

## 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>
```

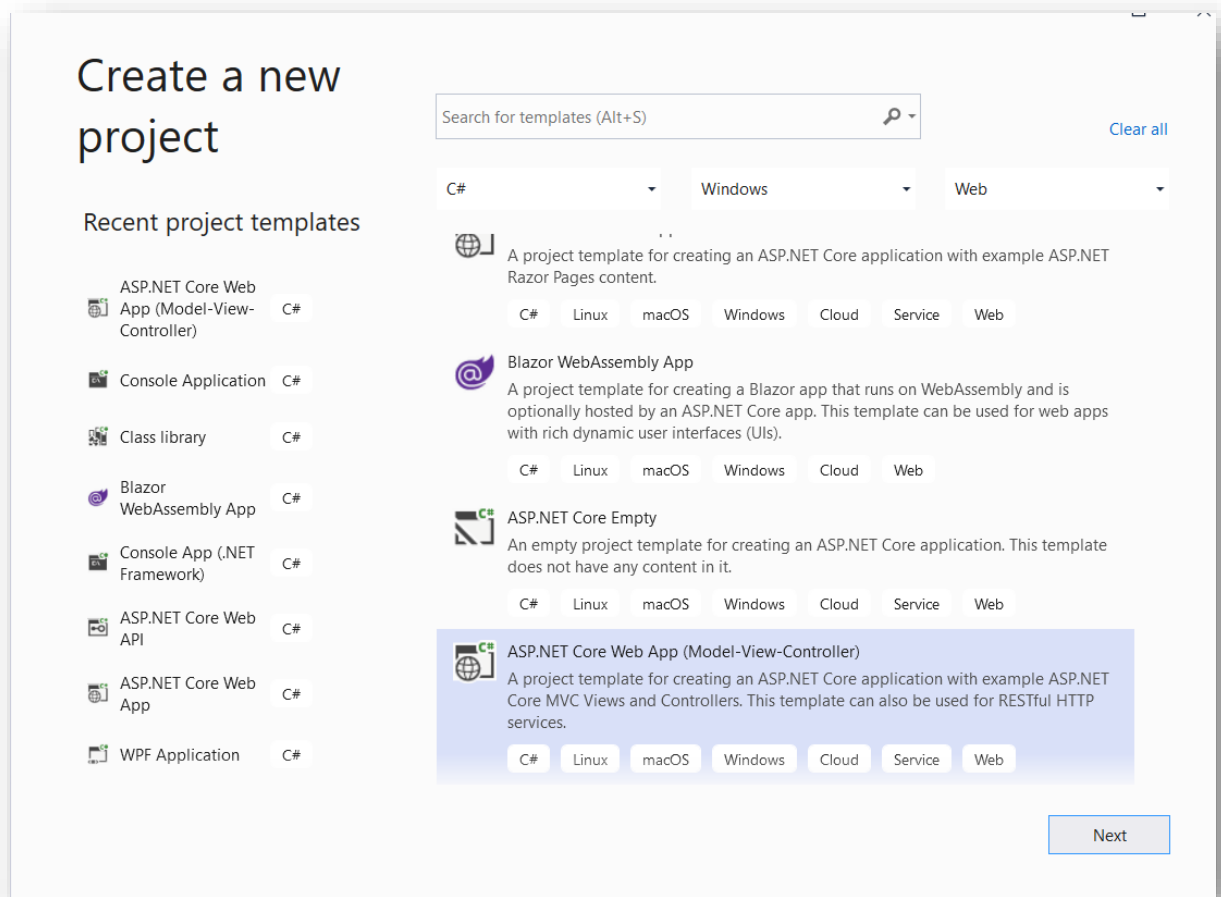


Рисунок 1 Создание проекта

Добавьте библиотеку стилей font-awesome. Это библиотека, содержащая большой выбор иконок. Подробную информацию можно найти на сайте производителя ( <https://fontawesome.com/> ) . Кликните правой кнопкой по папке ~wwwroot/lib и выберите Add-Client-side library (Рисунок 2)

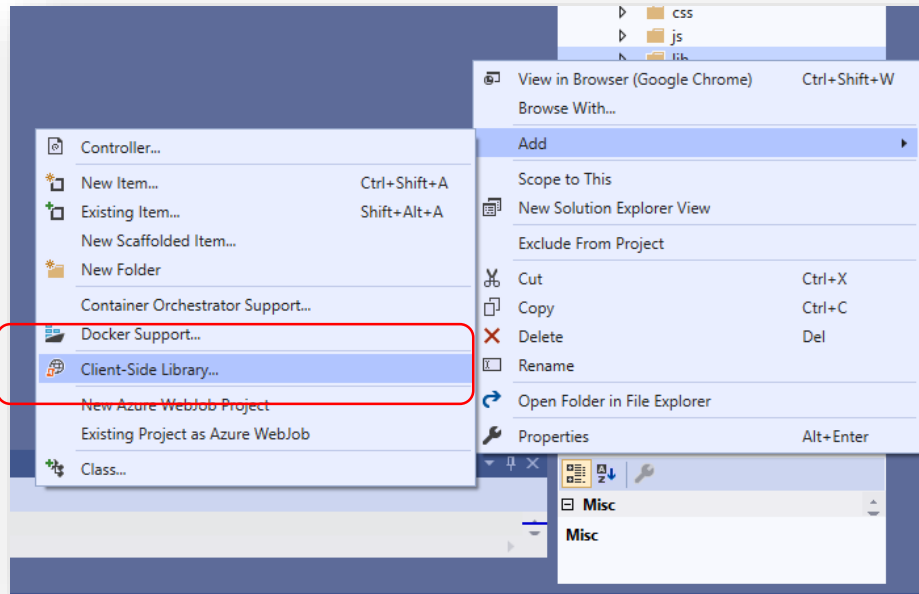


Рисунок 2 Добавление библиотеки

В открывшемся окне введите font-awesome (Рисунок 3).

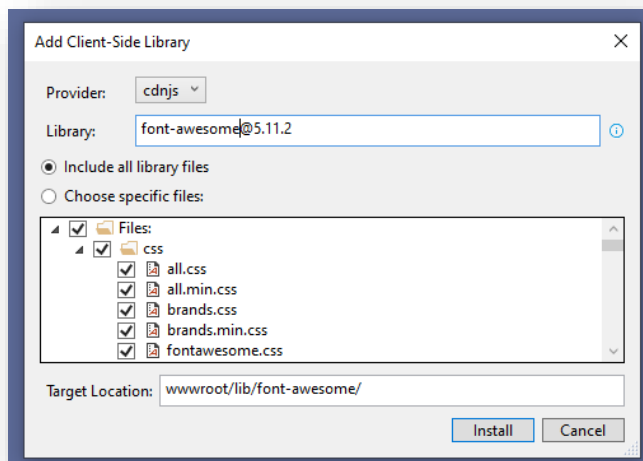


Рисунок 3 Выбор библиотеки

## 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 font-awesome --provider cdnjs --destination wwwroot/lib/font-  
awesome
```

