**Case Study: Product Catalog for TechTrend Innovations**

**Problem Statement:**

**Background**: TechTrend Innovations is a growing e-commerce startup specializing in electronics, gadgets, and accessories. To enhance their online presence, they need a **Product Catalog** page for their website to showcase their featured products in a clean, organized, and visually appealing manner. The catalog should display static data without user interaction, focusing on presenting key information about the store and its products to attract potential customers. As a React developer, your task is to build a reusable, static Product Catalog component using React and JSX, leveraging Tailwind CSS for styling, and ensuring the implementation avoids runtime errors (e.g., Tailwind CDN issues) by using a production-ready CSS file.

**Objective**: Create a React-based Product Catalog page that renders:

* A store name (scalar data).
* A list of product categories (array data).
* Store details such as location, founding year, and rating (object data).
* A grid of featured products with details like name, price, and category (array of objects data).

The implementation should use React’s declarative JSX syntax, handle different data types efficiently, , modern styling with Tailwind CSS v4.1.12 (optional).

**Requirements**:

1. **Data Structure**:
   * **Scalar**: Display the store name as a prominent page title.
   * **Array**: Render a list of product categories (e.g., Electronics, Accessories, Gadgets) as styled tags.
   * **Object**: Show store information, including location (string), founded year (number), and rating (number), in a dedicated section.
   * **Array of Objects**: Display a list of products, each with an ID (number), name (string), price (number), and category (string), in a responsive grid layout.
2. **UI/UX**:
   * Use Tailwind CSS v4.1.12 via the production-ready CDN (https://cdn.jsdelivr.net/npm/tailwindcss@4.1.12/dist/tailwind.min.css) to avoid runtime warnings.
   * Style sections with borders, shadows, and appropriate spacing for a clean, professional look.
3. **Technical Constraints**:
   * Include unique key props for lists to ensure efficient rendering.

**Sample Data** (for reference):

* **Store Name**: "TechTrend Innovations"
* **Categories**: ["Electronics", "Accessories", "Gadgets"]
* **Store Info**: { location: "San Francisco, CA", founded: 2023, rating: 4.8 }
* **Products**:
  + { id: 1, name: "Wireless Earbuds", price: 59.99, category: "Electronics" }
  + { id: 2, name: "Smart Watch", price: 199.99, category: "Gadgets" }
  + { id: 3, name: "Phone Case", price: 19.99, category: "Accessories" }

**Acceptance Criteria**:

* The page renders all data types correctly without errors.
* Tailwind classes are used effectively for styling,
* The code is clean, readable, and follows React best practices (e.g., unique keys for lists).

**Constraints**:

* Must be a static component—no interactivity (e.g., no buttons to add products or filter categories).
* Should be beginner-friendly, using straightforward JSX and Tailwind classes.

**Bonus (Optional)**:

* Include a fallback message if the product list is empty (e.g., “No products available”).
* Primary focus on functionality and secondary focus on Styles.
* You can use either Tailwind/Bootstrap for CSS.
* Add more no. of products
* Maintain React Keys concept while processing Arrays.