

Runbook: the docs site (launch and after)

🕒 8 min read · Updated 12 October 2026 · Verified against [4adc454](#) on 2026-09-26

Launching docs.gurbanisoul.com and keeping it honest: one-time setup, the staging walkthrough, DNS cut-over, production smoke, rollback, and the routine after launch.

<https://docs.gurbanisoul.com/process/runbooks/docs-site/>

The wiki (docs-site/ over docs/, [ADR-0012](#)) deploys through `deploy-docs.yml` with the product’s posture: CI builds, deploys, smokes by commit, promotes, rolls back; the Vercel project’s own git deployments are off. This runbook is the launch checklist and the routine that follows it.

1. One-time setup (owner)

The Vercel project, the domain, the staging alias and the two secrets — steps 1–5 of [the deploy runbook’s wiki section](#). Until they exist, every push to integration still runs the full docs job (the required check), and `deploy-staging` fails clearly at preflight.

Then apply the ruleset that makes the docs job required:

```
REPO=Algorythmos-AI/sggs-platform bash scripts/gh/apply_rulesets.sh --dry-run # prints the change
REPO=Algorythmos-AI/sggs-platform bash scripts/gh/apply_rulesets.sh
```

The pin bot’s token (DOCS_BOT_TOKEN)

docs-pins opens its pull request with a fine-grained personal access token, so the required checks run on it (a pull request opened with the workflow’s own token would trigger none). At <https://github.com/settings/personal-access-tokens/new>:

Field	Value
Resource owner	Algorythmos-AI — not your personal account (it cannot be changed later)
Repository access	Only select repositories → sggs-platform
Repository permissions	Contents and Pull requests : Read and write (nothing else)
Expiration	90 days, with a renewal reminder

If the organization requires approval, approve it under Organization settings → Personal access tokens → Pending requests. Then `gh secret set DOCS_BOT_TOKEN --repo Algorythmos-AI/sggs-platform` (a hidden prompt; never paste a token into chat

or a file).

Before it writes anything, the workflow checks the token with a dry-run push. A token that authenticates but cannot push — a personal resource owner, pending approval, read-only Contents — does not fail silently: the bump arrives as the issue *docs: the sibling docs changed — bump the pins*, which names the cause. An expired or removed token does the same.

2. Staging walkthrough (before the first promotion)

Most of this walkthrough is automated now. After every staging deploy `deploy-docs` runs the **live suite** (`docs-site/e2e-live/live.spec.ts`) against the staging alias with the real API: the status strip, the search simulator, the Ang explorer on Angs 1, 712 and 1256, a line read from `/api/ang/1` verifying as `VERIFIED_EXACT`, the API console, ੬K search, axe on five pages as deployed, and no Content-Security-Policy errors. Run it by hand with `make docs-live URL=https://docs.gurbanisoul.com` (add `VERCEL_BYPASS=...` for the staging alias). What stays human is taste: one visual pass on a new release.

Open `https://sggs-docs-staging.vercel.app` logged in to Vercel. Run the visual pass the build cannot judge — the same list the repository's QA script shoots (`cd docs-site && node scripts/visual-qa.mjs <url>` writes light, dark and phone screenshots of every section under `docs-site/qa-shots/`):

Check	Where
The landing, the live status strip shows production's version and health	/
A poster walkthrough: Next, arrow keys, a step chip, Full size, Escape returns focus	/architecture/request-lifecycle/
The Ang explorer on Angs 1, 712, 1256; the English toggle; the outage message with the network off	/data/line-record/
The search simulator with <code>sat nam, waheguru, mercy</code> , a Gurmukhi query; the tier lights on poster 05	/architecture/search-waterfall/
The verify playground with a line from memory; the rung lights on poster 07	/search/verification-engine/
The API console: an Ang, a search, copy as curl	/api/
A pinned page's canonical banner and edit link	/data/architecture/database-schema/, /ios/nitnem/spec/
Hover-cards and a quiz; the two cited lines stack Gurmukhi, transliteration, citation	/scripture/what-sggs-is/
A learning path's ticks survive a reload	/learning-paths/platform-engineer/
Search (੬K) finds a technical term and a scripture term	any page
Dark mode on every section; phone width with no horizontal scroll	every section
The 404 page; a page's Open Graph card (<code>/og/<id>.png</code>)	/nope/, /og/index.png

Anything wrong is a normal pull request into `integration`; the site redeploys to staging on merge.

3. Production

1. Cloudflare: `CNAME docs` → Vercel's recommended target, DNS-only. Vercel: the domain is attached to `sggs-docs` and shows *Valid Configuration*.
2. Merge the next release PR `integration` → `main` (a merge commit, as always). `deploy-docs`'s `deploy-production` job records the deployment the domain serves now (the rollback target), builds, deploys unaliased, smokes by commit, promotes to `docs.gurbanisoul.com`, smokes again — and rolls back if the public smoke fails. Every failure opens an issue that says whether production changed.

A project's first Vercel deployment became production. On 2026-09-26 the docs project's first deployment — a *preview* from `deploy-staging` — was recorded as production, and `docs.gurbanisoul.com` attached to it, so the domain served an integration build until the 1.3.10 release. `deploy-staging` now refuses to run while the project has no production deployment (`scripts/ci/vercel_api.py` has `-production`) and fails if its own deployment is not a preview. For a new Vercel project, the first deployment must come from `deploy-production`.

