

#See <https://aka.ms/customizecontainer> to learn how to customize your debug container and how Visual Studio uses this Dockerfile to build your images for faster debugging.

```
FROM mcr.microsoft.com/dotnet/aspnet:6.0 AS base
WORKDIR /app
EXPOSE 80
```

```
FROM mcr.microsoft.com/dotnet/sdk:6.0 AS build
WORKDIR /src
COPY ["PractieProjDocker/PractieProjDocker.csproj", "PractieProjDocker/"]
RUN dotnet restore "PractieProjDocker/PractieProjDocker.csproj"
COPY . .
WORKDIR "/src/PractieProjDocker"
RUN dotnet build "PractieProjDocker.csproj" -c Release -o /app/build
```

```
FROM build AS publish
RUN dotnet publish "PractieProjDocker.csproj" -c Release -o /app/publish
/p:UseAppHost=false
```

```
FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "PractieProjDocker.dll"]namespace PractieProjDocker.Models
{
```

```
    public class User
    {
        public int Id { get; set; }
        public string Username { get; set; }
        public string Password { get; set; }
    }
}
```

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

```
namespace PractieProjDocker.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 && model.Username == _user.Username &&
model.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);
    }
}
using Microsoft.AspNetCore.Mvc;

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

@model PractieProjDocker.Models.User
@{
    ViewData["Title"] = "Login";
}
<h1>Login</h1>
<form asp-controller="Account" asp-action="Login" 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>
@{
    ViewData["Title"] = "Index";
}
<h1 style="text-align:center">Welcome to Dashboard</h1>
using Microsoft.AspNetCore.Authentication.Cookies;
using PractieProjDocker.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();
```