Project Brief: olivermann.studio

> **About this document.** This is the brief I used to build olivermann.studio with Claude. I'm sharing it as an example from my Lunch and Learn, *Building a Simple Website with AI* (Innovation UBC, October 9, 2026). Use it as a model for your own: keep the structure, then swap in your goals, tools, pages and content. For this public version, I removed my revision notes and personal content.
>
> It is written as instructions for an AI builder, which is why it is so specific.

Prepared for	Oliver Mann
Purpose	Rebuild a personal portfolio site (communications, photography and a blog)
Handoff to	Claude, for the build
Status	Light launch (v1), designed to be extended with more content over time

---

1. Project summary

Oliver is a UBC communications strategist and freelance portrait and event photographer. His previous portfolio site was taken down because it was expensive to maintain (WordPress hosting) relative to its traffic. He has purchased a new domain, **olivermann.studio**, and wants a replacement site that:

- Costs effectively **$0 per month** to run beyond the domain itself
- Is **minimal, editorial and clean**, continuing the visual language of the old site, refined
- Covers three purposes: a **communications portfolio**, a **photography portfolio** and a **blog** for case studies and links to published work
- Launches as a **light v1 fast**, with clear room to add content (more posts, more photos) over time without a redesign
- Can be updated by Oliver **without needing to code or run a local development environment**

2. Tech stack and hosting

LAYER	CHOICE	WHY
Site framework	**Astro** (current major version, content layer API)	A static site generator that builds fast, lightweight pages. It optimizes images at build time, which matters because photos are most of the page weight. Content is written as Markdown files with front matter, so adding a blog post means adding one file. Free, actively maintained, no vendor lock-in.
Hosting	**GitHub Pages**	Free for public repositories, with automatic HTTPS. One account to manage. Soft limits (about 1 GB site size, 100 GB bandwidth per month) are far beyond what this site needs. **The repository is public**, so nothing confidential goes in it.
Build and deploy	**GitHub Actions**	Rebuilds and republishes the site automatically on every change to the `main` branch, using the official Astro workflow.
Domain registrar	**Porkbun**	Inexpensive, with straightforward DNS management.
Email	**hello@olivermann.studio** through Porkbun's free email forwarding	Forwards to Oliver's existing inbox at no cost.
Contact method	**mailto: link only**, no form	Keeps the site 100% static, with nothing to break.
Analytics	**None for v1**	Out of scope. A free, privacy-friendly option can be added later without changing the architecture.
Image hosting	**Inside the repository**, optimized by Astro at build time	At about 20 to 40 curated photos, the whole site fits comfortably within GitHub Pages' size guidance. **Because the repository is public, photos are committed at web resolution only (about 2000 px on the long edge, about 85% JPEG quality), never as full-resolution originals.**
Fonts	**Self-hosted** with the `@fontsource` package (Inter only)	No third-party font requests, so it is faster and more private.
Lightbox	**One shared component**	Reused on the Photography page and for screenshots in blog posts, so there is only one piece of gallery code to maintain.

DNS setup (Porkbun to GitHub Pages)

1. In Porkbun's DNS settings, **first delete the default parked-domain records.** DNS will not resolve correctly until they are gone.
2. Add **A records** for the apex domain pointing to GitHub Pages: `185.199.108.153`, `185.199.109.153`, `185.199.110.153` and `185.199.111.153`.
3. Optionally add **AAAA records** for IPv6: `2606:50c0:8000::153`, `2606:50c0:8001::153`, `2606:50c0:8002::153` and `2606:50c0:8003::153`.
4. Add a **CNAME record for** `www` pointing to `<github-username>.github.io`, so www redirects to the main domain.
5. In the repository's Pages settings, set the custom domain. Because the site deploys through GitHub Actions, also commit a `CNAME` file containing exactly `olivermann.studio` to the `public/` folder so it lands in every build.
6. Turn on "Enforce HTTPS" once DNS has propagated (this can take up to 24 hours).

Check these IP addresses against GitHub's current documentation before you use them (verified as of July 2026).

---

3. Publishing workflow

Oliver is not a developer and does not want to run git commands or a local development environment. To add a new blog post:

1. Oliver sends Claude rough notes, bullet points or links.
2. Claude drafts the finished Markdown file, with correct front matter.
3. Oliver creates the post's folder and file using **GitHub's web editor**: Add file, Create new file, then type the path `src/content/blog/my-post-slug/index.md` in the filename field. Typing a `/` in the filename creates a folder. He pastes the content and commits to `main`.
4. If the post has images, he uploads them into the same post folder (Add file, Upload files) and commits. Two commits, one for the file and one for the images, is fine.
5. The GitHub Action rebuilds and republishes the site automatically.

