**POS & Northwind Business Simulation – React Requirements**

**1. Project Overview**

**Purpose:**  
A web-based simulation of a wholesale/retail trading company (like Northwind Traders) to manage products, inventory, customers, suppliers, orders, employees, shippers, and reports.

**Target Platform:**

* **Frontend:** React 18+ (Vite or CRA)
* **State Management:** Redux Toolkit / React Query
* **UI Libraries:** Bootstrap 5 / MUI (Material-UI)
* **Backend (optional):** Node.js + Express or ASP.NET Core API
* **Database (optional):** SQL Server / SQLite / In-memory JSON

**Core Modules:**

1. Product & Inventory Management
2. Customer Management
3. Supplier Management
4. Employee & Territory Management
5. Orders / Sales Management
6. Shippers / Logistics
7. Reporting & Analytics
8. Optional Enhancements: Authentication, Role-based access, Export reports

**2. Functional Requirements**

**2.1 Products & Inventory**

* CRUD Products: Add, update, delete, view details
* Attributes: ID, Name, Category, Supplier, Price, Stock
* Categories: CRUD management
* Suppliers: CRUD and link to products
* Stock Management: Increase/Reduce stock, low-stock alerts
* Filters & Search: By category, supplier, stock status

**Menu Workflow:**

* Products → List / Add / Edit / Delete / Stock Adjustments
* Categories → List / Add / Edit / Delete

**2.2 Customers**

* CRUD customer info: ID, Company, Contact, Address, City, Country, Phone
* Assign orders to customers
* Search & Filter by name, city, or country

**Menu Workflow:**

* Customers → List / Add / Edit / Delete
* Orders → Create order for selected customer

**2.3 Suppliers**

* CRUD suppliers
* Link products to suppliers
* Attributes: ID, Company, Contact, Address, City, Country, Phone

**Menu Workflow:**

* Suppliers → List / Add / Edit / Delete
* Products → Link/Unlink supplier

**2.4 Employees & Territories**

* CRUD employees: ID, Name, Title, HireDate
* CRUD territories: ID, Name
* Assign employees to territories
* Assign orders to employees by territory

**Menu Workflow:**

* Employees → List / Add / Edit / Delete
* Territories → List / Add / Edit / Delete
* Employee → Assign Territories

**2.5 Orders / Sales**

* Create orders workflow:
  1. Select customer
  2. Add products with quantity
  3. Validate stock availability
  4. Assign employee and shipper
  5. Confirm order → reduce stock
* Order Attributes: ID, Customer, Employee, Shipper, Products, Total, Date
* Track order history per customer
* Edit / Cancel orders

**Menu Workflow:**

* Orders → List / Add / Edit / Cancel / View Details

**2.6 Shippers / Logistics**

* CRUD shippers
* Assign shippers to orders
* Attributes: ID, Company, Contact, Phone

**Menu Workflow:**

* Shippers → List / Add / Edit / Delete
* Orders → Assign shipper

**2.7 Reporting & Analytics**

* Top-selling products
* Sales per customer / category / employee / territory
* Revenue and profit reports
* Inventory reports (stock levels, low-stock alerts)
* Export reports (CSV/Excel optional)

**Menu Workflow:**

* Reports → Products / Customers / Categories / Employees / Inventory
* Export → CSV / Excel

**3. Non-Functional Requirements**

* **Performance:** Handle 1000+ products, customers, orders efficiently
* **Usability:** Clean, responsive UI, intuitive navigation
* **Maintainability:** Modular components & services per business area
* **Persistence:** Optional database or localStorage for offline use
* **Security:** Optional role-based authentication (Admin, Cashier)

**4. Technical Requirements**

* **Language & Framework:** React 18+ (TypeScript optional)
* **State Management:** Redux Toolkit / React Query / Context API
* **UI Library:** Bootstrap 5 / MUI (Material-UI)
* **Data Handling:** Axios / Fetch API
* **Backend:** Optional Node.js / ASP.NET Core API
* **Database:** SQL Server / SQLite / In-memory

**Architecture:**

* Layered & modular:
* src/
* ├── components/ # UI components
* ├── pages/ # Page-level components
* ├── features/ # Redux slices / React Query hooks per module
* ├── services/ # API calls
* ├── utils/ # Helper functions
* ├── App.jsx
* └── main.jsx

**5. UI / Menu Structure**

**Main Menu / Sidebar:**

* Dashboard
* Products
  + List Products
  + Add Product
  + Categories
  + Stock Management
* Customers
  + List Customers
  + Add Customer
  + Order History
* Suppliers
  + List Suppliers
  + Add Supplier
  + Link Products
* Employees
  + List Employees
  + Add Employee
  + Territories
* Orders
  + List Orders
  + Add Order
  + Track Order Status
* Shippers
  + List Shippers
  + Add Shipper
* Reports
  + Sales Reports
  + Inventory Reports
  + Export Reports
* Settings (optional)
  + User Management
  + Roles & Permissions

**Page / Workflow Example: Add Order**

1. Select Customer → Dropdown / Search
2. Add Products → Table with search + add quantity
3. Assign Employee → Dropdown based on territory
4. Assign Shipper → Dropdown
5. Confirm Order → Reduce stock, generate order summary

**6. Optional Enhancements**

* Discounts / Promotions per order
* Multi-currency support
* Authentication & Role-based access (Admin / Cashier)
* Export reports to CSV/Excel
* Responsive UI (Desktop + Mobile)
* Notification system (stock alerts, order updates)

**✅ Suggested Development Steps**

1. **Products & Inventory Module** → CRUD + stock management
2. **Customer Module** → CRUD + order linking
3. **Supplier Module** → CRUD + product linking
4. **Employee & Territory Module** → CRUD + assignment
5. **Shippers Module** → CRUD
6. **Orders Module** → Full workflow
7. **Reports Module** → Sales, inventory, analytics
8. **Optional Enhancements** → Authentication, export, promotions

If you want, I can **draw a visual React sidebar/menu layout with all modules and sub-menus** showing workflows for your POS project—this makes it much easier for dev planning.

Do you want me to do that?