Forms Authentication Windows Authentication

🔐 1. What is Forms Authentication?

Definition

Forms Authentication is an ASP.NET authentication method where **users log in using a Web Form (Login.aspx)** with a **username and password** stored in:

- web.config
- a database
- or any custom store

How It Works

1. User tries to access a protected page
2. They are redirected to **Login.aspx**
3. User enters username + password
4. ASP.NET validates credentials
5. A **Forms Authentication Cookie** is created
6. User is now authenticated

Where It Is Used

- Internet websites
- Applications with **custom users**
- Apps where users are *not* in a Windows/Active Directory domain
- Public-facing web apps (e.g., shopping sites)

Why Use Forms Authentication?

- Fully customizable login screen
- Works for non-domain users
- Can integrate with any database
- More control over login logic

🖥️ 2. What is Windows Authentication?

Definition

Windows Authentication uses the **Windows operating system's user accounts** (or Active Directory domain accounts) to authenticate users.

How It Works

1. User tries to access the site
2. IIS checks Windows credentials (domain login)
3. If valid, the user is authenticated automatically
4. No login page required

Where It Is Used

- Intranet applications
- Corporate networks
- When all users are company employees
- Environments where every user logs into Windows domain

Why Use Windows Authentication?

- No login page required
- Uses existing Windows accounts
- Secure and centrally managed
- Integrates with Active Directory permissions

⚖️ 3. Key Differences (Simple Table)

Feature	**Forms Authentication**	**Windows Authentication**
User database	Custom (web.config / DB)	Windows / Active Directory
Login page	Required (Login.aspx)	Not required
Cookie	Uses Forms Auth cookie	Uses Windows security token
Environment	Internet or extranet	Intranet (internal network)
Customization	Fully customizable	Limited
Best for	Public websites	Internal office apps
Credential storage	Developer-controlled	Managed by OS/Domain

🧩 4. When to Use Which?

✓ Use Forms Authentication when:

- Your users are NOT Windows/Domain users
- You need a custom login experience
- Your app is public-facing
- You store users in a database

✓ Use Windows Authentication when:

- All users are internal employees
- PC login = website login (SSO-like experience)
- You want to use Windows groups & AD security
- You don’t want to maintain user accounts

⚙️ 5. How to Enable Them (web.config)

Forms Authentication

```
<authentication mode="Forms">
  <forms loginUrl="Login.aspx" />
</authentication>
```

Windows Authentication

```
<authentication mode="Windows" />
```

IIS also needs Windows Authentication enabled.

🧠 In One Sentence

- **Forms Authentication** → Custom login page for ANY user (best for internet apps).
- **Windows Authentication** → Uses Windows/AD credentials automatically (best for intranet apps).