**Session and state management**

**Session management:**

Session management used to store a data in a server for a specific session.

After the session the data will be erased.

**Session state can be configured in two main ways:**

**1.We can register the session in program.cs**

builder.Services.AddSession(options =>

{

options.IdleTimeout = TimeSpan.FromMinutes(20); // Session timeout

options.Cookie.HttpOnly = true; // To make the Http acces only

options.Cookie.IsEssential = true; //

});

1. We can register as Web.config:

<configuration>

<system.web>

<sessionState mode="InProc" cookieless="false" timeout="20"/>

</system.web>

</configuration>

1. Session ID: A unique identifier assigned to each user's session.
2. Session State: The data stored for a specific user during their visit.
3. Session Timeout: The length of time a session remains active before it

**Setting session values:  
  
Here I set the value in one controller action**

public IActionResult Index()

{

**HttpContext.Session.SetString("UserName", "Dhanush Ben");**

return View();

}

**Retaining the session values:**

var userName = HttpContext.Session.GetString("UserName");

**Important Considerations**

**Session Size** → Keep session data small to avoid memory pressure.  
 **Scalability** → In a web farm, consider using StateServer or SQLServer mode instead of InProc.  
 **Security** → Use HttpOnly and Secure cookies to protect session identifiers.  
 **Timeout Handling** → Implement graceful handling when sessions expire.