|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** \_ ***ИУК «Информатика и управление»\_\_\_\_\_\_\_\_\_***

**КАФЕДРА** \_\_ ***ИУК5 «Системы обработки информации»***

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе на тему:**

***Веб-приложение для покупки товаров в интернет-магазине бытовой техники.***

по дисциплине ***Базы данных***

Студент гр. ИУК5-52Б \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Хромов А.Е. )

(подпись) (Ф.И.О.)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

Оценка руководителя \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

30-50 (дата)

Оценка защиты \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

30-50 (дата)

Оценка проекта \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка по пятибалльной шкале)

Комиссия: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

Калуга, 2021

Калужский филиал   
федерального государственного бюджетного образовательного учреждения высшего образования

***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»   
(КФ МГТУ им. Н.Э. Баумана)***

УТВЕРЖДАЮ

Заведующий кафедрой **\_\_ИУК5\_\_\_**

\_\_\_\_\_\_\_\_\_\_\_(Е.В. Вершинин)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине ***Базы данных***

Студент Хромов А.Е. ИУК5-52Б \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, инициалы, индекс группы)

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, инициалы)

График выполнения проекта: 25% к\_4\_нед., 50% к\_7\_нед., 75% к\_10\_нед., 100% к\_14\_нед.

***1. Тема курсового проекта***

***Веб-приложение для покупки товаров в интернет-магазине бытовой техники.***

***2. Техническое задание***

*Разработать веб-приложение для покупки товаров в интернет-магазине бытовой техники с использованием базы данных.* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***3. Оформление курсового проекта***

3.1. Расчетно-пояснительная записка на\_\_\_\_\_\_\_ листах формата А4.

3.2. Перечень графического материала КП (плакаты, схемы, чертежи и т.п.)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021г.

Руководитель курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

(подпись) (Ф.И.О.)

Задание получил\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_2021г.

(подпись) (Ф.И.О.)

Примечание:

Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

**ВВЕДЕНИЕ**

1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ
   1. Общие сведения
      1. Полное наименование системы

Веб-приложение для покупки товаров в интернет-магазине бытовой техники.

* + 1. Плановые сроки начала и окончания работы по созданию системы

Начало работы: 01.09.2021

Окончание работы:

* 1. Назначение и цели создания (развития) системы
     1. Назначение системы

Назначением системы является автоматизация покупок в магазине бытовой техники с помощью веб-приложения с использованием базы данных.

* + 1. Цели создания системы

Цель создания системы – повысить оборот товара, и в следствии получить большую прибыль за счет удобства просмотра и заказа имеющейся в магазине техники.

* 1. Характеристики объекта автоматизации

Объектом автоматизации является магазин бытовой техники. Автоматизированию подлежит демонстрация имеющейся бытовой техники в магазине и ее продажа.

* 1. Требования к системе
     1. Требования к системе в целом

Программный продукт должен представлять собой веб-приложение.

* + 1. Требования к структуре и функционированию системы

Приложение должно быть разбито на 3 слоя:

- Слой клиентского приложения в виде веб-страницы.

- Слой бизнес-логики приложения.

- Слой базы данных.

Функционал приложения должен содержать:

Авторизацию пользователей, отображение списка товаров, актуализацию товаров, фильтрацию товаров, просмотр и актуализацию списка категорий и списка производителей, добавление товаров в корзину, хранение списка пользователей и их информации.

* + 1. Требования к надежности

Программа должна обеспечивать корректную обработку исключительных ситуаций, вызванных ,например ,вводом неверных пользовательских данных или неоднозначной информации в базу данных.

* 1. Состав и содержание работ по созданию системы

|  |  |  |
| --- | --- | --- |
| Стадии | Этапы работ | Сроки исполнения |
| 1. Формирование требований к АС | 1.1. Исследование объекта и подтверждение необходимости создания АС.  1.2. Формирование требований пользователя к АС.  1.3. Оформление отчёта о  выполненной работе и заявки на разработку АС. |  |
| 2. Разработка концепции АС. | 2.1. Изучение выбранного объекта.  2.2. Проведение научно- исследовательских работ.  2.3. Проектирования концепции АС, удовлетворяющей потребности пользователя.  2.4. Оформление отчёта о выполненной работе. |  |
| 3. Техническое задание. | Разработка и утверждение технического задания на создание АС. |  |
| 4. Разработка системы. | Разработка системы согласно  техническому заданию. |  |
| 5. Защита курсовой работы. | 5.1. Создание грамотной презентации и речи для защиты курсовой работы.   * 1. . Защита курсовой работы. |  |

* 1. Порядок контроля и приемки системы.
     1. Состав, объем и методы испытаний системы и ее составных частей.

