The best way to implement in C# code and LINQ, the Command Query Responsibility Segregation (CQRS) pattern in a .NET Web CORE API VS2022 solution using validations. Please display code examples for both command and query classes, utilizing Entity Framework Core, Repositories, Interfaces and DTOs and mocked in memory database data and Entity Framework Core Migrations. Include a test project in the solution. Also display a complete diagram showing the Visual Studio 2022 .NET CORE Web API solution’s folder structure for the above example as it would appear in Solution Explorer

To implement the **Command Query Responsibility Segregation (CQRS)** pattern in a C# .NET Web API project using **LINQ**, **Entity Framework Core**, **Repositories**, **Interfaces**, **DTOs**, **validations**, and **test projects** (with mock data and migrations), we need to structure the solution in a way that follows CQRS principles, separating the command (write) logic from the query (read) logic.

Below is a full implementation example, including code for both **command** and **query** classes, and the corresponding **folders** and **structure**.

**Folder Structure in Visual Studio Solution Explorer:**

Mathematica Solution

│

├── Command

│   ├── Commands

│   │   └── CreateItemCommand.cs

│   └── Handlers

│       └── CreateItemCommandHandler.cs

│

├── Query

│   ├── Queries

│   │   └── GetItemsQuery.cs

│   └── Handlers

│       └── GetItemsQueryHandler.cs

│

├── DTOs

│   └── ItemDto.cs

│

├── Domain

│   ├── Entities

│   │   └── Item.cs

│   └── Interfaces

│       └── IItemRepository.cs

│

├── Infrastructure

│   ├── Data

│   │   └── AppDbContext.cs

│   └── Repositories

│       └── ItemRepository.cs

│

├── API

│   ├── Controllers

│   │   └── ItemController.cs

│

├── Validation

│   └── ItemValidator.cs

│

├── Migrations

│   └── (Auto-generated migration files here)

│

└── Tests

    ├── CommandTests

    │   └── CreateItemCommandTests.cs

    ├── QueryTests

    │   └── GetItemsQueryTests.cs

    └── TestStartup.cs

