**Best Approach and Practices**

**Task:**

- API listing all customers

- API adding a customer

- API updating a customer

- Use .NET Core 3.1 or NET 5+ **Used (.netCore3.1)**

**Project Architecture**

It is a three-tier architecture in which outermost layer is the API layer which calls service layer which should contain business logic. Service layer calls database layer which is utilising repository pattern as explained below.

**Repository Pattern**

Reason to use:

* To perform crud operation generically for any entity.
* It centralizes data logic or business logic and service logic
* If you want to modify the data access logic or business access logic, you don't need to change the repository logic

**Dependency Injection**

Reason to use:

-Every time in class/controller no need to instantiate the object we simply inject the object of that class in constructor to avail the features.

- It reduce tight coupling between software components.

**Linq Approach**

* It helps in development time by catching errors at compile time and includes IntelliSense & Debugging support and also it is strongly type.
* Developer can write minimal code
* There is no such need to open close database every time while query to database operation Linq framework manages this automatically.
* In my view Linq can be used for any type of collection and store procedure precompiled query which only execute on SQL engine.