**WEEK-3**

**LAB-1**

STEP 1: -

⦁ What is ORM?

=> ORM (Object-Relational Mapping) is a programming technique that allows you to query and manipulate data from a database using an object-oriented paradigm.

⦁ How ORM Maps C# Classes to Database Tables

=> Each C# class represents a table in the database.

Each property in the class represents a column in the table.

Relationships (e.g., one-to-many, many-to-many) are represented using navigation properties.

⦁ Benefits of ORM

1. Productivity: Write less boilerplate code, auto-generate SQL.

2. Maintainability: Code-centric model makes refactoring easier.

3. Abstraction: No need to manually write raw SQL for every query.

STEP 2: -

**EF Core vs EF Framework:**

|  |  |  |
| --- | --- | --- |
| **FEATURE** | EF CORE | EF FRAMEWORK(EF6) |
| Platform Support | Cross-platform (.NET Core & .NET 5+) | Windows-only (.NET Framework) |
| Performance | Faster with compiled queries | Slower with large models |
| Features | Modern features (LINQ, async, etc.) | Mature but limited flexibility |
| Mapping Capabilities | Supports NoSQL (like Cosmos DB) | Relational only |
| Development Status | Actively developed | Maintenance only |

STEP3: -

**EF Core 8.0 Features:**

1. **JSON Column Mapping**

* Store structured data in a single column using JSON format.

1. **Compiled Models for Performance**

* Models can be precompiled at build time to reduce startup time.

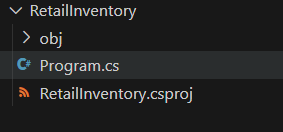
1. **Interceptors**

* Add custom logic when EF Core executes commands (e.g., logging, auditing).

1. **Improved Bulk Operations**

* Faster and more reliable support for AddRange, UpdateRange, etc.

**Step 4: Create the Console App**



**Step 5: Install EF Core Packages**

