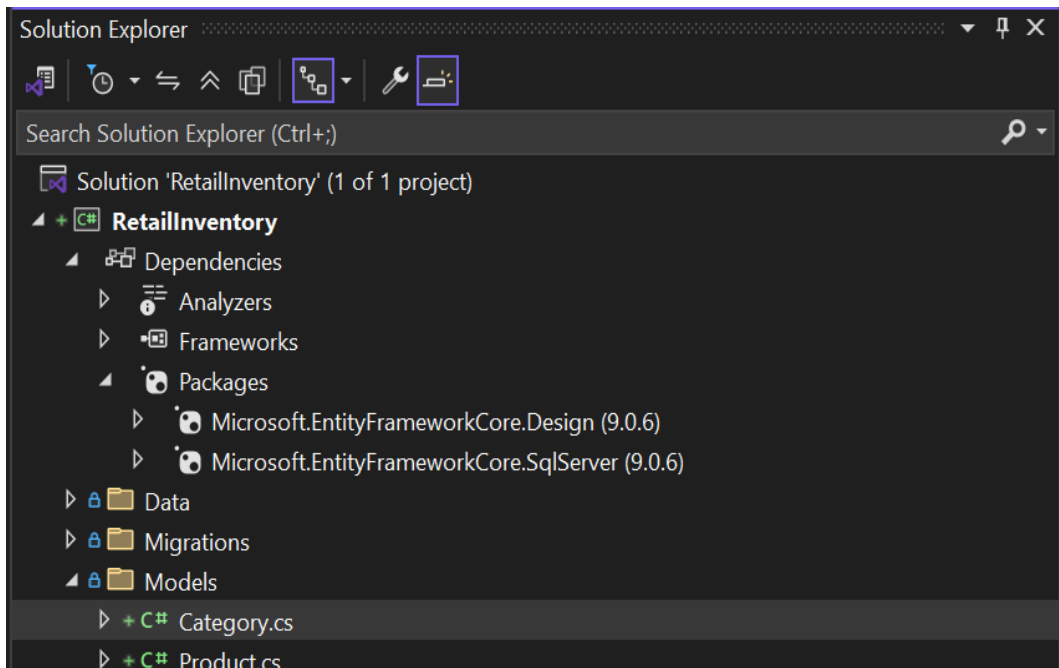
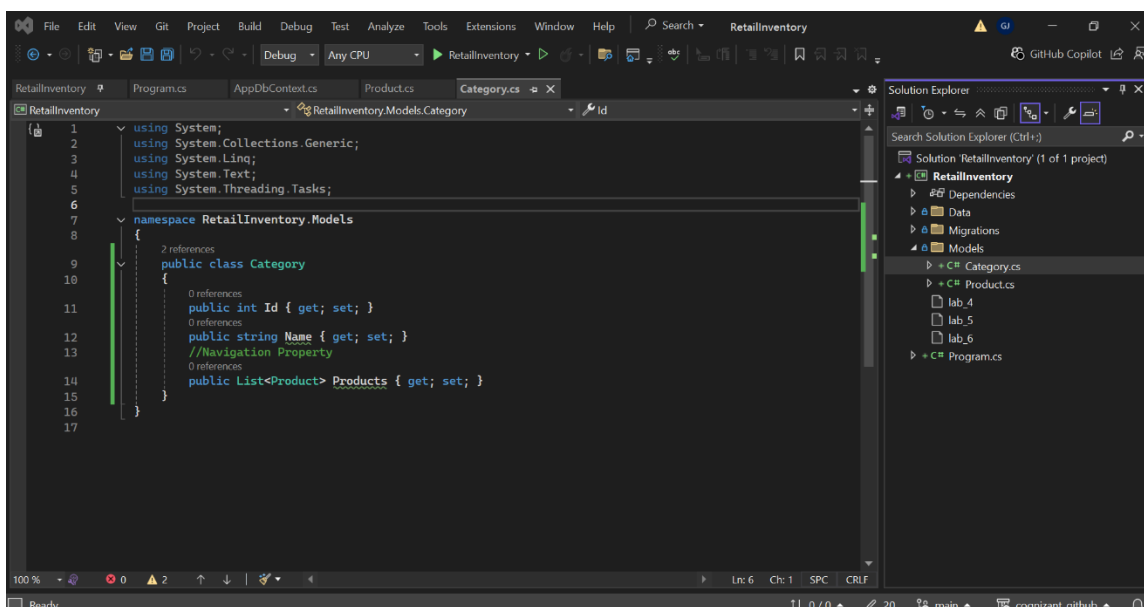


