



**МИНОБРНАУКИ РОССИИ**  
**федеральное государственное бюджетное образовательное учреждение**  
**высшего образования**  
**«Московский государственный технологический университет «СТАНКИН»**  
**(ФГБОУ ВО МГТУ «СТАНКИН»)**

---

**Институт**  
информационных технологий

**Кафедра**  
информационных систем

**Отчет по лабораторной работе №6**

по дисциплине **«Веб-программирование»**  
на тему: «Создание веб-приложений на основе MVC»

**Студент**  
группа ИДБ–21–06

**Музафаров К.Р.**

---

подпись

**Руководитель**  
старший преподаватель

**Перепелкина Ю.В.**

---

подпись

Москва 2023 г.

## ОГЛАВЛЕНИЕ

ВЫПОЛНЕНИЕ ЛАБОРАТОРНОЙ РАБОТЫ .....	3
ЛИСТИНГ КОДА.....	7
ВЫВОД.....	12
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ .....	13

# ВЫПОЛНЕНИЕ ЛАБОРАТОРНОЙ РАБОТЫ

## Задания

1. Создать новый проект ASP .NET MVC в среде Visual Studio (рис. 1).

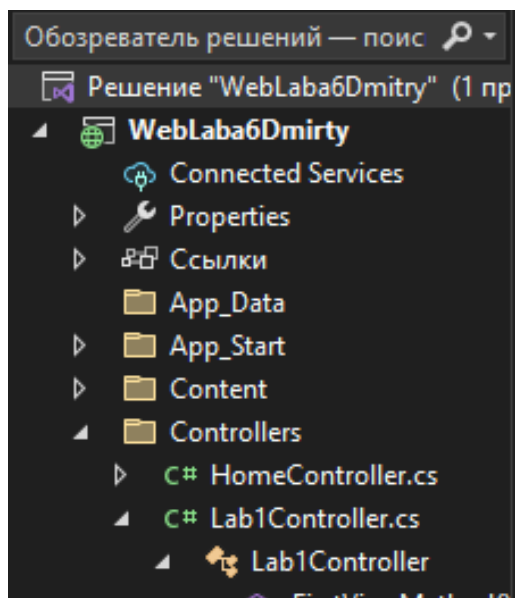


Рис. 1 Созданный проект

2. Создать новый контроллер (рис. 2).

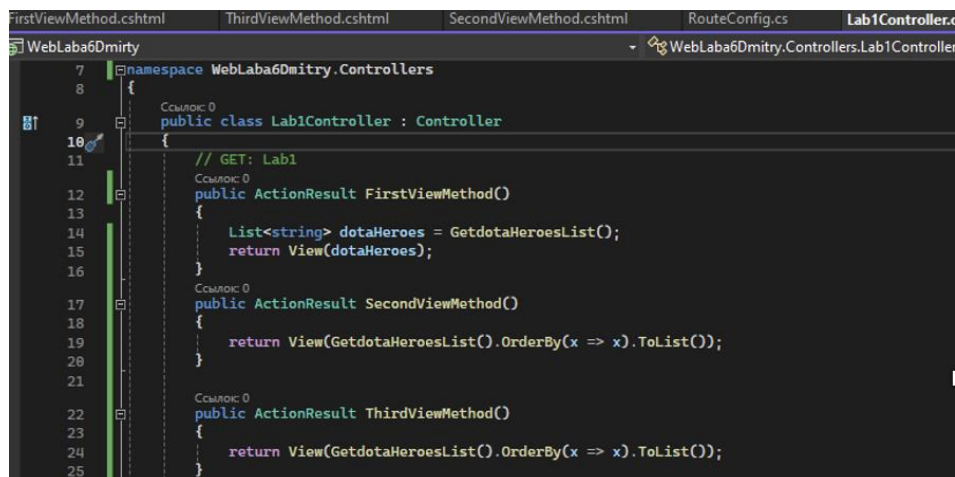


Рис. 2 Созданный контроллер

3. Создать метод, в котором осуществляется формирование списка строковых объектов (рис. 3).

```

public List<string> GetPlanetsList()
{
    List<string> planets = new List<string>();
    planets.Add("Мазерати");
    planets.Add("Вольфсваген");
    planets.Add("Зил");
    planets.Add("Мазерати2.0");
    planets.Add("Юниор");
    planets.Add("Сузуки");
    planets.Add("УАЗ");
    planets.Add("Ниссан");
    return planets;
}

```

Рис. 3 Метод с формированием списка

4. Создать методы и представления для отображения данных на веб-странице.

