Министерство образования и науки

Санкт-Петербургский национальный исследовательский университет  
информационных технологий, механики и оптики

Факультет инфокоммуникационных технологий

Отчет  
о выполнении лабораторных работ  
по теме «Разработка Web приложений с использованием ASP.NET MVC»  
Дисциплина: «Создание программного обеспечения инфокоммуникационных систем»

Выполнил: студент группы K34202 С. А. Сазонов  
Проверил: к.т.н., доцент Н. А. Осипов

Санкт-Петербург  
2022

# Лабораторная работа 1. Введение в разработку ASP.NET MVC Web Application

## Цель работы

Создать простой проект на основе контроллера и модели, изучить основные элементы веб-приложения.

## Ход работы

### Упражнение 1. Создание веб-приложения ASP.NET MVC на основе контроллера

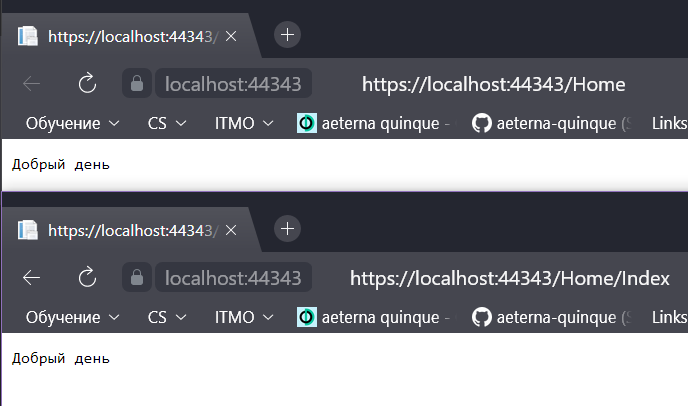


Рисунок 1 — Результат работы метода Index контроллера Home

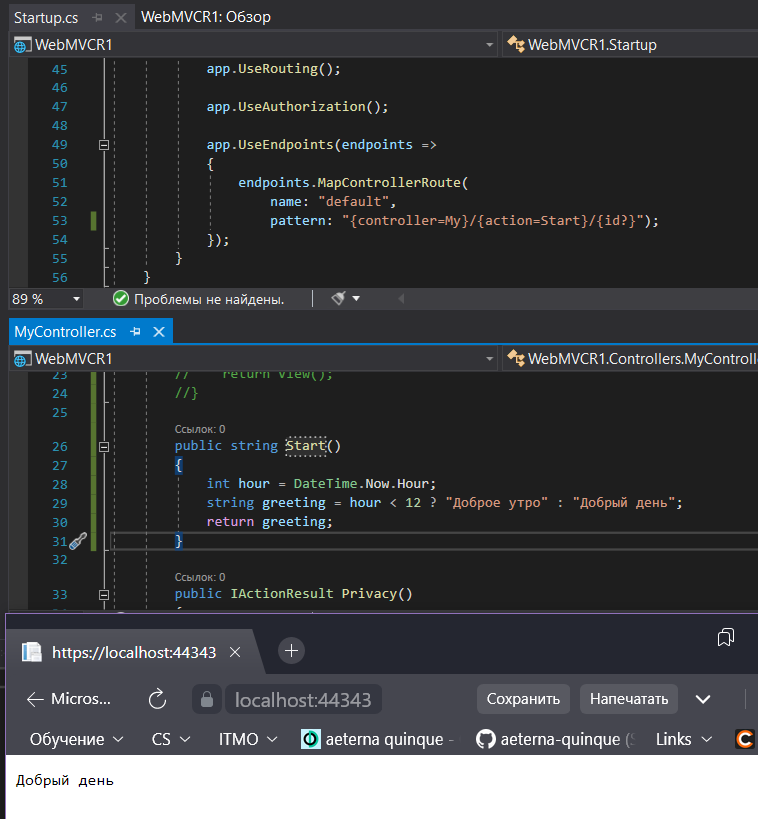


Рисунок 2 — Результат работы метода Start

### Упражнение 2. Передача параметров в контроллер

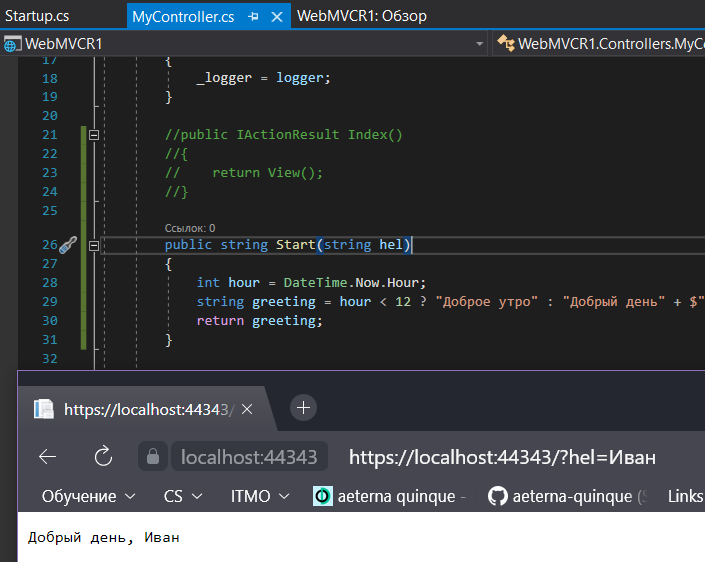


Рисунок 3 — Результат работы метода Start с входным параметром

### Упражнение 3. Реализация взаимодействия контроллера и модели

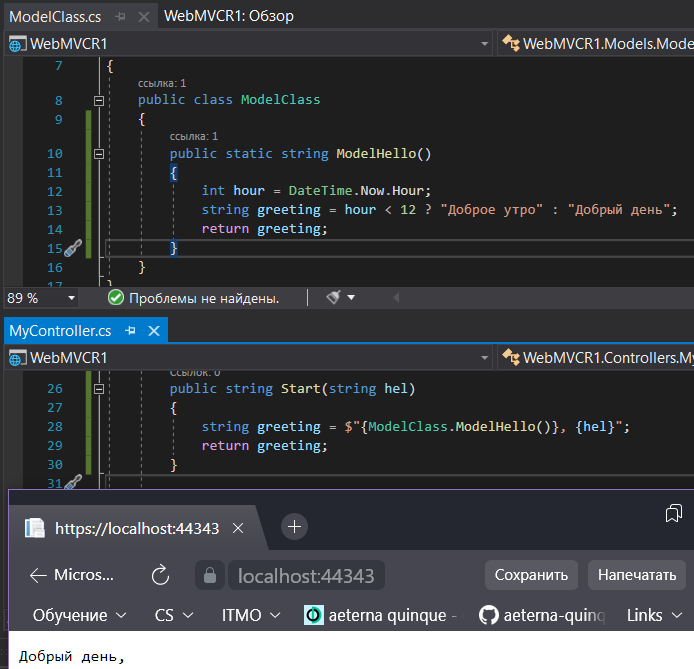


Рисунок 4 — Реализация взаимодействия контроллера и модели

## Выводы

В результате выполнения лабораторной работы был создан простой проект на основе элементов web-приложения. Был создан контроллер HomeController, модель ModelClass и реализовано взаимодействие между созданными элементами.

# 

# Лабораторная работа 2. Основы языка C#

## Цель работы

Изучить основы языка C# на основе созданного в прошлой лабораторной работе контроллера.

