Project Documentation: Product Catalog

**Overview**

The Product Catalog is a React-based web application that displays a list of products fetched from an API. Users can toggle between grid and list views for a better browsing experience.

**Technologies Used**

* React
* Framer Motion (for animations)
* React Icons
* Vite (for project setup and bundling)
* CSS (for styling)

**Features**

1. **Product Display**: Fetches and displays products from an external API.
2. **View Toggle**: Users can switch between grid and list views.
3. **Animations**: Smooth UI interactions using Framer Motion.
4. **Persistent Preferences**: Stores the last selected view mode in local storage.
5. **Toggle List View / Grid View**: Implements dynamic switching between list and grid views based on user preference.

**Project Structure**

project-root/

│── src/

│ ├── components/

│ │ ├── ProductCatalog.jsx

│ │ ├── ProductCard.jsx

│ ├── data/

│ │ ├── api.js

│ ├── App.jsx

│ ├── main.jsx

│── public/

│── index.html

│── styles.css

│── package.json

│── vite.config.js

**Component Breakdown**

**App.jsx**

* The main entry point of the application.
* Renders the ProductCatalog component.

**ProductCatalog.jsx**

* Fetches the product data from api.js.
* Manages the grid/list view toggle using useState.
* Stores user preferences in local storage.
* Renders ProductCard components for each product.

**ProductCard.jsx**

* Displays individual product details including title, price, and image.
* Includes hover animations.

**api.js**

* Contains functions to fetch product data from https://dummyjson.com/products.
* Provides methods to fetch all products and a single product by ID.

**Functional Requirements**

1. **Product Display**:
   * Uses an array named productsList with at least 10 product objects.
   * Renders products inside a <div class="product-container"> with different styles for both views.
2. **Toggle View Button**:
   * A button with id="toggleViewBtn" switches between List and Grid views.
   * List View: Products are displayed in a vertical list.
   * Grid View: Products are displayed in a multi-column grid.
   * Button text dynamically updates to "Switch to Grid View" or "Switch to List View".
3. **State Management**:
   * useState manages the isGridView state (boolean).
   * Default view is List View.
4. **CSS Styling**:
   * **List View**:
     + Products displayed in a row format.
     + Name and price aligned on the left, image on the right.
   * **Grid View**:
     + Products arranged in a flexbox or grid layout with multiple columns.
     + Image positioned above the name and price.

**API Endpoints**

* GET https://dummyjson.com/products - Fetches all products.
* GET https://dummyjson.com/products/{id} - Fetches product by ID.

**Installation & Setup**

1. Clone the repository:

git clone https://github.com/vamsikrishna-garisapati/product-catalog.git

1. Navigate to the project folder:

cd project-root

1. Install dependencies:

npm install

1. Run the development server:

npm run dev

**Bonus Features**

✅ Save user preference in local storage so the selected view persists on refresh.  
✅ Add animations when switching between views for smooth transitions.  
✅ Use icons instead of text (🗂 List View / 📦 Grid View) for better UI experience.

**Future Enhancements**

* Add product search functionality.
* Implement filtering and sorting options.
* Improve mobile responsiveness.

**Evaluation Criteria**

✔ Correct use of useState for managing the view state.  
✔ Proper toggle functionality between List and Grid view.  
✔ Clean and responsive design for both views.  
✔ Bonus: Persist user preference, animations, and enhanced UI elements.