**Objectives**:To Setup required Repository for the project ApexRestaurant .

**Theory:** The repository layer isolates Business layer from the Data Access Layer. The Repository contains Data Mapper entity. This entity can be used as a model entity for providing schema of the data for performing CRUD operations, by using the CRUD operations defined in the repository  Repository pattern C# is a way to implement data access by encapsulating the set of objects persisted in a data store and the operations performed over them, providing a more object-oriented view of the persistence layer.

**Group Discussion:**

ApexRestaurant\ and initialize projects:

Create an interface IGenericRepository.cs:

Create a class GenericRepository.cs

WhenEver we are follows same rules and some line are changed as per our requirement in different classes.

Create a class RepositoryModule.cs

After Create class RepositoryModule we have add different table as per our requirement in our projects.

Create a class RestaurantContext.cs:

Create a folder “Domain” and add a class “Customer.cs”,“Staff.cs”,”StaffRoles”,”Menu.cs”,”MenuItem.cs”,”Meal.cs”and” MealDish.cs” etc

Create a folder “Rcustomer” and other folder which are necessary for our project, where R stands for Repository.

Create an interface “ICustomerRepository.cs” inside “RCustomer” folder and create other Interface as follow IcustomerRepo .

Create a class “CustomerRepository.cs” insider “RCustomer” folder and create other class as per our requirement for a full project.

**At Finally,**

To add related packages:

cd ApexRestaurant.Repository

dotnet add package Microsoft.EntityFrameworkCore

dotnet add package Microsoft.EntityFrameworkCore.SqlServer

dotnet add package Newtonsoft.Json

And finally we have success fully build the project.

**Conclusion:** From this lab we have conclude that,Repository Pattern separates the data access logic and maps it to the entities in the business logic. It works with the domain entities and performs data access logic. In the Repository pattern, the domain entities, the data access logic and the business logic talk to each other using interfaces