```
@page @model PracticeProblem_5.Pages.LoginModel @{ }

<h1>Login</h1>

<form method=”post”> <div> <label for=”username”>Username:</label>
<input type=”text” id=”username” name=”username”> </div> <div>
<label for=”password”>Password:</label> <input type=”password”
id=”password” name=”password”> </div> <button type=”submit”>Login</button>
</form>

using Microsoft.AspNetCore.Mvc; using Microsoft.AspNetCore.Mvc.RazorPages;

namespace PracticeProblem_5.Pages { public class LoginModel : PageModel {
public void OnGet() { // Optional: Handle initial page load }

public IActionResult OnPost(string username, string password) { // Validate
username and password (not implemented in this example) if (username ==
”admin” && password == ”password”) { // Redirect to dashboard page upon
successful login return RedirectToPage(”/Dashboard”); } else { // Optionally
handle failed login attempt return Page(); } } } }

@page @model PracticeProblem_5.Pages.DashboardModel @{ }

<h1>Welcome to the Dashboard!</h1>

<p>Dashboard content goes here...</p>

<form method=”post”> <button type=”submit” name=”logout”>Logout</button>
</form>

using Microsoft.AspNetCore.Mvc; using Microsoft.AspNetCore.Mvc.RazorPages;

namespace PracticeProblem_5.Pages { public class DashboardModel : PageModel
{ public void OnGet() { // Optional: Handle initial page load }

public IActionResult OnPost(string logout) { if (!string.IsNullOrEmpty(logout))
{ // Redirect to login page upon logout return RedirectToPage(”/Login”); }
return Page(); } } }

var builder = WebApplication.CreateBuilder(args);

// Add services to the container. builder.Services.AddRazorPages();

var app = builder.Build();

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

app.UseRouting();

app.UseAuthorization();

app.MapRazorPages();

app.Run();
```

```
{ ”Logging”: { ”LogLevel”: { ”Default”: ”Information”, ”Microsoft.AspNetCore”:
”Warning” } }, ”AllowedHosts”: ”*” }
```