## Ход работы

### Упражнение 1. Создание и использование размерных типов данных

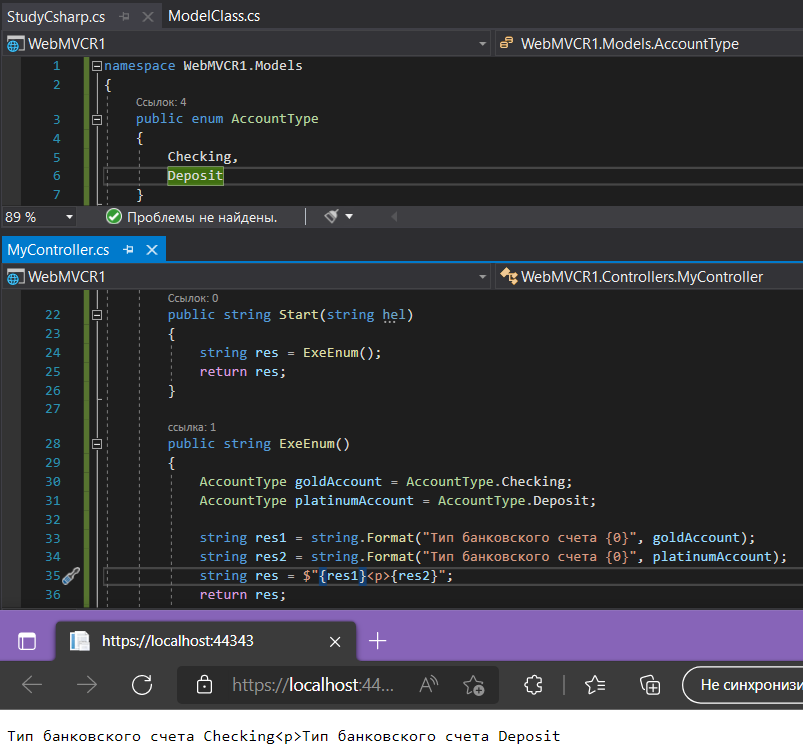


Рисунок 1 — Реализация метода ExeEnum

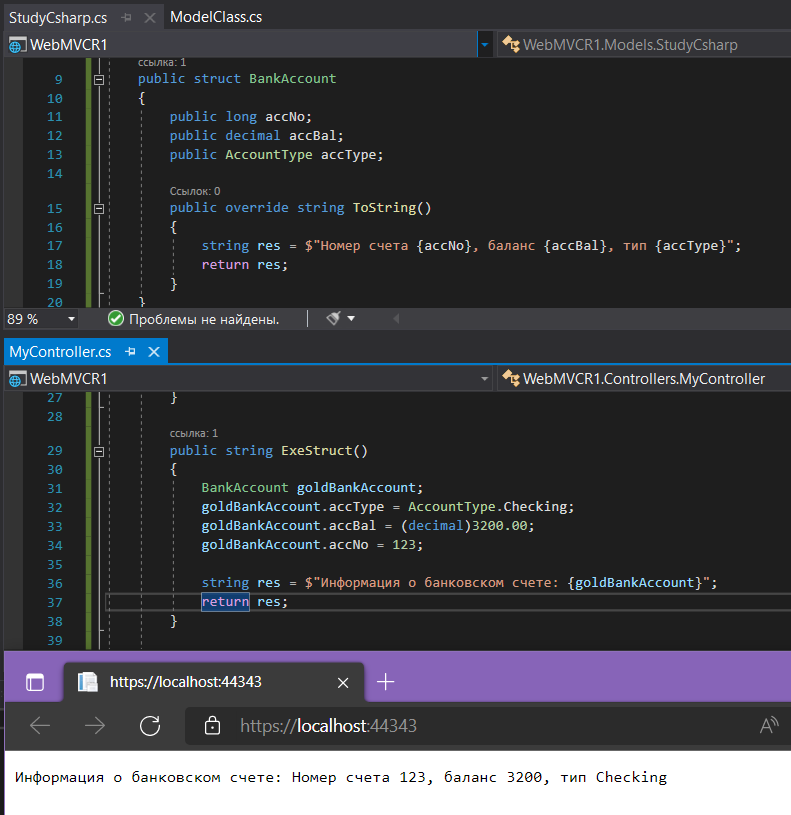


Рисунок 2 — Реализация метода ExeStruct

### Упражнение 2. Создание и использование методов

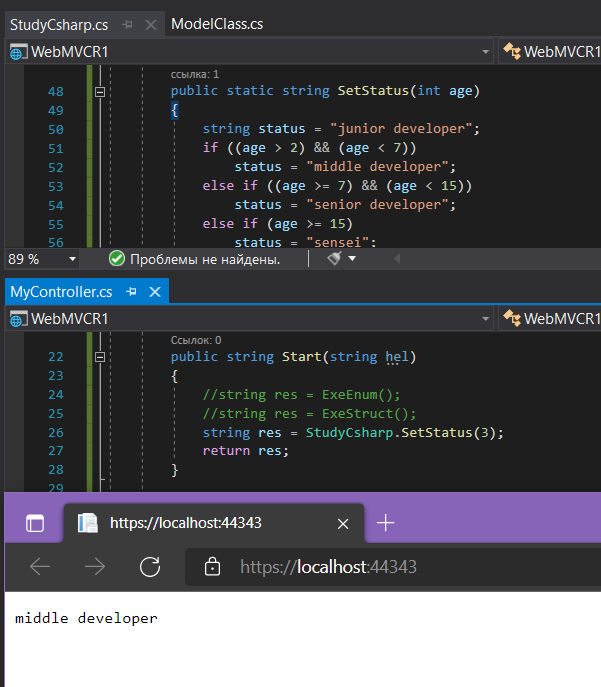


Рисунок 3 — Реализация метода SetStatus

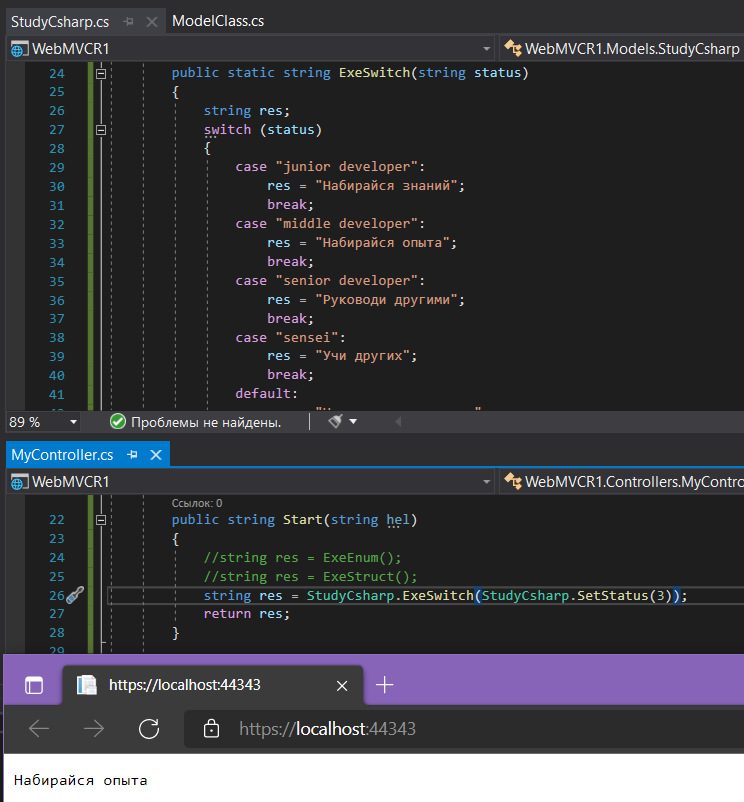


Рисунок 4 — Реализация метода ExeSwitch

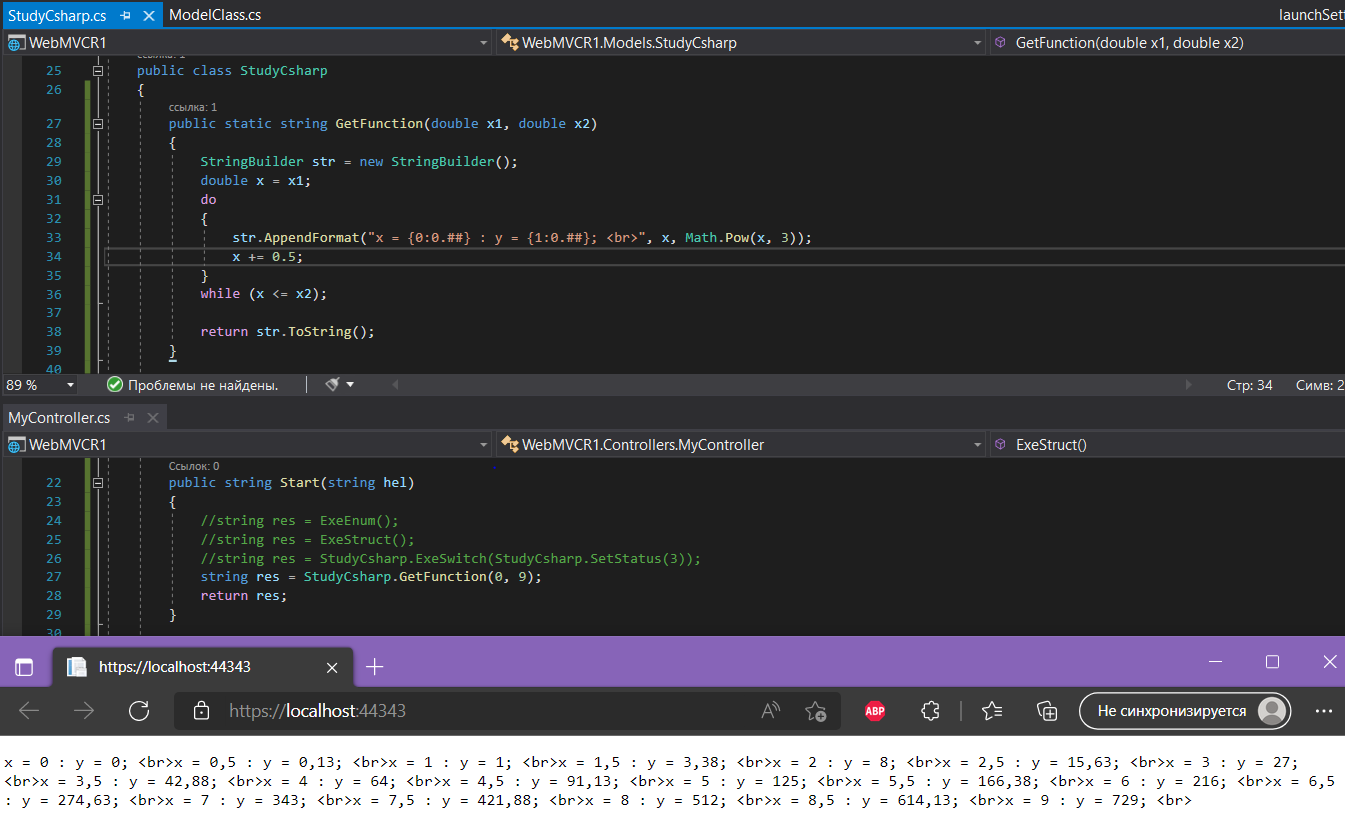


Рисунок 5 — Реализация метода GetFunction

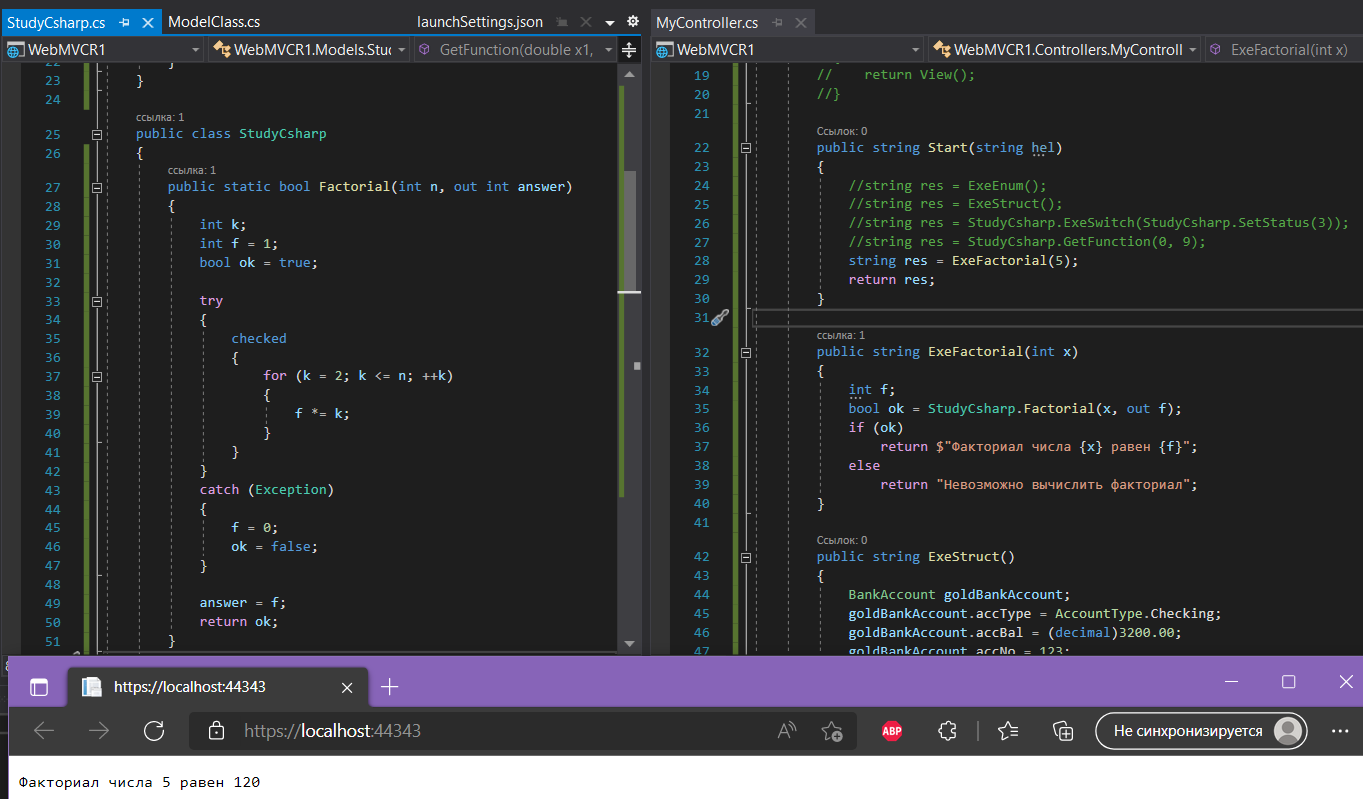


Рисунок 6 — Реализация метода ExeFactorial

### Упражнение 3. Реализация класса

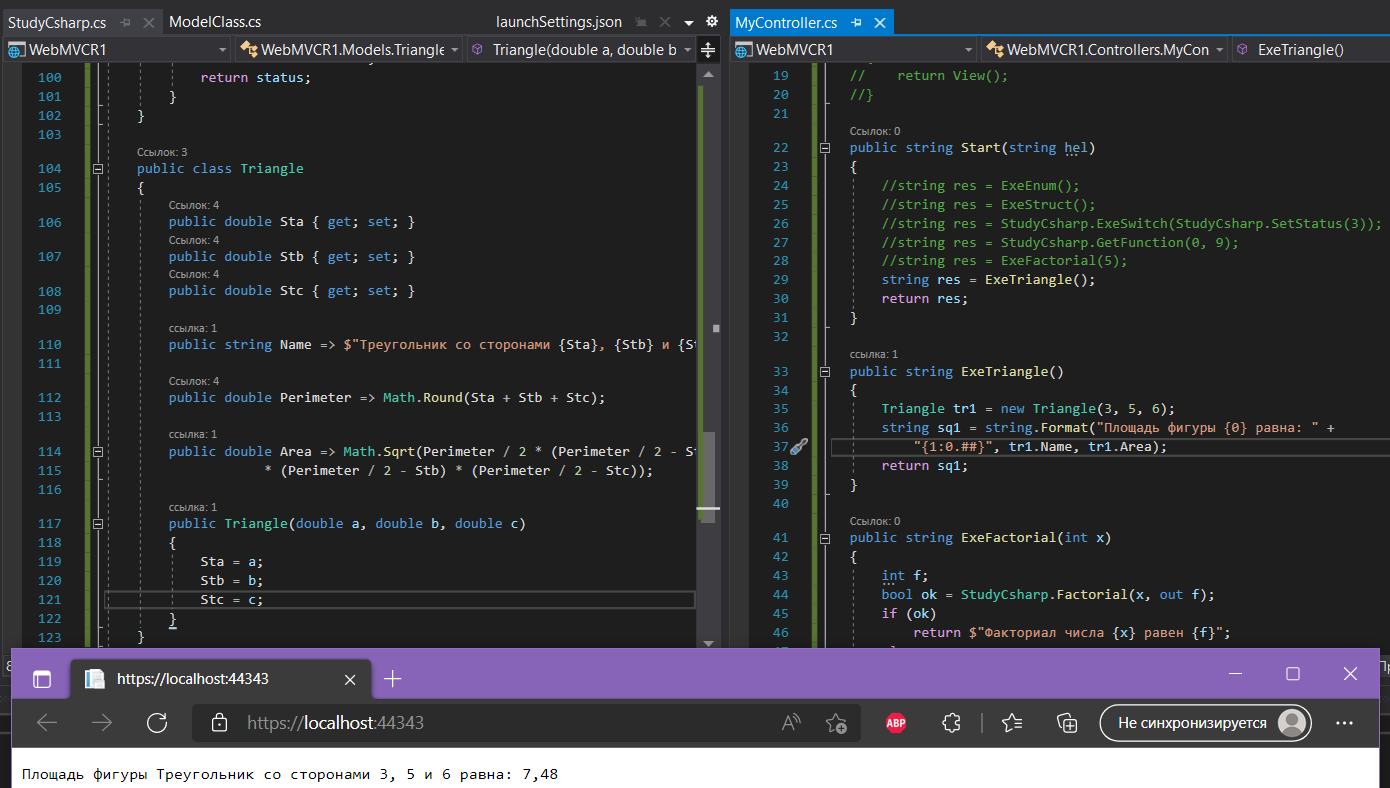


Рисунок 7 — Реализация класса Triangle

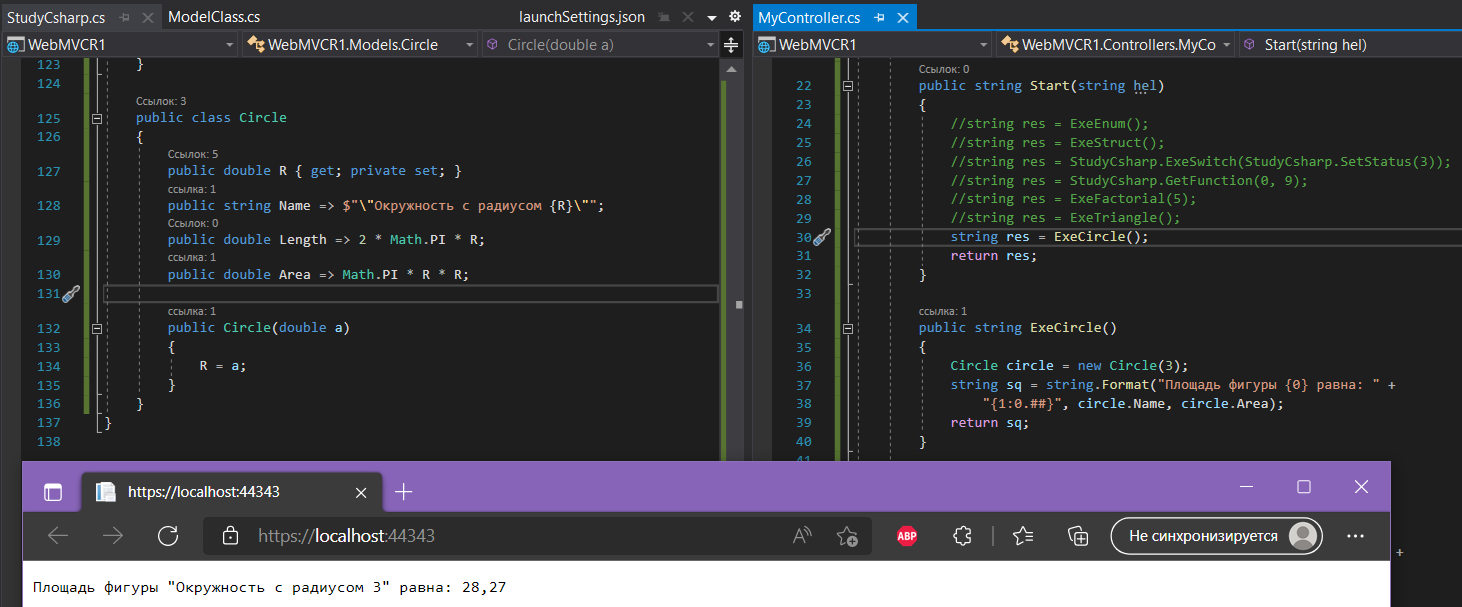


Рисунок 8 — Реализация класса Circle

### Упражнение 4. Наследование и реализация полиморфизма

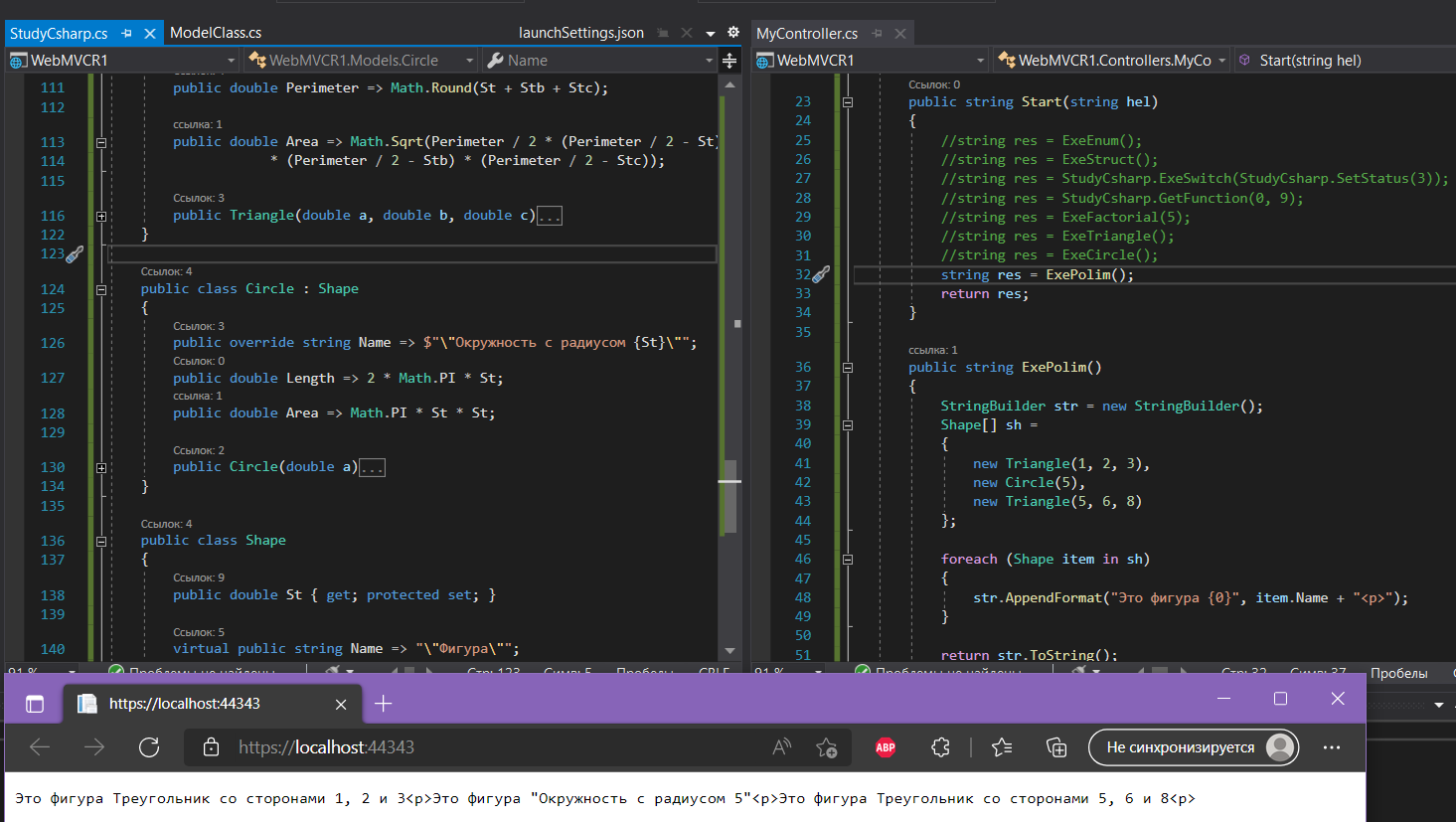


