# Programming Task: Generic Store Catalog Web App

This test will assess your ability to refactor and extend a modern React + TypeScript web application. You will transform the existing car store catalog into a generic, configurable store catalog that can support various store types (e.g., electronics, cars, pets) Bonus points for adding sorting, filtering, or pagination.

## Objectives

### 1. Make the Catalog Generic & Configurable

- Refactor the current car-specific implementation so the app can display catalogs for different store types.

- The store type and its configuration (fields, display, etc.) should be easily changeable.

- The UI should adapt to the selected store type, showing relevant fields and details for each item.

### 2. *\*Bonus\**: Sorting, Filtering OR Pagination

- Add at least one of the following features to the catalog UI:

  - **\*\*Sorting:\*\*** Allow users to sort items by price, name, or other relevant fields.

  - **\*\*Filtering:\*\*** Allow users to filter items by category, price range, or other attributes.

  - **\*\*Pagination:\*\*** Break the catalog into pages if there are many items.

## Deliverables

- Refactored React + TypeScript client code supporting multiple store types.

- completed ready-for-run configuration for at least 2 additional store types with detailed items.

-----

## Evaluation Criteria:

1. Functionality:

The app should successfully fetch and display items.

The selected feature (Order/filter or Pagination) must be fully functional.

Use correct folder structure to your benfits, functionality, code order, best practice and etc.

2. Code Quality:

The code should be generic enough to use for different kind of stores.

The code should be structured with reusable components (e.g., item card, item display sidebar, filter).

The code should be clean, easy to understand, and follow best practices (e.g., React hooks, state management).

3. Error and Loading States:

Handle errors appropriately and show loading states when the app is fetching data.

## Limitations:

None. you may use any library as you wish.

## Expected Time:

This task should take approximately 2-4 hours to complete. You should sent it by mail or through git platform of your chioce within 5 days. Focus on delivering clean and functional code with a good user experience.

**\*\*Good luck!\*\***