Create a PRD with the attached file for this professional service website for software development

TETRIX Product Requirements Document (PRD) provides a clear roadmap for designing and building a professional services website for a SaaS company. It defines goals, user needs, functional and non-functional requirements, technical constraints, and delivery milestones to guide the UI/UX and frontend development teams.

Overview

TETRIX is a Software-as-a-Service (SaaS) provider specializing in enterprise solutions. The website will serve as a marketing and lead-generation platform, featuring a gradient-driven landing page, detailed service descriptions, partner showcases, and a comprehensive contact hub. The site will be built with Astro (Islands architecture) leveraging Tailwind CSS and Lucide React icons, with stock imagery from Picsum.photos[1].

Goals and Objectives

- Drive qualified leads via clear CTAs ("Get in Touch," "View Works") on the landing page.
- Establish TETRIX brand identity with a centralized logo, consistent color palette, and gradient backgrounds.
- Showcase services, case studies, and technology partners to build credibility.
- Provide multiple contact touchpoints: phone, email, address, unified contact form, live chat.
- Ensure mobile-first responsiveness, fast load times, and SEO best practices.

Stakeholders

- Product Manager: Defines business priorities and acceptance criteria.
- UI/UX Designer: Crafts wireframes, component library, and visual identity.
- Frontend Developer: Implements Astro pages, Tailwind styles, and client-side components.
- Marketing Team: Supplies content, case studies, and partner logos.
- DevOps/Hosting: Deploys static assets to Netlify, Vercel, or equivalent.

User Personas

Startup Founder

- Needs concise overview of services and partner credibility.

- Expects clear CTAs and easy contact.

Technical Lead

- Seeks detailed solution architecture and case studies.

- Values performance metrics and partner technologies.

Support Seeker

- Requires multiple contact methods and quick response.

- Uses mobile devices for on-the-go inquiries.

Use Cases

- Visitor lands on `/` and engages with hero CTAs.

- Technical Lead navigates to `/solutions` for case studies and pricing.

- Prospective client submits contact form on `/contact`.

- Support Seeker taps phone number on `/contact-information-phone` for immediate call.

Functional Requirements

1. Navigation & Layout

- Global header with logo (top-left), main nav (Home, About, Solutions, Services, Contact), and CTA button (Get in Touch)[1].

- Left sidebar or sticky header variant on desktop, collapsible on mobile.

- 12-column grid with max-width 1200px, responsive padding (24px mobile, 48px desktop) [1].

2. Pages & Sections

- **Home (`/`)**: Gradient hero, subheading, partner logos grid (Framer, AWS, NVIDIA), dual CTAs.

- **About (`/about`)**: Company timeline, team profiles, mission and values.

- **Solutions (`/solutions`)**: Card-based filterable grid, case studies, comparison tables.

- **Services (`/services`)**: Detailed catalog with booking/inquiry forms, FAQ accordion.

- **Contact Hub (`/contact`)**: Universal contact form, live chat widget, method selector.

- **Contact Detail Routes**:

  - `/contact-information-phone`: Click-to-call phone list, business hours, callback form.

  - `/email`: Email directory, multi-step form with validation.

  - `/address`: Interactive map (50% viewport), directions sidebar, transport options.

3. Client-Side Components

- Use `"use client"` directive for interactive elements (menu toggle, forms, accordions) [1].

- Icons via `lucide-react` for nav toggles, contact methods, and social links.

- Image components configured with `remotePatterns` for `picsum.photos`.

Non-Functional Requirements

- **Performance**: Lighthouse score ≥ 90 on mobile and desktop.

- **Accessibility**: WCAG 2.1 AA compliance for contrast, keyboard navigation, ARIA labels.

- **SEO**: Semantic HTML, meta tags per page, structured data for services.

- **Security**: HTTPS only, form input sanitization.

- **Scalability**: Static site generation for fast global delivery, CDN caching.

Technical Constraints

- Framework: Astro Islands architecture with `@astrojs/tailwind` and `@astrojs/image`.

- Styling: Tailwind CSS utility classes; no external UI libraries beyond Lucide React.

- Code Structure:

```
src/
├── components/
│    ├── layout/ (Header, Navigation, Footer)
│    ├── features/ (Hero, Partners, CTASection)
│    └── shared/ (Buttons, FormControls)
├── content/ (Markdown/MDX for docs and pages)
├── hooks/ (useMenu, useForm)
└── utils/ (routes, config)
```

- Imports via `@/` path aliases.

- Remote image patterns whitelist `picsum.photos` domain.

Design Guidelines

- **Color Palette**: Gradient #FFF8E7 → #E8F5E9; text #1E1E1E; CTA #2A2A2A with light text; nav background rgba(42,42,42,0.8) [1].

- **Typography**: Sans-serif headings, 2.5rem hero title, 1.25rem body text.

- **Spacing**: 12px base unit, 24px mobile gutters, 48px desktop.

- **Grid**: 12 columns, 16px gutter; hero uses 2-column layout on tablet+.

Acceptance Criteria

- All routes render correct content and metadata.

- Navigation behaves responsively with client-side toggles.

- Forms validate and submit data to mock API endpoints.

- Images load from Picsum with correct aspect ratios.

- CI pipeline builds and deploys static assets without errors.

Delivery Timeline

- Week 1: Project setup, core layouts, header/navigation.

- Week 2: Home and About pages, gradient hero, partners grid.

- Week 3: Solutions and Services with dynamic filters and forms.

- Week 4: Contact hub, detail routes, form integrations, accessibility audit.

- Week 5: Testing, performance optimization, final review, and launch.

Risks and Mitigations

- **Delay in content**: Use placeholder Markdown/MDX with "Coming Soon" tags.

- **Image licensing**: Rely on Picsum.photos for royalty-free stock.

- **Form backend not ready**: Stub API endpoints with mock server.

This PRD aligns the TETRIX team around a unified vision, ensuring the front-end implementation meets business goals, user needs, and technical standards.

❈

1. Tetrix.txt