

Задания для экзамена

1. Создать приложение ASP.NET Core, используя шаблон MVC и способ аутентификации «индивидуальные учетные записи пользователей».
2. На странице макета внутри тега <body> оставьте только подключение файлов JavaScript и директивы @RenderBody() и @RenderSection().
3. Выберите любую предметную область и создайте класс, описывающий сущность выбранной области. Класс должен содержать свойство Id и не менее 3 свойств, характеризующих сущность.
4. Добавьте в контекст базы данных (AppDbContext) набор созданных в п.2 сущностных классов. Выполните миграцию базы данных.
5. Заполните базу не менее 3-мя записями (можно использовать SqlServer Object Explorer)

Индивидуальные задания

1. Представление Index должно выводить список созданных сущностей в виде таблицы. Названия свойств в заголовке таблицы должны выводиться на русском языке (**использовать механизм аннотации данных**)
2. Представление Index должно выводить список созданных сущностей в виде таблицы. Формирование одной строки таблицы оформить в виде частичного представления.
3. Представление Index выводит список названий объектов. Напротив каждого названия имеется кнопка «Подробнее». При нажатии на кнопку выводится представление с полной информацией об объекте (список всех свойств выбранного объекта)
4. Представление Index должно использовать страницу макета. На представлении выводится информация о первом объекте в базе данных. На странице макета предусмотреть необязательную секцию, в которой представление Index выведет текущее время.
5. Зарегистрировать маршрут, который при запросе вида ~/Product/Show/*x*/*y* вызывается метод GetProduct контроллера Home. Метод должен вернуть представление, которое «*y*» раз выведет название объекта, у которого id равен значению «*x*». Параметры *x* и *y* должны передаваться под именами «num» и «qty» соответственно.
6. Представление Index должно выводить список созданных сущностей в виде таблицы. На страницу должно выводиться не более 2 объектов. Номер страницы передавать в строке запроса, например, ~/Home/Index?page=2
7. Представление Index имеет поле ввода и кнопку «Показать». В поле вводится Id объекта. При нажатии на кнопку ниже выводится информация о выбранном объекте. Если объект с данным Id не найден, вывести соответствующее сообщение.
8. В контроллере Home опишите действия Index(), Sumary(), Show(int? id), About(). Разрешите неавторизованный доступ только к методу Index. База данных авторизации должна иметь имя «MyDataBase». У пользователя должно быть дополнительное поле string Education.
9. Представление должно выводить названия объектов базы данных в виде выпадающего списка (тег <select>) и кнопку «Отправить». При выборе объекта в списке и нажатии кнопки отправить возвращается представление, которое выведет Id выбранного объекта
10. Создайте представление для редактирования объекта базы данных. Выполните валидацию на стороне клиента. Название объекта должно быть обязательным для заполнения, длина строки названия должна быть не менее 5 и не более 15 символов. Сообщение об ошибке должно быть таким: «Длина слова от 5 до 15 символов».
11. Представление должно передать файл на сервер. Файл должен сохраниться в папке wwwroot/Files.
12. Создайте компонент представления, который выводит приветствие и информацию о текущей дате, например, «Добрый день (вечер). Сегодня 1 января 2020 года». Утро, день

или вечер – выбирается в зависимости от времени суток. Подключите компонент на страницу макета.

13. Создайте тэг-хелпер, который преобразует разметку `<div class="nav"> . . .</div>` в семантически более понятную разметку `<nav class="nav"> . . .</nav>`
14. Создайте «пустой» компонент Middleware с именем MyComponent. Опишите расширяющий метод UseMyComponent() для добавления созданного компонента в конвейер Middleware. Добавьте компонент в конвейер.
15. Создайте «пустой» класс. Зарегистрируйте созданный класс в качестве Transient сервиса. Внедрите созданный сервис в метод Index контроллера Home.
16. Имеется класс

```
public class Calculator
{
    int[] _data;
    public Calculator(int[] data)
    {
        _data = data;
    }

    public int GetMax() => _data.Max();
}
```

Метод GetMax() возвращает максимальное число из исходного массива. Напишите модульный тест для этого метода.

17. Создайте тэг-хелпер, преобразующий разметку

```
<user user-name="John"></user>
```

в вид

```
<h1>
    <span class="badge badge-secondary">John</span>
</h1>
```

18. Метод GetImage контроллера Home передает клиенту файл изображения. Файлы хранятся в папке wwwroot/Images. Представление Index содержит изображение, получаемое с помощью метода GetImage.
19. На страницу выводится поле ввода и кнопка «Показать». В поле вводится Id объекта. При нажатии на кнопку ниже выводится информация о выбранном объекте. Если объект с данным Id не найден, вывести соответствующее сообщение. **Реализовать с помощью Razor Page**
20. На страницу выводится список названий объектов. Напротив каждого названия имеется кнопка «Подробнее». При нажатии на кнопку выводится страница с полной информацией об объекте (список всех свойств выбранного объекта). **Реализовать с помощью Razor Page**