**Muhammad imran   (213038)**

**Assignment no :  06**

**Activity: 01**

**Home Controller:**

using Microsoft.AspNetCore.Mvc;

namespace WebApplication7.Controllers

{

    public class HomeController1 : Controller

    {

        public IActionResult Index()

        {

            return View();

        }

    }

}

**Index.cshtml :**

<html>

    <head>

    </head>

    <body>

        @model List<WebApplication7.Models.Product>

        <ul>

            @foreach(var product in Model)

            {

               <li>@product.Name</li>

            }

        </ul>

    </body>

</html>

**Product.cs:**

namespace WebApplication7.Models

{

    public class Product

    {

        public int ProductId { get; set; }

        public string Name { get; set; }

        public decimal Price { get; set; }

    }

}

\_**ViewStart.cshtml:**

@{

    Layout = "\_Layout";

}

**\_Layout.cshtml:**

<!DOCTYPE html>

<html>

    <head>

    <meta name="viewport" content="width=device-width" />

    <title>@ViewBag.Title</title>

    </head>

    <body>

        <div>

        @RenderBody()

        </div>

    </body>

</html>

**HomeController.cs :**

using Microsoft.AspNetCore.Mvc;

using System.Diagnostics;

using WebApplication7.Models;

using static WebApplication7.Controllers.ViewResult;

using static WebApplication7.Models.Order;

namespace WebApplication7.Controllers

{

    public class ViewResult : Controller

    {

        private readonly ILogger<ViewResult> \_logger;

        public ViewResult(ILogger<ViewResult> logger)

        {

            \_logger = logger;

        }

        public IActionResult Index()

        {

            return View();

        }

        public IActionResult Privacy()

        {

            return View();

        }

        [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]

        public IActionResult Error()

        {

            return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });

        }

        List<Product> products = new List<Product>

        {

            new Product

            {

                Name = "Mobile Phone",

                Price = 50000

            },

             new Product

            {

                Name = "Tablet ",

                Price = 100000

            },

              new Product

            {

                Name = "Laptoop",

                Price = 80000

            },

        };

    }

    Order order = new Order();

    order.Products Products;

    order.Total Products.Sum(product => product.Price);

        return View(order);

}

**ErrorViewModel.cs :**

namespace WebApplication7.Models

{

    public class ErrorViewModel

    {

        public string? RequestId { get; set; }

        public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);

    }

}

**Order.cs:**

namespace WebApplication7.Models

{

    public class Order

    {

        public int orderId { get; set; }

        public List<Product> products { get; set; }

        public decimal total { get; set; }

        internal class Products

        {

        }

    }

}

**Index1.cshtml:**

<html>

<head>

</head>

<body>

    @model List<WebApplication7.Models.Product>

    <ul>

       <tr>

           <th>

               Product Name

           </th>

           <th>

               Price

           </th>

       </tr>

       @foreach(var Product in Model.Products)

        {

            <tr>

                <td>@Product.Name</td>

                <td>@Product.Price</td>

            </tr>

        }

        <tr>

            <td><b>Total</b></td>

            <td>@Model.Total</td>

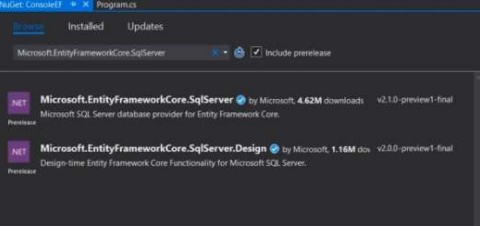
        </tr>

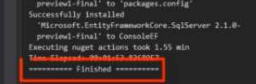
    </ul>

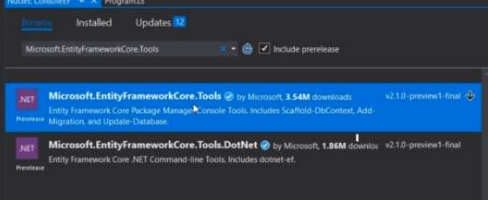
</body>

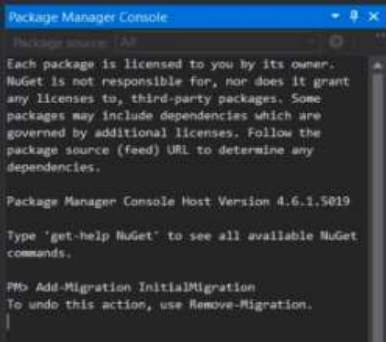
</html>

**Activity: 02**









**Code:**

using System;

using System.Data;

using System.Data.SqlClient;

class Employee

{

    public int EmployeeId { get; set; }

    public string EmployeeName { get; set; }

    public decimal Salary { get; set; }

    public string Description { get; set; }

}

class Program

{

    static string connectionString = "YourConnectionStringHere"; // Replace this with your actual connection string

    static void Main()

    {

        // Example usage:

        // Create an employee

        Employee newEmployee = new Employee

        {

            EmployeeName = "John Doe",

            Salary = 50000,

            Description = "Senior Developer"

        };

        int newEmployeeId = CreateEmployee(newEmployee);

        // Update an employee

        newEmployee.EmployeeId = newEmployeeId;

        newEmployee.Salary = 60000;

        UpdateEmployee(newEmployee);

        // Delete an employee

        DeleteEmployee(newEmployeeId);

    }

    static int CreateEmployee(Employee employee)

    {

        using (SqlConnection connection = new SqlConnection(connectionString))

        {

            string query = "INSERT INTO Employees (EmployeeName, Salary, Description) VALUES (@Name, @Salary, @Description); SELECT SCOPE\_IDENTITY();";

            SqlCommand command = new SqlCommand(query, connection);

            command.Parameters.AddWithValue("@Name", employee.EmployeeName);

            command.Parameters.AddWithValue("@Salary", employee.Salary);

            command.Parameters.AddWithValue("@Description", employee.Description);

            connection.Open();

            int newEmployeeId = Convert.ToInt32(command.ExecuteScalar());

            return newEmployeeId;

        }

    }

    static void UpdateEmployee(Employee employee)

    {

        using (SqlConnection connection = new SqlConnection(connectionString))

        {

            string query = "UPDATE Employees SET Salary = @Salary WHERE EmployeeId = @EmployeeId";

            SqlCommand command = new SqlCommand(query, connection);

            command.Parameters.AddWithValue("@Salary", employee.Salary);

            command.Parameters.AddWithValue("@EmployeeId", employee.EmployeeId);

            connection.Open();

            command.ExecuteNonQuery();

        }

    }

    static void DeleteEmployee(int employeeId)

    {

        using (SqlConnection connection = new SqlConnection(connectionString))

        {

            string query = "DELETE FROM Employees WHERE EmployeeId = @EmployeeId";

            SqlCommand command = new SqlCommand(query, connection);

            command.Parameters.AddWithValue("@EmployeeId", employeeId);

            connection.Open();

            command.ExecuteNonQuery();

        }

    }

}