Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

5

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| 1 | Design and implement an ASP.NET Core Web API with endpoints for GET (retrieve products), POST (add products), PUT (update products), and DELETE (remove products) methods to manage a product inventory system. |
| 2 | Create an ASP.NET Core Web API for managing student records, enabling CRUD operations (GET, POST, PUT, and DELETE) for adding, retrieving, updating, and deleting student information. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Submitted On:

**\_\_\_\_\_\_\_\_\_\_\_**

Task 01: Design and implement an ASP.NET Core Web API with endpoints for GET (retrieve products), POST (add products), PUT (update products), and DELETE (remove products) methods to manage a product inventory system.

Solution:

Product Model:

namespace CCLab5.Models

{

public class Product

{

public long Id { get; set; }

public string Name { get; set; }

public int Price { get; set; }

public int Quantity { get; set; }

public string Description { get; set; }

}

}

Product Context:

public class ProductContext : DbContext

{

public ProductContext(DbContextOptions<ProductContext> options) : base(options) { }

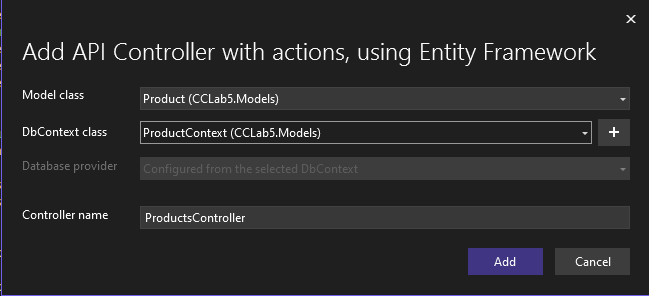
public DbSet<Product> Products { get; set; } = null;

}

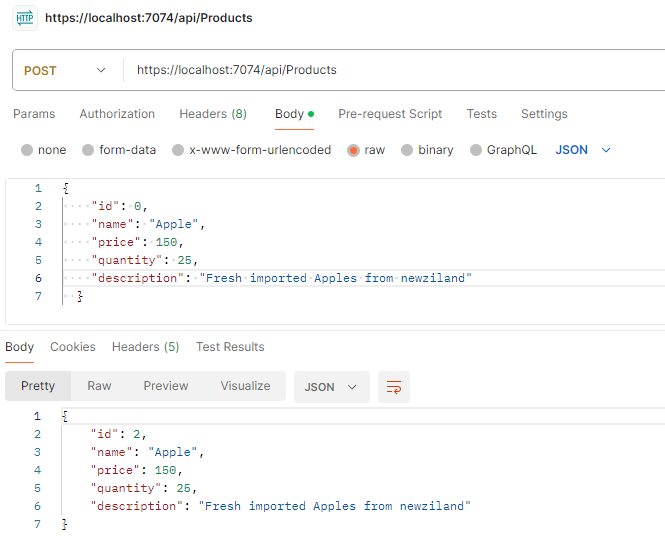
Program Class:

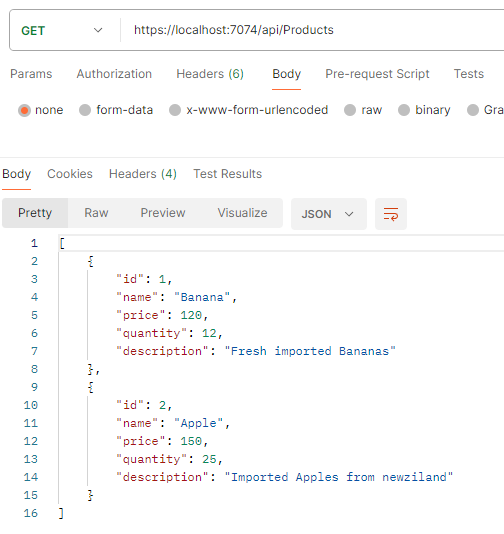
builder.Services.AddDbContext<ProductContext>(opt => opt.UseInMemoryDatabase("Product"));

Products Controller:

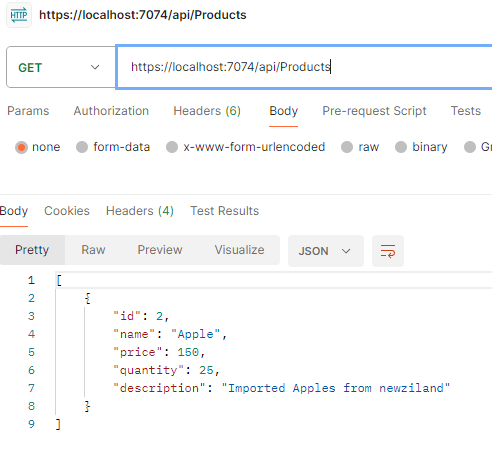


Output:









Task 02: Create an ASP.NET Core Web API for managing student records, enabling CRUD operations (GET, POST, PUT, and DELETE) for adding, retrieving, updating, and deleting student information.

Solution:

Student Model:

public class Student

{

public int Id { get; set; }

public string Name { get; set; }

public string Enroll { get; set; }

public string Department { get; set; }

public int Semester { get; set; }

public string Batch { get; set; }

}

Student Context:

public class StudentContext:DbContext

{

public StudentContext(DbContextOptions<StudentContext> options) : base(options) { }

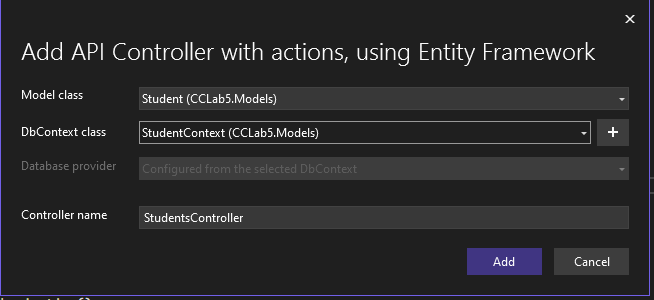
public DbSet<Student> Students { get; set; }

}

Program Class:

builder.Services.AddDbContext<StudentContext>(opt => opt.UseInMemoryDatabase("Student"));

Students Controller:



Output:

