Лабораторная работа №17
Разграничение прав доступа в веб-приложении

1 **Цель работы**

1.1 Научиться реализовывать разграничение прав доступа к БД в веб приложении ASP.Net.

2Литература

2.1 Руководство по ASP.NET Core и C#. metanit.com – Текст : электронный // metanit.com, 2025. – URL: https://metanit.com/sharp/aspnet6/ - гл.12-13

3Подготовка к работе

3.1 Повторить теоретический материал (см.п.2).

3.2 Изучить описание практической работы.

4Основное оборудование

4.1 Персональный компьютер.

5Задание

5.1 Создание веб-приложения

5.1.1 Создать проект «Веб-приложение ASP.NET Core».

5.1.2 Добавить в веб-приложение или отдельную библиотеку контекст БД и модели для таблиц БД (предметная область: кинотеатр + таблица пользователей из ЛР№15), объекты должны быть сгруппированы по папкам.

5.1.3 В веб-приложение добавить папки Films и Tickets, для каждой сгенерировать соответствующий набор страниц Razor (CRUD).

5.1.4 Реализовать переход к этим страницам из верхнего меню страницы _Layout.cshtml (вместо Privacy и Home):

Название

5.2 Добавление окна авторизации

5.2.1 Добавить в проект страницу Login (страница Razor, EF, на основе Create).

5.2.2 Модифицировать разметку страницы Login:

- оставить в разметке ввод логина и пароля (остальные поля БД убрать),
- изменить надпись «Создать» на «Войти»,
- добавить в разметку еще одну форму с кнопкой «Войти как гость».

У форм указать атрибут asp-page-handler для указания обработчика (Login у первой формы, Guest – у второй формы.

5.2.3 Добавить в код формы авторизации методы OnPostLogin и OnPostGuest. Проверить, что выполняется переход к этим методам при нажатии на соответствующие кнопки.

5.2.4 Модифицировать разметку верхнего меню страницы _Layout.cshtml:

<a asp-page="/Login" asp-page-handler="Login" class="btn btn-primary">Войти

5.3 Работа с сессиями

5.3.1 Добавить в Program.cs код для реализации работы с сессиями:

builder.Services.AddSession();

builder.Services.AddHttpContextAccessor();
…
app.UseSession();

5.3.2 Реализовать вход под ролью «Гость» в методе OnPostGuest:

HttpContext.Session.Clear();
HttpContext.Session.SetString("Role", "Гость");
return RedirectToPage("/директория/Index"); // на страницу фильмы

5.3.3 Реализовать вход под ролью пользователя БД в методе OnPostLogin:

Получить данные пользователя (из данных привязанного свойства User). Если пользователь найден и пароль корректен (с учетом регистра):

- сохранить логин и роль в сессии,
- перенаправить на страницу фильмы.

Иначе выполнять следующий код:

return Page();

5.3.4 Добавить в странице Login метод OnGetLogout для выхода

```
public IActionResult OnGetLogout()
{
   HttpContext.Session.Clear();
   return RedirectToPage("/Index");
}
```

5.3.5 Реализовать отображение в правом углу меню страницы _Layout.cshtml логина пользователя (если есть) и кнопки «Выйти», если пользователь авторизован, иначе – отображение кнопки «Войти».

5.4 Разграничение прав доступа

5.4.1 Добавить страницу AuthPage (пустая страница Razor).

5.4.2 Добавить на страницу набор свойств для получения роли и для проверки на соответствие роли:

```
public string UserRole => HttpContext.Session.GetString("Role");
public bool IsAdmin => UserRole == "Администратор";
…
```

5.4.3 Добавить метод для перенаправления в зависимости от наличия роли:

```
protected IActionResult HasRole()
{
   if (string.IsNullOrEmpty(UserRole))
      return RedirectToPage("/Login");

   return null;
}
```

5.4.4 Добавить метод IsInRole(string role) для перенаправления в зависимости от определенной роли и CanEdit() для перенаправления неадминистраторов по аналогии с методом HasRole.

5.4.5 Страницы:

- просмотра списка билетов – доступны администратору и билетеру,
- просмотра детальной информации о билете – доступны билетеру,
- просмотра списка фильмов и детальной информации о фильме – доступны всем,
- страницы редактирования доступны только администраторам.

У страниц просмотра и редактирования данных, доступных определенным ролям, указать родительский класс AuthPageModel и в методах OnGetAsync добавить проверку на возможность перехода, например:

```
if (HasRole() is IActionResult result)
   return result;
```

5.5 Изменение разметки в зависимости от роли

Реализовать скрытие кнопок редактирования, если роль не позволяет переход:

```
@if (Model.UserRole == "роль")
{
   Разметка, доступная роли
}
```

или

```
@if (Model.IsРоль)
{
   Разметка, доступная роли
}
```

6Порядок выполнения работы

6.1 Выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8Контрольные вопросы

8.1 Что такое «сессия» в веб-приложении?

8.2 Что такое «куки» в веб-приложении?

8.3 Каков срок жизни сессии и куки?

8.4 Как указать, что приложение ASP.Net будет поддерживать работу с сессиями?