

Interrogation Panel - Data Transferring

Description

Objective: The objective is to develop an application that utilizes ViewBag, ViewData, and TempData concepts to pass data between the controller and views.

InterrogationController.cs

```
using Microsoft.AspNetCore.Mvc;
```

```
namespace ASPNET6_Template.Controllers
```

```
{
```

```
    public class InterrogationController : Controller
```

```
    {
```

```
        //Implement your code here
```

```
        public ActionResult MemberName(int registrationNumber, string name)
```

```
        {
```

```
            ViewBag.Details = registrationNumber + " " + name;
```

```
            return View();
```

```
        }
```

```
        public ActionResult Solution(string solution)
```

```
        {
```

```
            TempData["Solution"]=solution;
```

```
            return RedirectToAction("FinalSolution");
```

```
        }
```

```
public ActionResult Problem(string problem)
{
    ViewData["Problem"]=problem;
    return View();
}

public ActionResult FinalSolution(){
    string final= TempData["Solution"] as string;
    return View();
}
}
```

FinalSolution.cshtml

<!-- Implement your code here -->

```
@{
    ViewData["Title"] = "Final Solution";
}
```

<h1> The Final Solution is: @TempData["Solution"] </h1>

MemberName.cshtml

<!-- Implement your code here -->

```
@{  
    ViewData["Title"] = "Member Name";  
}
```

<h2> Member Details </h2>

<p> @ViewBag.Details</p>

Program.cshtml

<!-- Implement your code here -->

```
@{  
    ViewData["Title"] = "Problem";  
}
```

<h2> @ViewData["Message"] </h2>

<p>@ViewData["Problem"] </p>

<a href="@Url.Action("MemberName", "Interrogation", new {registrationNumber=
"101126", name= "Linin Park"})">

Go to Member Name

Program.cs

```
var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllersWithViews();

var app = builder.Build();

// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
    // The default HSTS value is 30 days. You may want to change this for production
    scenarios, see https://aka.ms/aspnetcore-hsts.
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.MapControllerRoute(
    name: "default",
```

```
pattern: "{controller=Interrogation}/{action=MemberName}/{id?}");
```

```
app.Run();
```

Interrogation Panel - Component

Description

Objective: The objective is to develop an application that utilizes partial views and model binding concepts to achieve the desired functionality.

InterrogationController.cs

```
using Microsoft.AspNetCore.Mvc;
```

```
namespace ASPNET6_Template.Controllers
```

```
{
```

```
    public class InterrogationController : Controller
```

```
    {
```

```
        //Implement your code here
```

```
        public IActionResult GuideMe()
```

```
        {
```

```
            return View();
```

```
        }
```

```
        public ActionResult Form()
```

```

    {
        return View();
    }

    [HttpPost]
    [ValidateAntiForgeryToken]

    public ActionResult Form(IFormCollection collection)
    {
        try
        {
            return RedirectToAction(nameof(GuideMe));
        }
        catch
        {
            return View();
        }
    }
}

```

Program.cs

```

namespace ASPNET6_Template.Models
{
    public class Problem
    {

```

```
//Implement your code here
```

```
public int Id {get; set;}
```

```
public string InterrogatorsProblem {get; set;}
```

```
public string Choice1 {get; set;}
```

```
public string Choice2 {get; set;}
```

```
public string Choice3 {get; set;}
```

```
public string Choice4 {get; set;}
```

```
public string Solution {get; set;}
```

```
}
```

```
}
```

Form.cshtml

```
<!-- Implement your code here -->
```

```
@model ASPNET6_Template.Models.Problem
```

```
@{
```

```
    ViewData["Title"] = "Form";
```

```
}
```

```
<h2> Create a Problem </h2>
```

```
<form asp-action="Form" method="post">
```

```
<div>
```

```
<label asp-for="Id"></label>

<input asp-for="Id" />

</div>
```

```
<div>

<label asp-for="InterrogatorsProblem"></label>

<input asp-for="InterrogatorsProblem" />

</div>
```

```
<div>

<label asp-for="Choice1"></label>

<input asp-for="Choice1" />

</div>
```

```
<div>

<label asp-for="Choice2"></label>

<input asp-for="Choice2" />

</div>
```

```
<div>

<label asp-for="Choice3"></label>

<input asp-for="Choice3" />

</div>
```

```
<div>

<label asp-for="Choice4"></label>
```



```
<input asp-for="Choice4" />
```

```
</div>
```

```
<div>
```

```
<label asp-for="Solution"></label>
```

```
<input asp-for="Solution" />
```

```
</div>
```

```
<button type="Submit"> Create </button>
```

```
</form>
```

GuideMe.cshtml

```
<!-- Implement your code here -->
```

```
@{
```

```
    ViewData["Title"]="GuideMe";
```

```
}
```

```
<h1> Welcome to Interrogation Panel !!! </h1>
```

```
<partial name="_ReferenceButton" />
```

