

Website Framework

Front page:

Progressive Disclosure

Scannable hubs

- Show small, high-signal surface area first leading to depth behind clear intentional entry points

Intro sentence (wayfinding):

- **Role wedge:** “Job Title”
- **Domain:** “Job Domain”
- **Scale/proof:** 1–2 safe-to-share outcomes (availability, risk reduction, time saved, compliance posture & not sensitive details)

Placeholder: Job title + domain

*ROLE GATE: Are you a Recruiter, Hiring Manager, Peer/Guest? (With a welcome intro)
Selection -> opens a new page DEDICATED as a Secondary primary home page tailored to specifics. Page transitions and shows the first “artifact” of utmost relevance (recruiter = resume in the middle, hiring manager = work story and projects, peer = personal hobbies and stuff).
Each page shows relatively the same content in different orders and tonality, wording and structure.*

Resume is one click away in home page

Contact is one click away in home page

Each category page has an above-the-fold “TL;DR” that explains what’s inside and what a visitor can do next (download, view a project, email).

<https://www.semrush.com/blog/above-the-fold/>

The home page never forces recall; it offers recognition cues (“Resume”, “Projects”, “Writing”, “Certifications”, “Skills” etc.).

Top Level Nav (5 MAX):

- Work & Experience
 - Inside work 2 modes: Experience timeline (fast skim) and Project case studies (deep dive)
- Projects
- Writing
- Skills
- About

Utilities Top level:

- Resume (download + web view)
- Contact (button)

Above the fold (must be scannable in an F-pattern):

- Title: Name + role wedge + location preference (e.g., “Open to/ on-site”)
- 1 sentence positioning (one-line impact statement)
- 2 proof bullets (safe metrics or outcomes)
- 2 utilities: Resume + Contact (private inquires, which could extend to walkthrough)
- Github repo of the website/ For now we will just have our github profile
- Disclosure panel: “What im not sharing publicly” (Security risk)

Below the fold:

- The category tiles (“Work”, “Projects”, “Writing”, “Certifications”, “Skills”, “About”) with 1-line “what you’ll get” summaries
- “Featured” row: 2–3 projects (even if 2 are placeholders now, label them clearly as “In progress”)
- Latest writing teaser (1–3)
- Certifications snapshot (logos optional; text is safer for accessibility and load time)

This is explicitly about **visual hierarchy** and guiding attention in the intended order.

For each CATEGORY page, “rabbit hole” mechanic:

Each category page should implement progressive disclosure:

- Start with a “TL;DR panel” (what’s here, who it’s for, and how long it takes).
- Then show **Level 1 entries** (cards or list rows).
- Each entry expands into **Level 2 depth** (accordion/expand, or click-through detail page).
Progressive disclosure is a known pattern for making complex systems easier to learn without hiding depth.

For work & projects, core sections include the following:

- Context + problem statement
- Constraints (time, compliance, org context—sanitized)
- Architecture diagram (sanitized)
- Key decisions + tradeoffs
- Implementation highlights (patterns, not secrets)
- Results + what you’d do next

Design Token: need to lock in the color, type, spacing, radii, and shadows. Afterwards reused as variables for code when implementing in existing/new designs

Visual layout

Trust + Clarity themes > “stereotypical aesthetic”.

Requirements:

- plenty of whitespace
- strong typographic rhythm
- subtle gridlines/borders (low contrast)
- one accent color used sparingly for actions and highlights

For normal text, minimum contrast is commonly treated as 4.5:1 for WCAG AA (and higher for AAA). Use this as a hard constraint during palette selection.

Use a contrast checker as part of your workflow (this is non-negotiable if you want “professional outcome”)

One accent color: start with a psychological calming blue and then let neutral colors do most of the work.

Color “roles” defined to prevent random color usage and keeping the website coherent:

- **bg** (page background) => ?
- **surface** (cards/sections) => ?
- **text** (primary) => ?
- **muted** (secondary) => ?
- **Border** => ?
- **accent + accentHover** => ?
- **focusRing** (very important for keyboard usability) => ?
- optional: **success/warn/error** => ?

Ensure typography system is controlled by the criteria:

50-75 character target limit per line for body text

Headings are used as structural navigation

Body text set to 16-18px equivalent, line height set to around 1.5-1.6 depending on font

Long form content width is set with a constrain using ch so natural targets measure 65-80ch max

Pairing: one sans for prose + one mono for code snippets (usage of mono is used sparingly)

Utilize a 8px OR 4/8 rhythm for spacing decisions and Material guidance using 8dp spacing concepts and aligned measurements to maintain balance and rhythm

Need to set:

Global container width rules:

Breakpoints (mobile first):

One border radius set:

One shadow system or none.

Content Inventory (What we got to show):

-

-

Content Audit (What is good so far, missing and risky):

Good:

Missing:

RISKY:

Content types:

- **Experience entries** (job/company/timeframe + impact bullets + skills/tags)
- **Projects** (case-study style, even if placeholders)
- **Writing/posts**
- **Skills**
- **Certifications**
- **Static pages** (About, Contact, Security/Privacy stance)

Tech stack modeling

Astro

A pragmatic collections layout:

- **projects/** (case studies)
- **posts/** (blog)
- **books/**

- `certs/`
- optionally `experience/` (or keep experience as data if you prefer)

Each collection should have frontmatter fields like:

- `title, summary, date`
- `tags`
- `featured` (boolean)
- `draft` (boolean)
- `links` (repo/demo)
- `confidentiality` (enum: public / partial / private)

This sets you up to generate consistent cards, feeds, and indexes without rewriting templates later

ASTRO ISLANDS:

Focus on search bar within projects/posts (client side), a theme toggle (for light vs dark)

Everything else is static

Accessibility baseline: aim for WCAG AA behavior (practically)

Key requirements to bake into templates:

- **Headings are properly nested** (one H1 per page; do not skip heading levels).
- **Skip link / bypass blocks** so keyboard and screen reader users can jump to main content.
- **Focus order makes sense** (tab order preserves meaning).
- **Contrast meets minimums** for text and key UI elements.
- Use semantic structure / landmarks so structure conveyed visually is also programmatic (this is core WCAG intent).

Performance: design with Core Web Vitals in mind

Core Web Vitals are user-experience metrics for load, responsiveness, and visual stability (LCP, INP, CLS). Keep them in mind early because they're hard to "patch in" later if the design relies on heavy JS or layout shifts.

Portfolio-specific performance rules:

- default to static HTML; hydrate only necessary components (Astro islands)

- avoid layout jumps: reserve space for images/diagrams
- optimize images instead of shipping raw screenshots

This aligns with Astro's strengths and with web performance guidance.