Рисунок 9 — Реализация полиморфизма и наследования от класса Shape

### Упражнение 5. Применение коллекций для группировки объектов

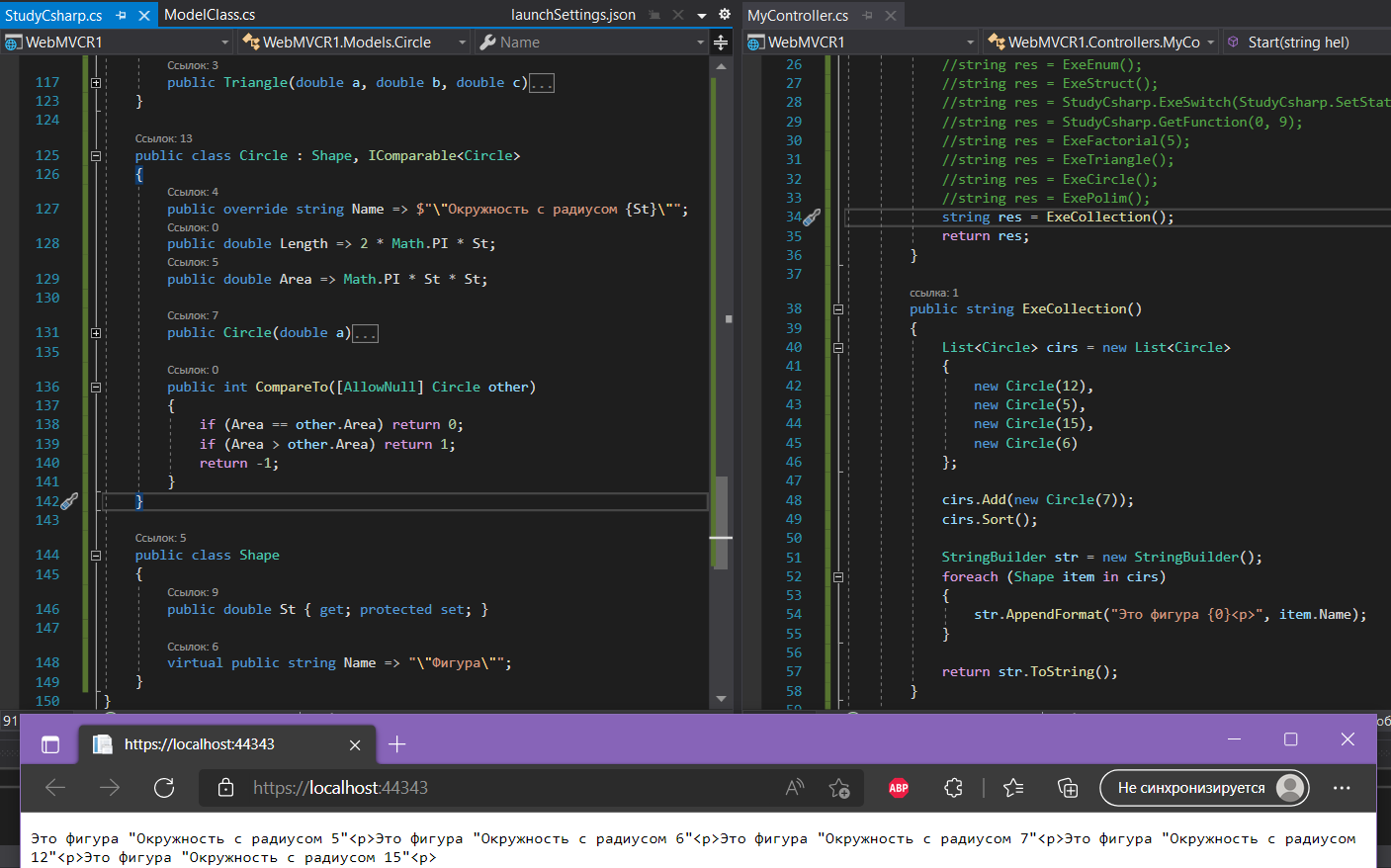


Рисунок 10 — Использование коллекции List для объектов класса Circle

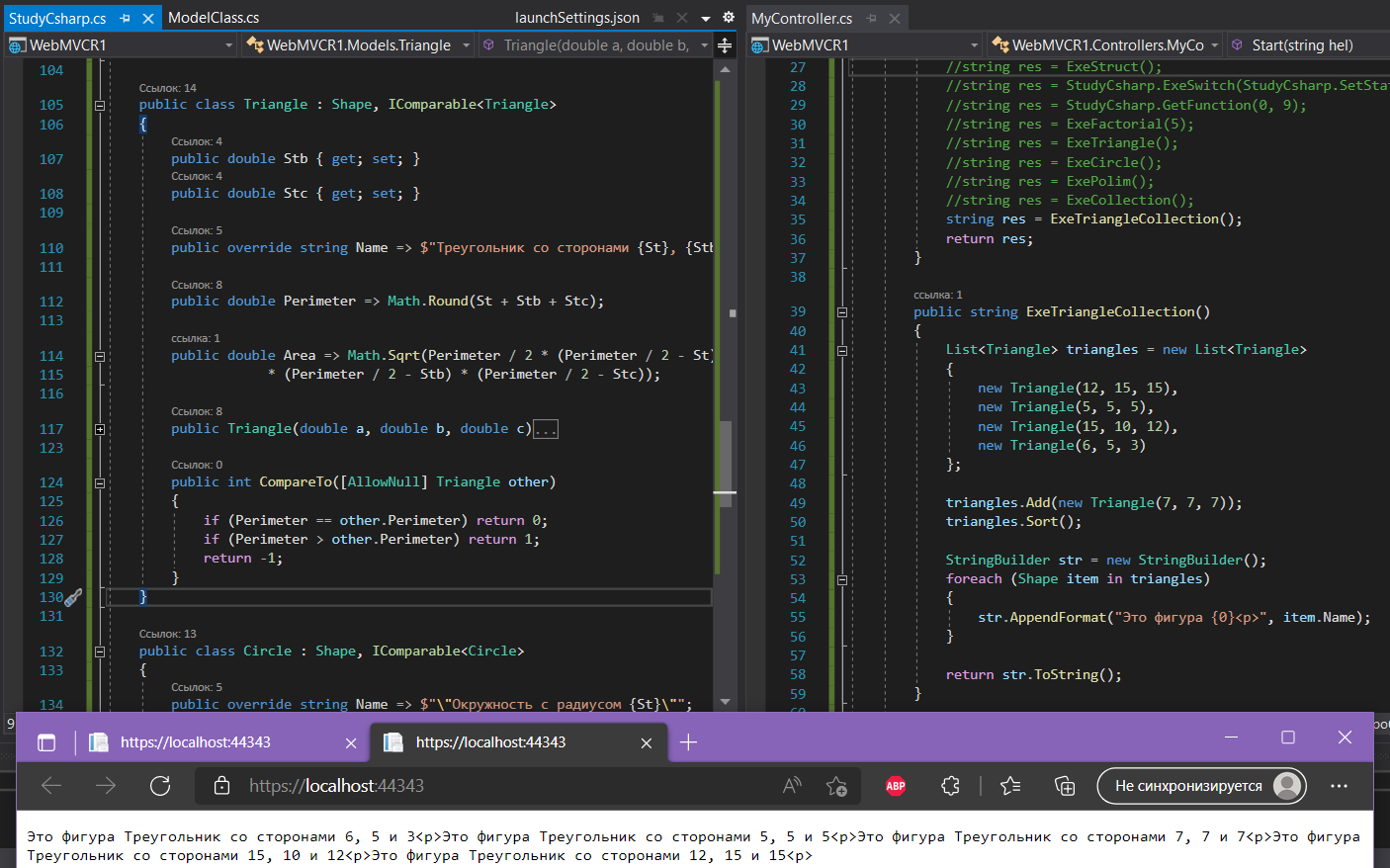


Рисунок 11 — Использование коллекции List для объектов класса Triangle

## Выводы

В результате выполнения лабораторной работы были изучены и использованы на практике такие элементы языка C#, как: перечисления, структуры, строки, классы, методы, операторы выбора, циклы, коллекции, интерфейсы. Также были изучены и реализованы наследование и полиморфизм — одни из основных принципов объектно-ориентированного программирования.

# 

# Лабораторная работа 3. Создание веб-приложения ASP.NET MVC

## Цель работы

Добавить в проект представление, модель и реализовать взаимодействие с ними в рамках шаблона MVC.

## Ход работы

### Упражнение 1. Реализация взаимодействия контроллера и представления

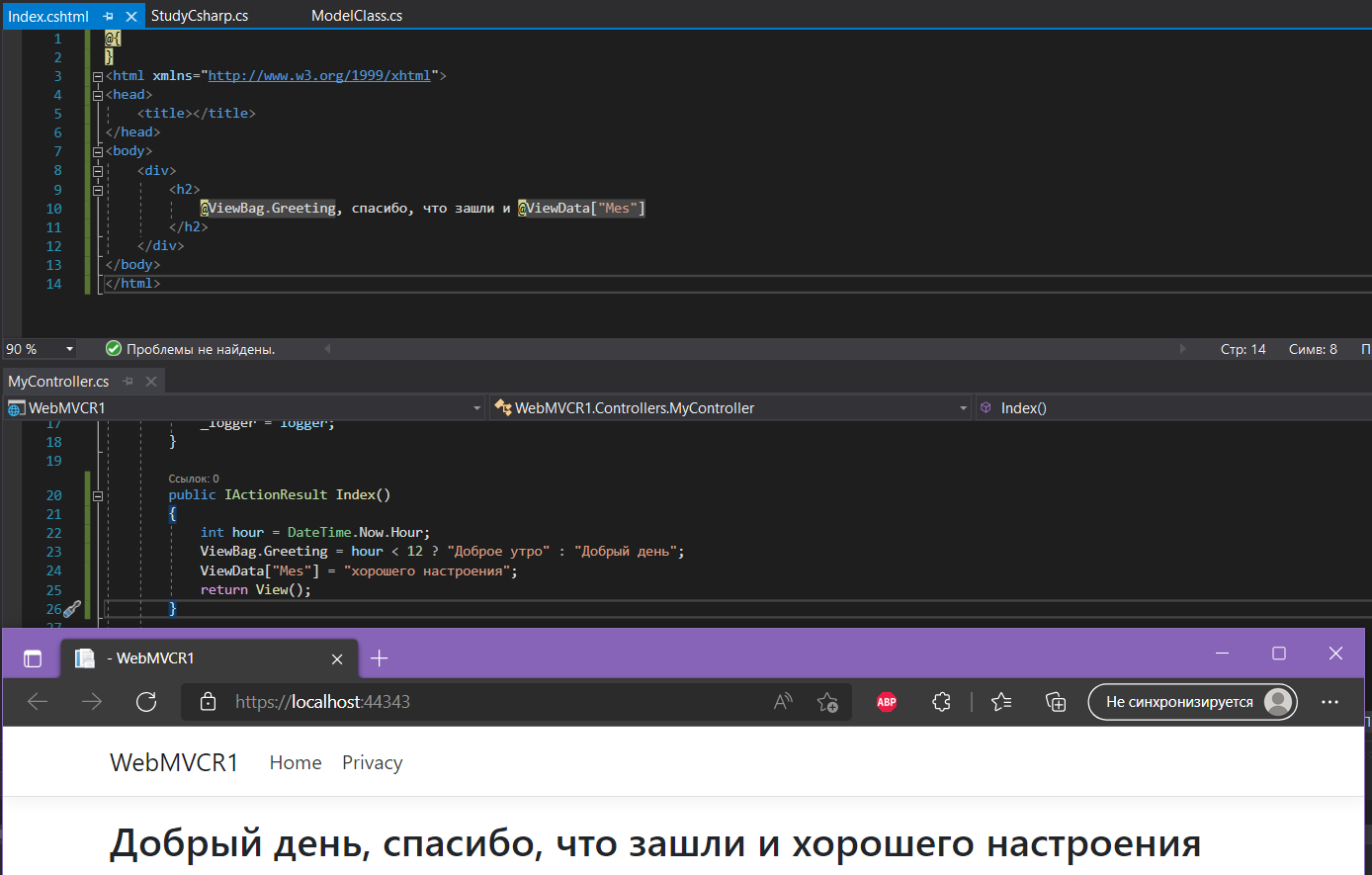


Рисунок 1 — Реализация взаимодействия контроллера My и представления Index

### Упражнение 2. Реализация взаимодействия модели, представления и контроллера в шаблоне MVC

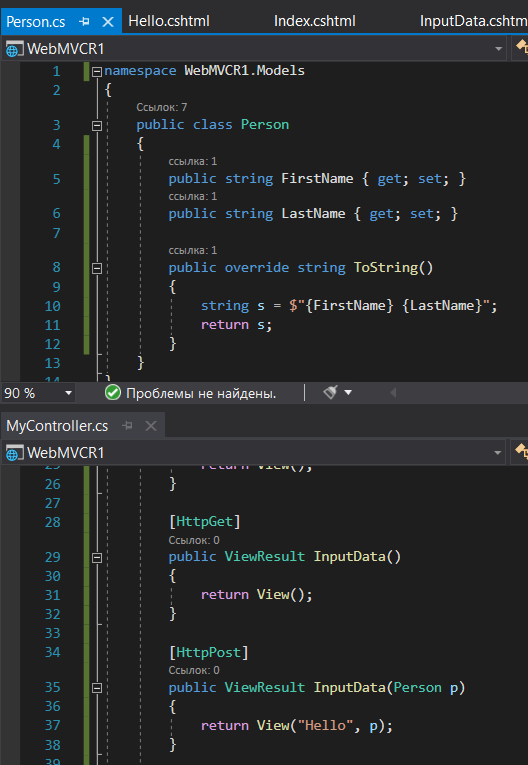


Рисунок 2 — Модель Person и методы InputData контроллера My

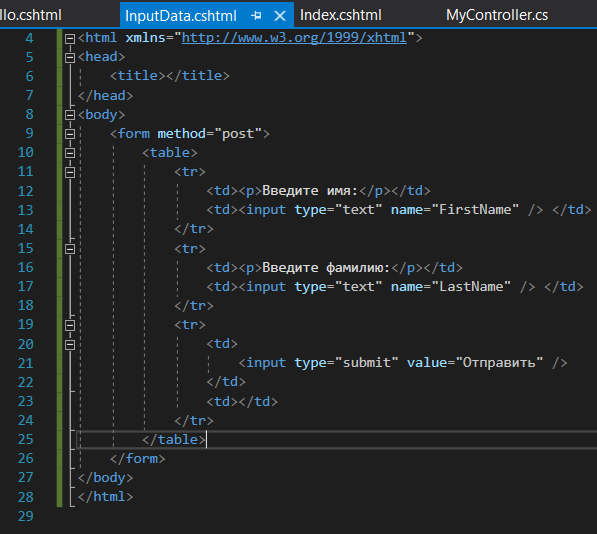


Рисунок 3 — Представление InputData

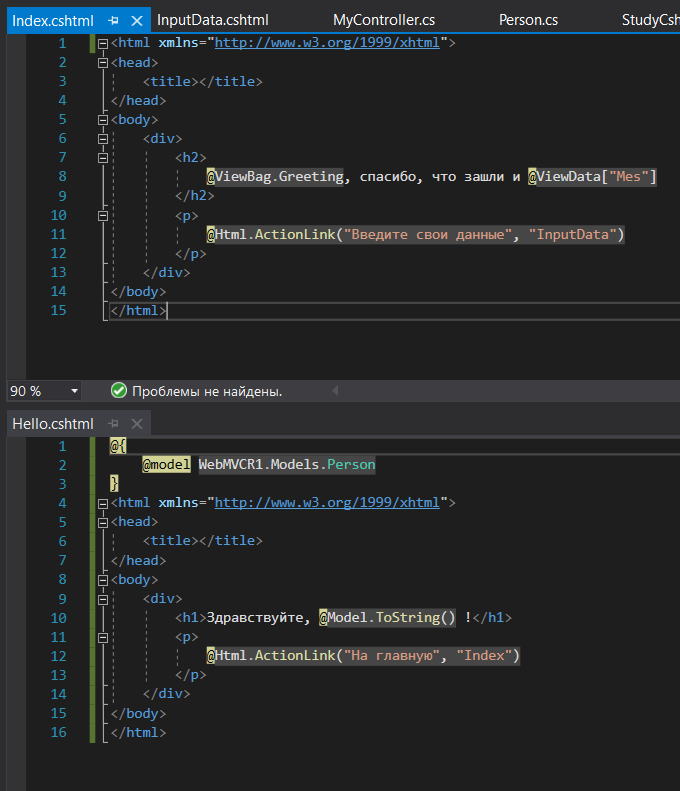


Рисунок 4 — Представления Index и Hello

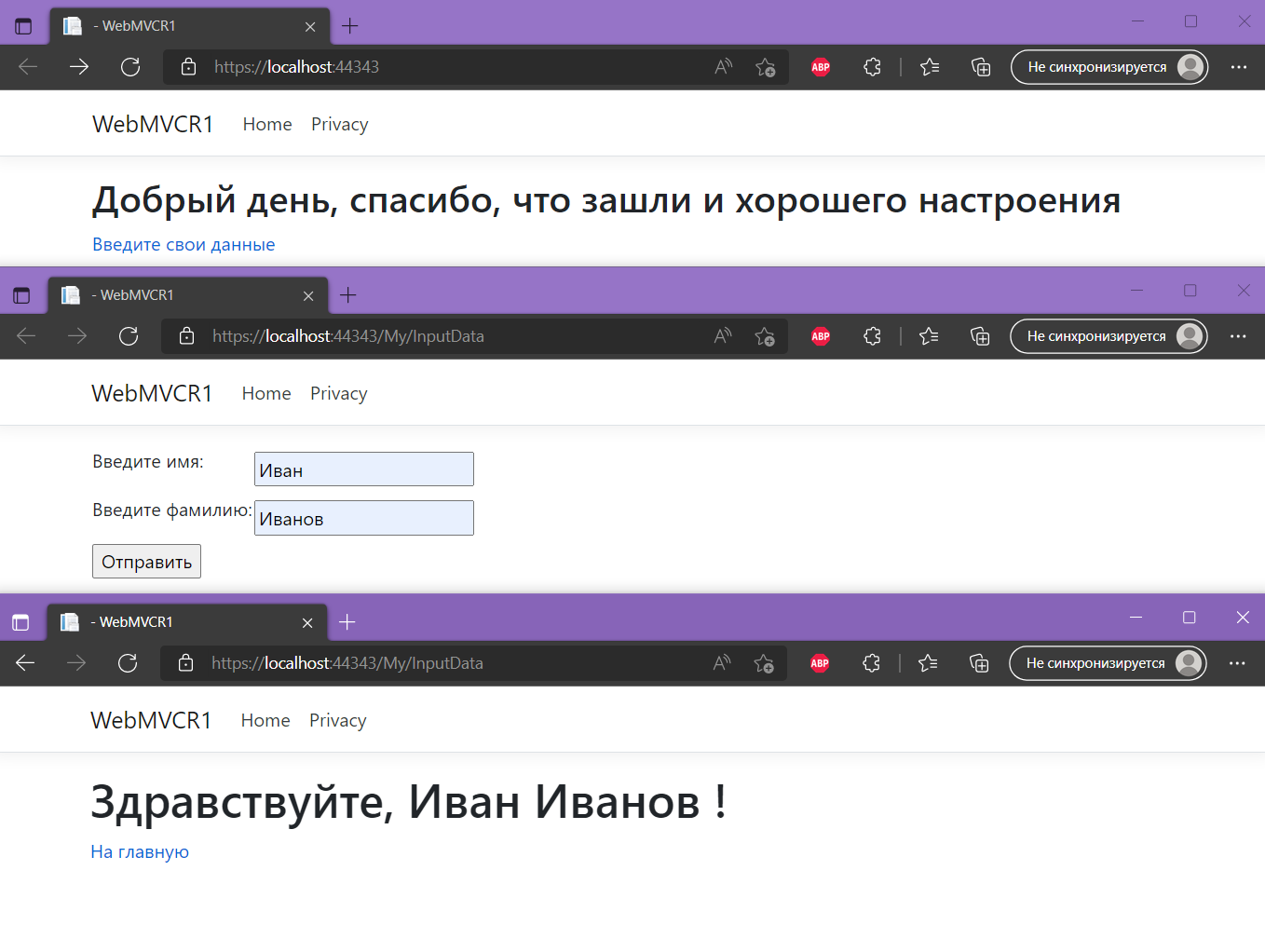


Рисунок 5 — Результат работы приложения

### Упражнение 3. Реализация модели-репозитория

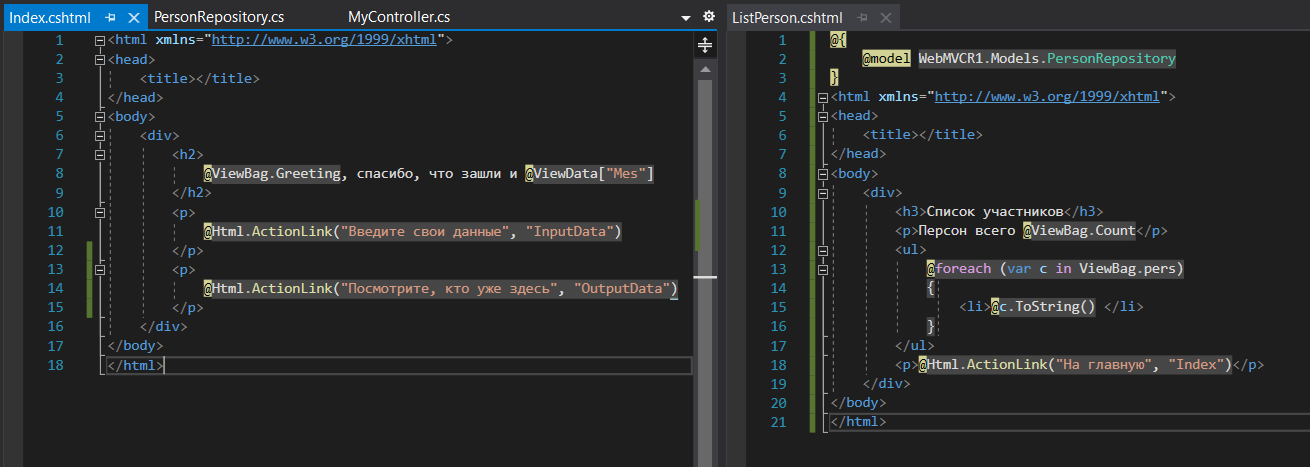


Рисунок 6 — Представления Index и ListPerson

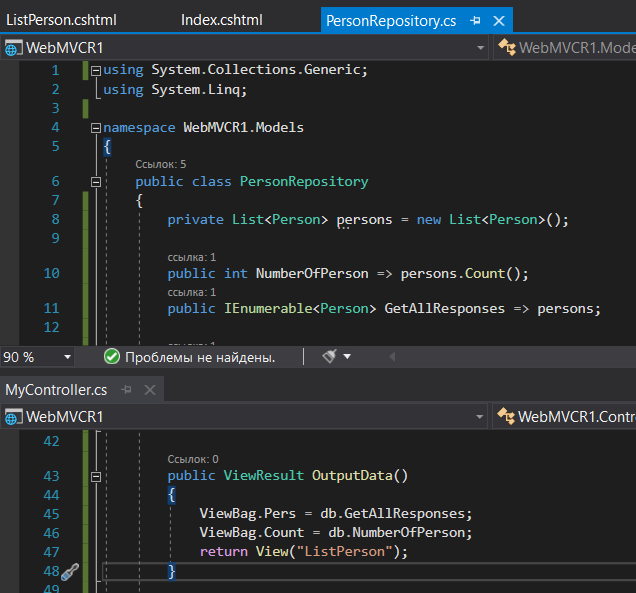


Рисунок 7 — Класс PersonRepository и его использование в методе OutputData контроллера My

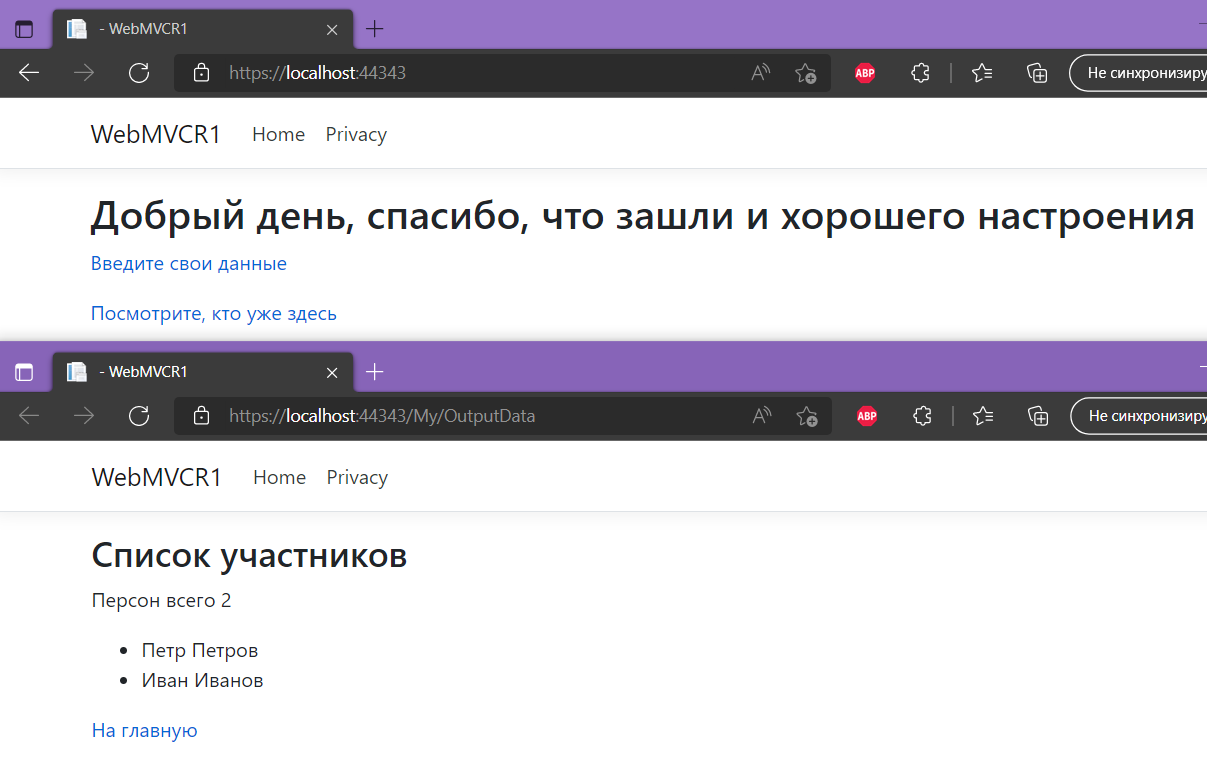


Рисунок 8 — Результат работы приложения

## Выводы

В результате выполнения лабораторной работы приложение было построено и запущено при помощи схемы MVC.

# Лабораторная работа 4. Разработка представлений

## Цель работы

Разработать представление с использованием шаблонных страниц и таблицы стилей.