_ReferenceButton.cshtml

<!-- Implement your code here -->

<p>

**<a asp-action="Form"> Create a new problem **

</p>

Program.cs

```
var builder = WebApplication.CreateBuilder(args);
```

```
// Add services to the container.
```

```
builder.Services.AddControllersWithViews();
```

```
var app = builder.Build();
```

```
// Configure the HTTP request pipeline.
```

```
if (!app.Environment.IsDevelopment())
```

```
{
```

```
    app.UseExceptionHandler("/Home/Error");
```

```
    // The default HSTS value is 30 days. You may want to change this for production  
    scenarios, see https://aka.ms/aspnetcore-hsts.
```

```
    app.UseHsts();
```

```
}
```

```
app.UseHttpsRedirection();  
app.UseStaticFiles();  
  
app.UseRouting();  
  
app.UseAuthorization();  
  
app.MapControllerRoute(  
    name: "default",  
    pattern: "{controller=Interrogation}/{action=GuideMe}/{id?}");  
  
app.Run();
```

InterrogationController.cs

```
using Microsoft.AspNetCore.Mvc;  
  
namespace ASPNET6_Template.Controllers  
{  
    public class InterrogationController : Controller  
    {  
        //Implement your code here  
        public ActionResult MemberName(int registrationNumber, string name)  
        {
```

```
ViewBag.Details = registrationNumber + " " + name;  
return View();  
}
```

```
public ActionResult Solution(string solution)  
{  
    TempData["Solution"]=solution;  
    return RedirectToAction("FinalSolution");  
}
```

```
public ActionResult Problem(string problem)  
{  
    ViewData["Problem"]=problem;  
    return View();  
}
```

```
public ActionResult FinalSolution(){  
    string final= TempData["Solution"] as string;  
    return View();  
}  
}  
}
```

FinalSolution.cshtml

<!-- Implement your code here -->

@{

 ViewData["Title"] = "Final Solution";

}

<h1> The Final Solution is: @TempData["Solution"] </h1>

MemberName.cshtml

<!-- Implement your code here -->

@{

 ViewData["Title"] = "Member Name";

}

<h2> Member Details </h2>

<p> @ViewBag.Details</p>

Problem.cshtml

<!-- Implement your code here -->

@{

 ViewData["Title"] = "Problem";

}

<h2> @ViewData["Message"] </h2>

<p>@ViewData["Problem"] </p>

<a href="@Url.Action("MemberName", "Interrogation", new {registrationNumber= "101126",
name= "Linin Park"})">

Go to Member Name

Program.cs

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllersWithViews();

var app = builder.Build();

// Configure the HTTP request pipeline.

if (!app.Environment.IsDevelopment())

{

 app.UseExceptionHandler("/Home/Error");

// The default HSTS value is 30 days. You may want to change this for production scenarios, see <https://aka.ms/aspnetcore-hsts>.

```
app.UseHsts();  
}
```

```
app.UseHttpsRedirection();  
app.UseStaticFiles();
```

```
app.UseRouting();
```

```
app.UseAuthorization();
```

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

```
app.Run();
```

Interrogation Panel - Razor Page

Description

Objective: The objective is to develop an application using the Model-View-Controller (MVC) architecture with Razor Pages.

```
using Microsoft.AspNetCore.Mvc;
using ASPNET6_Template.Data;
using ASPNET6_Template.Models;
using System.Linq;

namespace ASPNET6_Template.Controllers
{
    public class InterrogationController : Controller
    {
        InterrogationDbContext db;

        public InterrogationController(InterrogationDbContext _db){
            db = _db;
        }

        [HttpGet]
        public ActionResult Craft(){
            return View();
        }
    }
}
```


[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Craft(Problem model)

{

try

{

db.Problems.Add(model);

db.SaveChanges();

return RedirectToAction("Catalog","Interrogation");

}

catch

{

return RedirectToAction("Catalog","Interrogation");

}

}

[HttpGet]

public ActionResult Catalog()

{

return View(db.Problems.ToList());

}

}

}

InterrogationDbContext.cs

```
using Microsoft.EntityFrameworkCore;
using ASPNET6_Template.Models;

namespace ASPNET6_Template.Data
{
    public class InterrogationDbContext : DbContext
    {
        public InterrogationDbContext(DbContextOptions<InterrogationDbContext> options) :
        base(options)
        {
        }

        public DbSet<Problem> Problems { get; set; }
    }
}
```

Problem.cs

```
using System.ComponentModel.DataAnnotations;

namespace ASPNET6_Template.Models
{
    public class Problem
    {
        [Key]
```

```
public int Id { get; set; }
```

```
[Required(ErrorMessage = "Please provide Problem")]
```

```
public string InterrogatorsProblem { get; set; }
```

```
[Required(ErrorMessage = "Please provide Choice1")]
```

```
public string Choice1 { get; set; }
```

```
[Required(ErrorMessage = "Please provide Choice2")]
```

```
public string Choice2 { get; set; }
```

```
[Required(ErrorMessage = "Please provide Choice3")]
```

```
public string Choice3 { get; set; }
```

```
[Required(ErrorMessage = "Please provide Choice4")]
```

```
public string Choice4 { get; set; }
```

```
[Required(ErrorMessage = "Please provide Solution")]
```

```
public string Solution { get; set; }
```

```
}
```

```
}
```