**Примечание:** в приведенных ниже примерах используется библиотека font-awesome версии 6.2.1. Если в вашем проекте используется другая версия, то имена классов иконок могут не совпадать. Для информации – см. сайт производителя: <https://fontawesome.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 обработки 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 выводющую 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/font-awesome/css/all.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 5.1. 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

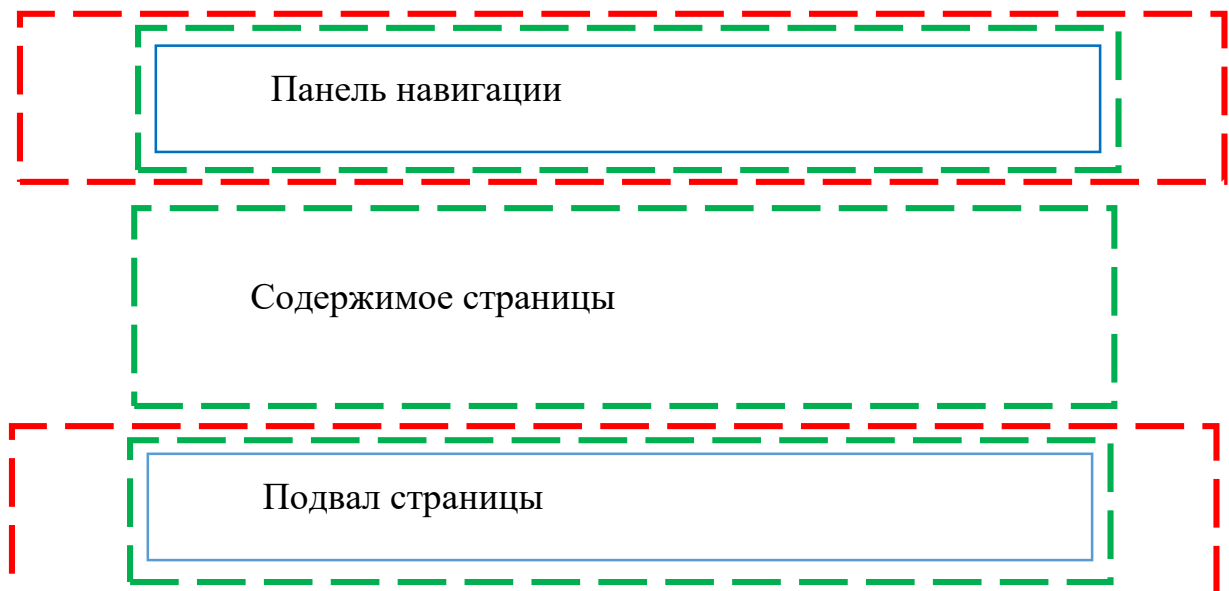


Рисунок 4 Структура страницы

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.

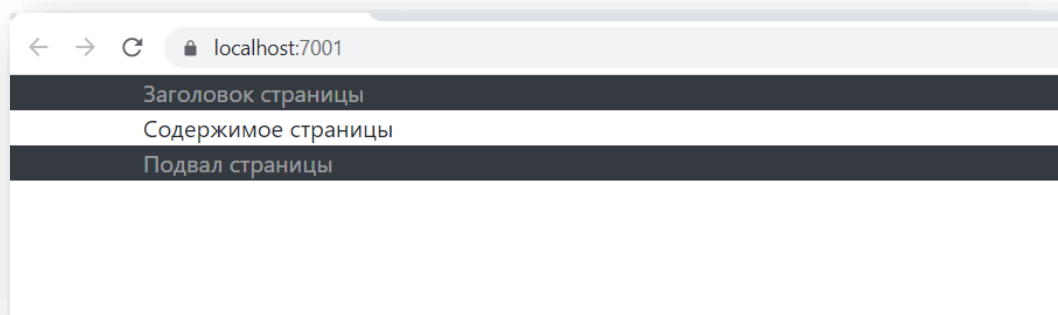


Рисунок 5 Базовый вид страницы

### 3.7. Оформление заголовка страницы

Заголовок страницы должен содержать следующее (Рисунок 6):

- меню сайта
- информация пользователя

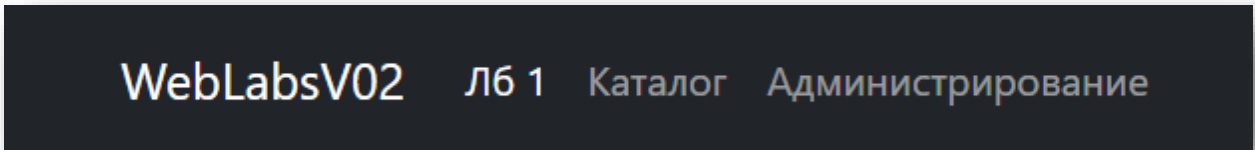


Рисунок 6 Заголовок страницы

Меню сайта (Рисунок 8) состоит из

Рисунок 7 Информация пользователя

- 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/5.1/components/navbar/>).

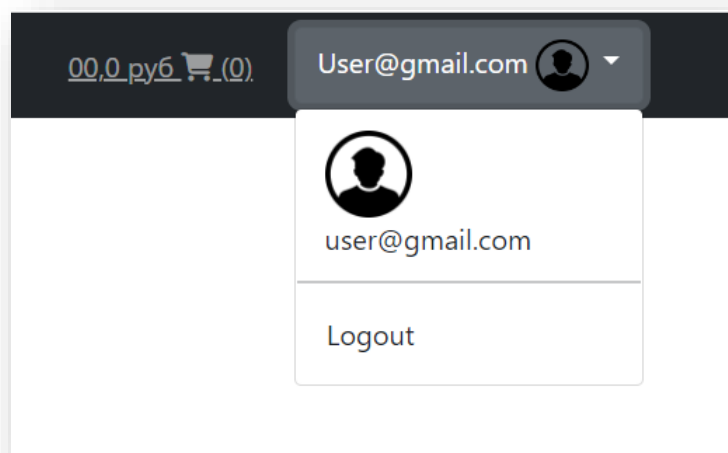


WebLabsV02 Лб 1 Каталог Администрирование

Рисунок 8 Меню сайта

Информация пользователя (Рисунок 7) состоит из:

- ссылка на страницу корзины заказа. Текст ссылки - информация о корзине заказа (общая стоимость и количество товаров). Для иконки использовать иконку font-awesome (класс «[fa-solid fa-cart-shopping](#)»)
- выпадающее меню в виде имени пользователя и его аватарки. ( см. <https://getbootstrap.com/docs/5.1/components/dropdowns/>) Меню состоит из аватарки, имени пользователя, кнопки «Logout» (или «Выход»)





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 navbar-expand-md navbar-dark bg-dark">
  <!-- меню сайта -->
  <a class="navbar-brand" href="#">WebLabsV02</a>
  <div class="navbar-nav">
    <a class="nav-item nav-link active" href="#">Л6 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="fa-solid fa-cart-shopping"></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 -->
</nav> <!-- Панель навигации - конец -->
```

### 3.8. Разметка подвала страницы

Подвал страницы (Рисунок 9) представляет собой навигационную панель, смещенную вправо (класс «ms-auto»). Кнопки панели представляют

собой иконки социальных сетей (классы font-awesome: «fa-brands fa-facebook-f», fa-brands fa-vk, fa-brands fa-twitter») увеличенного размера (класс «fa-2x»).

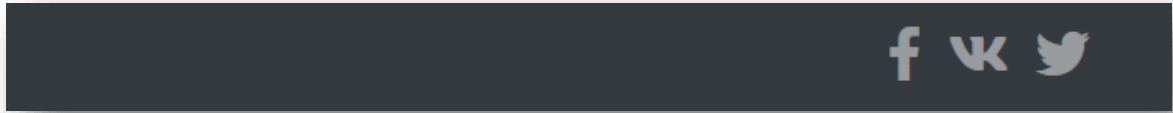


Рисунок 9 Подвал страницы

Пример разметки (внутри тэга `<div class="container">`):