## Ход работы

### Упражнение 1. Создание формы для ввода данных с помощью HTML Helpers

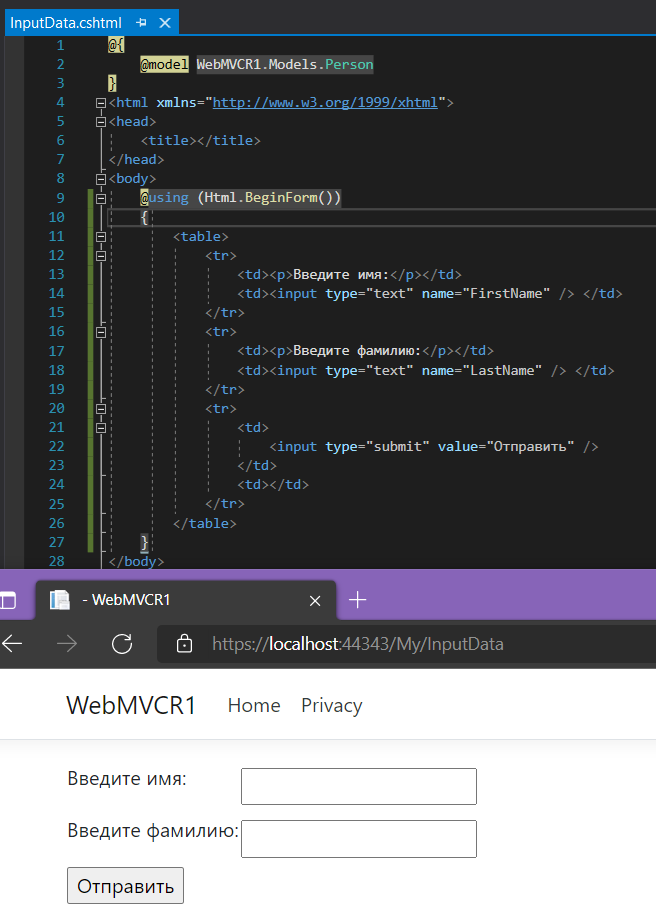


Рисунок 1 — Использование HTML хелпера BeginForm

### Упражнение 2. Применение шаблонных страниц

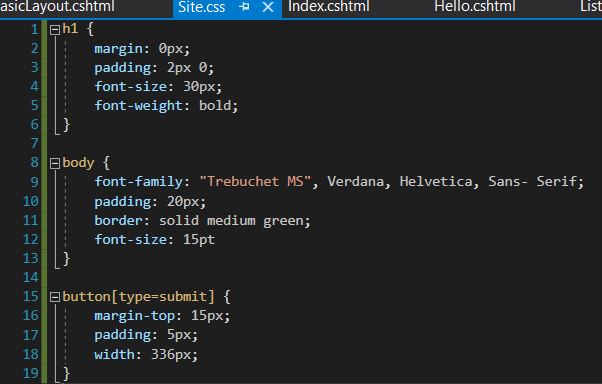


Рисунок 2 — Файл стилей Site

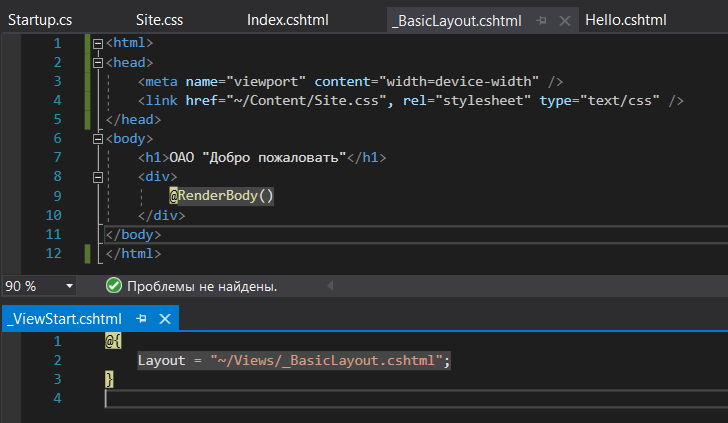


Рисунок 3 — Использование шаблона представления и файла стилей

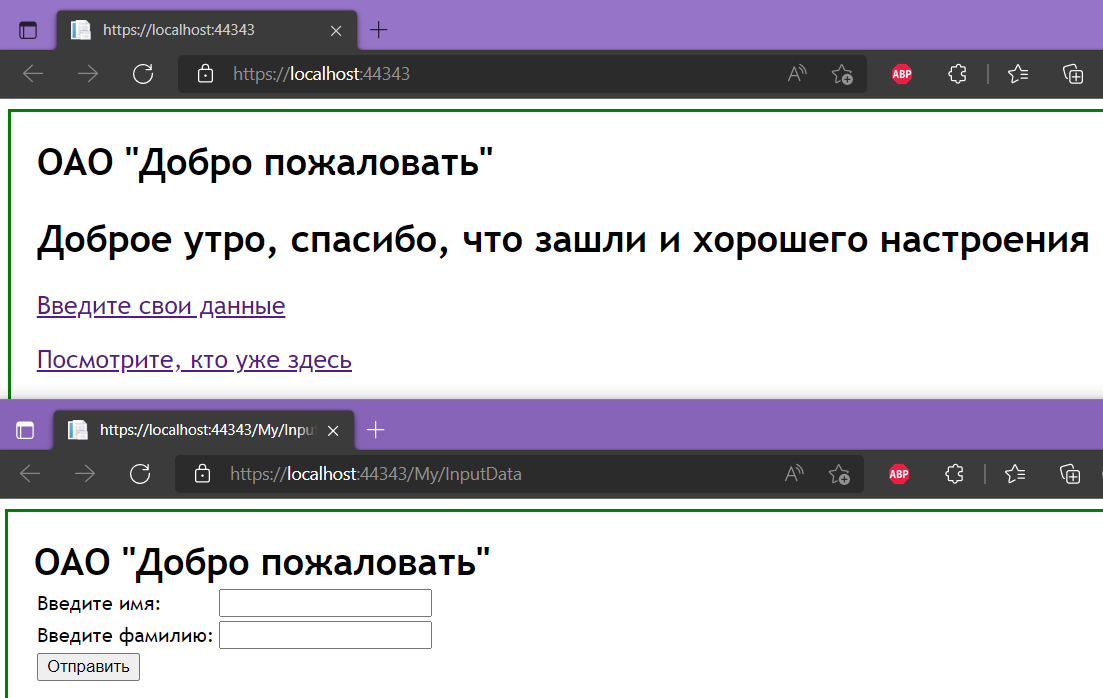


Рисунок 4 — Результат работы программы

### 

### Упражнение 3. Использование шаблонов представлений MVC с формированием шаблонов данных

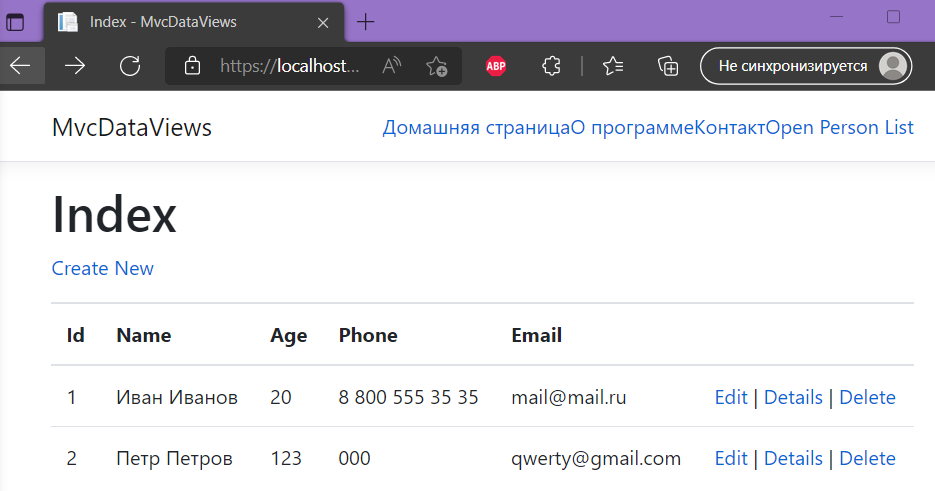


Рисунок 5 — Представление Index, созданное при помощи представлений с формированием шаблонов данных

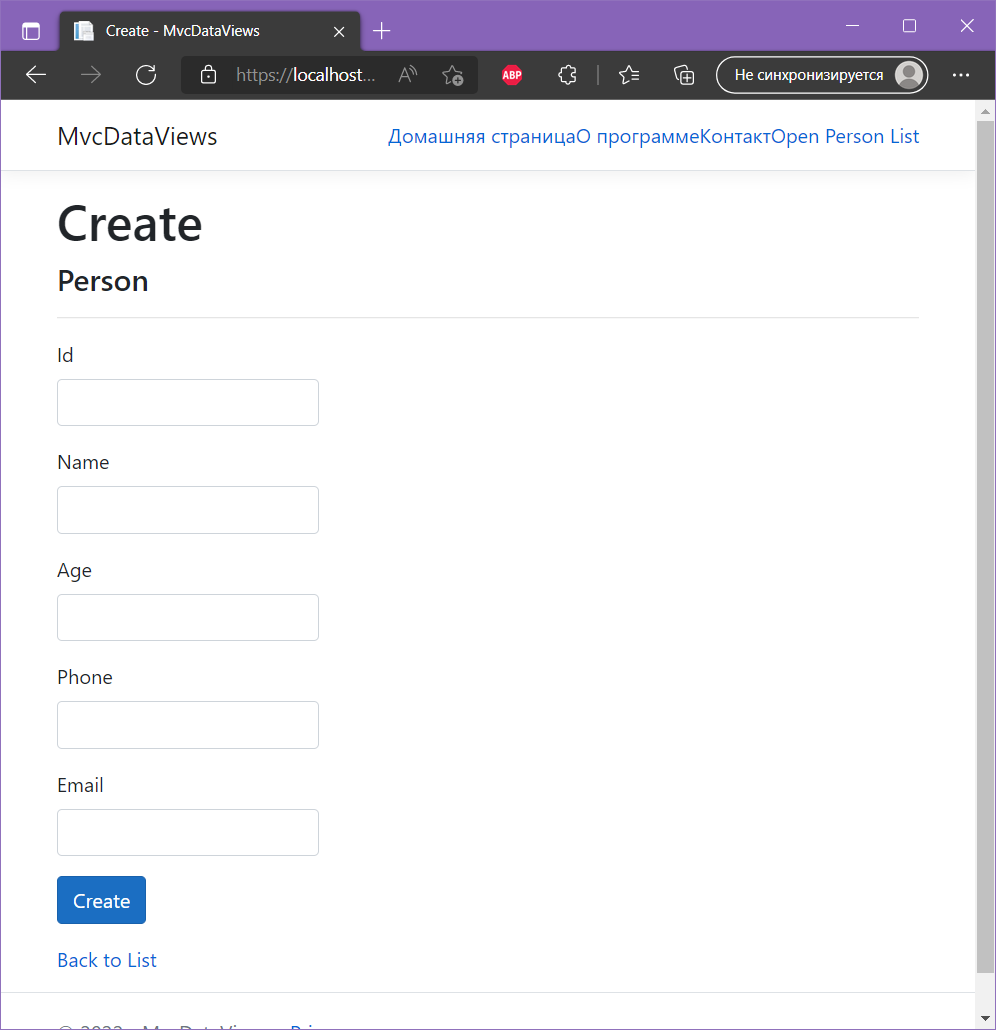


Рисунок 6 — Представление Create, созданное при помощи представлений с формированием шаблонов данных

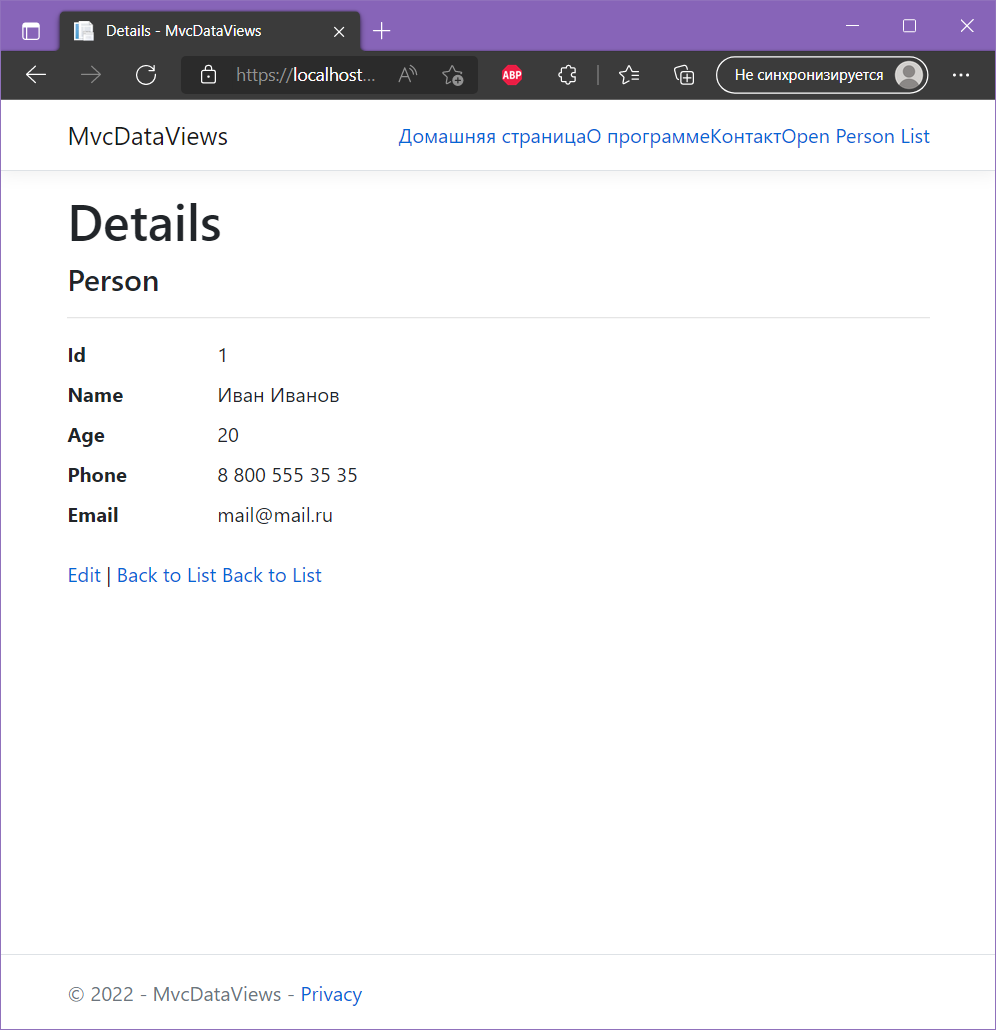


Рисунок 7 — Представление Details, созданное при помощи представлений с формированием шаблонов данных

