

## Лабораторная работа №1

### Знакомство с приложением ASP.NET Core

#### **1. Цель работы.**

Знакомство со структурой приложения ASP.NET MVC Core.

Знакомство с представлениями (Views) и страницами-макетами (Layouts).

#### **2. Задача работы**

Научиться создавать представления, использующие страницы-макеты

Время выполнения работы – 4 часа (2 занятия)

#### **3. Выполнение работы**

##### 3.1. Подготовка к работе

В среде **VisualStudio** создайте новый проект ASP.NET Core

Выберите шаблон «Web Application (Model-View-Controller)» (Рисунок 1).

Назначьте решению имя WEB\_GGGG\_NNN, а проекту в решении – имя WEB\_GGGG\_NNN.UI, где GGGG – номер группы, а NNN – фамилия.

##### **VisualStudio Code**

`dotnet new mvc -o <имя проекта>`

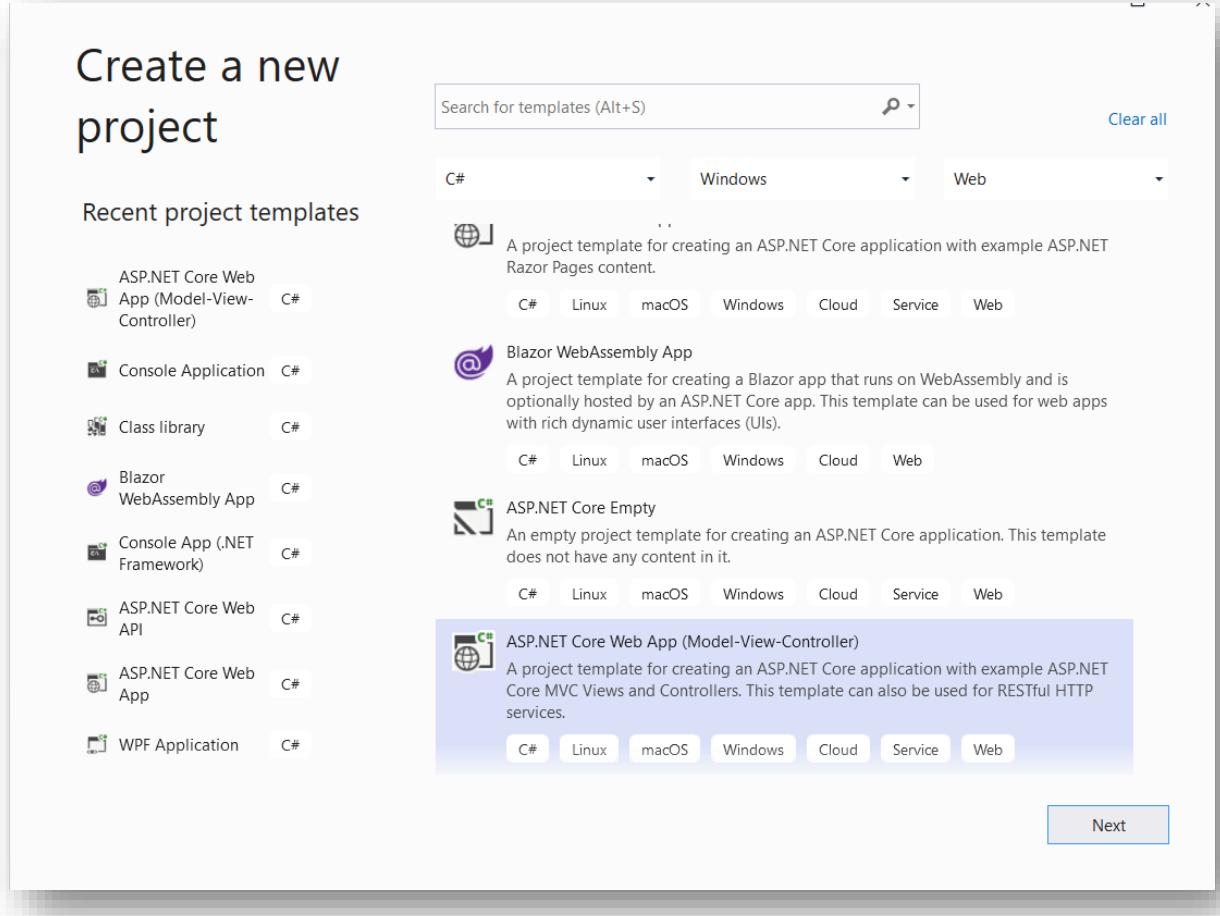


Рисунок 1 Создание проекта

Для отображения на страницах иконок добавьте любую соответствующую библиотеку стилей. Например: font-awesome или open-iconic (см. <https://fontawesome.com/> или <https://github.com/iconic/open-iconic/> соответственно). В предлагаемом примере используется библиотека bootstrap-icons. Подробную информацию можно найти на сайте производителя (<https://icons.getbootstrap.com/>). Кликните правой кнопкой по папке ~wwwroot/lib и выберите Add-Client-side library (Рисунок 2)

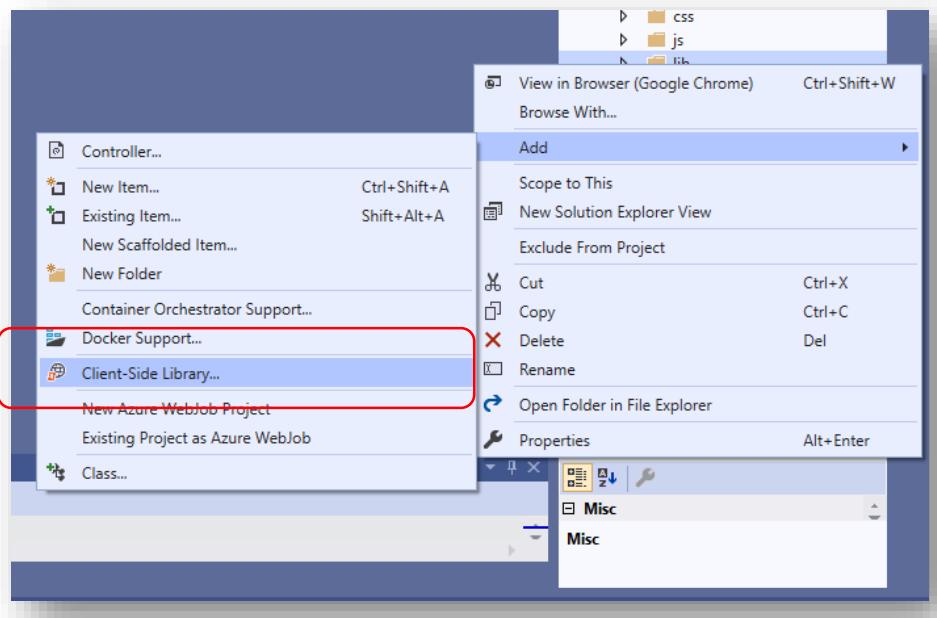


Рисунок 2 Добавление библиотеки

В открывшемся окне введите bootstrap-icons (Рисунок 3).

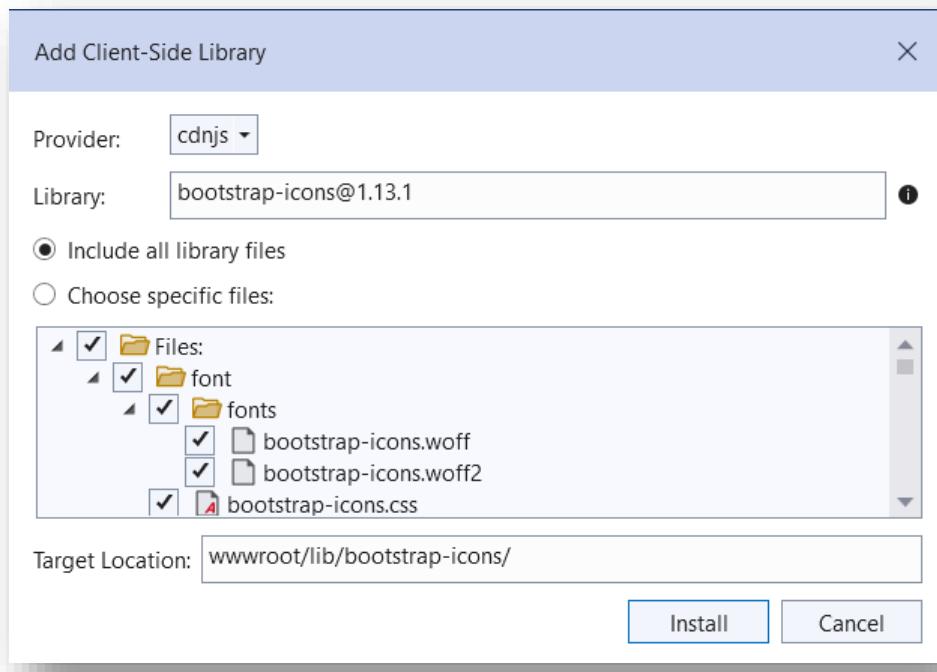


Рисунок 3 Выбор библиотеки

## VisualStudio Code

- 1) Установите LibMan CLI:

```
dotnet tool install -g Microsoft.Web.LibraryManager.Cli
```