To add, remove, reorder or recaption **photos**, Oliver uploads the image to `src/assets/photos/` and edits one data file, `src/data/photos.yaml`, in the web editor.

There is no CMS or admin interface for v1. The Markdown and front matter format must be simple enough that pasting a new file in via GitHub's website "just works". The repository README must document both workflows above, plus the YAML gotchas listed in section 8.

4. Site map and global design system

Navigation: Home / Communications (one combined page), Photography, Resume, Blog. Contact is not a nav item. It lives only in the footer.

Design tokens

- **Typography:** Inter everywhere. Headlines are bold or extra-bold with tight letter-spacing, body text is regular, and small UI labels are medium weight, uppercase, with wide tracking. No serif anywhere.
- **Colour palette:** strictly black and white. White background (`#FFFFFF`), near-black text (`#111111`), muted grey for secondary text (`#6b6b6b`), light grey for image placeholders (`#ECECEC`). No accent colour. Links and active states are black with underlines or bottom borders. Light mode only.
- **Logo:** a simple text wordmark, "Oliver Mann", in bold Inter. No icon.
- **Photography:** shown in natural colour as shot, with no filter applied site-wide.
- **Motion:** subtle only, such as gentle fade-ins on scroll and hover states. No parallax and no auto-playing carousels. **All animations must respect `prefers-reduced-motion`.**
- **Responsiveness:** mobile-first. Every layout must degrade gracefully to a single column on small screens.

Reusable layout components

1. **Flexible section block:** 1, 2 or 3 columns of cards or content.
2. **Featured block ("1 big + stacked"):** one large featured item with smaller items stacked beside it. It must look intentional with anywhere from 1 to 4 items.
3. **Filterable card grid:** a row of filter tabs above a responsive grid. Used for storytelling posts and the Photography page.
4. **Lightbox:** click-to-enlarge with previous and next navigation. Used on the Photography page and for screenshots in blog posts.

---

5. Page: Home / Communications

A single combined page. Sections, top to bottom:

- **Hero:** two elements only. The name, and a typed rotating line ("I am a communications strategist, marketer, photographer, and so on"). The typing animation must respect `prefers-reduced-motion`.
- **Featured case studies:** uses the featured block. Items are pulled automatically from the blog collection, using the `featured` field. The four newest featured posts show, with the newest in the big slot and the next three in the smaller slots. A "View more" link goes to the Blog page, pre-filtered. Launch scope: 1 to 2 case studies.

- **Storytelling:** a filterable card grid (All, Writing, Video, Podcast) showing the latest 6 posts, generated automatically from the blog collection. If a post has an `externalLink`, its card links straight out to the published piece and no page is generated on this site. Otherwise it links to the hosted post.
- **Research and Indigenous engagement:** the same layout as the featured block, with a photo on every item. Items are blog entries with the `research` type.
- **Speaking and leadership:** a card grid showing the latest 3 entries. Items are blog entries with the `speaking` type.

Email marketing and social media are intentionally not separate sections. They are folded into the case studies.

---

6. Page: Photography

A single gallery page, with no intro text.

- **Filter tabs:** All, Portraits, Events.
- **Order:** by date taken, newest first, using a `date` field on each photo.
- **Layout:** a masonry grid with large images and fewer per row than a typical grid.
- **Interaction:** clicking a photo opens the shared lightbox, with previous and next arrows through the currently filtered set. Event photos also show a quiet "View full album" link to an external album.
- **No self-hosted full galleries.** The site is a curated teaser. A quiet line near the gallery invites people to get in touch to see more from a shoot.
- **Target volume:** about 20 to 40 curated photos at launch.
- **Alt text:** every photo needs descriptive alt text, covering the subject and context, not just the filename.

Photo data model

Image files live in `src/assets/photos/`. One data file, `src/data/photos.yaml`, drives the gallery:

```
- src: "portrait-01.jpg"
  alt: "Model in a patterned blue jacket photographed against the sea and sky"
  category: portrait          # portrait | event
  date: 2017-05-01
- src: "cohort-day-feb-2026.jpg"
  alt: "Attendees networking in a bright lobby at a cohort day event"
  category: event
  date: 2026-02-04
  caption: "Cohort Day, February 2026"          # optional
  flickrAlbum: "https://www.flickr.com/..."     # optional, event photos only
```

The build must validate this file, so a typo in `category` or a missing `alt` fails the build with a clear error instead of silently breaking the page.

---

7. Page: Resume

A standalone, web-only page (no downloadable PDF for v1). It has three parts: **Experience** (reverse chronological, with a few achievement bullets per role), **Education**, and **Skills**. Content comes from Oliver's current resume.