```
<div class="navbar navbar-dark navbar-expand-md bg-dark">
  <div class="navbar-nav ms-auto">
    <a class="nav-item nav-link" href="http://www.facebook.com">
      <i class="fa-brands fa-facebook-f fa-2x"></i>
    </a>
    <a class="nav-item nav-link" href="http://www.vk.com">
      <i class="fa-brands fa-vk fa-2x"></i>
    </a>
    <a class="nav-item nav-link" href="http://www.twitter.com">
      <i class="fa-brands fa-twitter fa-2x"></i>
    </a>
  </div>
</div>
```

### 3.9. Разметка содержимого страницы

Требуется выполнить HTML разметку страницы, которая имеет вид как указано на Рисунок 10.

## Лабораторная работа 1

### Список

А. элемент 1 списка  
В. элемент 2 списка

---

### Форма

☐ Чек-бокс 1  
☐ Чек-бокс 2  
☒ Радио-1   ☐ Радио-2   ☐ Радио-3

Login:

Password:

Рисунок 10 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/5.1/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 11, использовался браузер Google Chrome):

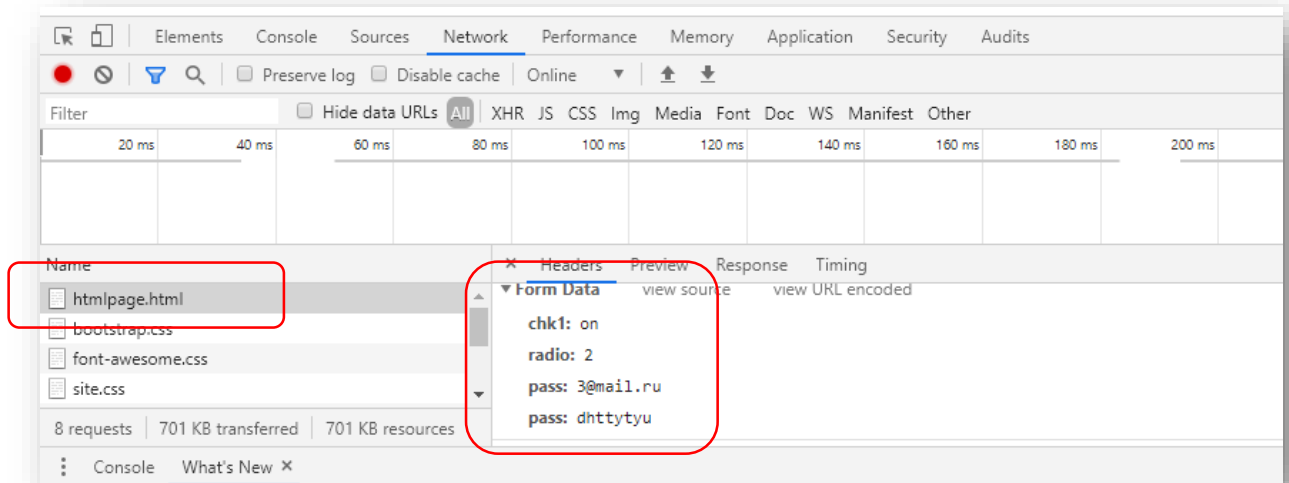


Рисунок 11 Режим разработчика

### 3.11. Окончательный вид страницы

Окончательный вид страницы приведен на Рисунок 12

The screenshot shows a web application interface. At the top, there is a dark header bar with the text 'WebLabsV02 Л6 1 Каталог Администрирование' on the left, '00\_0\_ру6 (0)' in the center, and 'User@gmail.com' with a user icon on the right. Below the header, the main content area has a title 'Лабораторная работа 1' in large bold font, followed by a subtitle 'Список'. Under 'Список', there is a list of five items: 