Entity Framework Core + LINQ Training Tasks

Starting from **the Restaurant solution (Server project)** you have to:

- Install **Microsoft.EntityFrameworkCore.SqlServer** to use entity framework core in your application
- Install **Microsoft.EntityFrameworkCore.Design** to use migrations capabilities
- Install **Microsoft.EntityFrameworkCore.Proxies** to use lazy loading in EF core
- Create a new context to access database (Restaurant context that inherits **DbContext**) and register it in the **Startup.cs like in the provided example.**

- Create a relationship one-to-many between **RestaurantBranch** and **OrderTable** (a restaurant branch will have a collection of **OrderTables**)

- Add a new entity called Waiter with the following properties: WaiterId, Name, Age and create a relation of type many-to-many between Waiter and **OrderTable** so that one or more waiters can serve to many tables (they share it).

- Add a new entity called **Manager/Director** with the following properties: ManagerId, Name, Surname and create a relation of type one-to-one between **RestaurantBranch** and **Manager**. Each restaurant has exactly one manager and the manager is unique for that restaurant.

- Create a **custom LINQ method** (an extension method that can be used on IEnumerable) that is going to filter **OrderTables** by any criteria that you want (maybe filter them by a **RestaurantBranch**, that is Open and is on a specific Address).

Mentions: in the provided API

Modify all methods that you created (get, post, patch, put, delete) to actually do these operations on a database.

Think where lazy / eager loading fits in every operation (maybe you can use lazy when retrieving all **OrderTables** but eager when you retrieve order tables by id).