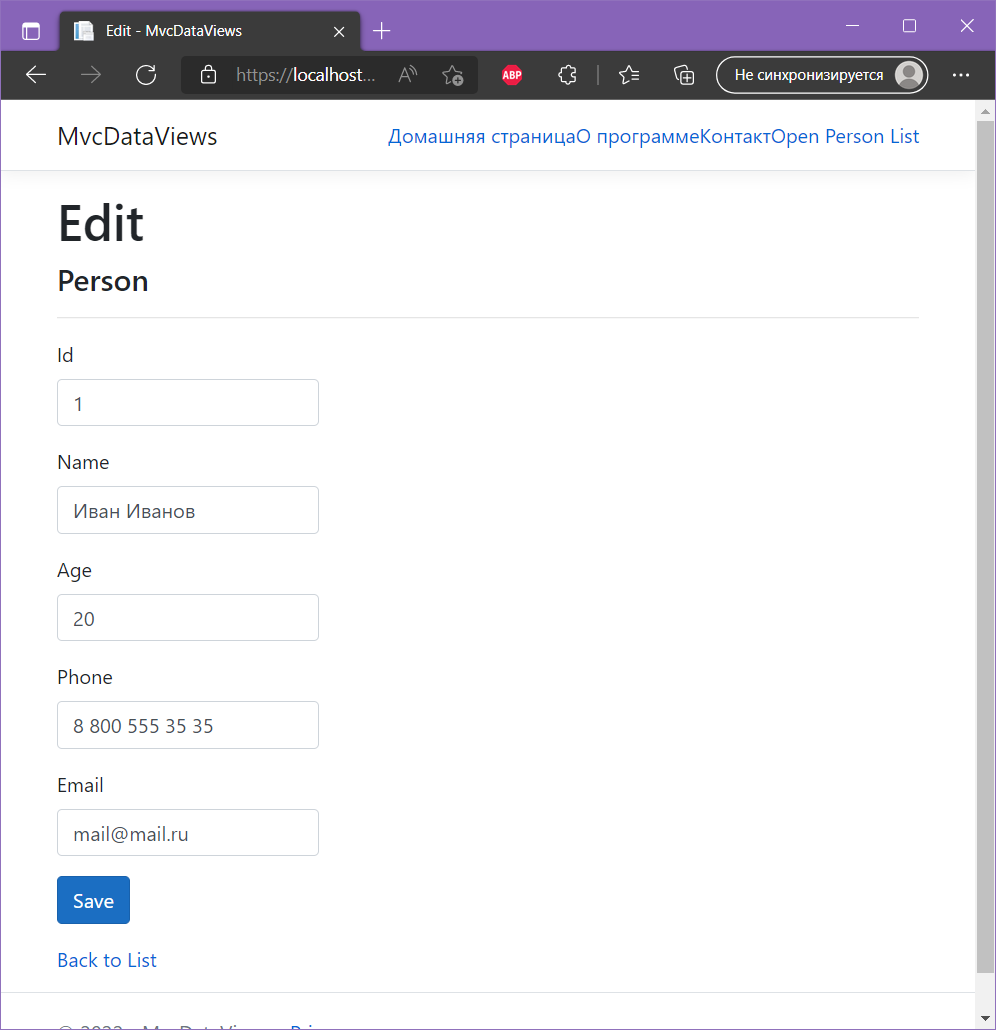


Рисунок 8 — Представление Edit, созданное при помощи представлений с формированием шаблонов данных

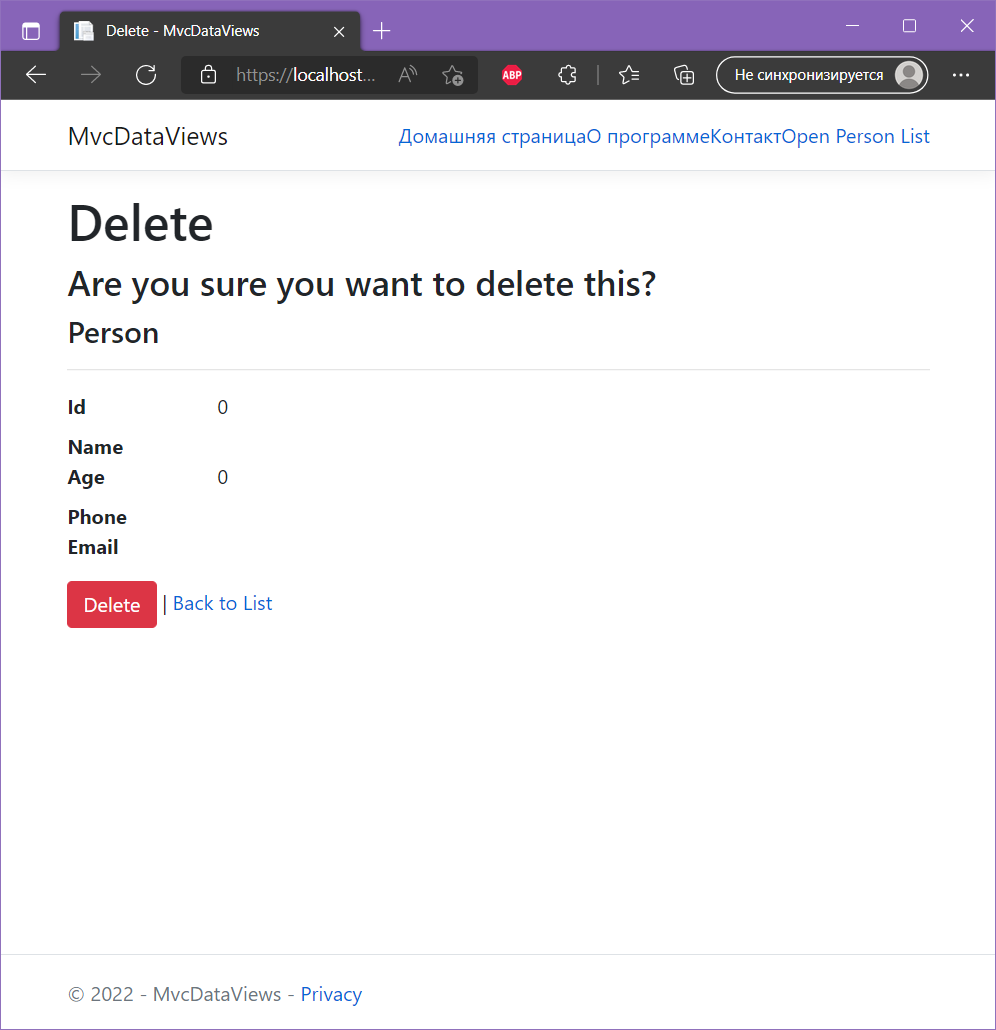


Рисунок 9 — Представление Delete, созданное при помощи представлений с формированием шаблонов данных

## Выводы

В результате выполнения лабораторной работы для представлений проекта был использован макет \_ViewStart с реализацией таблицы стилей. При помощи базовых возможностей Visual Studio был создан новый проект с представлениями, реализованными при помощи шаблонов.

# Лабораторная работа 5. Разработка модели

## Цель работы

Создать приложение с реализацией хранения данных.

## Ход работы

### Упражнение 1. Создание приложения с реализацией хранения данных

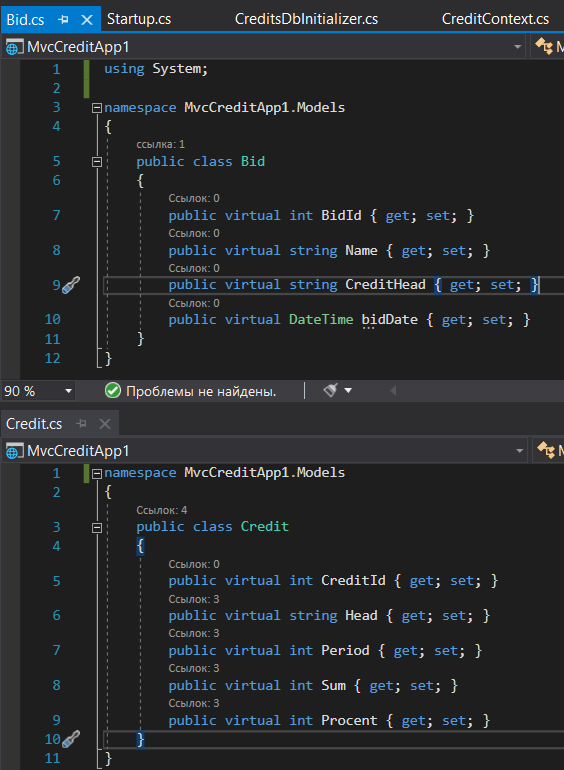


Рисунок 1 — Модели данных

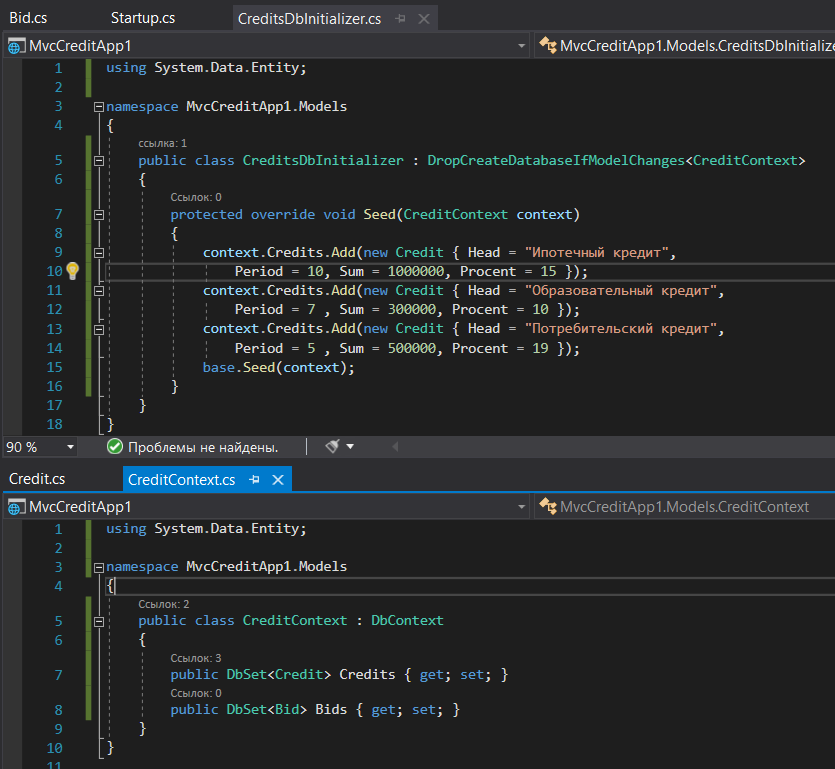


Рисунок 2 — Классы для взаимодействия приложения с базой данных

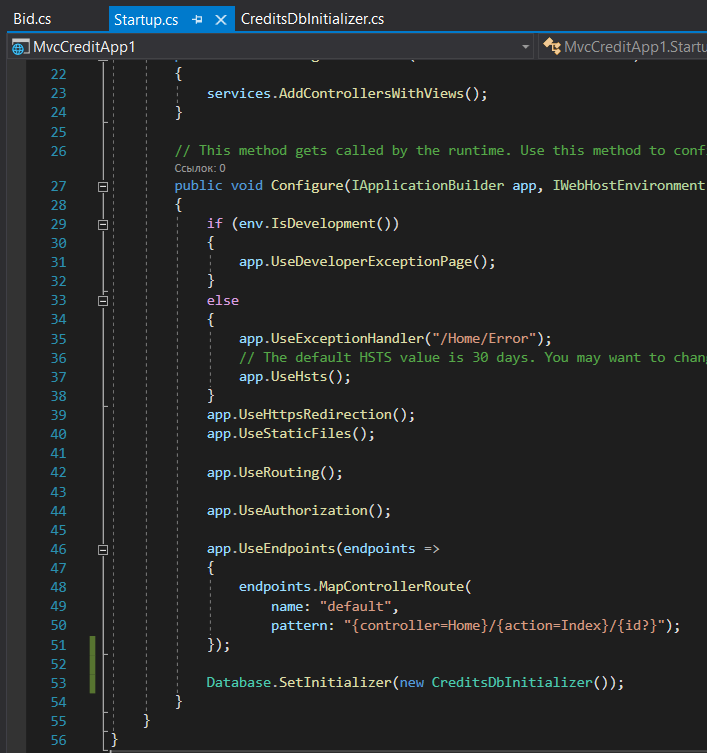


Рисунок 3 — Инициализация базы данных

### Упражнение 2. Настройка работы с данными

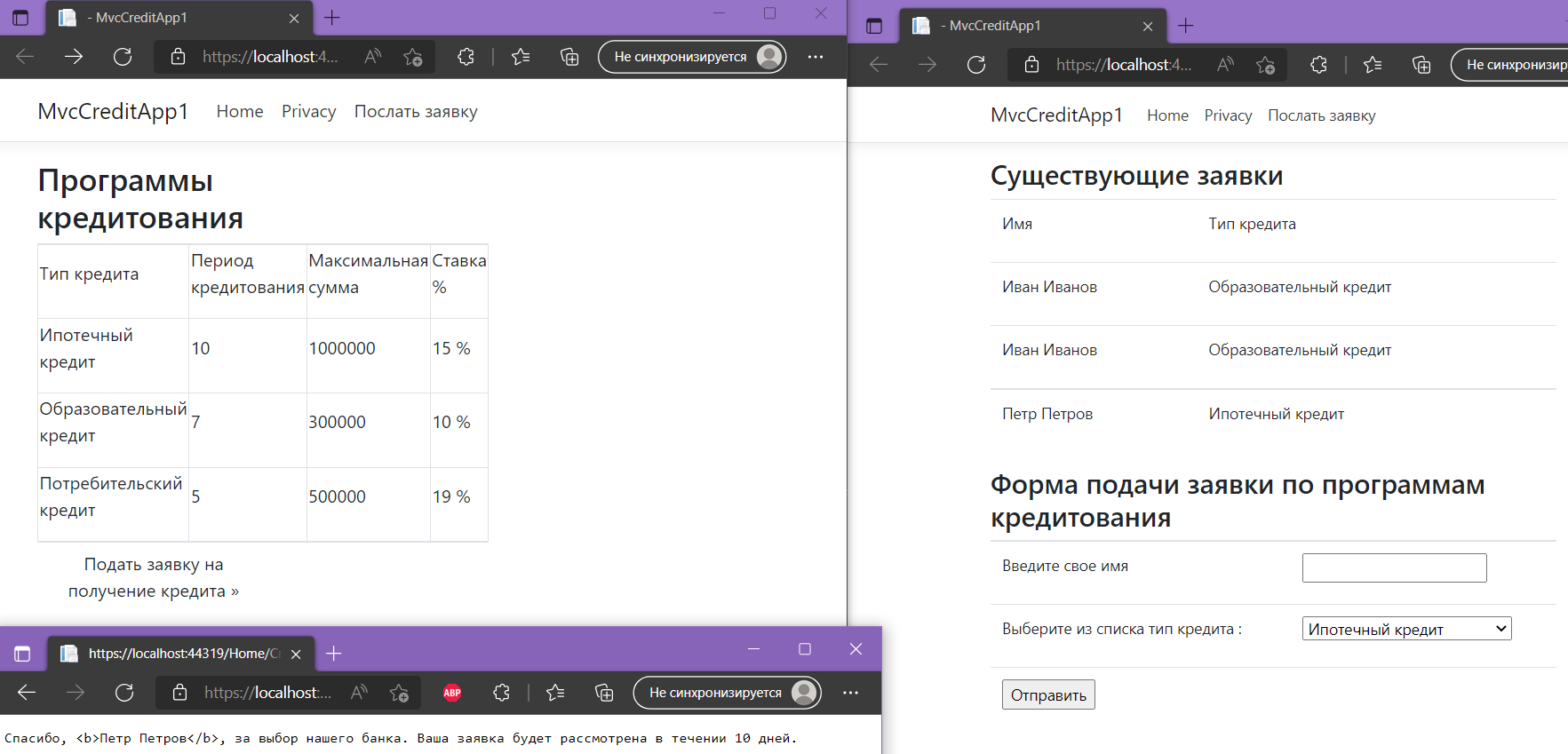


Рисунок 4 — Результат работы приложения с данными

## Выводы

В результате выполнения лабораторной работы было создано приложение, реализующее взаимодействие программы с базой данных.

# Лабораторная работа 6. Применение контроллеров для формирования шаблонов данных

## Цель работы

Реализовать работу с данными с помощью шаблонных контроллеров и представлений, использующих Entity Framework.

