the production site.

In a repository already connected to GitHub: use a feature branch to preview edits. Start with a clean working tree. 'git pull --ff-only' stops instead of merging if histories have diverged. Replace main if your developer uses a different production branch.

```
git status
git switch main
git pull --ff-only
git switch -c update-contact-details
# Edit your files, then check them locally
npm run build
git diff
git add path/to/changed-file
git diff --cached
git commit -m "Update contact details"
git push -u origin update-contact-details
```

Replace path/to/changed-file with a real edited file path; repeat git add for each intended file. Open a pull request on GitHub, check the Vercel preview, then merge after reviewing it. A production deployment follows changes to Vercel's configured production branch. For your next update, use a new branch name.

What to check in the pull request

Review the changed files, spelling, business details, image paths and mobile layout. Check the build result and the preview URL. A Preview deployment may be protected or use different email settings. Test with the intended environment, then merge when ready. Changes on the production branch can trigger a live deployment automatically.

Editing an article or replacing an image

In this package, blog posts live in content/blog as Markdown files. Copy an existing article as a starting point; preserve valid front matter and unique filenames. Media goes under public. The visible URL for public/images/photo.webp is /images/photo.webp. Match case exactly. A new post appears after a successful rebuild; editing the local ZIP alone does nothing online.

CHECKPOINT

Your changes are reviewed in a preview before they reach the production branch. Keep the last good version available.

Useful commands at a glance

These commands have different jobs. Read the current folder, branch and deployment target before acting.

Inspect without publishing

```
git status
git branch --show-current
git remote -v
git log --oneline -5
git diff
git diff --cached
```

These show your working files, current branch, remote destinations, recent checkpoints, unstaged changes and staged changes. Run `git diff --cached` before every commit if you are unsure what will be included.

Build and choose a different local port

```
npm run build
npm run dev -- --port 3001
```

The first checks whether the site builds. The second starts a development server on another port when the project uses Next.js scripts. Open the address it prints. These commands are separate workflows; stop an existing local server if you need its terminal.

Optional Vercel CLI route

```
npx vercel login
npx vercel link
npx vercel deploy
# Only after checking the preview and production settings:
npx vercel deploy --prod
```

The CLI is an alternative to Git-connected deployments, not a required extra step. Verify the team and project when linking, and read the prompts before confirming. The plain deploy command produces a preview by default; `--prod` publishes to production. Environment variables must already be configured for the intended target. Keep the source in Git even if you choose CLI deployment.

Use one team workflow consistently so a CLI deployment is not immediately replaced by an unexpected Git deployment. Do not install packages or accept command prompts you do not understand just to clear an error.

CHECKPOINT

A local build is not a deployment. A Git commit is not a push. A preview is not necessarily production.

Recover from a bad change

First identify whether the problem is in source code, deployment settings, DNS or email. Undoing the wrong layer can create a second problem.

If the live code deployment is broken

Open the Vercel project and find the last known working deployment. Use the available rollback controls for your plan and deployment type. Verify the target before making it production. Test the affected pages and the form afterwards. Vercel rollback is not a backup of your email inbox or external services.

Fix the repository as well

A hosting rollback does not remove the bad source commit. The next deployment can reintroduce it. Ask the developer to correct or revert the source in a reviewed branch. For an ordinary non-merge commit, the pattern below creates a new commit that undoes the earlier one without erasing shared history:

```
git log --oneline -5
git switch -c fix-recent-change
git revert COMMIT_ID
npm run build
git push -u origin fix-recent-change
```

Replace COMMIT_ID with the confirmed commit. Reverting a merge commit requires additional choices; ask your developer rather than guessing. If Git reports conflicts, stop and review them. Do not use `reset --hard` or `push --force` as a beginner recovery shortcut.

If the problem is DNS or credentials

A code rollback does not restore DNS records or rotate an exposed API key. Use your saved DNS record history for domain recovery and the provider dashboard for credential replacement. If a secret reached GitHub, revoke or rotate it immediately; deleting it from the latest file does not remove it from earlier commits.

CHECKPOINT

The live site works again and the repository explains the fix so the next deployment remains correct.

Fix Git and local setup errors

Use the exact message to choose a fix. Do not run a random sequence of commands until the error disappears.

Why does git say it is not recognised?

Install Git from git-scm.com, close and reopen the terminal, then run `git --version`. If it still fails, check the installer's PATH option. Do the same for Node.js if node or npm is missing. Do not change security settings or paste unrelated scripts to fix it.

How do I fix remote origin already exists?

Run `git remote -v` first. If origin already points to the correct repository, skip `git remote add origin`. If the address is wrong and you have confirmed the correct destination, use `git remote set-url origin` followed by that repository's HTTPS URL. Changing a remote does not move repository ownership.

Why was my Git push rejected?

Check the exact error. Authentication failures need account/repository access checked. A non-fast-forward rejection means the remote has commits you do not have. Do not force-push over them. For a first upload to a repository containing a README, clone that repository into a separate folder and copy your website into it while preserving its `.git` folder. For an existing collaborative project, ask your developer to reconcile the histories.

Why does localhost fail with HTTPS or an SSL error?

A normal `npm run dev` server uses HTTP. Open the exact Local URL printed in the terminal, including its port. HTTPS on an HTTP-only port will fail. If the browser saved an earlier redirect, use a fresh local port for development. Do not bypass certificate warnings for a public website.

How do I fix EADDRINUSE or port 3000 already in use?