2) Инициализируйте LibMan в проекте:

```
libman init
```

3) Загрузите font-awesome:

```
libman install bootstrap-icons --provider cdnjs --destination
```

```
wwwroot/lib/bootstrap-icons
```

**Примечание:** в приведенных ниже примерах используется библиотека bootstrap-icons версии 1.131. Если в вашем проекте используется другая версия, то имена классов иконок могут не совпадать. Для информации – см. сайт производителя: <https://icons.getbootstrap.com/>.

### 3.2. Знакомство с проектом

Найдите в проекте место размещения статических файлов (файлов css, js и др.)

Найдите в проекте папки для размещения контроллеров и представлений.

Откройте файл program.cs. Найдите регистрацию сервисов и настройку конвейера обработка запросов.

Откройте файл appsettings.json. Ознакомьтесь с его содержимым.

### 3.3. Подготовка проекта

Из папки Controllers удалите файл HomeController.cs.

Из папки Views удалите папку Home.

**Примечание:** в проекте будут использоваться порты 7001 и 5001

Откройте файл LaunchSettings.json в папке Properties. Измените порты в соответствии с примечанием:

```
"profiles": {  
    "WebLabs_BSUIR_V02": {  
        "commandName": "Project",  
        "dotnetRunMessages": true,  
        "launchBrowser": true,
```

```
"applicationUrl": "https://localhost:7001;http://localhost:5001",
"environmentVariables": {
  "ASPNETCORE_ENVIRONMENT": "Development"
}
```

### 3.4. Создание контроллера и представления

- a. Создайте контроллер с именем Home;
- b. Создайте представление для метода Index созданного контроллера, без использования страницы шаблона и выводящую статический текст “Hello World!”; (предварительно создайте папку Home в папке Views).
- c. *Запустите проект и проверьте полученный результат*

### 3.5. Разметка представления Index

На представлении Index подключите таблицы стилей

- lib/bootstrap/dist/css/bootstrap.min.css,
- lib/bootstrap-icons/font/bootstrap-icons.min.css,
- css/site.css,

а также файлы скриптов

- lib/jquery/dist/jquery.min.js
- lib/bootstrap/dist/js/bootstrap.bundle.min.js.

**Примечание:** в качестве примера можно взять страницу \_Layout.cshtml из папки Views/Shared

**Примечание:** в приведенных ниже примерах используется библиотека bootstrap версии 5.3. Если в вашем проекте используется другая версия, то имена классов могут не совпадать. Для информации – см. сайт производителя: <https://getbootstrap.com>.

Пример разметки:

```
@{
    Layout = null;
}
```

```

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Index</title>
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
    <link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />
    <link rel="stylesheet" href="~/lib/font-awesome/css/all.min.css" />
    <link rel="stylesheet" href="~/WebLabs_BSUIR_V02.styles.css" asp-append-
version="true" />
</head>
<body>
    Hello World!
    <script src="~/lib/jquery/dist/jquery.min.js"></script>
    <script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
    <script src="~/js/site.js" asp-append-version="true"></script>
</body>
</html>

```

В файле Site.css удалите все содержимое и опишите следующие стили:

```

header, footer {
    background-color: rgba(var(--bs-dark-rgb));
}

.nav-color {
    color: rgba(255, 255, 255, 0.5)
}

