**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 5**

по дисциплине «Разработка приложений баз данных для информационных систем»

на тему: «Разработка интерфейса приложения баз данных с использованием с использованием аутентификации и авторизации»

Вариант 29

Выполнил: студент гр. ИТП-31

Зайцев А.В.

Принял: доцент

Асенчик О. Д.

Гомель 2020

**Цель работы:** Научиться использовать ASP.NET MVC Сore для создания типовых web-приложений для работы с информацией из реляционных баз данных.

**Задание:** Используя разработанный ранее слой доступа к базе данным согласно своего варианта, спроектировать и создать интерфейс Web-приложения на основе ASP.NET Core MVC Framework и Entity Framework Core.

Web- приложение должно удовлетворять следующим требованиям:

1. Осуществлять ввод, редактирование, добавление и просмотр данных не менее чем из трех таблиц реляционной базы согласно варианту. Не менее, чем одна из таблиц должна находиться на стороне отношения «многие» в схеме базы данных.
2. Иметь единое стилевое оформление, основанное на использовании мастер-страниц.
3. Иметь удобную систему навигации (строка меню, гиперссылки, кнопки), которая обеспечивает оптимальный путь перехода между двумя произвольно выбранными страницами в соответствии с логикой приложения.
4. Пользователь для работы с приложением должен пройти аутентификацию.
5. Должно поддерживать реализацию не менее двух ролевых политик.
6. Администратор должен иметь возможность управлять пользователями: просмативать, создавать, удалять и редактировать данные учетных записей.
7. Представления для просмотра данных из таблиц должны предусматривать разбиение данных на страницы, фильтрацию по одному или нескольким полям.

***На оценку 9 или 10:***

1. *Осуществить кэширования данных для отображения с помощью встроенного инструмента кэширования - объекта ImemoryCache. Выводить кэшированные данные таблиц MemoryCache на соответствующие страницы на сайта, генрируемые с использованием представлений (Views). Данные в кэше хранить неизменными до проведения операций вставки, изменения или удаления данных. После проведения этих операций кэш должен формироваться заново.*
2. *Реализовать сохранение состояния (значений) элементов представлений, предназначенных для осуществления фильтрации, с использованием куки и (или) с объекта Session. Осуществить заполнение элементов представлений, предназначенных для осуществления фильтрации, при их загрузке данными, ранее сохранненными в объекте куки и (или) Session.*

**Ход выполнения:**

Сначала были созданы модели базы данных. Они были помещены в папку *Models*.

Далее был создан класс контекста базы данных. Он был помещён в папку *Data*.

После этого был написан класс, роль которого заключается в заполнении базы данных начальными записями. Он заполняет таблицу базы данных только в том случае, если в ней нет записей.

Далее в проект был добавлен *Identity.* Он необходим для авторизации и аутентификации пользователей. Был создан класс для первичной инициализации пользователей. Он создаёт первичного админа.

Далее были созданы классы контроллеров. Они все были помещены в папку *Controllers*. После были написаны классы моделей представлений для облегчения манипулирования данными приложения и выводом их пользователю.

Был написан сервис для кеширования. Кеширую только записи таблиц на стороне отношения многие.

Доступ к данным на стороне отношения один получает только админ. В будущем будет добавлена ещё одна роль – менеджер. Он сможет изменять записи на стороне отношения многие и просматривать записи на стороне отношения один.

Для всех контроллеров были созданы папки в папке *Views*. В каждом контроллере есть методы *Index, Create, Edit, Delete, Details*. И поэтому в каждой папке с представлениями были созданы соответствующие представления для этих методов.

Структура приложения расположена на рисунке 1.