Program.cs

```
using ASPNET6_Template.Data;
using Microsoft.EntityFrameworkCore;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddControllersWithViews();
builder.Services.AddDbContext<InterrogationDbContext>
    (options => options.UseSqlServer
        (builder.Configuration.GetConnectionString("DBConnection")));

var app = builder.Build();

// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
    // The default HSTS value is 30 days.
    // You may want to change this for production scenarios,
    // see https://aka.ms/aspnetcore-hsts.
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();
```

```
app.UseRouting();
```

```
app.UseAuthorization();
```

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

```
app.Run();
```

Interrogation Panel - DB Initialize

Description

Objective: The objective is to develop an application using the Model-View-Controller (MVC) architecture with the database as part of the initialization process.

DbIntializer.cs

```
using ASPNET6_Template.Models;
using Microsoft.EntityFrameworkCore;

namespace ASPNET6_Template.Data
{
    public class DbInitializer : IDbInitializer
    {
        private readonly InterrogationDbContext _context;

        public DbInitializer(InterrogationDbContext context)
        {
            _context = context;
        }
    }
}
```

```
public void Initialize()
{
    try
    {
        if(!_context.Database.GetPendingMigrations().Count() > 0)
        {
            _context.Database.Migrate();
        }
    }
    catch(Exception ex){ }
    if(!_context.Problems.Any())
    {
        var questions = new List<Problem>()
        {
            new Problem()
            {
                InterrogatorsProblem = "The First metal used by the
man was",
                Choice1 = "Iron",
                Choice2 = "Copper",
```

```
        Choice3 = "Alluminium",
        Choice4 = "Gold",
        Solution = "Copper"
    },
    new Problem()
    {
        InterrogatorsProblem = "Which of the following is a
balanced fertiliser for plants ?",
        Choice1 = "Urea",
        Choice2 = "Ammonia sulphate",
        Choice3 = "Nitrates",
        Choice4 = "Compost",
        Solution = "Compost"
    }
};
_context.Problems.AddRange(questions);
_context.SaveChanges();
}
}
}
```



```
}
```

IDbInitializer.cs

```
namespace ASPNET6_Template.Data
```

```
{
```

```
    public interface IDbInitializer
```

```
    {
```

```
        void Initialize();
```

```
    }
```

```
}
```

InterrogationDbContext.cs

```
using ASPNET6_Template.Models;
```

```
using Microsoft.EntityFrameworkCore;
```

```
namespace ASPNET6_Template.Data
```

```
{
```

```
    public class InterrogationDbContext : DbContext
```

```
    {
```

```
public
InterrogationDbContext(DbContextOptions<InterrogationDbContext> options) : base(options)
{

}

public DbSet<Problem> Problems { set; get; }
}
}
```

Problem.cs

```
using System.ComponentModel.DataAnnotations;

namespace ASPNET6_Template.Models
{
    public class Problem
    {
        [Key]
        public int Id { get; set; }

        [Required(ErrorMessage = "Please provide Problem")]
    }
}
```

```
public string InterrogatorsProblem { get; set; }  
[Required(ErrorMessage = "Please provide Choice1")]  
public string Choice1 { get; set; }  
[Required(ErrorMessage = "Please provide Choice2")]  
public string Choice2 { get; set; }  
[Required(ErrorMessage = "Please provide Choice3")]  
public string Choice3 { get; set; }  
[Required(ErrorMessage = "Please provide Choice4")]  
public string Choice4 { get; set; }  
[Required(ErrorMessage = "Please provide Solution")]  
public string Solution { get; set; }  
}  
}
```

Program.cs

```
using ASPNET6_Template.Data;  
using Microsoft.EntityFrameworkCore;  
  
var builder = WebApplication.CreateBuilder(args);  
  
// Add services to the container.
```

```
builder.Services.AddControllersWithViews();

builder.Services.AddDbContext<InterrogationDbContext>(options =>

    options.UseSqlServer(builder.Configuration.GetConnectionString("DBConnection"))

);

builder.Services.AddScoped<IDbInitializer, DbInitializer>(); //
Corrected class name spelling
```

```
var app = builder.Build();
```

```
// Configure the HTTP request pipeline.
```

```
if (!app.Environment.IsDevelopment())
```

```
{
```

```
    app.UseExceptionHandler("/Home/Error");
```

```
    app.UseHsts(); // Default HSTS value is 30 days. Consider
adjusting for production.
```

```
}
```

```
app.UseHttpsRedirection();
```

```
app.UseStaticFiles();
```

```
app.UseRouting();
```

SeedDatabase(app); // Pass 'app' explicitly to SeedDatabase

app.UseAuthorization();

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

app.Run();

**// Corrected SeedDatabase method with proper dependency
resolution**

**void SeedDatabase(WebApplication app)
{
 using var scope = app.Services.CreateScope();
 var dbInitializer =
scope.ServiceProvider.GetRequiredService<IDbInitializer>(); //
Fixed spelling
 dbInitializer.Initialize();
}**

