



sagarmatha

COLLEGE OF SCIENCE & TECHNOLOGY
TRIBHUVAN UNIVERSITY AFFILIATE

Lab Report On: NCC

Lab Report No: 04

Submitted By:

Bijay Shrestha

Roll no: 11

Group: BSc.CSIT

Submitted To:

CSIT Department

Date: 8 Dec

1.) NWP to show state management using session to set Session and retrieve Session Value in your ASP.NET Core Program.

⇒ PROCESS INVOLVED

→ First, ASP.NET Core Web application, named as 'Lab4' is created with MVC pattern.

→ In 'Startup.cs' file, the two line of code is added as :-

```
public void ConfigureServices (IServiceCollection  
services)
```

```
{
```

```
services.AddControllersWithViews ();
```

```
services.AddSession ();
```

```
}
```

```
public void Configure (IApplicationBuilder app,  
IWebHostEnvironment env)
```

```
{
```

```
    {
```

```
        app.UseSession ();
```

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

```
    }
```

```
}
```

→ A controller named as "TestSessionController.cs" was created.

→ The action methods to set and get session are written as :-

```
public class TestSessionController : Controller
{
    public ActionResult Index()
    {
        HttpContext.Session.SetString("Name", "Bijay Shrestha");
        return View();
    }

    public ActionResult getSessionData()
    {
        string sessData = HttpContext.Session.GetString("Name");

        ViewData["name"] = sessData;
        return View();
    }
}
```

→ The corresponding View Page for action methods are :-
Index.cshtml

```
@{
    ViewData["Title"] = "Index";
    layout = "~/Views/Shared/_Layout.cshtml";
}
<h1>Index </h1>
```

getSessionData.cshtml

```
@{
    ViewData["Title"] = "getSessionData";
    layout = "~/Views/Shared/_Layout.cshtml";
}
<h1>getSessionData </h1>
@if (ViewData["name"] != null) {
    <h2>The name from session is @ViewData["name"] </h2>
} else {
    <h2>There is no value in session. </h2>
}
```

2.) Write a simple program to show use of TempData from one action method 1 to action method 2

PROCESS INVOLVED

→ In a controller folder, a controller named as "TestTempDataController" is created with two action methods, 'Method1' and 'method2' to set TempData and get TempData.

```
public class TestTempDataController : Controller
{
    public IActionResult Method1 ()
    {
        TempData ["UsesId"] = 11;
        return View ();
    }
    public string Method2 ()
    {
        if (TempData ["UsesId"] == null)
        {
            return "There is no value in TempData";
        }
        string usesId = TempData ["UsesId"].ToString ();
        return $"The uses id from temp data is : {usesId}";
    }
}
```

→ The usesId in TempData is stored temporarily which can be seen from changing page from method1 to method2. When the method2 page is refreshed the TempData becomes empty.

3.) WAP in ASP.NET Core to process QueryStrings.

PROCESS INVOLVED

→ A controller "TestQueryStringController.cs" is created with action methods as

```
public class TestQueryStringController : Controller
{
    public IActionResult Index (int id, string name)
    {
        ViewData ["id"] = id;
        ViewData ["name"] = name;
        return View();
    }
    public string Display (int id, string name)
    {
        return $"Id is {id} In Name is {name}";
    }
}
```

→ The View page for Index is :-

```
@{
    ViewData ["Title"] = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h1> Index </h1>
<h2> Id is @ViewData ["id"] </h2>
<h2> Name is @ViewData ["name"] </h2>
```