**Complete Technical Documentation**

**Project Name:**

**Northwind Order Management System**

**Overview**

Northwind Order Management System is a modern, full-stack web application that enables users to manage sales orders with rich features like address validation, interactive maps, and PDF reporting.

This solution was built using React and ASP.NET Core following Clean Architecture principles, ensuring a scalable, maintainable, and testable application for real-world business use.

**Technology Stack**

|  |  |
| --- | --- |
| Layer | Technology |
| Frontend | React, TypeScript, Ant Design, Axios |
| Backend | ASP.NET Core, C#, MediatR, CQRS, Entity Framework Core |
| Database | PostgreSQL (adaptable to SQL Server) |
| External APIs | Google Maps API (Places + Geocoding) |
| PDF Generation | QuestPDF |
| Testing | XUnit, Moq (backend unit tests), optional React Testing Lib |
| Dev Tools | Postman, GitHub, Visual Studio, VS Code |

**Project Architecture (Clean Architecture + CQRS)**

/Northwind.OrderManagement\_JM.Server/

│

├── Application/ → Business Logic (CQRS Handlers, DTOs, Validators)

│ ├── Features/

│ ├── Interfaces/

│ └── Reports/

│

├── Domain/ → Core business entities (Order, OrderDetails, etc.)

├── Infrastructure/ → External integrations, services (Google Maps)

├── Persistence/ → DbContext, EF Core Migrations, Seeders

├── WebAPI/Controllers/ → HTTP endpoints (RESTful)

└── Tests/ → Unit Tests with XUnit & Moq

**Frontend Structure**

/frontend/form/

│

├── src/

│ ├── components/ → UI Components (OrderForm, OrderList, ValidatedCard)

│ ├── hooks/ → Custom hook: useOrderData.ts (centralized logic)

│ ├── pages/ → Main view: OrderManagementUI.tsx

│ ├── services/ → Address validation, geocoding, PDF, etc.

│ ├── utils/ → Google Maps Loader

│ └── api.ts → Axios configuration

├── public/

└── .env → Environment variables (.gitignored)

**Key Features**

**1. Full CRUD for Orders**

* Create, Read, Update, Delete operations for sales orders.
* Assign customers, employees, and shippers.
* Add multiple products with quantity, discount, price, and subtotal.
* Auto-calculate total price.

**2. Address Validation via Google Maps API**

* Address input field with autocomplete powered by Google Places.
* On validation, it displays:
  + Formatted address
  + City, State, Postal Code, Country
  + Latitude & Longitude

**3. Embedded Map View**

* Google Map centered on validated shipping address.
* Displays a marker with full location details.

**4. PDF Report Generation**

* All Orders Report: Full list of orders with customers, employees, dates, and totals.
* Order Details Report: For a single order, includes metadata and line items.

**5. Navigation & Filtering**

* **Buttons to move between first, previous, next, last order.**
* **Modal to search/filter orders by:**
  + Order ID
  + Customer
  + Employee

**6. Form Validation & Error Handling**

* Real-time form validation (required fields, numeric limits, etc.)
* Backend error messages shown on UI.
* Errors in API handled with toast messages or visual indicators.

**7. Backend Unit Testing (XUnit)**

* CQRS Handlers (queries and commands) have associated unit tests.
* Services like Google AddressValidator and ReportGenerator are tested using Moq.

**Database Design (simplified)**

|  |  |
| --- | --- |
| Table | Relationships |
| Orders | ↔️ Customers, ↔️ Employees, ↔️ Shippers |
| OrderDetails | ↔️ Orders, ↔️ Products |
| Products | ↔️ Suppliers, ↔️ Categories |

Seeded using Northwind schema adapted for MySQL Server.

**Security & Environment**

* All API keys and sensitive data are stored in .env (never committed to Git).
* Frontend reads the Google Maps API Key from process.env.REACT\_APP\_GOOGLE\_API\_KEY.

**How to Run Locally**

**# Backend**

cd Northwind.OrderManagement\_JM.Server

dotnet ef database update

dotnet run

**# Frontend**

cd frontend/form

npm install

npm start

Add .env files with:

REACT\_APP\_GOOGLE\_API\_KEY=your\_google\_key\_here

**What Has Been Completed**

|  |  |
| --- | --- |
| Feature | Status |
| Order Form (CRUD + product rows) | ✅ |
| Address Autocomplete + Validation | ✅ |
| Google Maps Embedding | ✅ |
| PDF Report Generation | ✅ |
| Modal Filtering + Navigation | ✅ |
| Hook-based state management | ✅ |
| Backend CQRS + Clean Architecture | ✅ |
| Unit Testing (Handlers, Services) | ✅ |
| Error Handling | ✅ |
| Environment-secure Configuration | ✅ |

**Project Highlights (Presentation Style)**

For Clients/Stakeholders:

* Easy-to-use order form with dynamic product selection.
* Address autocomplete saves time and ensures accuracy.
* See the real delivery location on an interactive map.
* Export your data in clean, printable PDFs.
* Robust search and filtering to locate orders instantly.

**For Developers / Tech Leads:**

* **Fully separated logic using Clean Architecture.**
* **CQRS pattern ensures clear separation of queries and commands.**
* **External integrations decoupled using interfaces.**
* **High test coverage for critical logic.**
* **Scalable, maintainable code with modern patterns.**

**GitHub & Deployment**

* **GitHub repository includes .gitignore to protect sensitive files.**
* **Easily deployable to Azure App Service, Vercel, Netlify, etc.**
* **Uses modern toolchain for continuous development.**