Skills are shown as grouped pill tags (for example Strategy and Leadership, Storytelling and Content, Digital and Analytics), with no percentage bars.

---

8. Page: Blog

Blog index

A reverse-chronological list of all published posts, all types mixed. Each row has a thumbnail, tag, title, date and a short excerpt. Posts with an `externalLink` link directly out and are marked with a small arrow.

Filter tabs: All, Case Studies, Writing, Video, Podcast, Research, Speaking. Filters support URL hash links (for example `/blog#case-studies`), so "View more" links on the Home page land pre-filtered.

Post layout

A single column with a capped reading width, built from one flexible template used for every post type:

1. Hero image (or a video embed)
2. Large headline
3. Short subhead
4. Byline and date
5. Body copy, with paragraphs, bold text, inline images and embeds
6. Screenshots, shown as thumbnails that open in the shared lightbox. Attach the lightbox with a small script that targets images in the post body, so posts never need component syntax.

No comments and no email signup for v1.

Front matter format

Each post lives at `src/content/blog/<slug>/index.md`, with its images in the same folder. The folder name becomes the URL (kebab-case, lowercase, no spaces).

```
---
title: "How I Built My Portfolio Website With AI in a Weekend"   # always in double quotes
date: 2026-07-01          # YYYY-MM-DD, no quotes
types: [case-study]       # one or more types (listed below)
draft: false              # true = ignored at build time
role: "Speaker"           # optional, shown in the byline (used on speaking posts)
skillsTags: ["Email Marketing", "Copywriting", "Analytics"]
heroImage: "./hero.jpg"   # relative path, image sits in the same folder
heroImageAlt: "Descriptive alt text"   # required whenever heroImage is set
heroVideoId: ""           # optional YouTube ID, used instead of heroImage
excerpt: "One-sentence summary."       # always in double quotes
featured: true            # appears in the Home page featured block
externalLink: ""          # if set, no page is generated; the card links out
---
```

The available types are `case-study`, `writing`, `video`, `podcast`, `research`, `speaking` and `indigenous-engagement`. A post can have more than one. The home page features the four newest posts with `featured: true`: the newest takes the big slot and the next three are stacked, so there is no ranking number to maintain. The schema must be validated, so typos fail loudly. `heroImageAlt` is required whenever `heroImage` is set, and only one of `heroImage` and `heroVideoId` may be used.

YAML gotchas to document in the README (Oliver pastes files in through GitHub's web editor):

- Always put `title` and `excerpt` in double quotes. Unquoted colons or apostrophes break the front matter.
- Dates must be exactly `YYYY-MM-DD`, unquoted.
- Folder names must be kebab-case, lowercase, with no spaces.
- If a build fails, the Actions tab shows which line caused it, and the previous version of the site stays live in the meantime.

---

9. Footer (every page)

- **Contact:** a mailto link (no contact form)
- **Social:** LinkedIn and Instagram
- **Copyright:** © Oliver Mann and the current year

---

10. Explicitly out of scope for v1

These were deliberately deferred, not forgotten:

- Photo galleries hosted on the site itself (event photos link out to external albums instead)
- A downloadable PDF resume
- Blog comments or an email signup
- Site analytics
- Any CMS or admin editing interface
- Dark mode

---

11. Recommended build process

Build in phases. This keeps rework cheap if the visual direction needs adjusting, and lets content arrive gradually.

1. **Design review (mockups only).** Build static HTML and CSS mockups of each page, one at a time, using placeholder content where real content isn't ready. No repository, scaffolding or deployment yet. Review and approve each page before moving on.
2. **Full build.** Scaffold the real Astro project, set up the repository and the deploy workflow, and connect the domain. Build every page as reusable components, including the content schema, the photo data model and the README publishing guide.
3. **Content population.** Add real content page by page as it becomes ready.
4. **QA against the definition of done** below.

---

12. Definition of done

- [ ] All pages (Home, Photography, Resume, Blog) built and responsive on mobile, tablet and desktop
- [ ] Site deployed on GitHub Pages, resolving at the custom domain with HTTPS enforced (and www redirecting to the main domain)
- [ ] Footer present and consistent, with a working mailto link and social links
- [ ] At least one blog post published and linked from the Home page
- [ ] Photography page live with filters, lightbox and the data file working
- [ ] Basic SEO in place: page titles, meta descriptions, a social sharing image, favicon, sitemap and robots.txt
- [ ] A custom 404 page that matches the design and links back to Home
- [ ] Animations respect `prefers-reduced-motion`
- [ ] All committed images are web-resolution exports
- [ ] The README documents the full publishing workflow, complete enough that Oliver can publish through GitHub's web editor without further help