1. **ManageOrderAndCustomerAPI** 🡺 this contains the API controllers.

ManageOrdersAndCustomers solution consist of 4 projects to apply DDD:

* 1. **CustomerController :**  to manage customer CRUD operations.
  2. **ProductController :** to manage product CRUD operations.
  3. **OrdersController :** to manage Orders done customers CRUD operations.

1. **ManageOrdersAndCustomerBL 🡺** this contains the DB Models, View Models, generic Repository Interface and generic Unit Of Work Interface.
   1. **DBModels :** those are the 4 models (Customer, Item, Order, Product) used to create the OrderManager DB using EF code first approach.
   2. **ViewModels :** those are 4 models that contains the logic used by the API controllers.
   3. **IRepository :** this is the Interface of the generic Repository pattern.
   4. **IUnitOfWork :**  this is the interface of the generic Unit Of Work pattern.
2. **ManageOrdersAndCustomersEF 🡺** this contains the EF code first approach, generic Repository and generic Unit Of Work implementation.
   1. **AppDBContext class :** is responsible to build the database when using EF first approach.
   2. **Repository class :** is a generic class that implements **IRepository.**
   3. **UnitOfWork class :** is a generic class that implements **IUnitOfWork.**
3. **ManageOrdersAndCustomersTest 🡺** this contains NUnit testing for the 3 controllers of the API project.

**PS**: API is been tested using **NUnit** test project in addition to a 3rd party tool called **Postman.**

Assumptions :

1. User can create/update/delete/get products & Customers by calling the methods created in **CustomerController** and **ProductController**.
2. To create an order, user has to call the **OrdersController** and provide a complex object of type **OrderVM.**
3. This **OrderVM** object contains the **ItemsVM** (product, quantity) that were selected by the customer.
4. Each selected Item contains a **ProductVM** data (name, price).
5. User adds all selected Items to an order at once.
6. User can edit any item in the order.
7. User can delete the order.

Database Diagram :

