

# Лабораторная работа №18

## Отображение и редактирование данных в веб-приложении

### 1 Цель работы

1.1 Научиться реализовывать настройку интерфейса пользователя для ввода и отображения данных БД в веб приложении ASP.Net.

### 2 Литература

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

### 3 Подготовка к работе

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

3.2 Изучить описание практической работы.

### 4 Основное оборудование

4.1 Персональный компьютер.

### 5 Задание

Задание выполнять в проекте ЛР №17, в который добавить страницы, сгенерированные для модели Сеанс и Посетитель (добавить их отображение в меню).

#### 5.1 Русификация интерфейса

##### 5.1.1 Русификация подписей модели

У свойств модели Сеанс и Посетитель указать аннотации Display:  
[Display(Name = "Название свойства")]

##### 5.1.2 Русификация подписей кнопок и ссылок

У кнопок и ссылок указать подписи на русском языке:

```
<input type="submit" value="Подпись кнопки" .../>
```

```
<a asp-page="Страница" ...>Подпись ссылки </a>
```

#### 5.2 Отображение данных связанных таблиц

##### 5.2.1 Добавление вспомогательных свойств

Для отображения кинотеатра и номера зала одним значением добавить в Сеанс открытое необязательное строковое свойство на чтение, в котором соединить требуемые данные.

##### 5.2.2 Отображение данных на страницах просмотра

В разметке страниц просмотра сеансов вместо внешнего ключа отображать свойства связанной таблицы, понятные пользователю (название фильма, кинотеатр и номер зала).

Пример:

```
@Html.DisplayFor(modelItem => item.Category.Name)
```

Вместо id фильма – название, вместо id зала – кинотеатр и номер зала.

##### 5.2.3 Отображение данных в выпадающих списках

В методах OnGet/OnGetAsync страниц редактирования внести изменения, чтобы вместо идентификаторов отображались свойства связанной таблицы, понятные пользователю (название фильма, кинотеатр, номер зала):

```
ViewData["Столбец FK"] = new SelectList(_context.ИсточникДанных, "Столбец PK", "Отображаемый столбец");
```

Пример:

```
ViewData["CategoryId"] = new SelectList(_context.Categories, "Id", "Name");
```

### 5.3 Настройка навигационных свойств для валидации

Проверка валидации ModelState.IsValid выдает сообщение из-за неуказанных навигационных свойств.

Для решения проблемы:

- сделать навигационные свойства в модели необязательными:

```
public virtual Тип? НавигационноеСвойство { get; set; }
```

- перед валидацией убрать проверку навигационных свойств:

```
ModelState.Remove("Модель.НавигационноеСвойство");
```

```
...
```

```
if (!ModelState.IsValid)
```

```
{
```

```
    return Page();
```

```
}
```

### 5.4 Настройка элементов для редактирования данных

#### 5.4.1 Смена локали

В Program.cs перед созданием app сменить настройки для ввода вещественных чисел:

```
CultureInfo.DefaultThreadCurrentCulture =
```

```
    new CultureInfo("ru-RU") { NumberFormat = { NumberDecimalSeparator = "." } };
```

#### 5.4.2 Смена стиля отображения элементов

Для смены стиля отображения элементов управления использовать аннотации модели:

```
[DataType(DataType.Тип)]
```

Типы DataType: Currency (для цены), Date (только дата), Time (только время), DateTime (дата и время), MultilineText (многострочное поле ввода - для описания), Password (для пароля).

Смену отображения также можно настроить в атрибутах html-разметки, например для элемента input: type="date" (только дата), type="password" (пароль).

### 5.5 Настройка аннотаций валидации

Настроить аннотации для того, чтобы пользователь видел сообщения об ошибках на русском языке (сообщение указывается в ErrorMessage).

Примеры аннотаций:

```
[Required(ErrorMessage = "Поле обязательно для заполнения")]
```

```
[StringLength(100, MinimumLength = 3,
```

```
    ErrorMessage = "Длина {0} должна быть от {2} до {1} символов")]
```

```
[Range(0.01, 10000.00, ErrorMessage = "Цена должна быть от {1} до {2}")]
```

```
[RegularExpression(@"^[a-zA-Za-яA-ЯёЁ\s]+$",
```

```
    ErrorMessage = "Только буквы и пробелы")]
```

```
[EmailAddress(ErrorMessage = "Некорректный email адрес")]
```

```
[Phone(ErrorMessage = "Некорректный номер телефона")]
```

## **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 Как настроить подписи свойств в удобном для пользователя виде?
- 8.2 Как отобразить данные связанных таблиц в интерфейсе?
- 8.3 Как настроить в модели сообщения об ошибках?
- 8.4 Какие аннотации позволяют настроить валидацию данных?
- 8.5 Что такое «валидация данных»?