```

### 3.6. Разметка структуры страницы

Выполните разметку страницы, чтобы получить структуру как на

Рисунок 4

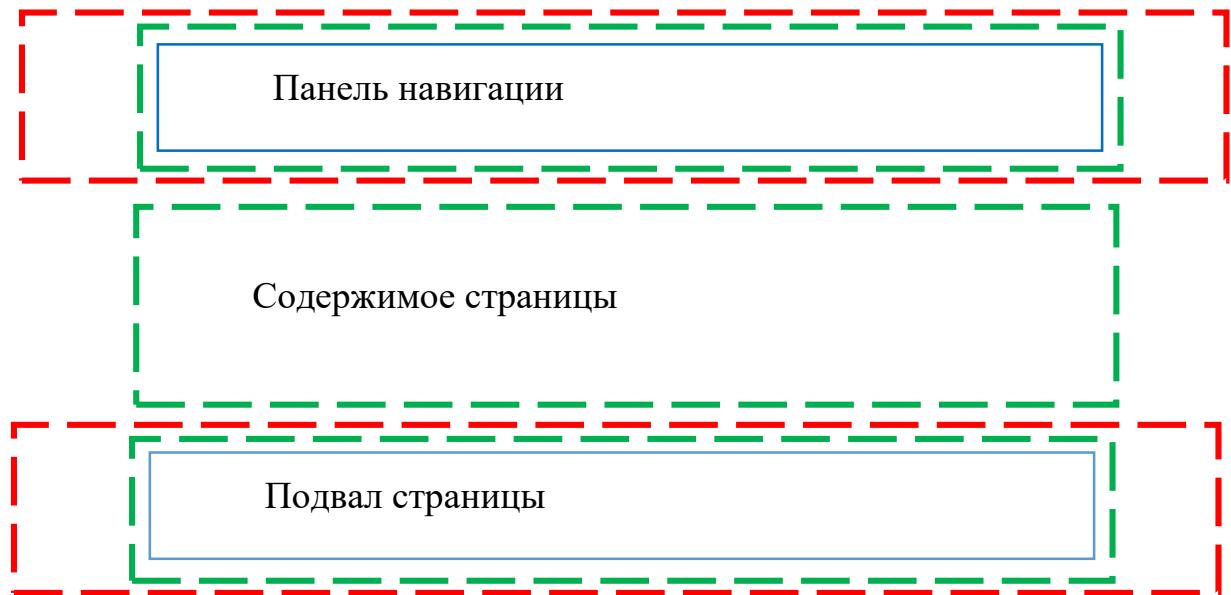


Рисунок 4 Структура страницы

На рисунке красными прямоугольниками обозначены тэги `<header>` (верхний прямоугольник) и `<footer>` (нижний прямоугольник). Зелеными прямоугольниками обозначены тэги `<div>`, имеющие класс css «container». Этот класс ограничивает ширину блока в зависимости от размера экрана.

Для такой структуры разметка в тэге body будет следующая:

```
<header>
    <div class="container">
        <!-- Панель навигации -->
        <span class="nav-color">Заголовок страницы</span>
    </div><!-- container - конец -->
</header><!-- header - конец -->
<main class="container">
    Содержимое страницы
</main>
<footer>
    <div class="container">
        <span class="nav-color">Подвал страницы</span>
    </div>
</footer>
```

Откройте страницу в браузере.

Вы должны получить вид ка на Рисунок 5.

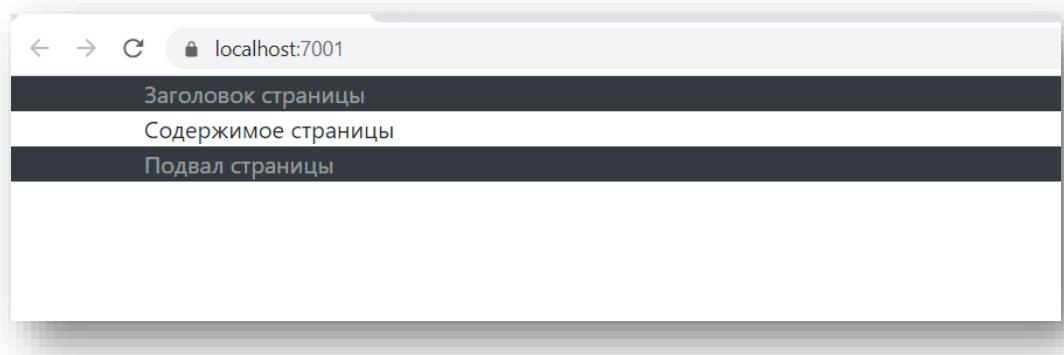


Рисунок 5 Базовый вид страницы

### 3.7. Оформление заголовка страницы

Заголовок страницы должен содержать следующее (Рисунок 6):

