# **App Purpose and Core Functionality**

**Prompt:**

You are a technical project manager helping a developer organize a system design document.

The following content is from \*\*Chapter X\*\* of a document generated through multiple prompts. Each chapter contains implementation guidance for building a full-stack web app.

Summarize this chapter into the following sections:

Action Items – Checklist of specific tasks the developer should complete to implement the suggestions and group them based on their category.

**Result:**

I. Core Frontend Development

* Implement Service Showcase:
  + Develop pages/Services.tsx for a comprehensive list of services.
  + Create pages/ServiceDetail.tsx for individual service pages, including a "Before/After image slider" for dynamic content (initially static, then integrate with backend).
* Develop Information Pages:
  + Build pages/About.tsx for company information.
  + Create pages/FAQ.tsx for frequently asked questions.
  + Develop pages/Testimonials.tsx to display customer reviews (initially static, then integrate with backend for dynamic content).
* Design and Implement Navigation:
  + Construct components/Header.tsx to include the main navigation and "Get a Quote" CTA.
  + Develop components/Footer.tsx for additional navigation and contact information.
* Create Visual Portfolio:
  + Implement pages/Gallery.tsx to showcase past projects (initially static, then integrate with backend for dynamic content).
* Integrate UI Component Library:
  + Utilize client/src/components/ui/ to ensure responsive design elements across the application.

II. Dynamic Feature Planning & Backend Integration

* Contact & Lead Generation Forms:
  + Develop the contact form on pages/Contact.tsx to capture user information (name, email, phone, service, message).
  + Implement the "Get a Quote" form, accessible via the CTA, to gather detailed service needs.
  + Backend Requirement: Plan and develop backend endpoints to receive and process data from both the contact and "Get a Quote" forms (e.g., send emails, save to database).
* Chatbot Placeholder & Future Integration:
  + Implement components/ChatbotPlaceholder.tsx as a temporary solution.
  + Backend Requirement: Begin planning for future chatbot integration, which will require significant backend support for NLP and potential CRM integration.
* Dynamic Content Management:
  + Backend Requirement: Design a backend system (e.g., a CMS) to manage and serve dynamic content for testimonials, gallery images, and service detail pages.
* Analytics Integration:
  + Integrate client-side analytics tools like Google Analytics and Meta Pixel as per the PRD.
  + Backend Consideration: Evaluate if backend integration with analytics platforms is needed for robust data collection and reporting.

# **Identify Missing or Incomplete Parts**

# **Component & Layout Audit**

# **Component Data Needs and Logic**

# **Group Features by Intended Role**

# **Roles & Access Control Logic**

# **Authentication Strategy**

# **Protecting Routes by Role or Status**

# **Managing Auth State in the Frontend**

# **Backend Validation and Integration**

# **Entity & Resource Modeling**

# **API Route and Backend Endpoint Suggestions**

# **Database Schema and Relationships**

# **Backend Folder Structure and Starter Boilerplate**