

- Source code in AccountController.cs

```
using BankDockerLogin.Models;
using Microsoft.AspNetCore.Mvc;

namespace BankDockerLogin.Controllers
{
    public class AccountController : Controller
    {
        // Simulated user data (replace with your actual authentication logic)
        private readonly User _user = new User
        {
            Id = 1,
            Username = "maximus",
            Password = "perseus"
        };

        [HttpGet]
        public IActionResult Login()
        {
            return View();
        }

        [HttpPost]
        public IActionResult Login(User model)
        //{
            //if (ModelState.IsValid && Username == _user.Username && Password
            // == _user.Password)
            {
                // Successful login, redirect to dashboard
                return RedirectToAction("Index", "Dashboard");
            }
            // Invalid login, show error message
            // ModelState.AddModelError("", "Invalid username or password");
            //return View(model);
        }
    }
}
```

- Source code in Dashboard.cs

```
using Microsoft.AspNetCore.Mvc;

namespace BankDockerLogin.Controllers
{
    public class DashboardController : Controller
    {
        public IActionResult Index()
        {
            return View();
        }
    }
}
```

- Source code in HomeController.cs

```
using BankDockerLogin.Models;
using Microsoft.AspNetCore.Mvc;
using System.Diagnostics;

namespace BankDockerLogin.Controllers
{
    public class HomeController : Controller
    {
        private readonly ILogger<HomeController> _logger;

        public HomeController(ILogger<HomeController> logger)
        {
            _logger = logger;
        }

        public IActionResult Index()
        {
            return View();
        }

        public IActionResult Privacy()
        {
            return View();
        }

        [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore
= true)]
        public IActionResult Error()
        {
            return View(new ErrorViewModel { RequestId = Activity.Current?.Id ??
HttpContext.TraceIdentifier });
        }
    }
}
```

- Source Code in ErrorViewModel.cs

```
namespace BankDockerLogin.Models
{
    public class ErrorViewModel
    {
        public string? RequestId { get; set; }

        public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);
    }
}
```

- Source Code in User.cs

```
namespace BankDockerLogin.Models
{
    public class User
    {
        public int Id { get; set; }
        public string? Username { get; set; }
        public string? Password { get; set; }
    }
}
```

- Source code in Login.cshtml

```
@*
    For more information on enabling MVC for empty projects, visit
    https://go.microsoft.com/fwlink/?LinkID=397860
*@
@{
}
@model BankDockLogin.Models.User
@{
    ViewData["Title"] = "Login";
}
<h1>Login</h1>
<form asp-controller="Dashboard" asp-action="Index" method="post">
    <div class="form-group">
        <label asp-for="Username">Username:</label>
        <input asp-for="Username" class="form-control" />
        <span asp-validation-for="Username" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label asp-for="Password">Password:</label>
        <input asp-for="Password" type="password" class="form-control" />
        <span asp-validation-for="Password" class="text-danger"></span>
    </div>
    <button type="submit" class="btn btn-primary">Login</button>
</form>
```

- Source code in Index.cshtml

```
@*
    For more information on enabling MVC for empty projects, visit
    https://go.microsoft.com/fwlink/?LinkID=397860
*@
@{
}
@{
    ViewData["Title"] = "Index";
}
<h1 style="text-align:center">Welcome to Dashboard</h1>
```

- Source code in Program.cs

```
using Microsoft.AspNetCore.Authentication.Cookies;
using BankDockerLogin.Models;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddControllersWithViews();
builder.Services.AddAuthentication(CookieAuthenticationDefaults.AuthenticationScheme)
    .AddCookie(options =>
    {
        options.LoginPath = "/Account/Login"; // Redirect to the login page for
        unauthenticated

    });
var app = builder.Build();

// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
}
app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.UseEndpoints(endpoints =>
{
    endpoints.MapControllerRoute(
        name: "dashboard",
        pattern: "dashboard",
        defaults: new { controller = "Dashboard", action = "Index" });
    endpoints.MapControllerRoute(
        name: "default",
        pattern: "{controller=Account}/{action=Login}/{id?}");
});

app.Run();
```