- меню сайта
- информация пользователя

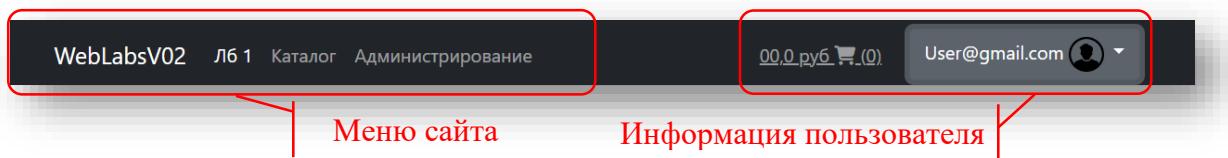


Рисунок 6 Заголовок страницы

Меню сайта (Рисунок 7) состоит из

- заголовка (Ваш номер зачетки - ссылка на страницу Лаб1, класс Bootstrap «badge»)
- кнопок навигации по страницам сайта (Лб1, Каталог и Администрирование) - используйте классы для оформления навигационных панелей bootstrap (<https://getbootstrap.com/docs/5.3/components/navbar/> ).



Рисунок 7 Меню сайта

Информация пользователя (**Ошибка! Источник ссылки не найден.8**)  
состоит из:

- ссылка на страницу корзины заказа. Текст ссылки - информация о корзине заказа (общая стоимость и количество товаров). Для иконки использовать иконку bootstrap-icons (класс «`bi bi-cart`»)
- выпадающее меню в виде имени пользователя и его аватарки. ( см. <https://getbootstrap.com/docs/5.3/components/dropdowns/> ) Меню состоит из аватарки, имени пользователя, кнопки «Logout» (или «Выход»)

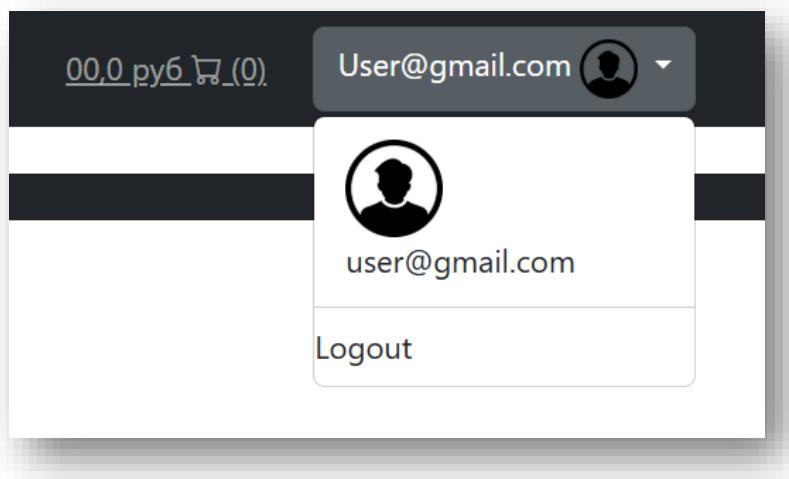


Рисунок 87 Информация пользователя

Для хранения аватарки создайте папку **Images** в папке **wwwroot** проекта.

**Важно:** Пункт меню Logout должен представлять собой форму, отправляющую данные методом Post

Разметка заголовка сайта может выглядеть так (внутри тэга `<div class="container">`):

```
<!-- Панель навигации -->
<nav class="navbar bg-dark navbar-expand-md border-bottom border-body"
     data-bs-theme="dark">
    <!-- меню сайта -->
    <a class="navbar-brand" href="#">LabsV03</a>
    <div class="navbar-nav">
        <a class="nav-item nav-link active" href="#">Лб 1</a>
        <a class="nav-item nav-link" href="#">Каталог</a>
        <a class="nav-item nav-link" href="#">Администрирование</a>
    </div>
    <!-- меню сайта - конец -->
    <!-- Информация пользователя -->
    <a href="#" class="navbar-text ms-auto">
        00,0 руб <i class="bi bi-cart"></i> (0)
    </a>

    <div class="dropdown ms-4 nav-color">
        <button class="btn btn-secondary dropdown-toggle"
               type="button" id="dropdownMenuButton"
               data-bs-toggle="dropdown"
               aria-expanded="false">
            User@gmail.com
            
        </button>
        <ul class="dropdown-menu" aria-labelledby="dropdownMenuButton">
            <li>
                <span class="dropdown-item-text">
```

```

        
        user@gmail.com
    </span>
</li>
<li><hr class="dropdown-divider"></li>
<li>
        <form id="logoutForm" class="form-inline">
            <button id="logout" type="submit" class="nav-link
btn btn-link text-dark">Logout</button>
        </form>
    </li>
</ul>
</div>
<!-- Информация пользователя – конец --&gt;
&lt;/nav&gt; &lt;!-- Панель навигации – конец --&gt;
</pre>

```

### 3.8. Разметка подвала страницы

Подвал страницы (Рисунок 8) представляет собой навигационную панель, смещенную вправо (класс «ms-auto»). Кнопки панели представляют собой иконки социальных сетей.

