



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет Информационных технологий
Кафедра Информатики и информационных технологий

направление подготовки
09.03.02 «Информационные системы и технологии»

ЛАБОРАТОРНАЯ РАБОТА № 15

Дисциплина: Backend разработка

Тема:

Выполнил(а): студент(ка) группы 231-336

Канищев И.М
(Фамилия И.О.)

Дата, подпись _____
(Дата) (Подпись)

Проверил: _____
(Фамилия И.О., степень, звание) (Оценка)

Дата, подпись _____
(Дата) (Подпись)

Замечания:

Москва
2025

1. Создание ролей и пользователей

- Используем **ASP.NET Core Identity** для управления пользователями и ролями.
- Созданы роли: Admin и User.
- Созданы тестовые пользователи:

Пользователь	Роль	Пароль
admin@example.com	Admin	Admin123!
user@example.com	User	User123!

Код инициализации ролей и пользователей находится в Program.cs с использованием RoleManager и UserManager.

2. Настройка middleware для перенаправления запросов

- Создан кастомный middleware RoleRedirectMiddleware, который:
 - Проверяет роль пользователя при доступе к /Admin и /User.
 - Перенаправляет пользователя на страницу /Home/AccessDenied, если роль не подходит.

```
if (context.Request.Path.StartsWithSegments("/Admin") &&
    !context.User.IsInRole("Admin"))
{
    context.Response.Redirect("/Home/AccessDenied");
    return;
}
```

Middleware подключается после UseAuthentication() и UseAuthorization() в Program.cs.

3. Настройка маршрутов и страниц логина

- Для неаутентифицированных пользователей добавлен контроллер AccountController с действиями Login и Logout.
- Настройка Identity Cookie для правильного перенаправления:

```
builder.Services.ConfigureApplicationCookie(options =>
{
    options.LoginPath = "/Account/Login"; // перенаправление
    неаутентифицированных

    options.AccessDeniedPath = "/Home/AccessDenied"; // перенаправление
    при недостаточных правах

});
```

- Создана страница `Login.cshtml` для ввода имени пользователя и пароля.
- После успешного логина происходит редирект на исходную страницу (`ReturnUrl`).

4. Ограничение доступа через контроллеры

- **AdminController** — доступ только для роли Admin.
- **UserController** — доступ для ролей User и Admin.
- **HomeController/AccessDenied** — страница для сообщений о запрещенном доступе.

```
[Authorize(Roles = "Admin")]

public class AdminController : Controller { ... }

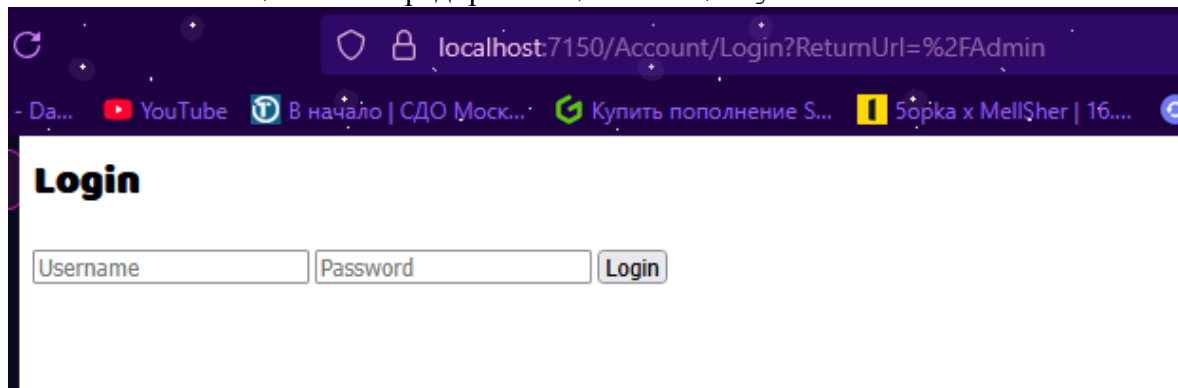
[Authorize(Roles = "User,Admin")]

public class UserController : Controller { ... }
```