## Ход работы

### Упражнение 1. Добавление шаблонного контроллера

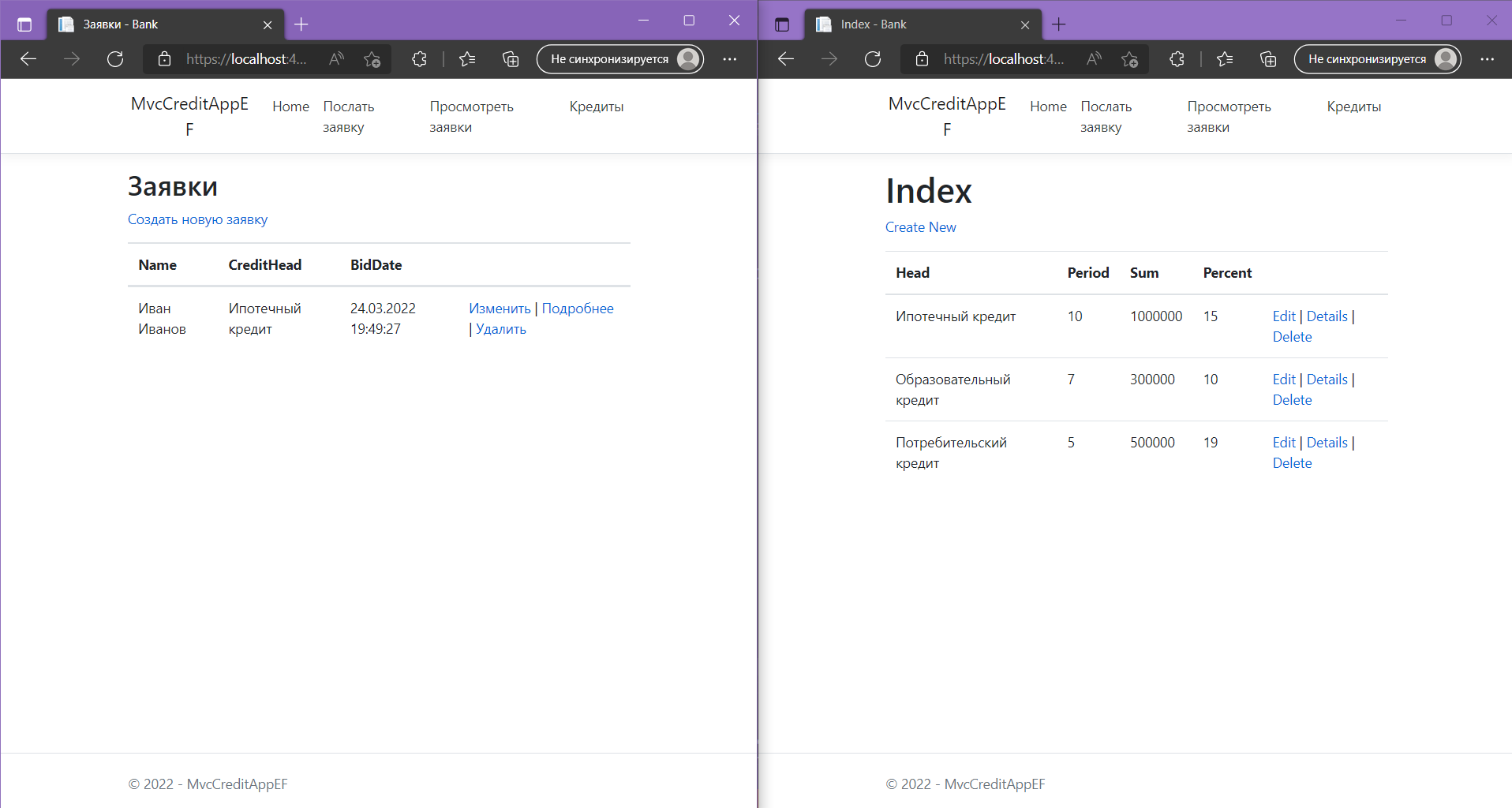


Рисунок 1 — Результат работы приложения с добавленным шаблонным контроллером для Credits

### Упражнение 2. Применение аннотации данных в модели

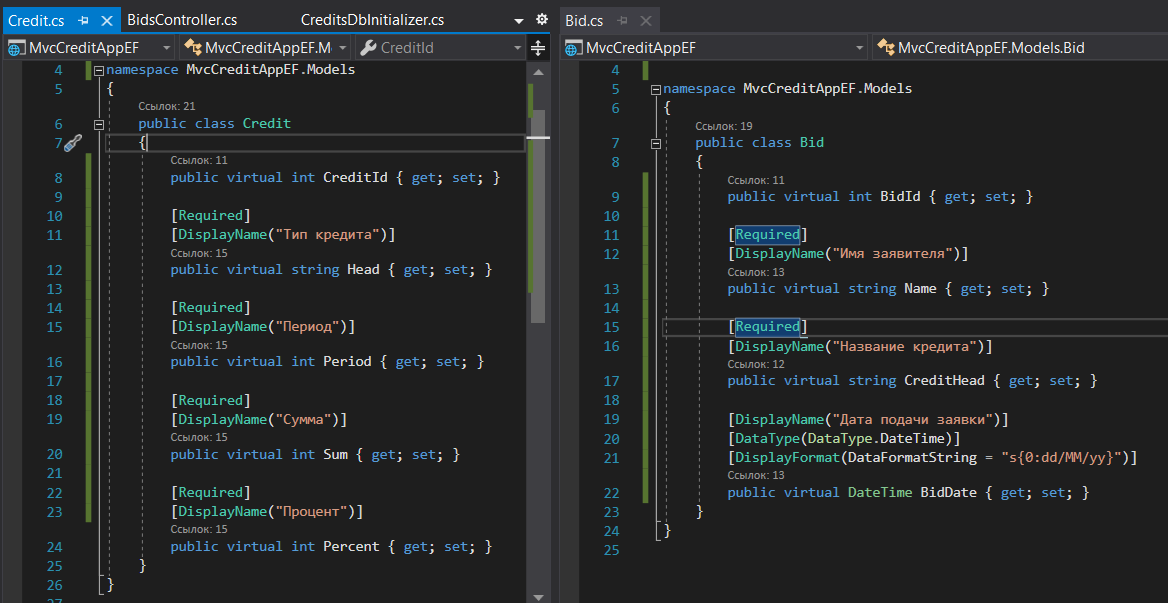


Рисунок 2 — Использование аннотаций

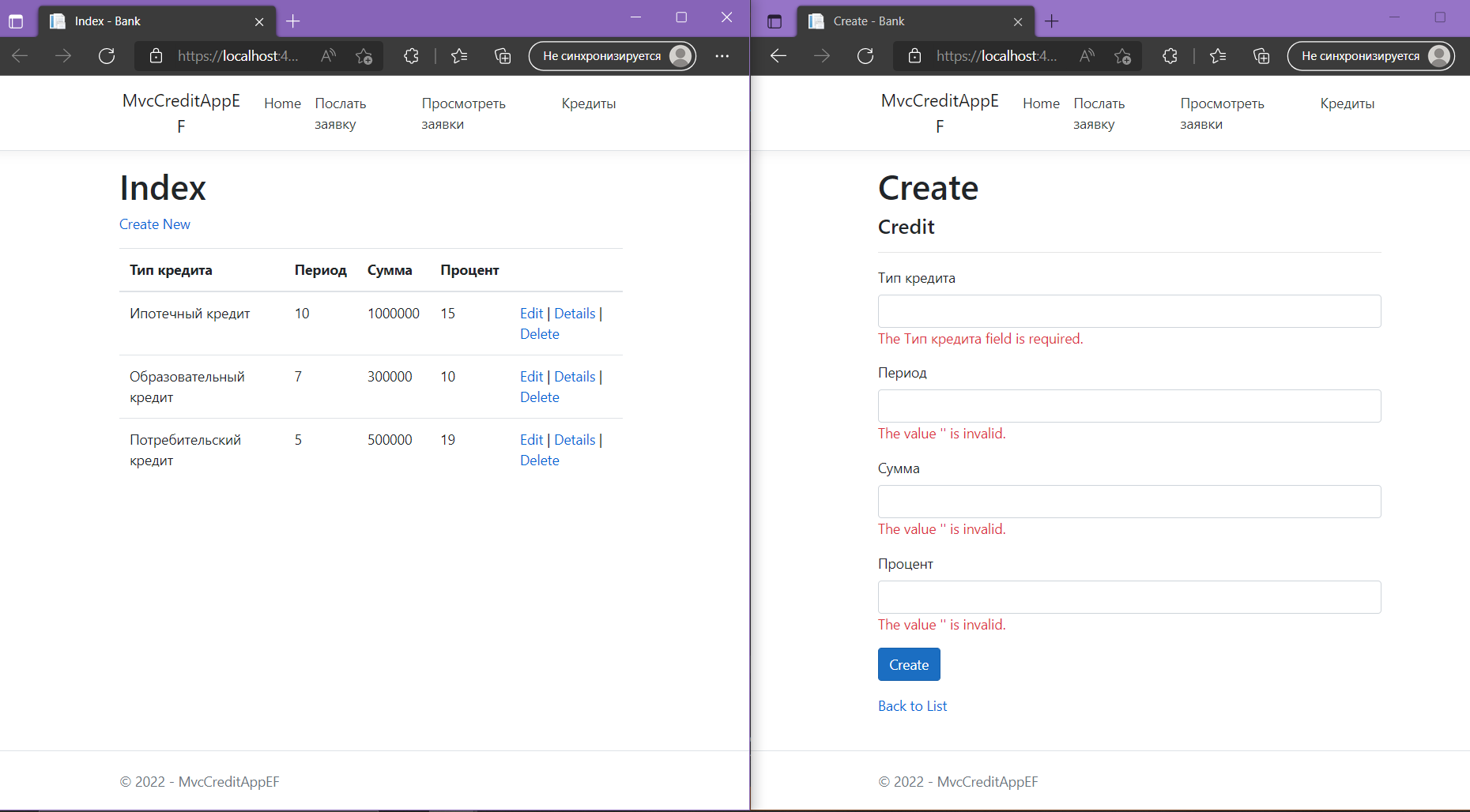


Рисунок 3 — Результат работы аннотаций

### Упражнение 3. Применение кэширования

Убедиться в работе кэширования можно, просмотрев код состояния в заголовках страницы в инструментах разработчика.

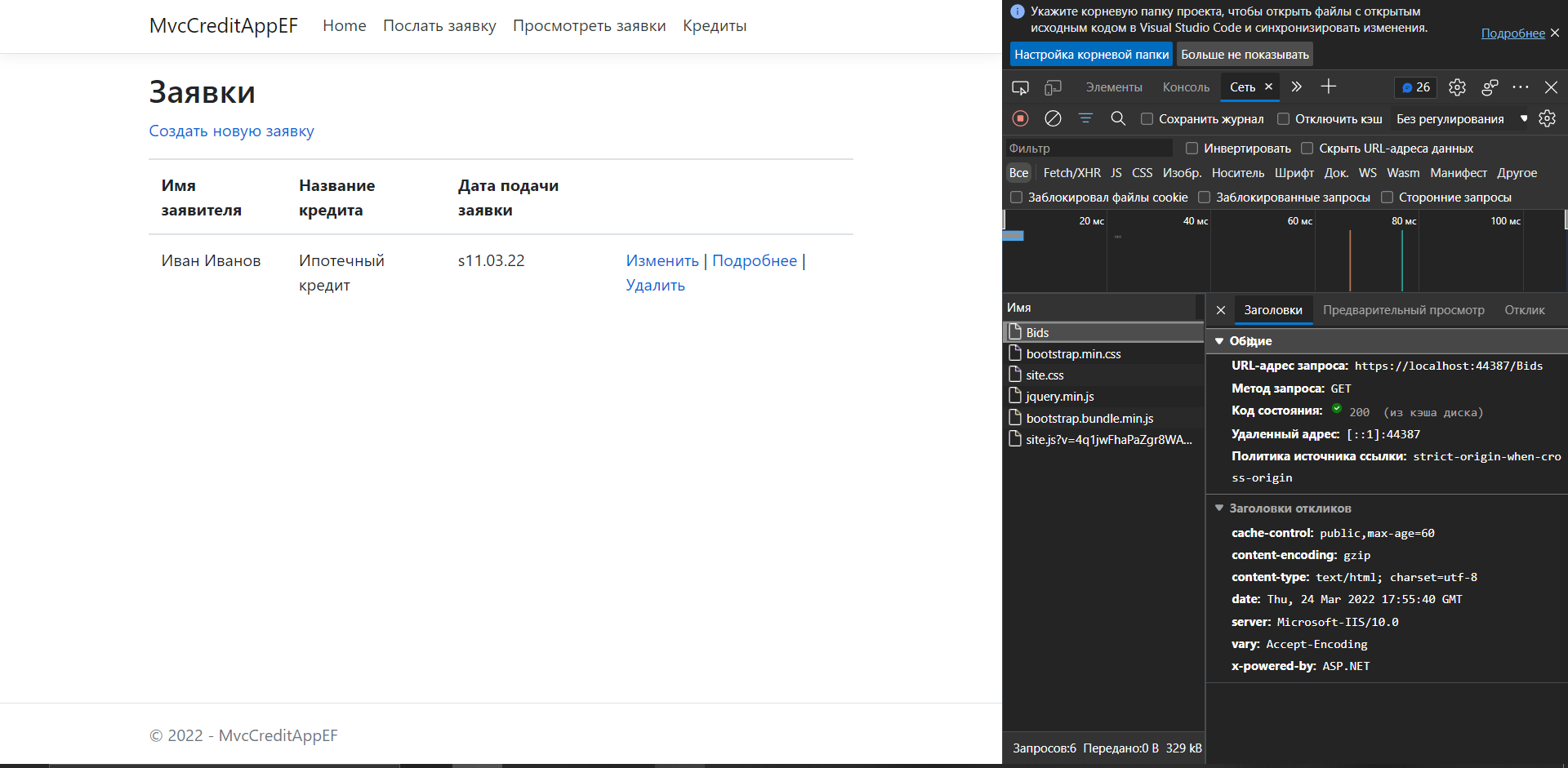


Рисунок 4 — Работа кэширования

## Выводы

В результате выполнения лабораторной работы в приложении была реализована работа с данными с помощью контроллеров и представлений, использующих Entity Framework.

# Лабораторная работа 7. Создание интерактивных страниц в ASP.NET MVC

## Цель работы

Добавить в проект предыдущей работы возможность использования AJAX и частичного обновления страниц.

## Ход работы

### Упражнение 1. Использование AJAX и частичных страниц

AJAX в новых версиях ASP NET Core MVC заменен на ViewComponents, в связи с чем в данном проекте не удалось поддержать работу AJAX по принципу, описанному в задании к лабораторной работе. Однако, похожий функционал был реализован при помощи частичных страниц ASP NET.