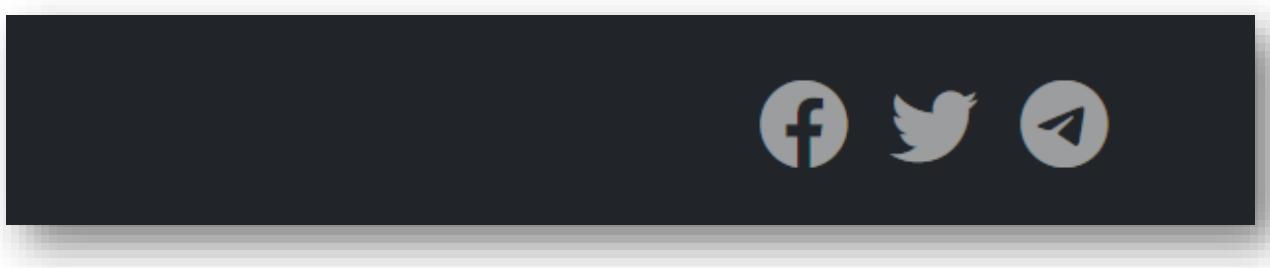


Рисунок 8 Подвал страницы

Пример разметки

```

<footer>
    <div class="container">
        <div class="navbar bg-dark navbar-expand-md border-bottom border-body"
            data-bs-theme="dark">
            <div class="navbar-nav ms-auto">
                <a class="nav-item nav-link" href="http://www.facebook.com">
                    <i class="bi bi-facebook" style="font-size: 2rem"></i>
                </a>
                <a class="nav-item nav-link" href="http://www.vk.com">
                    <i class="bi bi-twitter" style="font-size: 2rem"></i>
                </a>
                <a class="nav-item nav-link" href="http://www.twitter.com">
                    <i class="bi bi-telegram" style="font-size: 2rem"></i>
                </a>
            </div>
        </div>
    </div>
</footer>

```

### 3.9. Разметка содержимого страницы

Требуется выполнить HTML разметку страницы, которая имеет вид как указано на Рисунок 9.

**Список**

- A. элемент 1 списка
- B. элемент 2 списка

---

**Форма**

Логин:

Пароль:

**Отправить**

Рисунок 9 Содержимое страницы

**Список** формируется из тэгов `<ol>` (упорядоченный список), в котором для нумерации строк используются заглавные буквы.

Радио-кнопки, чек-боксы, поля ввода должны отправляться на сервер. Для этого они должны находиться внутри тэга `<form>`. Также не забудьте обязательно присвоить имена элементам формы (атрибут `name`)

Для наглядности в тэге `<form>` укажите метод отправки «`get`». Так вы сможете увидеть передаваемые формой данные в строке запроса.

Для разметки элементов формы используйте классы bootstrap (<https://getbootstrap.com/docs/5.3/forms/layout/#horizontal-form> )

### 3.10. Использование режима «Developer tools»

Измените метод отправки формы на POST

Откройте страницу в браузере. Нажмите клавишу F12. Откроется окно разработчика. Выберите закладку Network (сеть). Введите данные формы и нажмите «Отправить». Выберите в списке вашу страницу Html. Перейдите в окно Headers (заголовки) и найдите Form Data (данные формы). Убедитесь,

что данные отправились на сервер (Рисунок 10, использовался браузер Google Chrome):