Приложение должно пройти предварительные испытания, состоящие из отладки и минимального набора тестов.

В результате предварительных испытаний, должны быть исправлены недочёты, замечания на которые были получены в ходе предварительных испытаний.

Для проверки корректной работы внесённых изменений должны быть проведены повторные испытания разработанной программы.

* + 1. Общие требования к приемке работ.

В процессе приемки работ должна быть осуществлена проверка на соответствие требованиям настоящего «Технического задания». По результатам испытаний возможны доработки и исправления.

* 1. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие.

При подготовке объекта автоматизации к вводу системы в действие следует выполнить:

1. Обеспечение работ по вводу данных в систему
2. Развертывание системы на сервере.
3. Настройка системы доступа и создание учетных записей.
   1. Требования к документированию

По окончанию работы предъявлена расчетно-пояснительная записка в состав которой входят:

* техническое задание;
* научно-исследовательская часть;
* проектно-конструкторская часть;
* проектно-технологическая часть.

Также должна быть предоставлена графическая часть работы, выполненная формате А1 на 2 листах, в которую входят:

* демонстрационные чертежи;
* алгоритмические схемы.
  1. Источники разработки
* Гост 34.601-90
* Гост 34.602-89

1. ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ
   1. Постановка задачи проектирования.

Задачей данного проекта является разработка и проектирование интернет-магазина бытовой техники, отвечающего следующим требованиям:

1. Возможность просмотра товара и категорий
2. Возможность добавления товара в корзину и оформления заказа
3. Возможность фильтрации товара по характеристикам, соответствующим категории товара.
4. Актуализация данных категории, товара и производителей.
   1. Описание предметной области

В данном приложении существуют категории товаров и сами товары, распределенные по категориям. Категории имеют определенные характеристики с разными значениями, а товары имеют характеристики, присущие их категориям. Товары так же имеют разных производителей. Товары могут быть добавлены в заказ, в корзину или в избранное. Все исходные данные (категории, производители и товары) могут быть изменены администратором.

Описание и функционал со стороны пользователя:

* Пользователь может просматривать и фильтровать товар не авторизовываясь в веб-приложении.
* Фильтрация товара происходит по характеристикам конкретной категории (например, у телевизоров: диагональ, наличие smart-tv, тип дисплея и др.) а так-же по цене и производителю.
* При попытки добавить товар в корзину пользователю требуется авторизоваться.
* При оформлении заказа пользователю должна быть предоставлена возможность внести личные данные (такие как адрес, индекс, номер телефона).
* У пользователя может быть несколько заказов с разными состояниями (оплачен, ожидает оплаты, доставлен, и др.)
* Рассматривается возможность рассылки на электронную почту пользователя рекламных предложений, или предложений, основанных на товарах в корзине пользователя.

Описание и функционал со стороны администратора системы:

* Актуализация данных веб-приложения. Добавление, изменение и удаление товаров, категорий и производителей.
* Смена статусов заказов пользователя.

Для данного проекта была сформирована концептуальная схема (Рисунок 2.1.0).

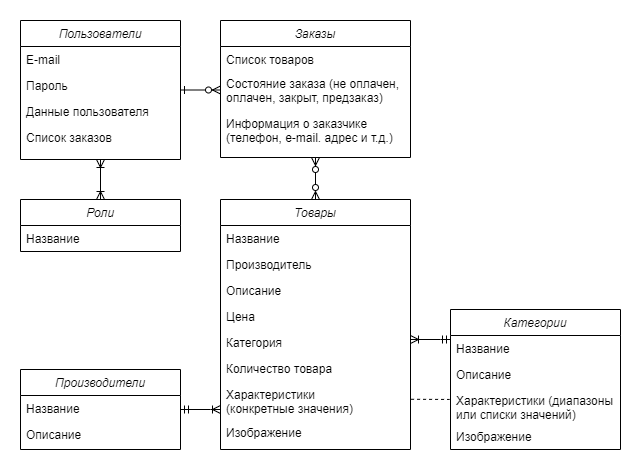


Рисунок 2.1 – Концептуальная схема

* 1. Анализ аналогов

В качестве аналога функционирования отображения товара и его фильтрации был рассмотрен интернет-магазин DNS.

