# PRD 3: Technical Architecture

## 1. Tech Stack

* **Framework:** React 18+ (Vite).
* **Language:** TypeScript.
* **Styling:** Tailwind CSS.
* **Icons:** Lucide React.
* **State Management:** React Context API or Zustand (No Redux needed).
* **Routing:** React Router DOM.

## 2. Data Layer (No Database)

### A. Configuration (src/config/)

Instead of a DB, we use static files:

* categories.ts: Defines available categories and their IDs.
* questions.ts: Defines the wizard steps, questions, and mapping to product specs.

### B. API Service (src/services/)

Create a powerApi.ts file.

* **Pattern:** Adapter Pattern.
* **Functionality:**
  + fetchProducts(category: string): Fetches raw data.
  + **Mock Mode:** Must include a USE\_MOCK\_DATA flag. If true, returns hardcoded JSON files (essential for development without live API keys).
* **Data Normalization:** Transform the external API response into our internal Product interface.

## 3. Data Models (TypeScript Interfaces)

// Internal Product Model  
interface Product {  
 id: string;  
 sku: string;  
 name: string;  
 brand: string;  
 price: number;  
 image: string;  
 category: string;  
 specs: Record<string, string | number | boolean>; // e.g. { capacity: 9, noise: 44 }  
 features: string[]; // e.g. ["AutoDose", "WiFi"]  
}  
  
// Wizard Config Model  
interface WizardQuestion {  
 id: string;  
 text: string;  
 options: {  
 label: string;  
 value: any;  
 // Rules for filtering  
 filterCriteria?: {  
 field: string;  
 operator: 'gte' | 'lte' | 'includes' | 'equals';  
 };  
 }[];  
}

## 4. Refactoring Plan (For AI Agent)

1. **Cleanup:** Delete prisma/, server/ folders. Remove @prisma/client and database drivers from package.json.
2. **Setup Config:** Create the static data structure for questions.
3. **Setup API:** Create the mock-capable API service.
4. **Frontend Logic:** Rewrite the Wizard component to read from local config instead of API/DB.