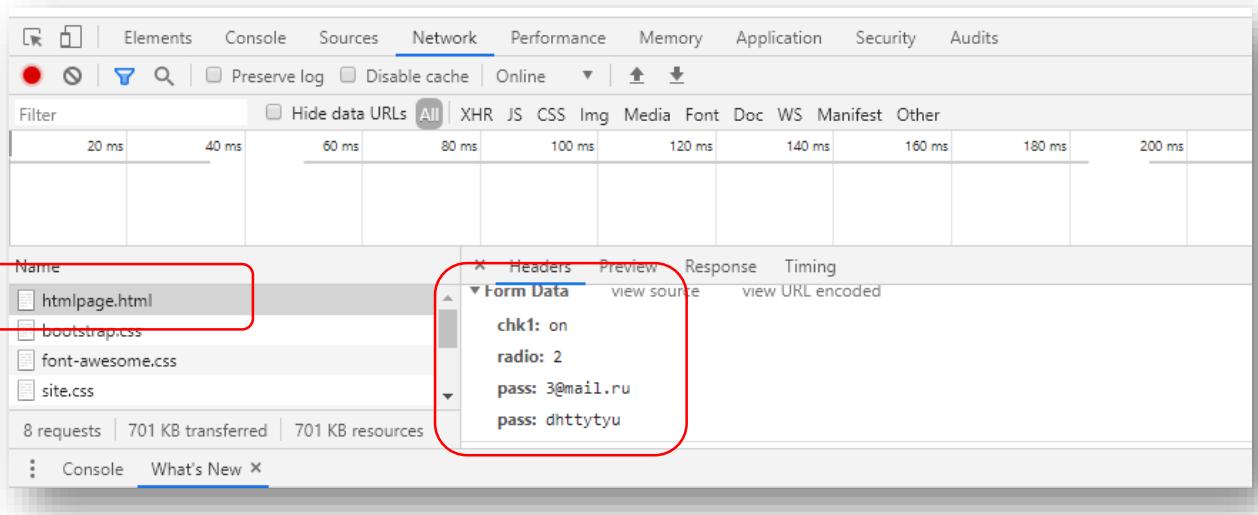


Рисунок 10 Режим разработчика

### 3.11. Окончательный вид страницы

Окончательный вид страницы приведен на Рисунок 11

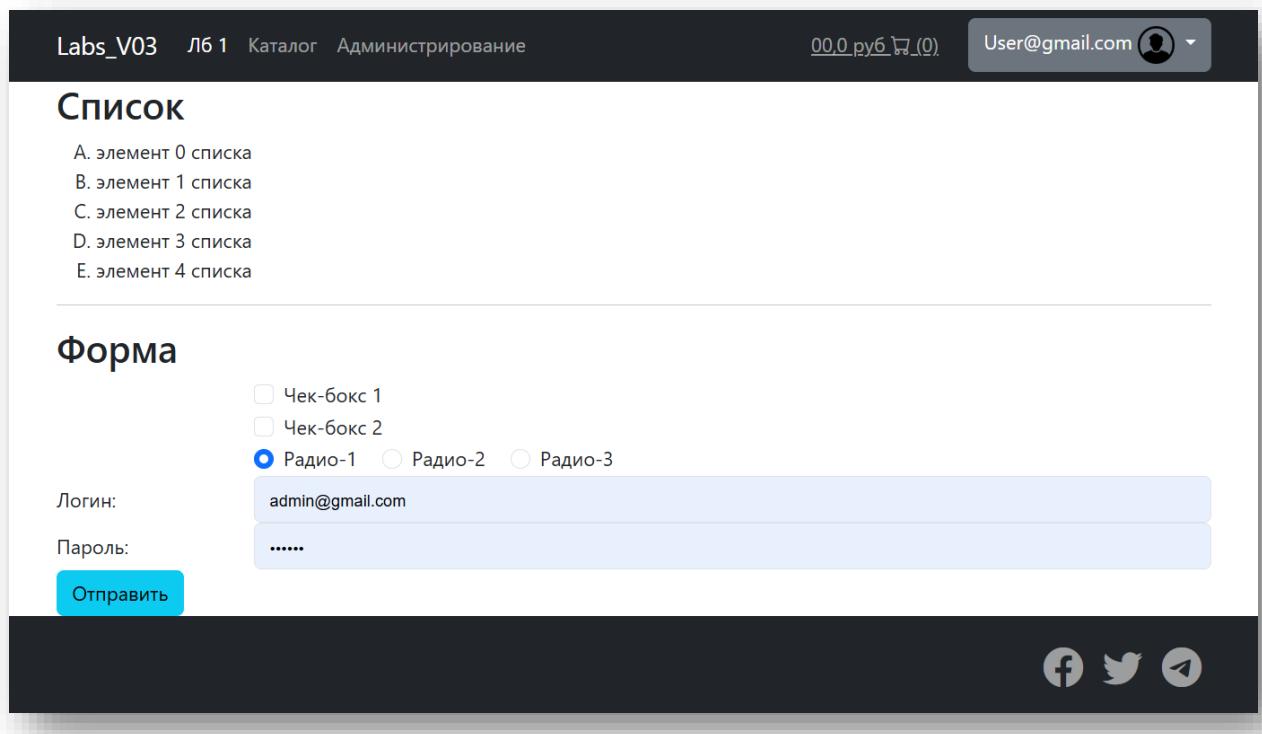


Рисунок 11 Окончательный вид страницы

### 3.12. Создание страницы-макета

- В папке Views/Shared удалите (или переименуйте) страницу макета (layout page) с именем \_Layout.cshtml.
- Создайте новую страницу макета

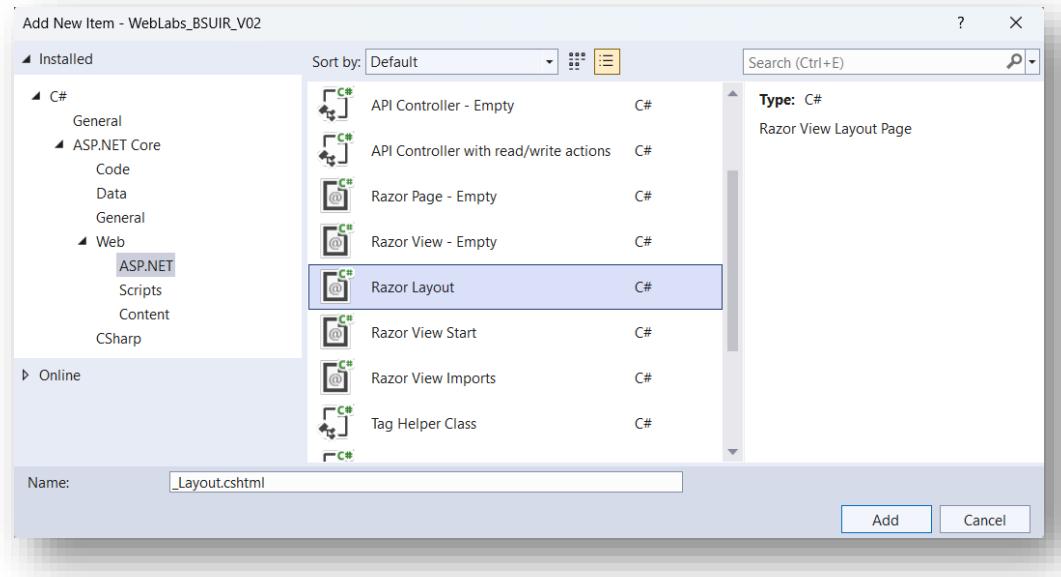


Рисунок 12 Добавление страницы-макета

- В тэге <head> добавьте подключение стилей (как на представлении Index)
- Поместите в тэг <body> разметку тэга <body> из представления Index, за исключением следующего: **внутри** тэга <main> удалите всю разметку и поместите @RenderBody()
- В конце тэга <body>, после загрузки скриптов, добавьте **необязательную асинхронную** секцию с именем «Scripts»
- Откройте в папке Views представление с именем \_ViewStart.cshtml и убедитесь, что в нем указано использование макета \_Layout.cshtml для всех страниц проекта;

### 3.13. Использование страницы-макета

- В представлении Index удалите всю разметку, за исключением разметки **внутри** тэга <main>.

b) Добавьте в представление Index кодовый блок

```
@{  
    ViewBag.Title = "Index";  
}
```

c) Запустите проект и проверьте, что внешний вид страницы не изменился и соответствует рис. 13.

#### 4. Вопросы для самопроверки

1. В какой папке проекта хранятся файлы представлений контроллера?
2. Где в проекте должен находиться статический контент?
3. Где в проекте создается конвейер обработки запросов?
4. Где в проекте размещаются файлы классов контроллеров?
5. Для чего используется файл \_ViewStart.cshtml?
6. Для чего используется файл \_ViewImports.cshtml?
7. Для чего используется файл макета (Layout)?
8. Как в Layout указать место, куда будет помещена разметка представления?
9. Что такое «Section» на макете (Layout)? Как описать секцию?
10. Как использовать секцию в представлении?
11. Как указать, что представление использует макет (Layout)?
12. Как указать, что представление НЕ использует макет (Layout)?
13. Что означает утверждение: «Протокол HTTP не поддерживает сохранение состояний»?
14. Как в тэге form указать адрес отправки данных?
15. В каком виде передаются данные от клиента на сервер?
16. Чем отличается передача данных а методах Post и Get?