DNS (ООО «ДНС Ритейл», англ. CSN Retail LLC) — российская компания, владелец розничной сети, специализирующейся на продаже компьютерной, цифровой и бытовой техники, а также производитель компьютеров, в том числе ноутбуков, планшетов и смартфонов (сборочное производство). По итогам 2019 года стала 6-й крупнейшей ритейл-компанией в России, в 2021 году — 22-й крупнейшей частной компанией России. В 2021 году сеть насчитывала более 2 тысяч магазинов.[[1]](#литра1)

Сайт DNS имеет очень удобную навигацию, яркий дизайн и информативное оформление товаров (Рисунок 2.2). В данном проекте было решено сделать подобное оформление списка товаров.

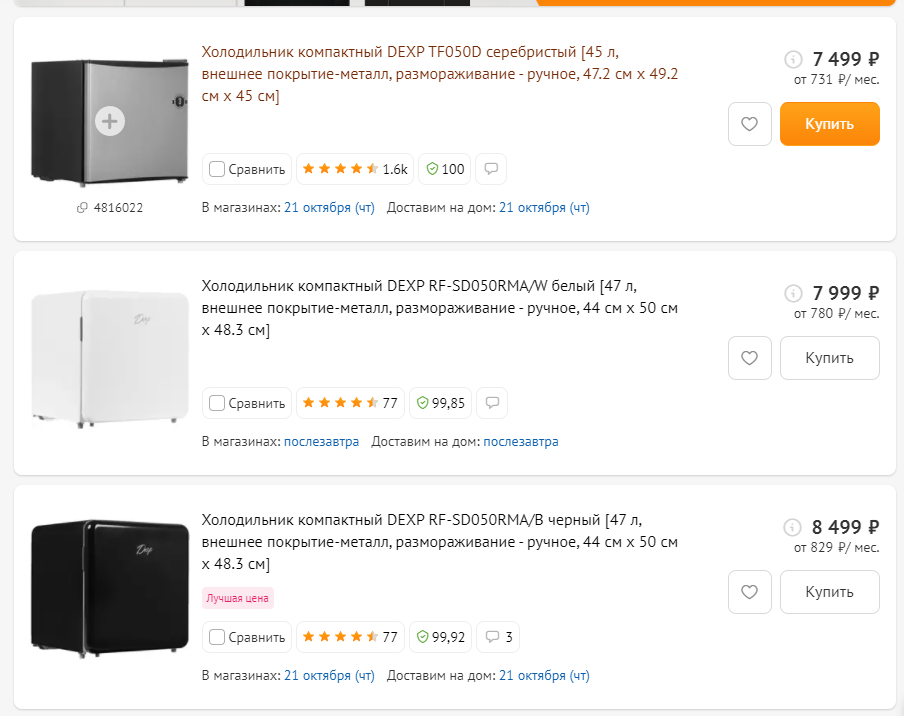


Рисунок 2.2 – Пример списка товаров в интернет-магазине DNS [[2]](#литра2)

Главный функционал, значимый для данного приложения и используемый на сайте DNS и аналогичных магазинах техники – фильтрация по характеристикам. DNS имеет крайне удобную и функциональную фильтрацию (Рисунок 2.3). Существует возможность выбора как и списка отдельных характеристик, так и диапазон значений (например: цена и диагональ у телевизоров).

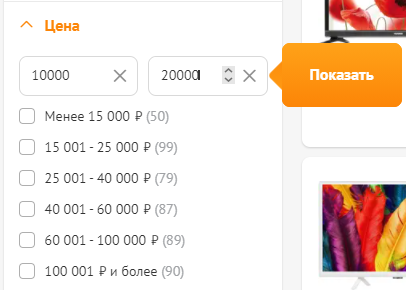


Рисунок 2.3 – Фильтрация по цене (имеется и диапазон и список значений)[[2]](#литра2)

В качестве еще одного аналога рассматривается часть “Список заказов” на сайте Wildberries.

