🧱 1. Create a New Razor Pages Project with Individual Accounts

Via CLI:

```
bash
CopyEdit
dotnet new razor -auth individual -o MyIdentityApp
cd MyIdentityApp
```

Or in Visual Studio:

- Choose **ASP.NET Core Web App (Model-View-Controller)** or **Razor Pages**.
- Under **Change Authentication**, select **Individual User Accounts**.
  This adds necessary Identity references automatically [tektutorialshub.com+15learnrazorpages.com+15csharp.com+15learnrazorpages.com+1dotnettutorials.net+1](#).

🛠️ 2. Review `Program.cs`

Open `Program.cs`, it should now include:

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

// Add EF Core + Identity
builder.Services.AddDbContext<ApplicationDbContext>(options =>

options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConn
ection")));
builder.Services.AddDefaultIdentity<IdentityUser>(options =>
{
    options.Password.RequiredLength = 6;
})
.AddEntityFrameworkStores<ApplicationDbContext>();

builder.Services.AddRazorPages();

var app = builder.Build();

app.UseStaticFiles();
app.UseRouting();
app.UseAuthentication();
app.UseAuthorization();

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

This configures Identity & Razor pages out of the box .

📦 3. Set Up Database

Open `appsettings.json`, provide your connection string (e.g. LocalDB or SQL Server).

Then run migrations:

```
bash
CopyEdit
dotnet ef migrations add InitialIdentity
dotnet ef database update
```

This creates the Identity tables via EF Core [medium.comreddit.com+4reddit.com+4medium.com+4dotnettutorials.net+3learnrazorpages.com+3reddit.com+3](#).

☑ 4. Scaffold Identity (Optional Customization)

If you want to customize the UI:

```
bash
CopyEdit
dotnet tool install -g dotnet-aspnet-codegenerator
dotnet aspnet-codegenerator identity -dc ApplicationDbContext \
  --files "Account.Register;Account.Login;Account.Logout" --useDefaultUI
```

This adds:

```
swift
CopyEdit
Areas/Identity/Pages/Account/Register.cshtml*
Areas/Identity/Pages/Account/Login.cshtml*
Areas/Identity/Pages/Account/Logout.cshtml*
```

You can then modify these as needed [toxigon.com+2reddit.com+2andrewlock.net+2](#).

📝 5. Customize `Register.cshtml` (Optional)

Here's a simple version of `Areas/Identity/Pages/Account/Register.cshtml`:

```
cshtml
CopyEdit
@page
@model RegisterModel
@{
    ViewData["Title"] = "Register";
```

```
}

<h2>Register</h2>
<form method="post">
  <div asp-validation-summary="All" class="text-danger"></div>
  <div class="form-group">
    <label asp-for="Input.Email"></label>
    <input asp-for="Input.Email" class="form-control" />
  </div>
  <div class="form-group">
    <label asp-for="Input.Password"></label>
    <input asp-for="Input.Password" type="password" class="form-control" />
  </div>
  <div class="form-group">
    <label asp-for="Input.ConfirmPassword"></label>
    <input asp-for="Input.ConfirmPassword" type="password" class="form-
control" />
  </div>
  <button type="submit" class="btn btn-primary">Register</button>
</form>
@section Scripts {
    <partial name="_ValidationScriptsPartial" />
}
```

The `PageModel` logic is handled in `Register.cshtml.cs` (`OnPostAsync`). Scaffolding sets it up by default [dotnettutorials.net+15codemurals.blogspot.com+15andrewlock.net+15andrewlock.net+2learnrazorpages.com+2learnrazorpages.com+2].

✍️ 6. Customize `Login.cshtml`

Similarly:

```
cshtml
CopyEdit
@page
@model LoginModel
@{
    ViewData["Title"] = "Login";
}

<h2>Login</h2>
<form method="post">
  <div asp-validation-summary="All" class="text-danger"></div>
  <div class="form-group">
    <label asp-for="Input.Email"></label>
    <input asp-for="Input.Email" class="form-control" />
  </div>
  <div class="form-group">
    <label asp-for="Input.Password"></label>
    <input asp-for="Input.Password" type="password" class="form-control" />
  </div>
  <div class="form-group">
    <input asp-for="Input.RememberMe" /> <label asp-
for="Input.RememberMe"></label>
```

```
  </div>
  <button type="submit" class="btn btn-primary">Login</button>
</form>
@section Scripts {
    <partial name="_ValidationScriptsPartial" />
}
```

The `LoginModel.OnPostAsync` validates credentials and signs the user in .

---

7. Add Logout Page

Edit or scaffold `Logout.cshtml` and `Logout.cshtml.cs`:

```
cshtml
CopyEdit
@page
@model LogoutModel
@{
    ViewData["Title"] = "Logout";
}

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

@if (User.Identity.IsAuthenticated)
{
    <form method="post">
        <button type="submit" class="btn btn-link">Logout</button>
    </form>
}
else
{
    <p>You have been logged out.</p>
}
```

Its PageModel calls `await _signInManager.SignOutAsync();` on POST toxigon.com+15tektutorialshub.com+15reddit.com+15.

---

8. Include Links in `_LoginPartial.cshtml`

In `Pages/Shared/_LoginPartial.cshtml` (automatically scaffolded):

```
cshtml
CopyEdit
@inject SignInManager<IdentityUser> SignInManager
@inject UserManager<IdentityUser> UserManager
<ul class="navbar-nav">
@if (SignInManager.IsSignedIn(User))
{
  <li class="nav-item">
    <a class="nav-link">Hello @User.Identity.Name!</a>
  </li>
```

```
  <li class="nav-item">
    <form asp-area="Identity" asp-page="/Account/Logout" method="post">
      <button type="submit" class="nav-link btn btn-link">Logout</button>
    </form>
  </li>
}
else
{
  <li class="nav-item"><a class="nav-link" asp-area="Identity" asp-
page="/Account/Register">Register</a></li>
  <li class="nav-item"><a class="nav-link" asp-area="Identity" asp-
page="/Account/Login">Login</a></li>
}
</ul>
```

This ensures your layout shows proper login/register/logout links reddit.com+5tektutorialshub.com+5reddit.com+5codemurals.blogspot.com.

☑ 9. Run the App

1. Build and run (`dotnet run` or F5 in VS).
2. Visit `/Identity/Account/Register` → fill in details → submit.
3. Then go to `/Identity/Account/Login` → sign in.
4. Once logged in, `_LoginPartial` shows your username and logout option.

☑ Summary

Step	Description
1	`dotnet new razor -auth individual`
2	Identity configured in `Program.cs`
3	Migrations: `add-migration`, `database update`
4	Scaffold Identity views (optional)
5	Customize the Razor pages as needed
6	Update `_LoginPartial.cshtml` for navigation
7	Run & test registration, login, logout