**VIJAY M**

**Mini project Module-1 - date 08-08-25**

**WebApp using .net mvc Model**

create a mvc webapp with the ability to add product and view in the table

**AppContext.cs**

```
using Microsoft.EntityFrameworkCore;
using WebApplication1.Models;



namespace WebApplication1.Context
{
    public class AppDbContext: DbContext
    {
        public AppDbContext(DbContextOptions<AppDbContext> options) : base(options)
        {

        }

       public DbSet<student> students { get; set; }
        public DbSet<Product> prod { get; set; }
    }
}
```

Product.cs

```
using System.ComponentModel.DataAnnotations;
```

```
namespace WebApplication1.Models
{
    public class Product
    {
        [Required]
        public int Id { get; set; }
        public string Name { get; set; }
        public string Description { get; set; }
        public int Price { get; set; }
    }
}
```

Product.chtml

```
@{
    ViewData["Title"] = "Product    Page";
}

@model List<Product>

<div class="d-flex justify-content-between align-items-center mb-4">
    <h1 class="display-4 m-0 mx-auto">Welcome to Product</h1>
    <a class="btn btn-primary" asp-action="CreateOrEditProd">Add Product</a>
</div>



<table class="table">
    <thead>
        <tr>
            <th>Id</th>
            <th>Name</th>
            <th>Description</th>
            <th>Price</th>
        </tr>
    </thead>
    <tbody>
        @{
            foreach (var prod in Model)
            {
                <tr>
                    <td>@prod.Id</td>
                    <td>@prod.Name</td>
                    <td>@prod.Description</td>
                    <td>@prod.Price</td>
                </tr>
            }
        }
    </tbody>

</table>
```

CreateOrEditProd.chtml

```
@{
    ViewData["Title"] = "create Product Page";
}

@model WebApplication1.Models.Product
<div class="text-center">
    <h1 class="display-4"> create or edit Product</h1>
</div>
```

```
<form asp-action="AddProd">
    @* <lable class="mt-4">StudentId</lable>
    <input class="form-control" type="text" placeholder="" asp-for="StudentID" /> *@

    <lable class="mt-4">Name</lable>
    <input class="form-control" type="text" placeholder="" asp-for="Name" />

    <lable class="mt-4">Description</lable>
    <input class="form-control" type="text" placeholder="" asp-for="Description" />

    <lable class="mt-4">Price</lable>
    <input class="form-control" type="text" placeholder="" asp-for="Price" />



    <button class="btn" type="submit">Submit</button>
</form>
```

HomeController.cs

```
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using System.Diagnostics;
using WebApplication1.Context;
using WebApplication1.Models;

namespace WebApplication1.Controllers
{
    public class HomeController : Controller
    {
        private readonly ILogger<HomeController> _logger;

        private AppDbContext appDbContext { get; set; }
        }




        public HomeController(ILogger<HomeController> logger,AppDbContext appDbContext)
        {
            _logger = logger;
            this.appDbContext = appDbContext;
        }

        public IActionResult Index()
        {
            return View();
        }

        public IActionResult AddStudent(student stud)
        {
            appDbContext.students.Add(stud);
            appDbContext.SaveChanges();
            return RedirectToAction("Student");
        }


        public IActionResult Student()
        {

            var allstudents=appDbContext.students.ToList();
            var totalAge = allstudents.Sum(x => x.Age);
            ViewBag.totalAge = totalAge;
            return View(allstudents);
        }

        [HttpPost]
```

```
public IActionResult UpdateAction(int id, string field, string value)
{
    var student = appDbContext.students.FirstOrDefault(s => s.StudentID == id);
    if (student == null) return NotFound();

    switch (field)
    {
        case "FirstName":
            student.FirstName = value;
            break;
        case "LastName":
            student.LastName = value;
            break;
        case "Age":
            if (int.TryParse(value, out int age))
                student.Age = age;
            break;
        case "Gender":
            student.Gender = value;
            break;
        case "Department":
            student.Department = value;
            break;
        default:
            return BadRequest("Invalid field");
    }

    appDbContext.SaveChanges();
    return RedirectToAction("Student");
}
```

```
public IActionResult Product()
{

    var allprod = appDbContext.prod.ToList();
    return View(allprod);
}

public IActionResult AddProd(Product prod1)
{
    appDbContext.prod.Add(prod1);
    appDbContext.SaveChanges();
    return RedirectToAction("Product");
}
public IActionResult Privacy()
{
    return View();
}
public IActionResult Createoreditstudent()
{
    return View();
}

public IActionResult CreateOrEditProd()
{
    return View();
}

public IActionResult DeleteAction(int id)
{
    var stud = appDbContext.students.FirstOrDefault(a => a.StudentID == id);
```