Wildberries (Вайлдбериз) (с англ. — «дикие ягоды») — международный интернет-магазин одежды, обуви, электроники, детских товаров, товаров для дома и других товаров.[[3]](#литра3)

Данный интернет-магазин имеет удобный, простой и информативный список заказов, в котором показывается сам заказ, сумма заказа а так же статус заказа (Рисунок 2.4). Было решено сделать аналогичную систему списка заказов.

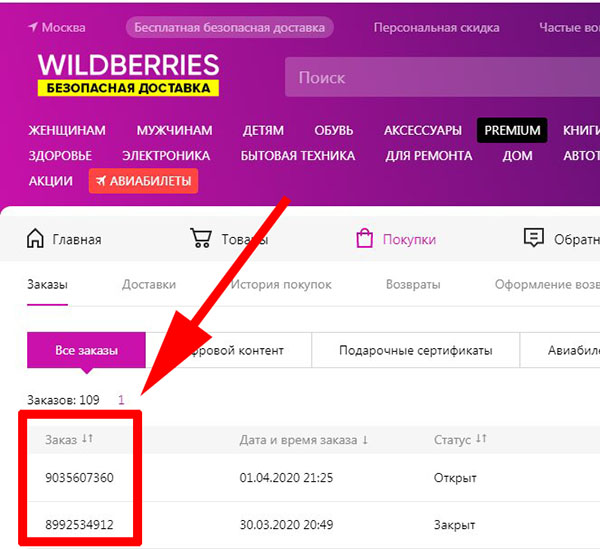


Рисунок 2.4 – Список заказов Wildberries[[4]](#литра4)

* 1. **Обоснование выбора инструментов и платформы для разработки.**

Для создания базы данных был выбран Microsoft SQL Server. Для разработки системы был выбрана среда разработки Visual Studio с кросс-платформенным фреймворком ASP.NET Core с использованием Entity Framework.

Основным преимуществом платформы .NET в отношении баз данных является наличие LINQ.

Аббревиатура LINQ обозначает целый набор технологий, создающих и использующих возможности интеграции запросов непосредственно в язык C#. Традиционно запросы к данным выражаются в виде простых строк без проверки типов при компиляции или поддержки IntelliSense. Кроме того, разработчику приходится изучать различные языки запросов для каждого типа источников данных: баз данных SQL, XML-документов, различных веб-служб и т. д. Технологии LINQ превращают запросы в удобную языковую конструкцию, которая применяется аналогично классам, методам и событиям.[[5]](#литра5)

ASP.NET Core - это кросс-платформенный фреймворк с открытым исходным кодом, разработанный совместно компанией Microsoft и ее сообществом. Это тотальное превращение ASP.NET, которое объединяет структуру MVC и Web API в единый фреймворк.[[6]](#литра6)

Основные преимущества ASP.NET Core:

* Архитектура MVC. ASP.NET Core упрощает разработку, компиляцию и тестирование разработчиками в модели, представлении или контроллере.
* Функциональность Razor Pages. Razor Pages - это новый элемент ASP.NET Core, который делает программные сценарии, основанные на веб-страницах, более производительными. С технической точки зрения, Razor Pages - это модель кодирования, основанная на веб-страницах, которая упрощает создание веб-интерфейса. С применением Razor Pages каждая веб-страница становится автономной с компонентом View, код также четко налажен.
* Улучшение совместной работы и кросс-платформенной поддержки. ASP.NET Core - это кросс-платформенный фреймворк, что означает, что приложения, построенные с использованием этого фреймворка, могут работать в операционных системах Windows, Linux и Mac. Кроме того, разработчики также могут свободно выбирать свои ОС разработки.

1. ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ
   1. Разработка структуры приложения.