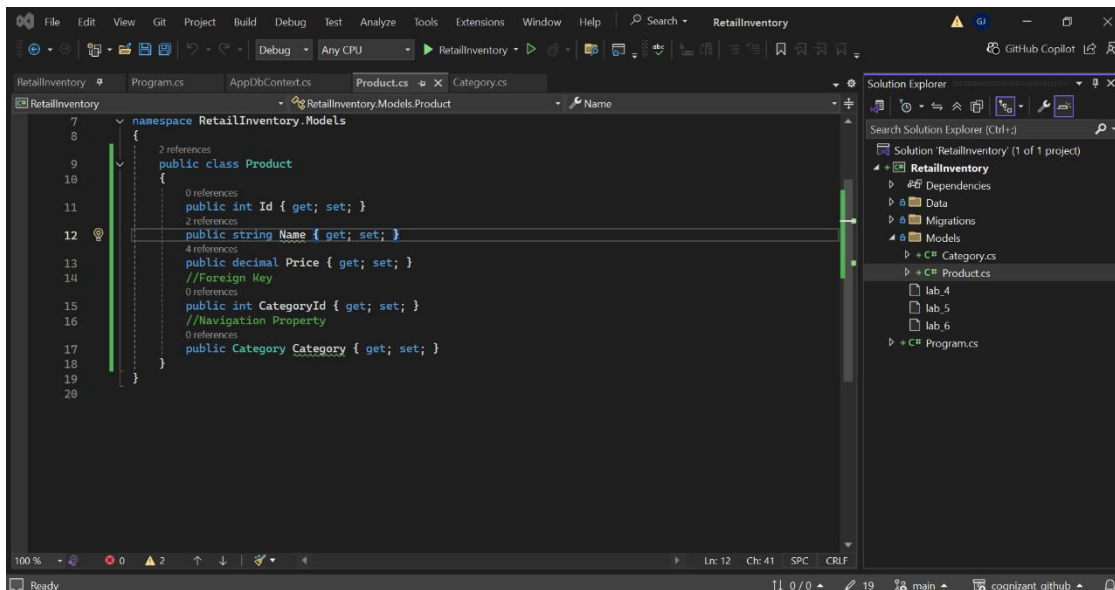
# Week-3 Ef Core

## Lab-1

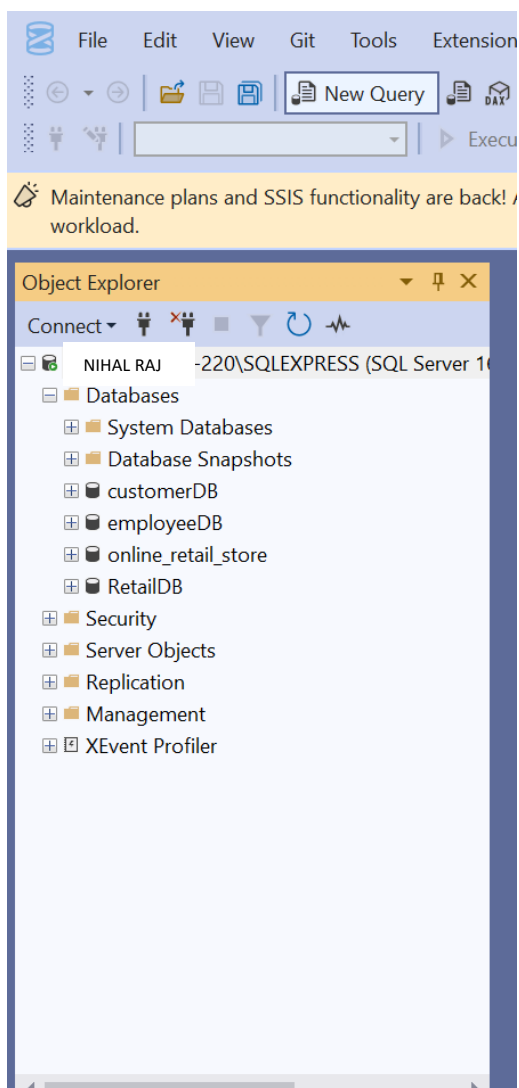


## Lab-2





## Lab-3



```

Developer PowerShell
+ Developer PowerShell
PS C:\Users\KIIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory> dotnet build
>>
Restore complete (0.5s)
RetailInventory succeeded with 4 warning(s) (1.2s) -> bin\Debug\net8.0\RetailInventory.dll
  C:\Users\KIIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory\Models\Product.cs(12,23): warning CS8618: Non-nullable property 'Name' must contain a non-null value when exiting constructor. Consider adding the 'required' modifier or declaring the property as nullable.
  C:\Users\KIIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory\Models\Product.cs(17,25): warning CS8618: Non-nullable property 'Category' must contain a non-null value when exiting constructor. Consider adding the 'required' modifier or declaring the property as nullable.
  C:\Users\KIIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory\Models\Category.cs(12,23): warning CS8618: Non-nullable property 'Name' must contain a non-null value when exiting constructor. Consider adding the 'required' modifier or declaring the property as nullable.
  C:\Users\KIIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory\Models\Category.cs(14,30): warning CS8618: Non-nullable property 'Products' must contain a non-null value when exiting constructor. Consider adding the 'required' modifier or declaring the property as nullable.