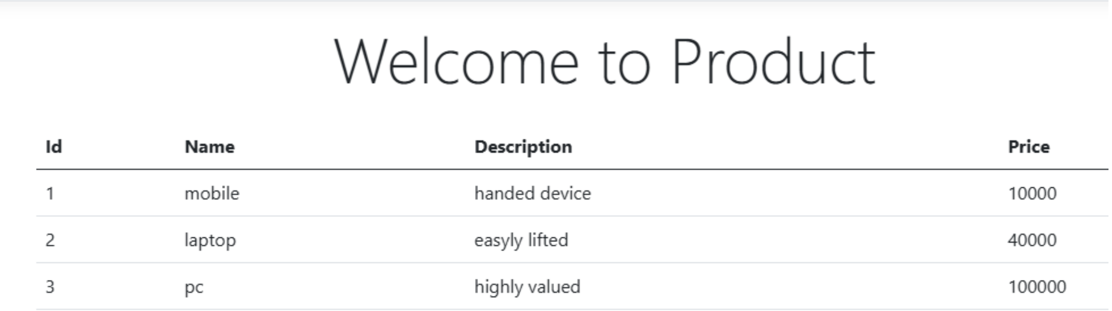
```
            if (stud != null)
            {
                _logger.LogWarning($"{stud.FirstName} is being deleted");
                appDbContext.students.Remove(stud);
                appDbContext.SaveChanges();
            }
            return RedirectToAction("Student");
        }

        [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
        public IActionResult Error()
        {
            return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });
        }
    }
}
```

## Program.cs

```
using Microsoft.EntityFrameworkCore;
using Serilog;
using WebApplication1.Context;

var builder = WebApplication.CreateBuilder(args);

Log.Logger=new LoggerConfiguration()
    .MinimumLevel.Information()
    .WriteTo.Console()
    .WriteTo.File("Logs/log.txt",rollingInterval:RollingInterval.Minute)
    .Enrich.FromLogContext()
    .CreateBootstrapLogger();
```

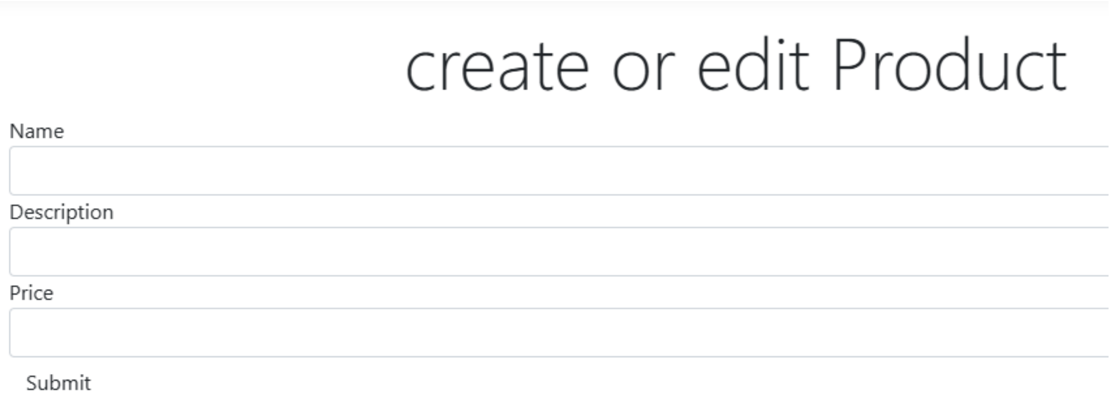
```
builder.Host.UseSerilog();

// Add services to the container.
builder.Services.AddControllersWithViews();



const string connectionstring= "Data Source=PTPLL605;" +
                    "Initial Catalog=sampledb;" +
                    "Integrated Security=True;" +
            "TrustServerCertificate=True;";

builder.Services.AddDbContext<AppDbContext>(x=>x.UseSqlServer(connectionstring));

var app = builder.Build();


// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
}
app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.MapControllerRoute(
    name: "default",
        pattern: "{controller=Home}/{action=Index}/{id?}");

app.Run();
```

**Output**



# Welcome to Product

| Id | Name | Description | Price |
|---|---|---|---|
| 1 | mobile | handed device | 10000 |
| 2 | laptop | easyly lifted | 40000 |
| 3 | pc | highly valued | 100000 |



# create or edit Product

Name

Description

Price