# **EF Core 8.0 Guided Hands-On Exercises**

## Lab 1: Understanding ORM with a Retail Inventory System

#### 1. What is ORM?

ORM stands for Object-Relational Mapping. It is a way to connect the code we write in C# with the database tables, without having to write SQL queries manually.

- A class in C# represents a table in the database.
- The properties of the class act as columns in the table.
- Each object (created using that class) is like a row in the table.
- Relationships like one-to-many or many-to-many can also be handled easily.

```
public class Student
{
   public int Id { get; set; }
   public string Name { get; set; }
}
```

### 2. Benefits of using ORM:

- Increases productivity (less code, faster development)
- Easier to maintain the code
- No need to write complex SQL everything is done using C# objects

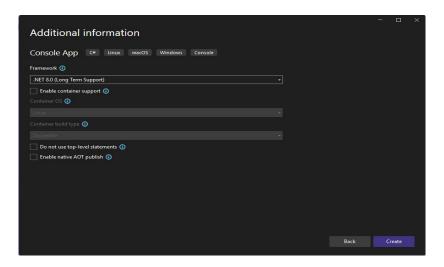
#### 3. EF Core vs EF Framework (EF6)

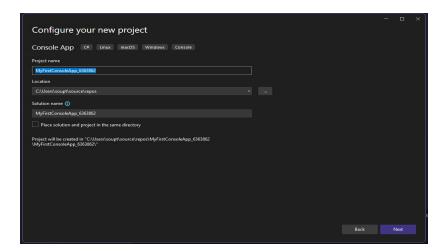
Feature	EF Core	EF Framework (EF6)
Platform	Cross-platform (Windows, Linux, Mac)	Windows-only
Performance	Lightweight and fast	Slower and heavier
Modern Features	LINQ, async/await, compiled queries	Limited modern features
Flexibility	Highly customizable	Less flexible
Maturity	Newer, still evolving	More mature, stable for old apps

#### 3. New features in EF Core 8.0

- <u>JSON column mapping</u>: You can store and query JSON data directly in a column, which is useful when data doesn't fit into normal tables easily.
- <u>Compiled models</u>: The models are pre-compiled, which makes the app start faster and improves performance.
- <u>Interceptors</u>: These allow you to monitor or change what EF is doing behind the scenes good for logging, security, etc.
- <u>Better bulk operations</u>: Makes it easier and faster to insert, update, or delete lots of data at once.

### 4. Create a .NET Console App:





# 5. Install EF Core Packages:

