Entity Framework(Week -3)

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

### ****1.What is ORM?****

***ORM*** *stands for* ***Object-Relational Mapping****.  
It is a programming technique that allows developers to interact with a database using* ***C# classes*** *instead of writing raw SQL queries.*

* *ORM maps* ***C# classes*** *to* ***database tables****, and class* ***properties*** *to* ***table columns****.*
* *This means you can* ***query****,* ***insert****,* ***update****, and* ***delete*** *records using* ***C# code****, and the ORM handles the SQL under the hood*.

***Benefits of ORM:***

* **Productivity** – Speeds up development.
* **Maintainability** – Code is easier to manage and refactor.
* **Abstraction** – Hides complex SQL and DB interactions.

### ****2: EF Core vs EF Framework****

| **Feature** | ****EF Core**** | ****EF Framework (EF6)**** |
| --- | --- | --- |
| **Platform** | **Cross-platform** | **Windows-only** |
| **Lightweight** | **Yes** | **No** |
| **Performance** | **Improved (supports compiled models, async)** | **Slower compared to EF Core** |
| **Modern Features** | **LINQ, async/await, compiled queries** | **Limited modern features** |
| **Flexibility & Extensibility** | **High** | **Less flexible** |
| **Release Year** | **Newer (EF Core 1.0 → 8.0)** | **Older, more mature** |

### ****3: EF Core 8.0 Features****

EF Core 8.0 includes new features that make development faster and more efficient:

* **JSON Column Mapping** – Store and query JSON in columns easily.
* **Compiled Models** – Faster startup time and performance.
* **Interceptors** – Hook into EF events (e.g., before save).
* **Better Bulk Operations** – Performance enhancements for large data changes.

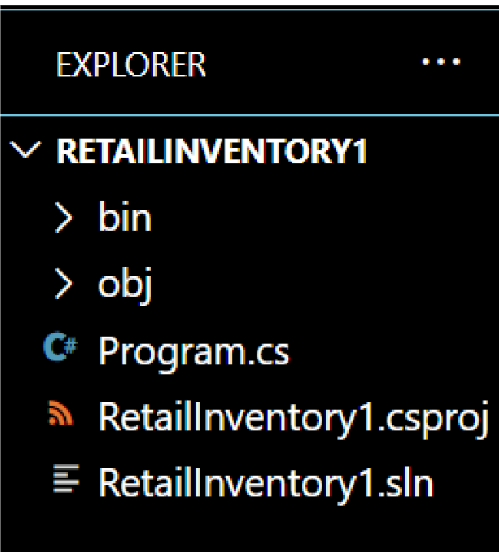
**4: Create a .NET Console App**

* *dotnet new console -n RetailInventory*

**5: Install EF Core Packages**

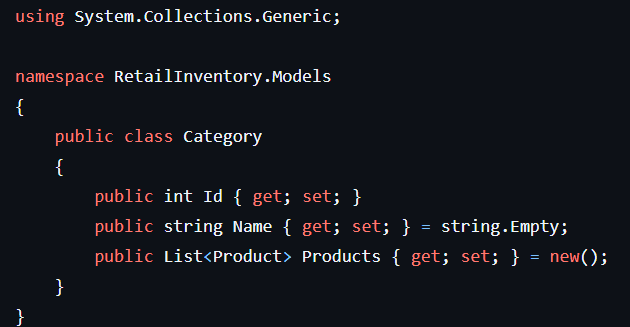
* dotnet add package Microsoft.EntityFrameworkCore.SqlServer
* dotnet add package Microsoft.EntityFrameworkCore.Design

After Setup

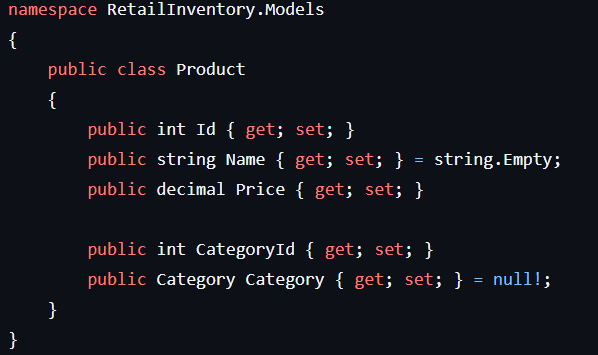


**Lab 2: Setting Up the Database Context for a Retail Store**

Category.cs



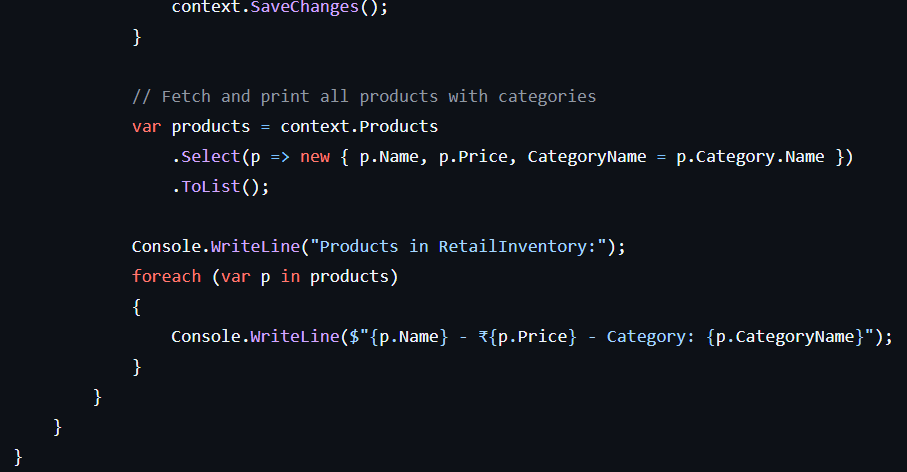
Product.cs



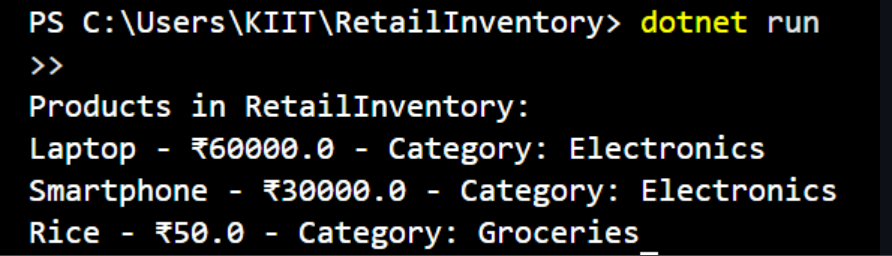
AppDbContext



Program.cs



Output

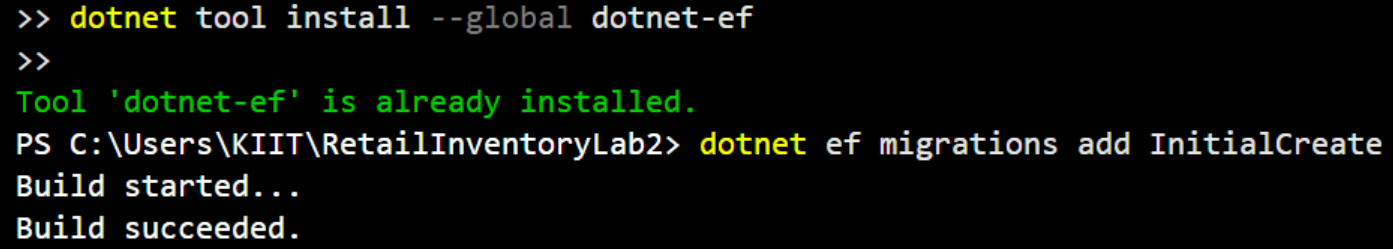


**Lab 3: Using EF Core CLI to Create and Apply Migrations**

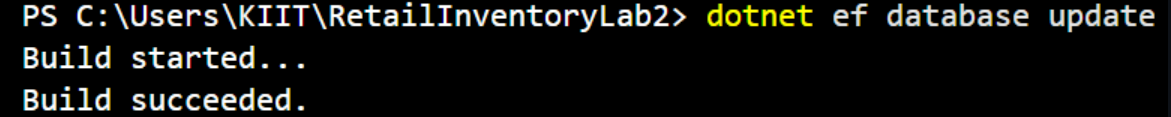