Another process is using that port. If it is your earlier development server, stop it in its terminal with `Ctrl+C`. Otherwise choose another port: `npm run dev -- --port 3001`. Open the new URL printed in the terminal. Do not stop an unknown process just to free a port.

npm ci reports package-lock mismatch

Confirm you are in the project root, using the intended Node/npm version and the supplied `package-lock.json`. Ask the developer to reconcile the lockfile with `package.json`. `npm install` can change dependency resolution, so do not silently replace the lockfile just to make a client handover run.

PowerShell blocks npm.ps1

On Windows, try `npm.cmd` in place of `npm`, for example `npm.cmd ci` or `npm.cmd run dev`, if the `npm PowerShell` wrapper is blocked by policy. This uses the installed command wrapper; it does not require disabling your system execution policy. On a managed computer, follow your administrator's rules.

Fix hosting, DNS and email issues

Check one layer at a time: build, deployment, domain, then email. Capture the first relevant error with private values removed.

Can I upload a Next.js ZIP straight to ordinary hosting?

This delivery needs a platform that supports its Next.js runtime and server-side contact endpoint. Shared hosting that only serves static files cannot run that endpoint. Static export requires code changes and a replacement for server features; renaming files or uploading the source to `public_html` will not make it work.

Why is my contact form not sending emails?

Check `RESEND_API_KEY`, `CONTACT_TO_EMAIL` and `CONTACT_FROM_EMAIL` in the right Vercel environment. The sender must use a verified Resend domain, and the key must have sending permission. Redeploy after changes. Look at Vercel logs, Resend delivery status and the receiving spam folder. This example's endpoint also limits repeated test submissions.

Why does my domain still show the old website?

Check that you changed records in the active DNS manager and copied the values from the correct Vercel project. Existing TTL values and caches can delay the change. Check Vercel's domain status, both root and `www` addresses, and another network. Do not delete business email records or repeatedly switch nameservers while waiting.

Why did my Vercel deployment fail?

Open the failed deployment's build logs and find the first meaningful error. Confirm the root contains `package.json`, required variables exist, and the Node version is compatible. Run `npm ci` and `npm run build` locally to reproduce it. A missing file may be a filename-case mismatch that Windows tolerated but the deployment does not.

Images work locally but disappear online

Confirm the image is committed, its path begins at the correct public URL and filename case matches exactly. Windows may accept `Photo.webp` and `photo.webp` as the same file where the deployment does not. Also check file size, supported image formats and any remote-image allowlist if the source is an external URL.

A new article does not appear

Confirm the Markdown filename, valid front matter, publication settings and image references. Check that the changes reached the production branch and produced a successful deployment. Browser cache and build failure are different problems; identify which version is actually serving the page.

Keep ownership simple

A good handover gives you access, a repeatable update path and a way to recover, not just a ZIP file.

What a complete handover record includes

- Repository URL, owner, default branch and production branch.
- Vercel team, project, domain list, deployment workflow and billing owner.
- Domain registrar, active DNS manager and renewal owner.
- Resend account, verified sending domain and notification recipient.
- Names of required environment variables, with secret values stored separately.
- Analytics and advertising account owners, plus any required consent configuration.
- Asset licences and any external service dependencies.
- Original delivery, latest source backup and a tested recovery contact.

A practical maintenance rhythm

After each change, test the changed page, phone layout, navigation and enquiry form. Periodically review billing and usage, domain renewal, delivery failures and broken links. Keep a current repository backup. Dependency and framework upgrades should be tested on a separate branch with a production build; routine content editing does not require upgrading everything.

When to ask for a developer

Get help with build errors you cannot explain, conflicting Git history, domain migrations, contact-form changes, dependency upgrades, complex layouts and business details repeated across schema and content. For frequent non-technical content updates, discuss adding a CMS rather than assuming this delivery already includes one.

What ongoing costs may remain

Domain registration renews separately from hosting. Hosting, email delivery, mailbox services, paid images and monitoring may each have their own billing. Review current plan terms and limits before subscribing; this guide does not promise permanently free hosting or a fixed provider price.

CHECKPOINT

You know how to access the site, publish a reviewed change, receive enquiries and ask for help with an exact, useful error report.

Official references

Technical guidance was checked against the linked official documentation on 26 October 2026. Interfaces, supported runtimes, limits and plan terms can change.

GitHub: add locally hosted code

<https://docs.github.com/en/migrations/importing-source-code/using-the-command-line-to-import-source-code/adding-locally-hosted-code-to-github>

GitHub: authentication

<https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/about-authentication-to-github>

GitHub: manage remotes

<https://docs.github.com/en/get-started/git-basics/managing-remote-repositories>

Next.js: installation

<https://nextjs.org/docs/app/getting-started/installation>

Vercel: GitHub deployments

<https://vercel.com/docs/git/vercel-for-github>

Vercel: environment variables

<https://vercel.com/docs/environment-variables>

Vercel: custom domains

<https://vercel.com/docs/domains/working-with-domains/add-a-domain>

Vercel: Hobby plan terms

<https://vercel.com/docs/plans/hobby>

Vercel: rollback

<https://vercel.com/docs/instant-rollback>

Vercel: CLI deployment

<https://vercel.com/docs/cli/deploy>

Resend: verified domains

<https://resend.com/docs/dashboard/domains/introduction>

Google: sitemaps

<https://developers.google.com/search/docs/crawling-indexing/sitemaps/overview>

Project-specific facts were checked against the supplied Next.js handover README, package.json, .env.example and contact API source. No client credentials or private account data are included. Commands use example identities and destinations.