# Requirements for Order Management System Web Application

## 1. Project Overview

Title: Order Management System Web API  
Objective: Develop a REST API for managing orders, implementing two models (e.g., Orders and Products), with data stored in memory and exposed via a Swagger-enabled interface.  
Technology Stack: ASP.NET Core, Entity Framework Core (EF Core), Swagger, Inline Memory for in-memory storage.

## 2. Folder Structure Setup

Define the folder structure for an organized and scalable project:  
- /Controllers: Houses the API controllers.  
- /Models: Defines the models used (e.g., Order, Product).  
- /Data: Contains Entity Framework-related configurations and in-memory data context.  
- /Services: Holds any business logic or services used by controllers.  
- /Migrations: For potential database migrations if needed.  
- /wwwroot: Optional for static files.  
- /Properties: Configuration settings (e.g., launchSettings.json).  
- Program.cs and Startup.cs: Entry point and configuration files.

## 3. Models Definition

Specify the two models:  
- Order Model:  
 - Properties: OrderId, OrderDate, OrderAmount, OrderStatus, ProductList, etc.  
 - Relationships: Link to the Product model.  
- Product Model:  
 - Properties: ProductId, ProductName, Price, StockQuantity.  
 - Relationships: Linked to orders if required.

## 4. Steps for Source Code Setup

Detailed steps for creating and organizing the source code:  
- Initialize the Project:  
 - Use 'dotnet new webapi' to create a base project.  
- Create Models:  
 - Add 'Order' and 'Product' classes in /Models.  
- Set up Controllers:  
 - Generate 'OrderController' and 'ProductController' in /Controllers.  
- Set up Data Context:  
 - Configure 'ApplicationDbContext' in /Data with in-memory database setup.  
- Enable Dependency Injection:  
 - Configure services in Startup.cs.

## 5. Entity Framework and Swagger Integration

Define how to enable and configure Entity Framework and Swagger:  
- Entity Framework Core:  
 - Use In-Memory provider to manage data during runtime.  
 - Add ApplicationDbContext in Startup.cs and configure as the service provider.  
- Swagger:  
 - Enable Swagger in Startup.cs for API documentation.  
 - Define Swagger options, including API versioning and endpoints.

## 6. Inline Memory for Data Storage

Configure EF Core to use in-memory storage by setting up 'AddDbContext' with options.UseInMemoryDatabase('OrderManagement').