FirstViewMethod (рис. 4-5).

```

@model List<string>
<h2>Список героев</h2>
@foreach (var dotaHeroes in Model)
{
    <div>@dotaHeroes</div>
}

```

Рис. 4 FirstViewMethod

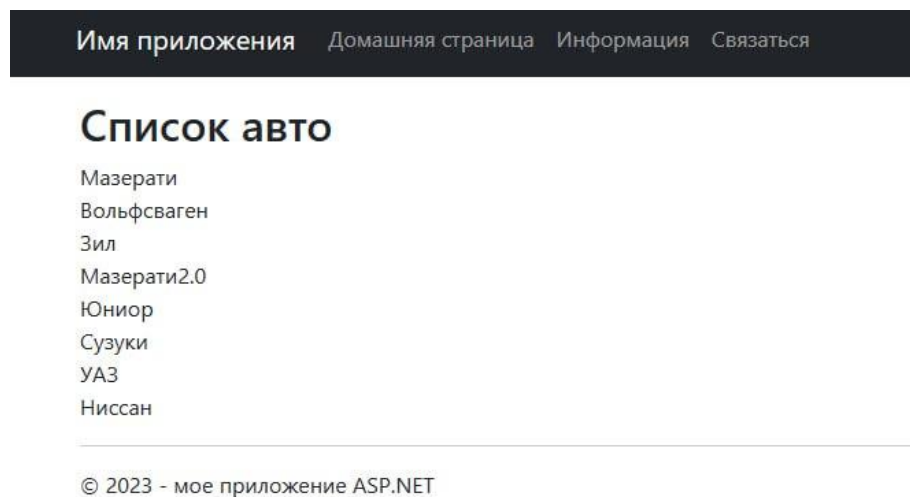


Рис. 5 Представление со списком данных

SecondViewMethod (рис. 6-7).

```

@model List<string>

<h2>Отсортированный список героев(таблица)</h2>

<table class="table" style="width:20%">
  <thead>
    <tr>
      <th class="text-center">Наименование</th>
    </tr>
  </thead>
  <tbody>
    @foreach (var dotaHeroes in Model)
    {
      <tr class="text-center">
        <td>@dotaHeroes</td>
      </tr>
    }
  </tbody>
</table>

```

Рис. 6 SecondViewMethod

## Отсортированный список авто(таблица)

Наименование
Вольфсваген
Зил
Мазерати
Мазерати2.0
Ниссан
Сузуки
УАЗ
Юниор

© 2023 - мое приложение ASP.NET

Рис. 7 Веб-страница второго представления

ThirdViewMethod (рис. 8-9).

```
@model List<string>

<h2>Группировка героев (по таблицам)</h2>

@{
    var currentModel = Model.GroupBy(x => x.Substring(0, 1)).ToList();
    for (int letterIndex = 0; letterIndex < currentModel.Count(); letterIndex++)
    {
        <table class="table" style="width:20%; border: 1px solid;">
            <thead>
                <tr>
                    <th class="text-center">@currentModel[letterIndex].Key</th>
                </tr>
            </thead>
            <tbody>
                @for (int objectIndex = 0; objectIndex < currentModel[letterIndex].Count(); objectIndex++)
                {
                    <tr class="text-center">
                        <td>@currentModel[letterIndex].ElementAt(objectIndex)</td>
                    </tr>
                }
            </tbody>
        </table>
    }
}
```

Рис. 8 ThirdViewMethod

## Группировка машин (по таблицам)

<b>В</b>
Вольфсваген

<b>З</b>
Зил

<b>М</b>
Мазерати
Мазерати2.0

<b>Н</b>
Ниссан

<b>С</b>
Сузуки

<b>У</b>
УАЗ

<b>Ю</b>
Юниор

---

© 2023 - мое приложение ASP.NET

Рис. 9 Веб-страница третьего представления

### ЛИСТИНГ КОДА

#### Задание 1

#### Lab1Controller.cs

```
using System;
```

```

using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace WebLaba6Karim.Controllers
{
    public class Lab1Controller : Controller
    {
        // GET: Lab1
        public ActionResult FirstViewMethod()
        {
            List<string> dotaAuto = GetdotaAutoList();
            return View(dotaAuto);
        }

        public ActionResult SecondViewMethod()
        {
            return View(GetdotaAutoList().OrderBy(x => x).ToList());
        }

        public ActionResult ThirdViewMethod()
        {
            return View(GetdotaAutoList().OrderBy(x => x).ToList());
        }

        public List<string> GetdotaAutoList()
        {
            List<string> dotaAuto = new List<string>();
            dotaAuto.Add("Мазератти");
            dotaAuto.Add("Вольксваген");
            dotaAuto.Add("Зил");
        }
    }
}