Данное приложение имеет архитектуру “клиент-сервер” с использованием MVC (Model-View-Controller) (Рисунок 3.1)

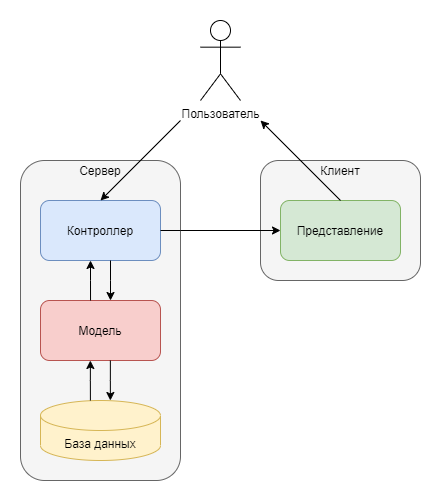


Рисунок 3.1 – Архитектура MVC

База данных хранит и исходные данные заполняемые администратором и данные пользователей и их заказы. Запросы к базе данных формируются в контроллерах. После получения данных, контроллер передает данную модель представлению на клиентскую сторону. Клиент из браузера может передавать определенные данные в контролеры на обработку и сохранение в базе данных.

Бизнес логика приложения находится в методах моделей (сущностей) и осуществляется в контроллерах. В представлениях на стороне пользователя нет бизнес логики.

Основное преимущество архитектуры MVC – модульность. Из-за этого свойства очень просто создавать множество представлений для одинаковых моделей (например представление модели “Категории” для администратора и пользователя). Так же с данной архитектурой намного проще отлаживать код (если, например, возникла ошибка в обработке данных, то сразу отбрасывается модель и представление, и ошибка ищется в контроллере).

* 1. Разработка алгоритмов обработки информации

Основные нетривиальные манипуляции с информацией – это функции связанные с характеристиками товаров и фильтрация по ним.

Основная преимущество хранения характеристик в виде JSON поля – возможность создавать и редактировать неограниченное количество характеристик, привязанных конкретно к одной категории, без создания дополнительных сущностей для характеристик. Из-за этого появляется другая проблема – обработка и создание этих характеристик.

Имеется класс CategoryCharacteristics – он находится в модели категории (сущность Category. Не записывается в базу данных), в виде List< CategoryCharacteristics> - так как может существовать много характеристик. Содержит поля string charactName (название самой характеристики) и класс CategoryCharacteristicsBool\_Value (объединяющий значения характеристики и значение, отвечающее за то, что являются ли эти значения числовыми и можно ли использовать для них фильтрация по диапазону) поля: List<string> charactValues, bool isNumeric. Данная странная на первый взгляд структура позволяет весьма просто сериализовать класс CategoryCharacteristics в поле типа JSON. При помощи словаря, вида Dictionary<string, CategoryCharacteristicsBool\_Value>, в который вносятся на место ключа название характеристик. После этого буквально одной встроенной функцией JsonConvert.SerializeObject из библиотеки Newtonsoft.Json производится сериализация данного обьекта в JSON. Далее данное значение сохраняется в базе данных в виде поля nvarchar(max); Десериализация происходит в обратном порядке с помощью функции JsonConvert.DeserializeObject.

Имеется класс ItemCharacteristics – он полностью зависит от класса рассмотренного выше (CategoryCharacteristics). Данный класс находится в модели товара (сущность Item. Не записывается в базу данных) в виде List< ItemCharacteristics> - так как может существовать много характеристик. Имеет поле string charactItemName – для имени характеристики, и поле string charactItemValue для значения характеристики. При создании нового товара (объекта Item) администратором – ему выдается список характеристик с их значениями из рассмотренного класса CategoryCharacteristics, после он выбирает для каждой характеристики одно значение (т.к. у товара может быть, например, только одна диагональ или один цвет). Далее название характеристики и ее значение записывается в объект ItemCharacteristics. При записи в базу данных с помощью объекта словаря вида Dictionary<string,string> записывается характеристика и ее значение, далее с помощью той же функции JsonConvert.SerializeObject происходит сериализация в JSON. Аналогично производится десериализация.