'А. элемент 0 списка', 'В. элемент 1 списка', 'С. элемент 2 списка', 'D. элемент 3 списка', and 'Е. элемент 4 списка'. Below this list is a horizontal line, followed by the title 'Форма'. Under 'Форма', there are three checkboxes: 'Чек-бокс 1', 'Чек-бокс 2', and 'Радио-1' (which is selected). To the right of 'Радио-1' are two more radio buttons: 'Радио-2' and 'Радио-3'. Below these are two input fields: 'Введите имя' and 'Введите пароль'. To the left of these fields are the labels 'Логин:' and 'Пароль:'. Below the input fields is a blue button labeled 'Отправить'. At the bottom of the page, there is a dark footer bar with social media icons for Facebook, VK, and Twitter.

Рисунок 12 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

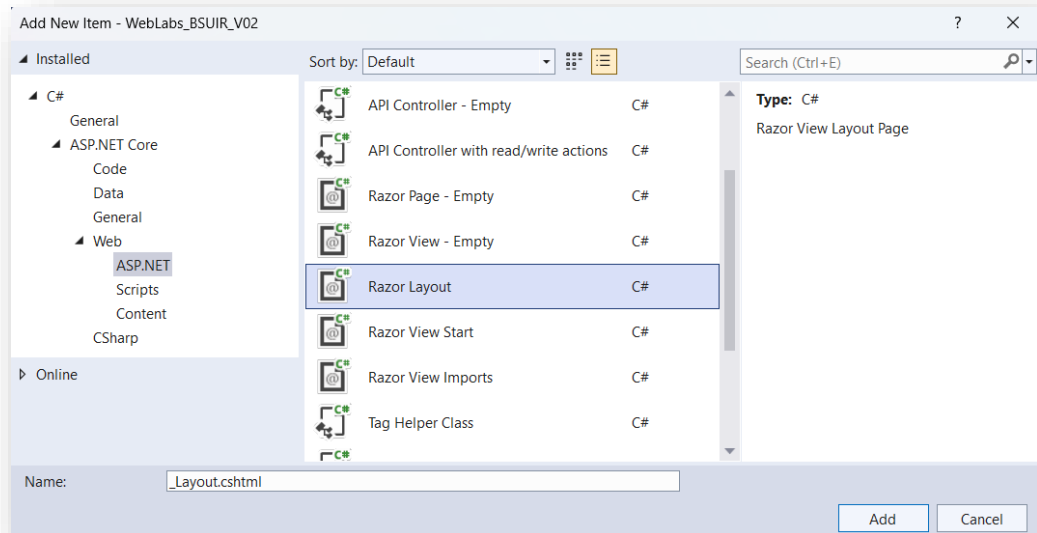


Рисунок 13 Добавление страницы-макета

- с) В тэге `<head>` добавьте подключение стилей (как на представлении Index)
- д) Поместите в тэг `<body>` разметку тэга `<body>` из представления Index, за исключением следующего: **внутри** тэга `<main>` удалите всю разметку и поместите **@RenderBody()**
- е) В конце тэга `<body>`, после загрузки скриптов, добавьте **необязательную асинхронную секцию** с именем «Scripts»
- ф) 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

- а) В представлении Index удалите всю разметку, за исключением разметки **внутри** тэга `<main>`.
- б) Добавьте в представление Index кодовый блок

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

с) *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 методами Post и Get?