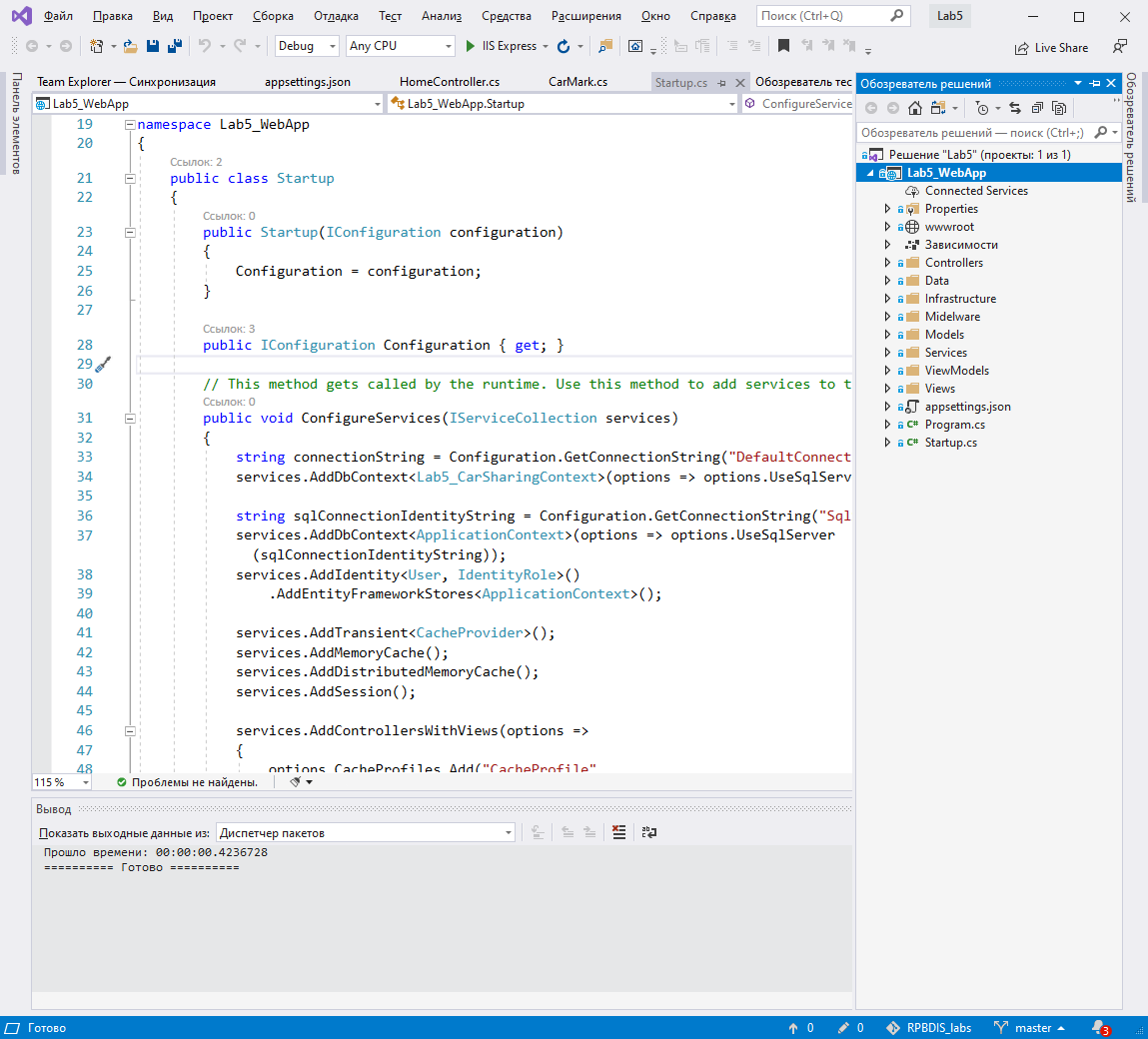


Рисунок 1 – Структура созданного приложения

Внешний вид созданного веб-приложения на рисунке 2.

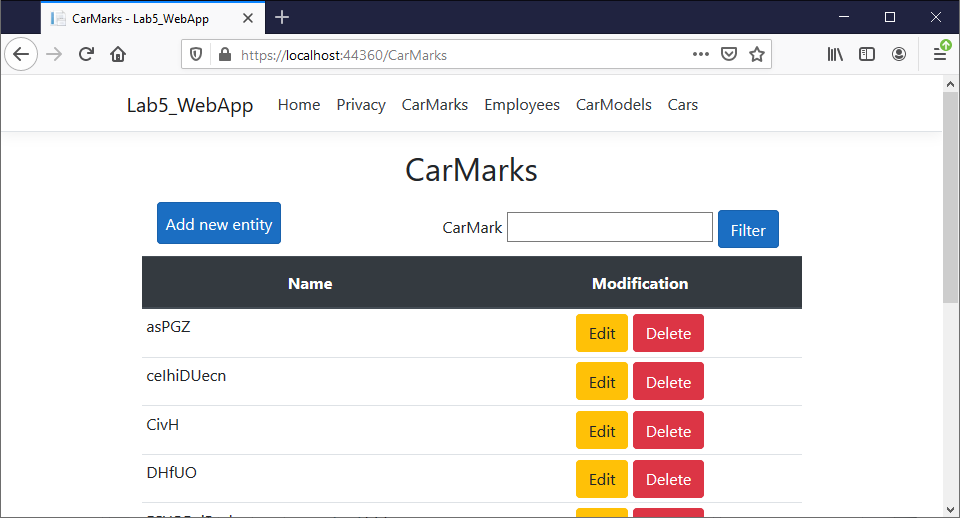


Рисунок 2 – Внешний вид созданного веб-приложения

На GitHub был создан репозиторий для проекта, где хранится весь код приложения. Ссылка: https://github.com/Zaitzev-Alexandar/RPBDIS\_labs/tree/master/Lab5

**Вывод:** В ходе выполнения был создан проект ASP.NET CORE MVC. Были созданы модели, контроллеры, модели представлений и представления. Была применена технология *Identity* для авторизации и аутентификации пользователей. Был написан сервис для кеширования.