Build succeeded with 4 warning(s) in 2.0s
PS C:\Users\KIIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory> dotnet ef migrations add InitialCreate
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'
PS C:\Users\KIIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory> dotnet ef database update
Build started...
Build succeeded.
Acquiring an exclusive lock for migration application. See https://aka.ms/efcore-docs-migrations-lock for more information if this takes too long.
Applying migration '20250706121255_InitialCreate'.
Done.
PS C:\Users\KIIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory>

```

## Lab – 4

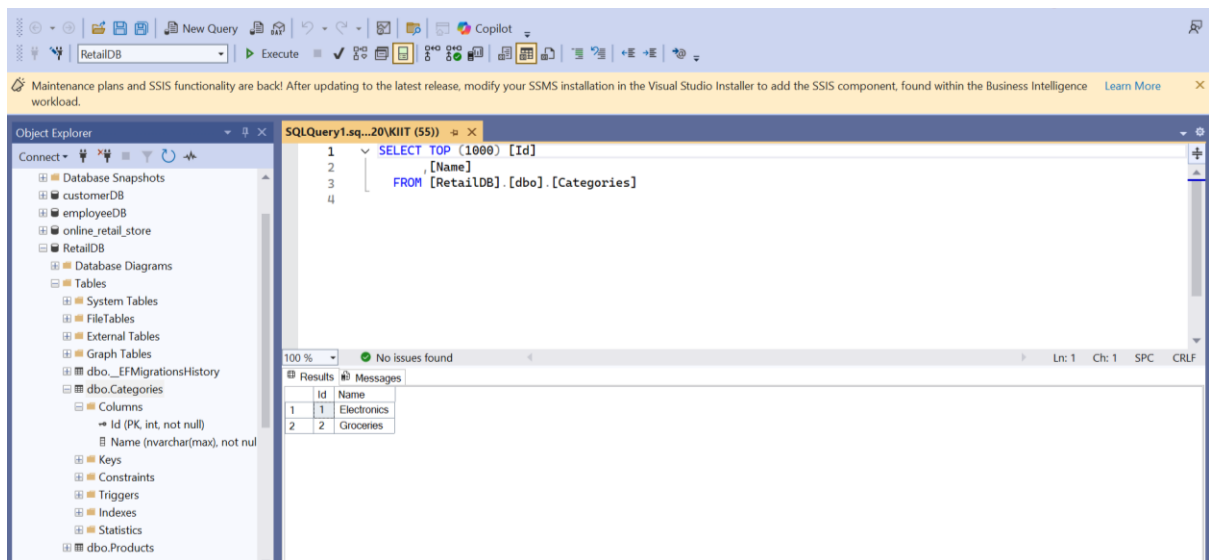
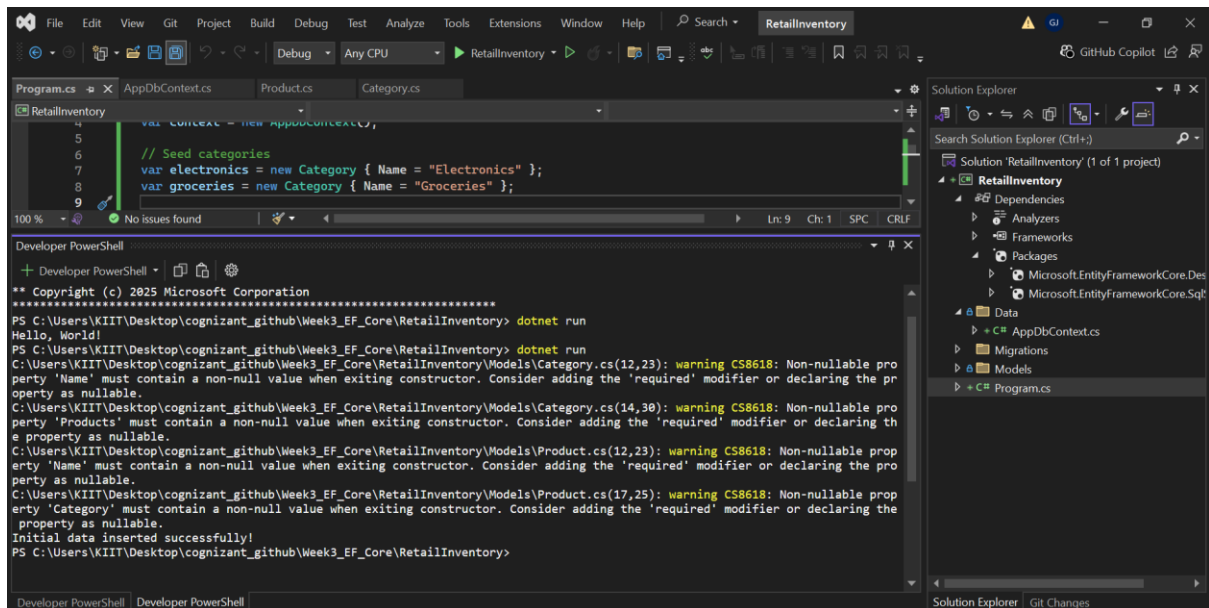
### Code:-

```

lab 4.cs
Miscellaneous Files
1  using RetailInventory.Data;
2  using RetailInventory.Models;
3
4  var context = new AppDbContext();
5
6  // Seed categories
7  var electronics = new Category { Name = "Electronics" };
8  var groceries = new Category { Name = "Groceries" };
9
10 await context.Categories.AddRangeAsync(electronics, groceries);
11
12 // Seed products
13 var product1 = new Product { Name = "Laptop", Price = 75000, Category = electronics };
14 var product2 = new Product { Name = "Rice Bag", Price = 1200, Category = groceries };
15
16 await context.Products.AddRangeAsync(product1, product2);
17
18 // Save changes
19 await context.SaveChangesAsync();
20
21 Console.WriteLine("Initial data inserted successfully!");
22