3. Prove it:

```
python3 scripts/ci/docs_smoke.py https://docs.gurbanisoul.com -
-commit <sha>
```

The smoke checks the landing, a Mermaid render, the search index, `/api/health` and `/api/meta` through the rewrite, the poster the architecture page links as “open full size” and an Open Graph card, and that `<meta name="sggs-docs-commit">` equals the commit. Nothing may redirect: a redirect on `/api` is how the widgets broke before launch (`trailingSlash`).

4. Owner walkthrough of section 2 on the public domain, once.

4. Rollback

The job rolls back on its own when the public smoke fails **after** promotion, to the deployment the domain served before the run (read from Vercel by the domain, never “the newest production deployment”: a build deployed with `--skip-domain` whose smoke failed is also a READY production deployment, but it never served anyone). A failure before promotion changes nothing and says so in its issue. By hand: `python3 scripts/ci/vercel_api.py live docs.gurbanisoul.com` names the serving deployment; Vercel → `sggs-docs` → Deployments → the previous good deployment → *Instant Rollback*, or `vercel rollback <previous-url> --scope <team>` with the docs

project's token. After a rollback Vercel stops assigning the domain automatically; the next `deploy-production` run's `vercel promote` restores it. The wiki carries no state, so a rollback costs nothing but the newer pages.

If the PDFs fail a build. `build-pdfs` names the page that could not be rendered, and `check-pdf` names the file and the promise it broke (most seriously: Gurmukhi set outside Sant Lipi, which means the font did not load). Fix the page or the stylesheet; never deploy around the gate. To ship without PDFs while that is investigated, remove `DOCS_PDF: "1"` from the three build steps of `deploy-docs.yml`: the build then makes none, the menu shows no *Download PDF*, and the build stops announcing `pdf1`.

Switching the page menu off. If the page menu (copy, view as Markdown, the assistant links, print) ever misbehaves, it can be removed without a code change: set `DOCS_PAGE_ACTIONS=0` in the docs project's build environment and deploy again. The build then leaves the menu out of every page and stops announcing `actions1`, so the live checks do not look for it (`docs-site/plugins/features.mjs`, [ADR-0013](#)).

Visual baselines

The docs check compares a picture of the top of six kinds of page (wide and phone, light and dark: 24 pictures in `docs-site/e2e-visual/__screenshots__`) with what the pull request's build draws. The sidebar, “On this page”, the dates under a title and live widgets are masked, so a new page or a re-verified stamp does not trip it; a change to the layout, the type or one of the six pages' opening content does.

When it fails:

1. Open the run, download the **visual-baselines** artifact. `__screenshots__` holds what this build drew; `test-results/` holds expected, actual and diff for each picture that changed.
2. If the change is a mistake, fix it. If it is meant, replace the files in `docs-site/e2e-visual/__screenshots__` with the artifact's and commit them in the same pull request, saying in the description what changed to the eye.

Never commit a baseline made on your own machine: fonts differ between systems, and the comparison runs only in the pinned Playwright container (`PW_IMAGE` in `deploy-docs.yml`; raise it together with the `playwright` version in `docs-site/package.json`, and refresh the baselines in that pull request).

5. After launch: the routine

When	What
every pull request	the docs check (required on integration and main): gates, build, links, budgets, CSP, e2e, axe on every page, the visual baseline, Lighthouse on five pages (all twenty nightly and on main) — about nine minutes
a dataset bump lands	nothing to do: the docs job installs the pinned database and re-verifies the cited lines
every 15 minutes	uptime probes the wiki's landing page and <code>/api/health</code> through its rewrite; issue <i>uptime: docs site probe failing</i>

When	What
every 6 hours	docs-watch : the domain serves the last deployed commit, the smoke and the live suite pass; issue <i>docs-watch: production wiki check failing</i> (after a manual rollback it stays open until the next release — expected)
every night	the docs job re-runs on integration ; issue <i>docs: nightly wiki build failing</i> while red
Mondays	docs-links checks every external link (issue <i>docs: broken external links</i>); security scans the history; docs-pins opens <i>docs(wiki): bump the sibling docs pins</i> when a sibling changed a published file (or issue <i>docs: the sibling docs changed — bump the pins</i> until <code>DOCS_BOT_TOKEN</code> exists); docs-freshness rewrites issue <i>docs: wiki pages to re-verify</i>
a page is in <i>wiki pages to re-verify</i>	re-read it against the code it cites, fix what drifted, move its verified stamp; the issue closes itself when every page is fresh
a scholar review is due	make review-pack writes one PDF — Scripture 101, the glossary’s scripture and script sections, the contributors — with a sign-off sheet
a new page	frontmatter, links, the scripture rule and the fallback rule are gated; a poster or a widget follows <u>the contributing guides</u>

6. Sibling README links

Done (2026-09-26): both sibling READMEs open with a line pointing at the wiki’s published copy of their docs (sggs-data#11¹, gurbani-soul-ios#19²).

7. Open review items

- The scholar review of **Scripture 101**, the glossary’s scripture terms and the contributors chronology. **make review-pack** builds the PDF with a sign-off sheet.
- Resolved: sggs-data’s `database-schema.md` now states the database’s counts (4,527 compositions, `max(comp_id)` 5,376), and `pipeline/tests/test_docs_facts.py` there holds them to the database.

Links

1. <https://github.com/Algorythmos-AI/sggs-data/pull/11>
2. <https://github.com/Algorythmos-AI/gurbani-soul-ios/pull/19>