State Management in ASP.NET

Objective:

Understand **State Management** and explore different techniques used in **ASP.NET Web Forms** to persist data between requests.

What is State Management?

In ASP.NET, **state management** is the technique to **maintain user data** across multiple requests/pages since HTTP is stateless.

Types of State Management

ASP.NET provides **two types**:

A. **Client-Side State Management**

Data is stored **on the client's browser**.

Technique	Description
Hidden Fields	Store data in a hidden form field
ViewState	Stores data in page's HTML
Cookies	Store small data in the browser
Query Strings	Pass data via URL

B. **Server-Side State Management**

Data is stored **on the server**.

Technique	Description
Session	Per-user session data
Application	Shared data across all users
Cache	Temporarily stores data for performance

Client-Side Techniques

1. Hidden Field

```
<asp:HiddenField ID="hfUserId" runat="server" Value="123" />
```

```
string id = hfUserId.Value;
```

2. ViewState

```
// Save
ViewState["Name"] = "Alice";

// Retrieve
string name = ViewState["Name"].ToString();
```

3. Cookies

```
// Add
Response.Cookies["UserName"].Value = "Bob";
Response.Cookies["UserName"].Expires = DateTime.Now.AddDays(7);

// Read
string name = Request.Cookies["UserName"]?.Value;
```

4. Query String

```
// Sending
Response.Redirect("Page2.aspx?user=Tom");

// Receiving on Page2.aspx
string user = Request.QueryString["user"];
```

Server-Side Techniques

1. Session

```
// Store
Session["UserName"] = "David";

// Retrieve
string name = Session["UserName"].ToString();
```

```
// Remove
Session.Remove("UserName");
```

2. Application

```
// Set
Application["AppVersion"] = "1.0";

// Get
string version = Application["AppVersion"].ToString();
```

3. Caching (optional intro)

```
// Insert into Cache
Cache["ListData"] = dataObject;

// Retrieve
var data = Cache["ListData"];
```

Comparison Table

Method	Scope	Storage	Size Limit	Persistence	Use Case
ViewState	Per Page	Client	Medium	Per request	Page-level data
Hidden Field	Per Page	Client	Small	Per request	ID, Flags
Cookies	Cross Pages	Client	4KB	Days/Weeks	Preferences
Query String	Cross Pages	Client	URL Limit	One-time	Navigation
Session	Per User	Server	Large	Session	Login, Cart
Application	All Users	Server	Large	App-wide	Counters

Sample Use Case: Session State

```
protected void btnLogin_Click(object sender, EventArgs e)
{
    Session["UserEmail"] = txtEmail.Text;
    Response.Redirect("Dashboard.aspx");
}
```

```
// Dashboard.aspx.cs
string email = Session["UserEmail"].ToString();
```

Best Practices

- Use **Session** wisely; large objects can impact server memory.
- Avoid storing sensitive info in **ViewState** or **Query Strings** unless encrypted.
- Use **Cookies** for small preferences (e.g., theme).
- Use **Application** for global/shared counters, configs.