```

### Output:-



## Lab-5

## Code:-

```
lab_5.cs  X
Miscellaneous Files

1  using Microsoft.EntityFrameworkCore;
2  using RetailInventory.Data;
3  using RetailInventory.Models;
4
5  var context = new AppDbContext();
6
7  // Retrieve all products
8  var products = await context.Products.Include(p => p.Category).ToListAsync();
9  Console.WriteLine("\n All Products:");
10 foreach (var p in products)
11     Console.WriteLine($"{p.Name} - ₹{p.Price} ({p.Category?.Name})");
12
13 // Find product by ID
14 var product = await context.Products.FindAsync(1);
15 Console.WriteLine($"{\n Product with ID = 1: {product?.Name ?? "Not Found"}");
16
17 // First product with price > ₹50,000
18 var expensive = await context.Products.FirstOrDefaultAsync(p => p.Price > 50000);
19 Console.WriteLine($"{\n Expensive Product: {expensive?.Name ?? "None"}");
20
```

## Output:-

```
Developer PowerShell
+ Developer PowerShell
PS C:\Users\KIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory> dotnet run
C:\Users\KIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory\Models\Category.cs(12,23): warning CS8618: Non-nullable property 'Name' must contain a non-null value when exiting constructor. Consider adding the 'required' modifier or declaring the property as nullable.
C:\Users\KIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory\Models\Category.cs(14,30): warning CS8618: Non-nullable property 'Products' must contain a non-null value when exiting constructor. Consider adding the 'required' modifier or declaring the property as nullable.
C:\Users\KIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory\Models\Product.cs(12,23): warning CS8618: Non-nullable property 'Name' must contain a non-null value when exiting constructor. Consider adding the 'required' modifier or declaring the property as nullable.
C:\Users\KIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory\Models\Product.cs(17,25): warning CS8618: Non-nullable property 'Category' must contain a non-null value when exiting constructor. Consider adding the 'required' modifier or declaring the property as nullable.

All Products:
laptop - ₹75000.00 (Electronics)
Rice Bag - ₹1200.00 (Groceries)

Product with ID = 1: Laptop

Expensive Product: Laptop
PS C:\Users\KIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory>
```