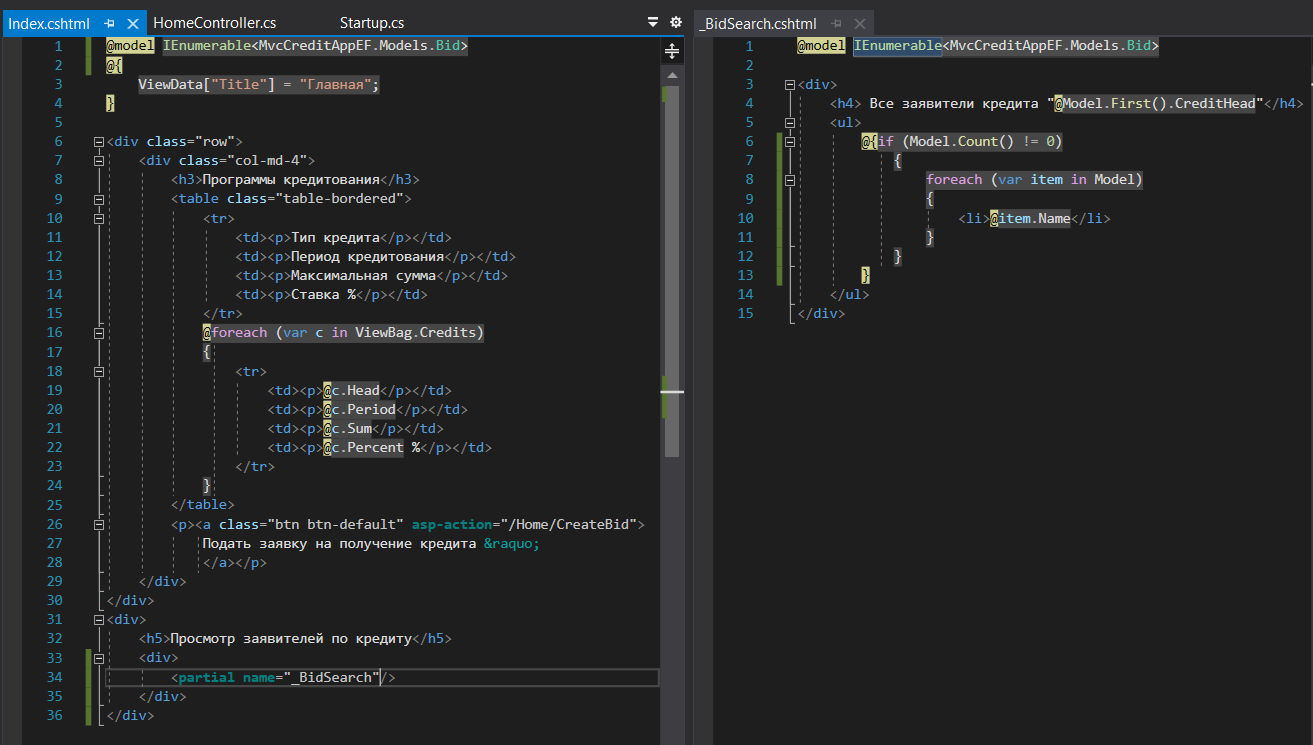


Рисунок 1 — Частичное представление \_BidSearch и его интеграция

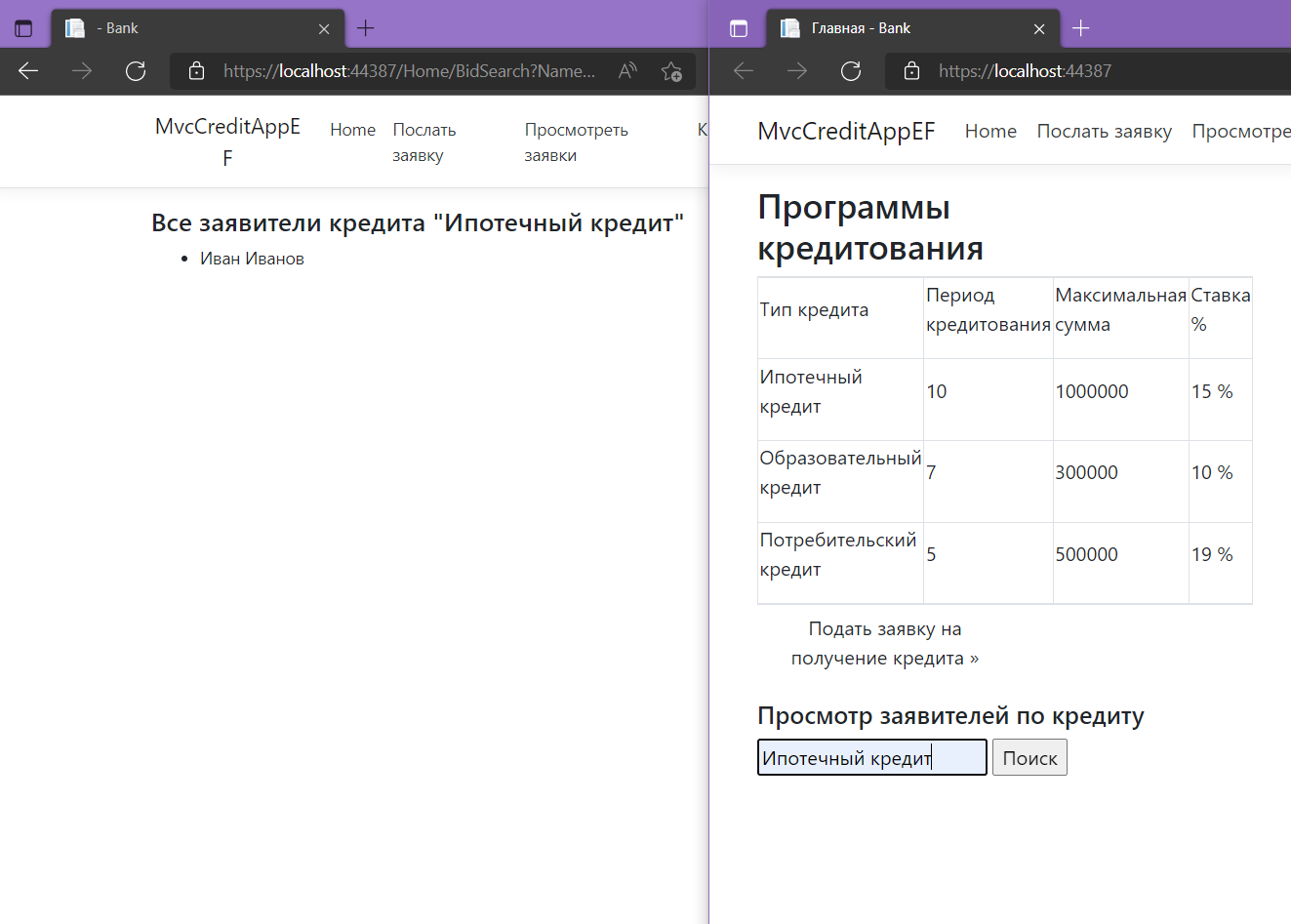


Рисунок 2 — Реализация работы частичного представления

## Выводы

В результате выполнения лабораторной работы было создано частичное представление \_BidSearch и реализовано его взаимодействие с программой.

# Лабораторная работа 8. Реализация авторизации и аутентификации в приложении ASP.NET MVC 5

## Цель работы

Реализовать авторизацию и аутентификацию при помощи ASP.NET Identity.

## Ход работы

### Упражнение 1. Использование типа аутентификации Individual User Accounts

Новая версия ASP.NET Core MVC не поддерживает OWIN. Однако, функционал авторизации и аутентификации уже встроен в ASP.NET Core качестве библиотеки AspNetCoreFramework.Identity.

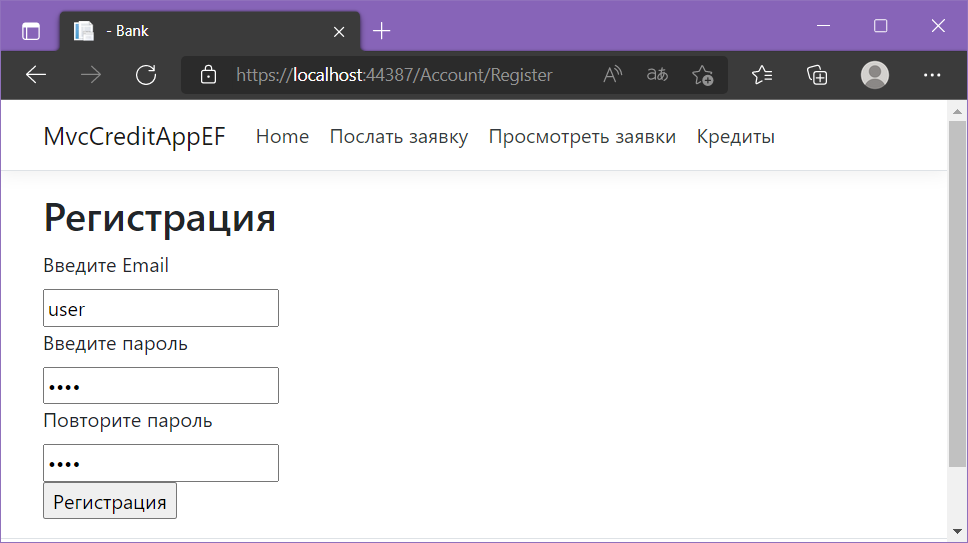


Рисунок 1 — Страница регистрации

### Упражнение 2. Ограничение входа для незарегистрированных пользователей

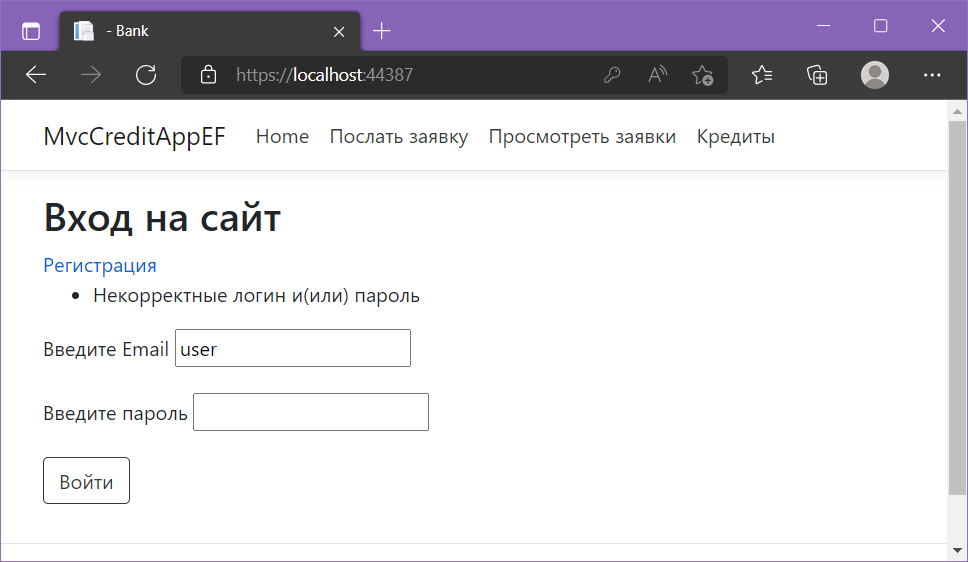


Рисунок 2 — Реализация ограничения входа

### 

### Упражнение 3. Создание ролей и их использование для разграничения доступа

### 

Рисунок 3 — Работа с сайтом под учетной записью admin

При попытке зайти на страницы, которые не поддерживают учетную запись user, пользователя возвращает на страницу входа.

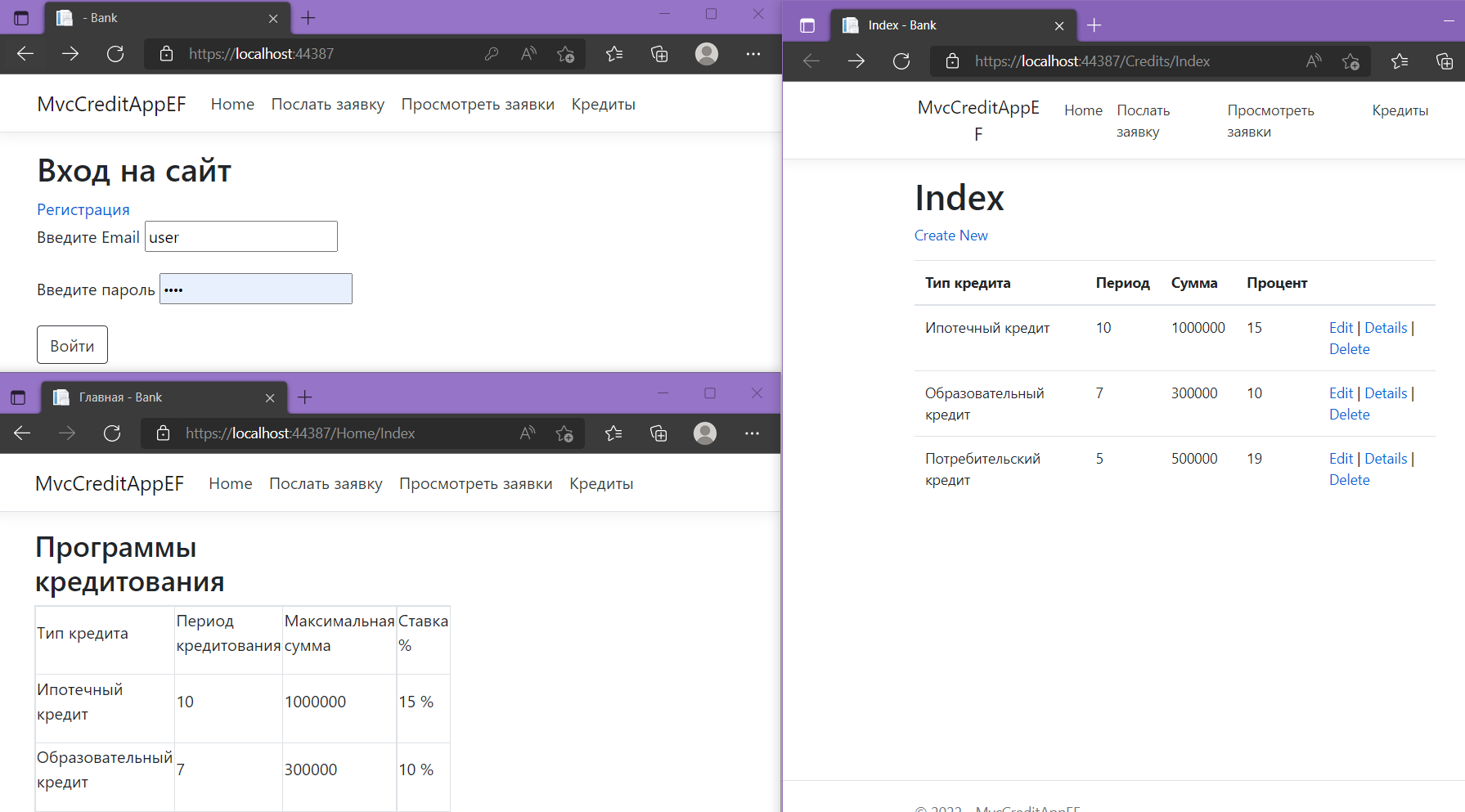


Рисунок 4 — Работа с сайтом под учетной записью user

## Выводы

В результате выполнения лабораторной работы была реализована аутентификация и авторизация пользователей, а также система ролей, разграничивающая доступ к функционалу программы для пользователей.