Фильтрация по описанным выше характеристикам происходит в классе FullFilter и Filter. Класс Filter имеет поля List<bool> exactValue, List<string> exactValueString, double? from и double? to. Поле фильтра exactValue – список значений bool, так как к каждому значению характеристики создается checkbox. При выборе значений характеристики пользователь меняет значения в этом списке с 0 на 1. Поля from и double нужны, если характеристика представляется диапазоном значений. Класс FullFilter включает в себя лист Filter – это и есть характеристики категории, также он включает схожие поля, описанные выше, только для цены и производителя (так как эти характеристики свойственны любому товару). Пользователь, применив фильтр, совершает POST запрос на сервер, передавая класс FullFilter. Он обрабатывается, находя те характеристики, значения которых были затронуты (или exactValue – есть хотя бы одно значение равное единице, или заполнены to и from). После этого формируется URL для GET запроса (например: … /filter?price=10000;15000&Разрешение%20экрана=HD;FullHD). Это нужно для того, чтобы пользователь мог обновить страницу и не отправлять повторный POST запрос, плюс это дает пользователю возможность сохранить фильтрацию (скопировав полный URL), если потребуется. Далее, совершается GET запрос, происходит обработка URL. URL “разбирается” и сопоставляется модели фильтра. После этого происходит фильтрация по известным значениям. Из-за такой архитектуры хранения характеристик приходится сначала получать весь список товаров категории, в которой происходит фильтрация, а далее, уже выбирать те товары, характеристики которых удовлетворяют полученным фильтрам. Сформированный список товаров, вместе с фильтром (для дальнейшей фильтрации) поступает в представление пользователя.

* 1. Логическая схема базы данных

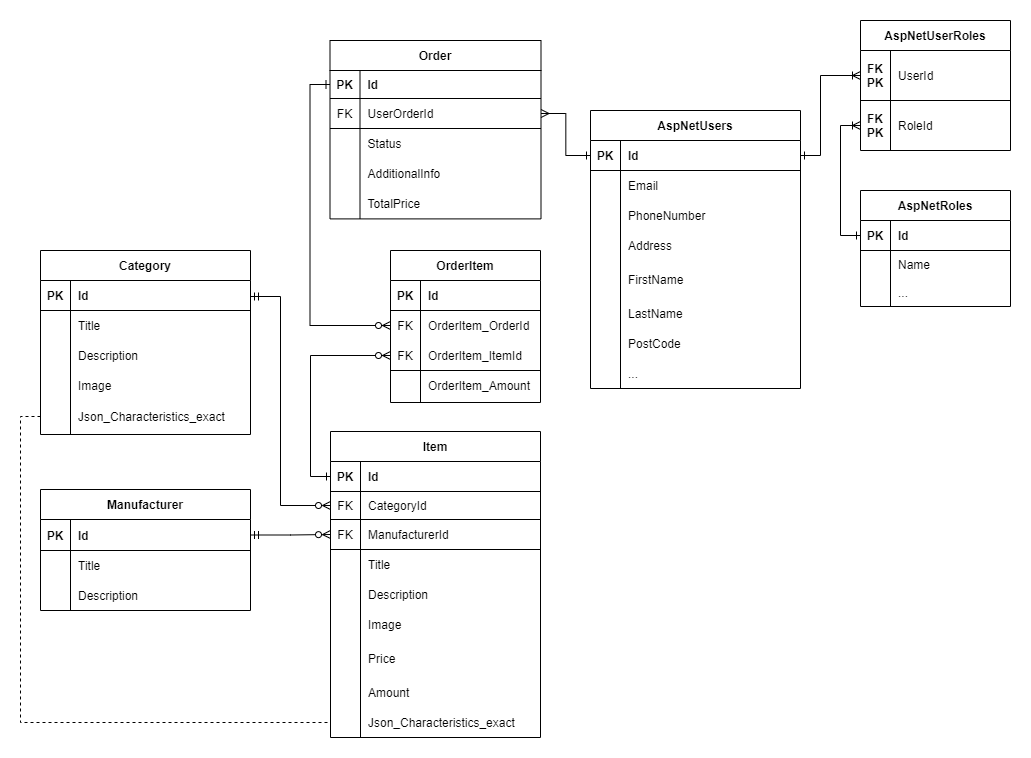


Рисунок 3.3 – Логическая схема базы данных

* 1. Разработка интерфейса взаимодействия пользователя с системой

Были разработаны макеты веб-страниц приложения:

* Страница для просмотра категорий (домашняя страница) (Рисунок 3.4.1). При добавлении новых категорий страница будет более заполненной.

