**Atelier Penha — Website Development Plan**

**Project Overview**

Website Name: Atelier Penha  
  
Objective: Build a multi-page, portfolio-style website matching the structure and interactive, animated experience of unseen.co. The site will keep the same page architecture (home, projects, project detail pages, about, contact, legal pages, etc.) and replicate its motion-driven UX while retaining unique branding for Atelier Penha.

**Approach & Key Principle**

Primary approach: First build all pages and populate them with real content and images (static HTML structure and semantic markup). After page content is completed and approved, implement animations, transitions, and advanced styles to match the feel of unseen.co. This staged approach avoids rework and ensures animations are applied consistently across finalized content.

**Sitemap / Pages (match unseen.co structure)**

* - Home / Landing
* - Projects / Works (index)
* - Project Detail (one template used for all project pages)
* - About
* - Contact

**Deliverables**

1. Fully functional static pages with responsive HTML structure and placeholder content/images.  
2. A single reusable Project Detail template that will be used for all projects.  
3. Animation & interaction layer implemented after final content: smooth scroll, scroll-triggered reveals, hover micro-interactions, project scrolling effects, and any WebGL/canvas assets used by unseen.co.  
4. CSS architecture (modular, e.g., BEM or utility-first such as Tailwind) and JS modules for interactions.  
**Technical Stack Recommendations**

- HTML5, semantic markup  
- CSS: Tailwind CSS or modular SCSS (choose based on team familiarity)  
- JavaScript: ES6+, module bundler (Vite or Webpack)  
- Animation libraries: GSAP (ScrollTrigger), Locomotive Scroll or Smooth Scroll alternatives, three.js or custom WebGL for canvas effects where required  
- Optional framework: React/Vite if you want component reusability and easier state handling; otherwise a high-performance static build is fine.  
**Animation Strategy (after pages completed)**

1. Audit each page to identify animation needs and interaction points.  
2. Create an animation spec sheet with timing, easing, triggers, and fallbacks.  
3. Implement animations progressively: global scroll behavior, then per-section reveals, then micro-interactions (buttons, links, hover states).  
4. For heavy visuals (WebGL/canvas): implement optimized low- and high-resolution fallbacks and ensure graceful degradation on low-power devices.  
5. Performance budget: keep main-thread work low, use will-change carefully, and avoid layout thrashing.

**Content & Asset Checklist**

- Final copy for all pages (headlines, body text, metadata)  
- Project images (high-res + web-optimized versions)  
- Logos and brand assets (SVG preferred)  
- Fonts (license confirmation)  
- Any videos or audio assets  
- SEO metadata and alt text for images

**Acceptance Criteria**

- All pages built and content in place.  
- Animations match approved animation spec and run smoothly on target devices.  
- Lighthouse performance and accessibility thresholds met or documented exceptions.

Notes: As requested, development will proceed by building full pages with content first; animations and styling will be the second pass.