## Lab-6

## Code:-

```
lab_6.cs*  lab_5.cs
Miscellaneous Files

4      var context = new AppDbContext();
5      // Update the price of the "Laptop"
6      var product = await context.Products.FirstOrDefaultAsync(p => p.Name == "Laptop");
7      if (product != null)
8      {
9          Console.WriteLine($"Updating price of: {product.Name}");
10         product.Price = 70000;
11         await context.SaveChangesAsync();
12         Console.WriteLine("Price updated!");
13     }
14     else
15     {
16         Console.WriteLine("Laptop not found.");
17     }
18     //Delete the product "Rice Bag"
19     var toDelete = await context.Products.FirstOrDefaultAsync(p => p.Name == "Rice Bag");
20     if (toDelete != null)
21     {
22         Console.WriteLine($"Deleting product: {toDelete.Name}");
23         context.Products.Remove(toDelete);
24         await context.SaveChangesAsync();
25         Console.WriteLine("Product deleted!");
26     }
27     else
28     {
29         Console.WriteLine("Rice Bag not found.");
30     }
31 }
```

Output:-

```
Developer PowerShell
+ Developer PowerShell
** Visual Studio 2022 Developer PowerShell v17.14.7
** Copyright (c) 2025 Microsoft Corporation

PS C:\Users\KIIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory> dotnet run
C:\Users\KIIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory\Models\Category.cs(12,23): warning CS8618: Non-nullable property 'Name' must contain a non-null value when exiting constructor. Consider adding the 'required' modifier or declaring the property as nullable.
C:\Users\KIIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory\Models\Category.cs(14,30): warning CS8618: Non-nullable property 'Products' must contain a non-null value when exiting constructor. Consider adding the 'required' modifier or declaring the property as nullable.
C:\Users\KIIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory\Models\Product.cs(12,23): warning CS8618: Non-nullable property 'Name' must contain a non-null value when exiting constructor. Consider adding the 'required' modifier or declaring the property as nullable.
C:\Users\KIIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory\Models\Product.cs(17,25): warning CS8618: Non-nullable property 'Category' must contain a non-null value when exiting constructor. Consider adding the 'required' modifier or declaring the property as nullable.

Updating price of: Laptop
Price updated!

Deleting product: Rice Bag
Product deleted!
PS C:\Users\KIIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory>
```

Lab-7

Code:-

```
lab_7.cs x lab_6.cs lab_5.cs
Miscellaneous Files RetailInventory.Lab7 Main(string[] args)

10 class Lab7
11 {
12     static async Task Main(string[] args)
13     {
14         using var context = new AppDbContext();
15
16         Console.WriteLine("\nFiltered Products (Price > ₹1000, sorted by Price descending):");
17
18         var filtered = await context.Products
19             .Where(p => p.Price > 1000)
20             .OrderByDescending(p => p.Price)
21             .ToListAsync();
22
23         foreach (var p in filtered)
24         {
25             Console.WriteLine($"- {p.Name} - ₹{p.Price}");
26         }
27
28         Console.WriteLine("\nProduct DTOs (Name + Price only):");
29
30         var productDTOs = await context.Products
31             .Select(p => new { p.Name, p.Price })
32             .ToListAsync();
33
34         foreach (var dto in productDTOs)
35         {
36             Console.WriteLine($"• {dto.Name}: ₹{dto.Price}");
37         }
38     }
}
```

## Output:-

```
Developer PowerShell
+ Developer PowerShell v17.14.7
** Visual Studio 2022 Developer PowerShell v17.14.7
** Copyright (c) 2025 Microsoft Corporation
*****
PS C:\Users\KIIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory> dotnet run
C:\Users\KIIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory\Models\Category.cs(12,23): warning CS8618: Non-nullable property 'Name' must contain a non-null value when exiting constructor. Consider adding the 'required' modifier or declaring the property as nullable.
C:\Users\KIIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory\Models\Category.cs(14,30): warning CS8618: Non-nullable property 'Products' must contain a non-null value when exiting constructor. Consider adding the 'required' modifier or declaring the property as nullable.
C:\Users\KIIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory\Models\Product.cs(12,23): warning CS8618: Non-nullable property 'Name' must contain a non-null value when exiting constructor. Consider adding the 'required' modifier or declaring the property as nullable.
C:\Users\KIIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory\Models\Product.cs(17,25): warning CS8618: Non-nullable property 'Category' must contain a non-null value when exiting constructor. Consider adding the 'required' modifier or declaring the property as nullable.

Filtered Products (Price > ₹1000, sorted by Price descending):
- Laptop - ₹70000.00

Product DTOs (Name + Price only):
• Laptop: ₹70000.00
PS C:\Users\KIIIT\Desktop\cognizant_github\Week3_EF_Core\RetailInventory>
```