```



```

        dotaAuto.Add("Мазератти2.0");
        dotaAuto.Add("Юниор");
        dotaAuto.Add("Сузуки");
        dotaAuto.Add("Уаз");
        dotaAuto.Add("Ниссан");

        return dotaAuto;
    }

}
}

RouteConfig.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Routing;
namespace WebLaba6Ksenia
{
    public class RouteConfig
    {
        public static void RegisterRoutes(RouteCollection routes)
        {
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

            routes.MapRoute(
                name: "Default",
                url: "{controller}/{action}/{id}",

```

```

        defaults: new { controller = "Lab1", action = "FirstViewMethod", id =
UrlParameter.Optional }

    );
}
}
}

```

### **FirstViewMethod.cshtml**

```

@model List<string>
<h2>Список Авто</h2>
@foreach (var dotaAuto in Model)
{
    <div>@dotaAuto</div>
}

```

### **SecondViewMethod.cshtml**

```

@model List<string>
<h2>Отсортированный список Авто(таблица)</h2>
<table class="table" style="width:20% ">
    <thead>
        <tr>
            <th class="text-center">Наименование</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var dotaAuto in Model)
        {
            <tr class="text-center">
                <td>@dotaAuto</td>
            </tr>
        }
    </tbody>

```

```
</table>
```

### **ThirdViewMethod.cshtml**

```
@model List<string>
```

```
<h2>Группировка Авто (по таблицам)</h2>
```

```
@{
```

```
    var currentModel = Model.GroupBy(x => x.Substring(0, 1)).ToList();
```

```
    for (int letterIndex = 0; letterIndex < currentModel.Count(); letterIndex++)
```

```
    {
```

```
        <table class="table" style="width:20%; border: 1px solid;">
```

```
            <thead>
```

```
                <tr>
```

```
                    <th class="text-center">@currentModel[letterIndex].Key</th>
```

```
                </tr>
```

```
            </thead>
```

```
            <tbody>
```

```
                @for (int objectIndex = 0; objectIndex <
```

```
currentModel[letterIndex].Count(); objectIndex++)
```

```
                {
```

```
                    <tr class="text-center">
```

```
                        <td>@currentModel[letterIndex].ElementAt(objectIndex)</td>
```

```
                    </tr>
```

```
                }
```

```
            </tbody>
```

```
        </table>
```

```
    }
```

```
}
```

## **ВЫВОД**

В ходе лабораторной работы сформировали знания, умения и навыки работы с современными системами контроля версий и базовыми инструментами создания веб-приложений, используя современный пакет Фреймворков типа MVC. Освоить механизмы создания простейших приложений с использованием ASP.NET MVC без подключения источника данных.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста. 2-е изд. — СПб.: Питер, 2012. — 304 с.: ил. (ссылка на Яндекс-диск <https://disk.yandex.ru/i/d-ky8jRcWqHR6g>)
2. Чакон С., Штрауб Б. Git для профессионального программиста. — СПб.: Питер, 2016.— 496 с.:ил. — (Серия «Библиотека программиста»). (ссылка на Яндекс-диск [https://disk.yandex.com/i/1WE8ZrQ27\\_b-Dw](https://disk.yandex.com/i/1WE8ZrQ27_b-Dw)).
3. Решения и проекты в Visual Studio [Электронный ресурс] / Хранилище документации Майкрософт — Режим доступа: <https://learn.microsoft.com/ru-ru/visualstudio/ide/solutions-and-projects-in-visual-studio?view=vs-2022&source=recommendations>
4. Что такое MVC: рассказываем простыми словами [Электронный ресурс] / Хекслет — Режим доступа: <https://ru.hexlet.io/blog/posts/chto-takoe-mvc-rasskazyvaem-prostyimi-slovami>