5. Тестирование механизма

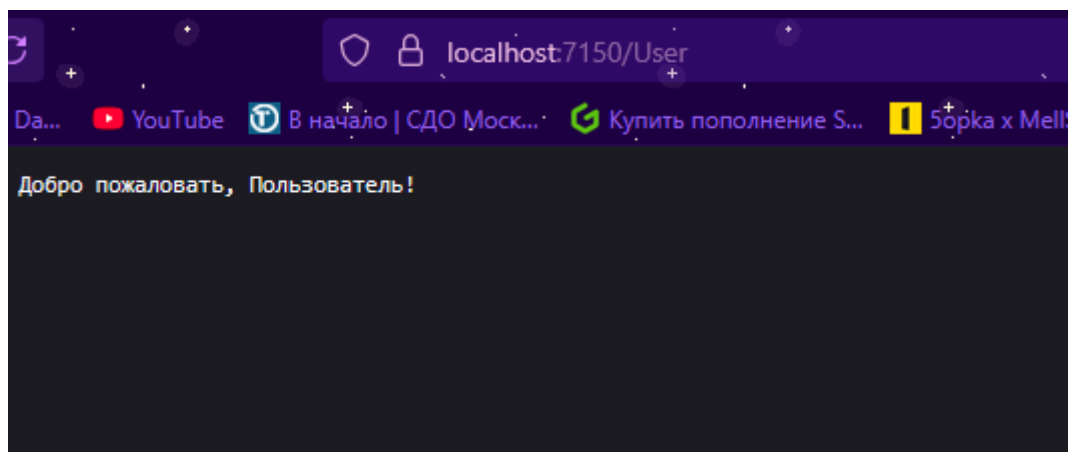
1. Неаутентифицированный пользователь:

- Попытка зайти на `/Admin` → редирект на `/Account/Login`.

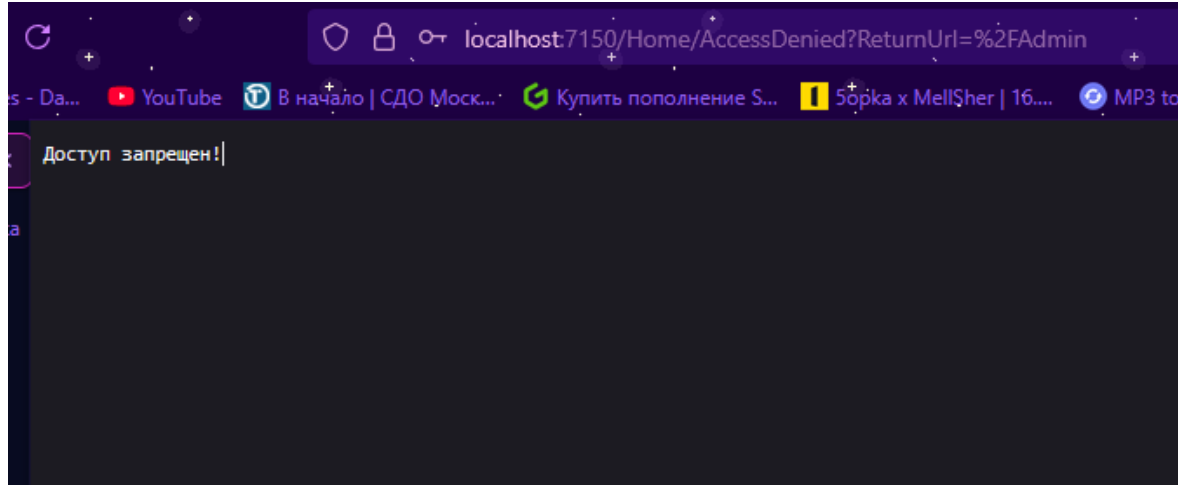


2. Пользователь с ролью User:

- `/User` → доступ разрешен

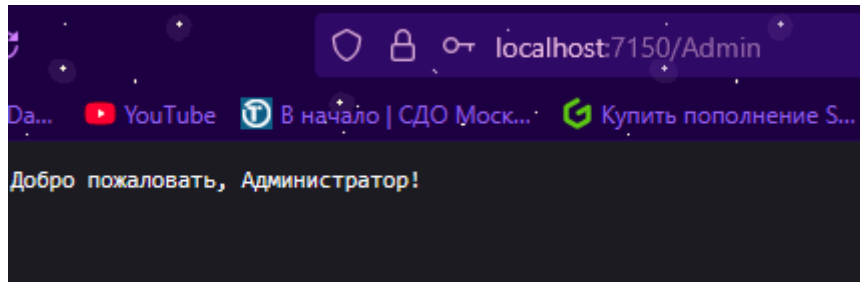


- /Admin → редирект на /Home/AccessDenied

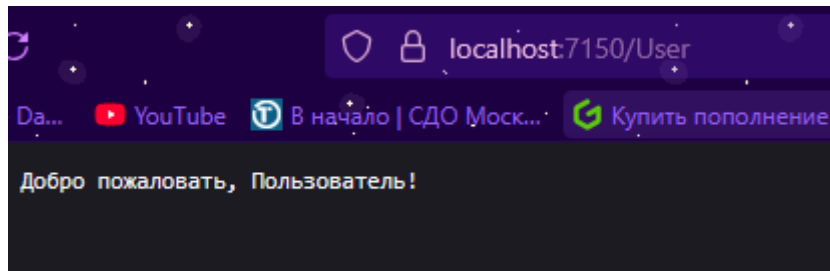


3. Пользователь с ролью Admin:

- /Admin → доступ разрешен



- /User → доступ разрешен



- Все сценарии проверены, перенаправления работают корректно.