1. **Install EF Core CLI**



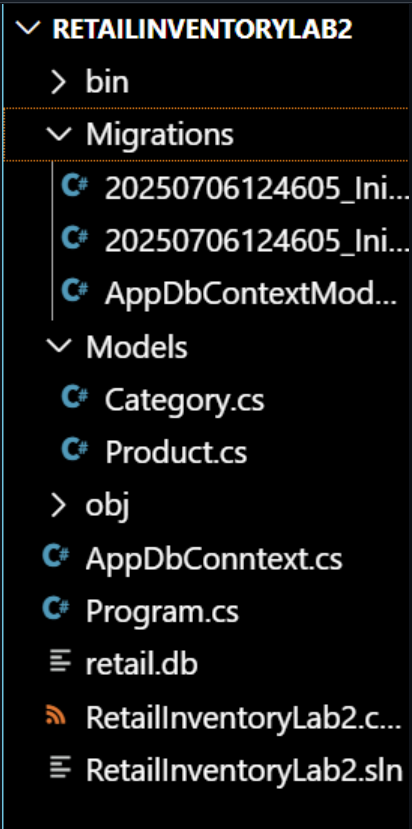
**2. Create Initial Migration:**



**3. Apply Migration to Create Database**

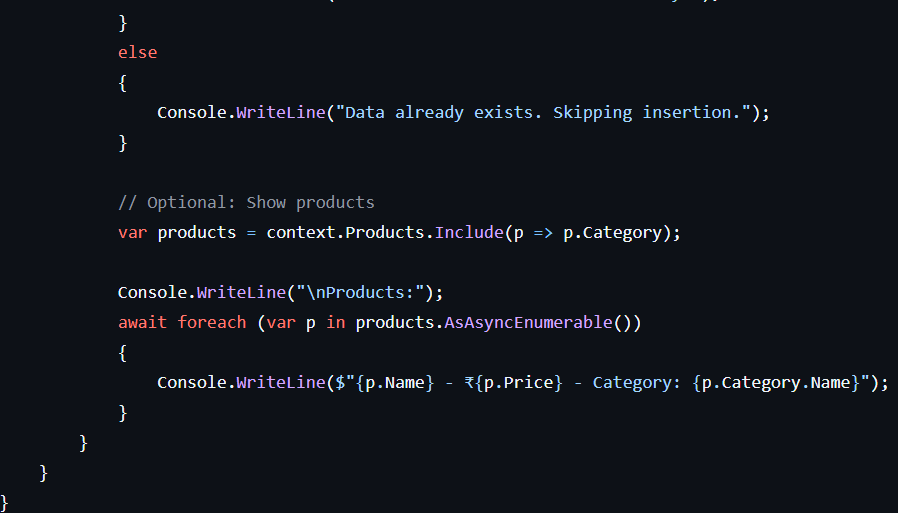
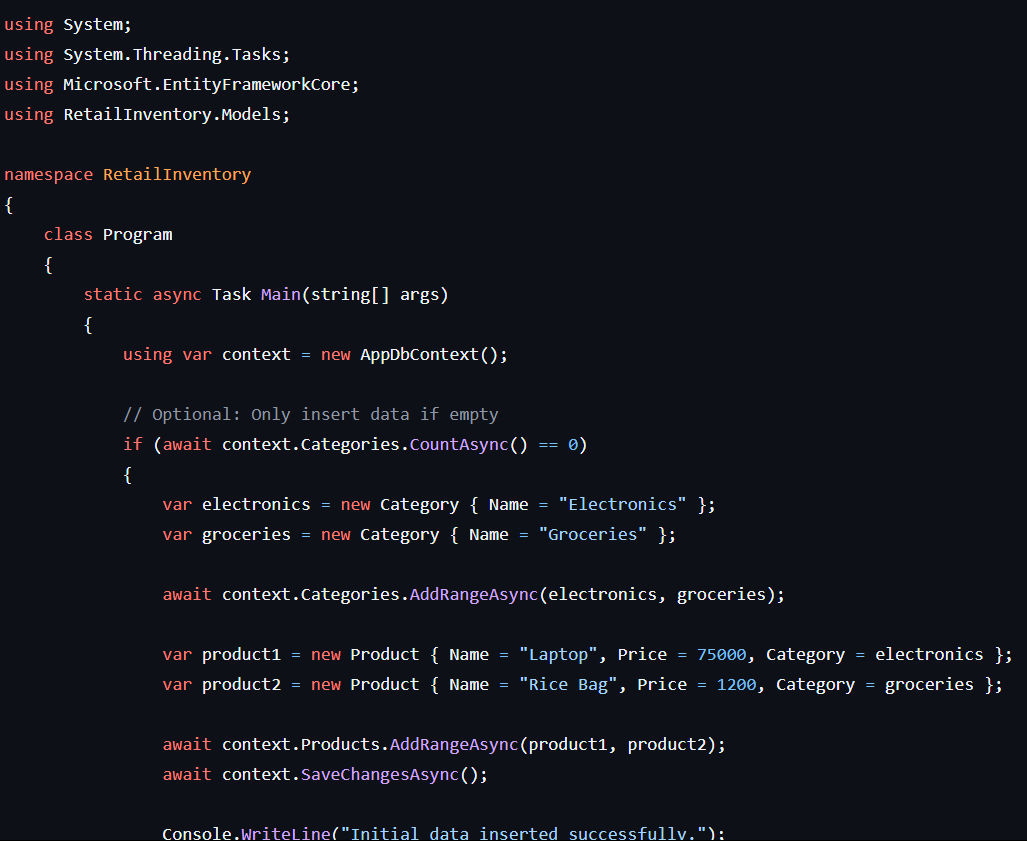


**Output**

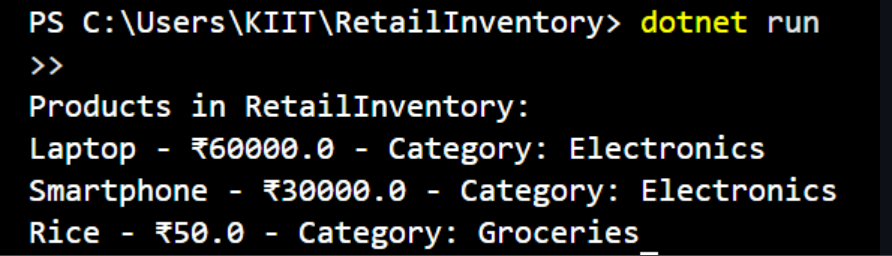


**Lab 4: Inserting Initial Data into the Database**

Program.cs



Output



**Lab 5: Retrieving Data from the Database**

Program.cs



Output