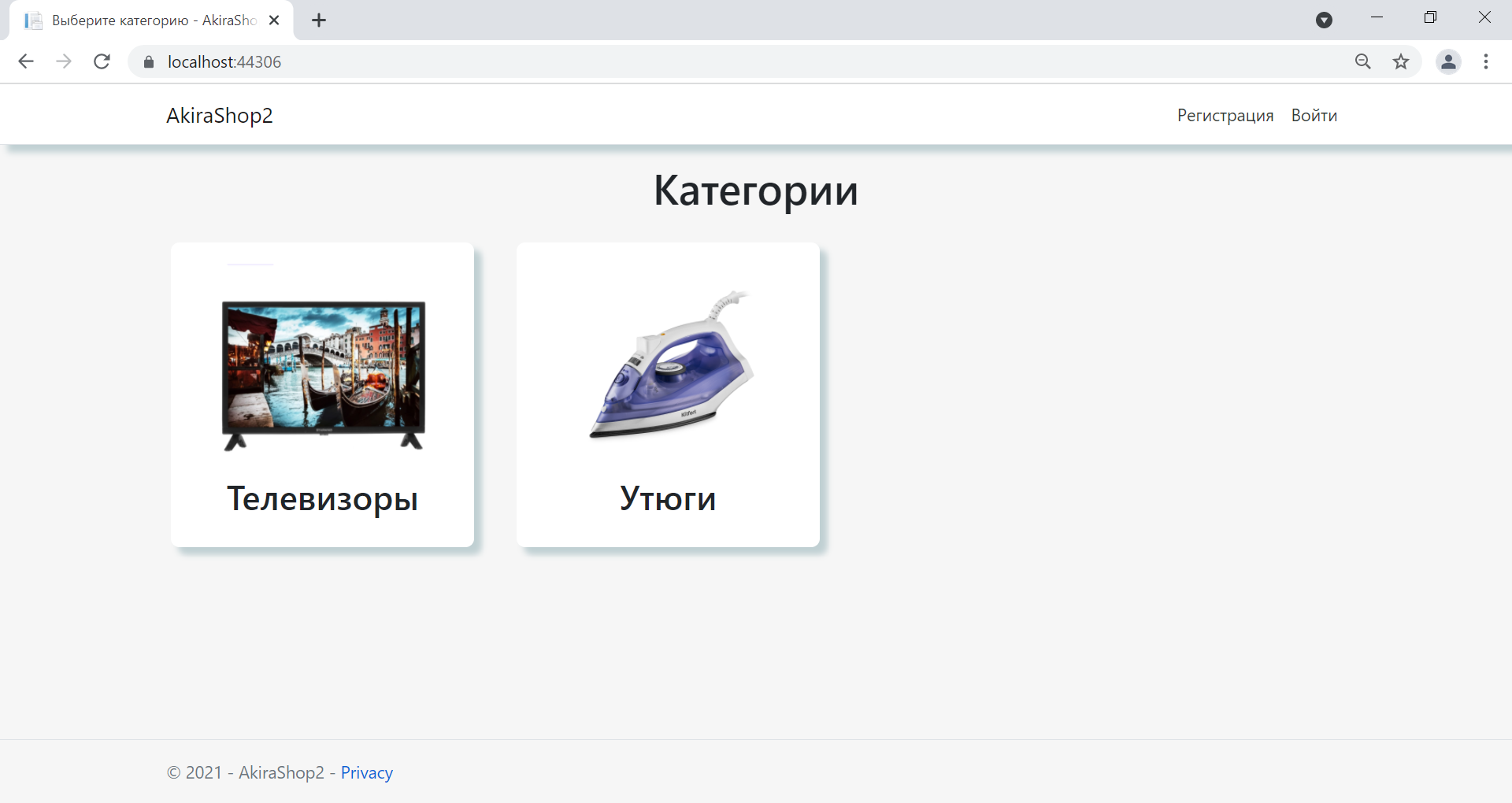


Рисунок 3.4.1 – Категории товаров

* Страница для просмотра товаров и фильтрации определенной категории (Рисунок 3.4.2).

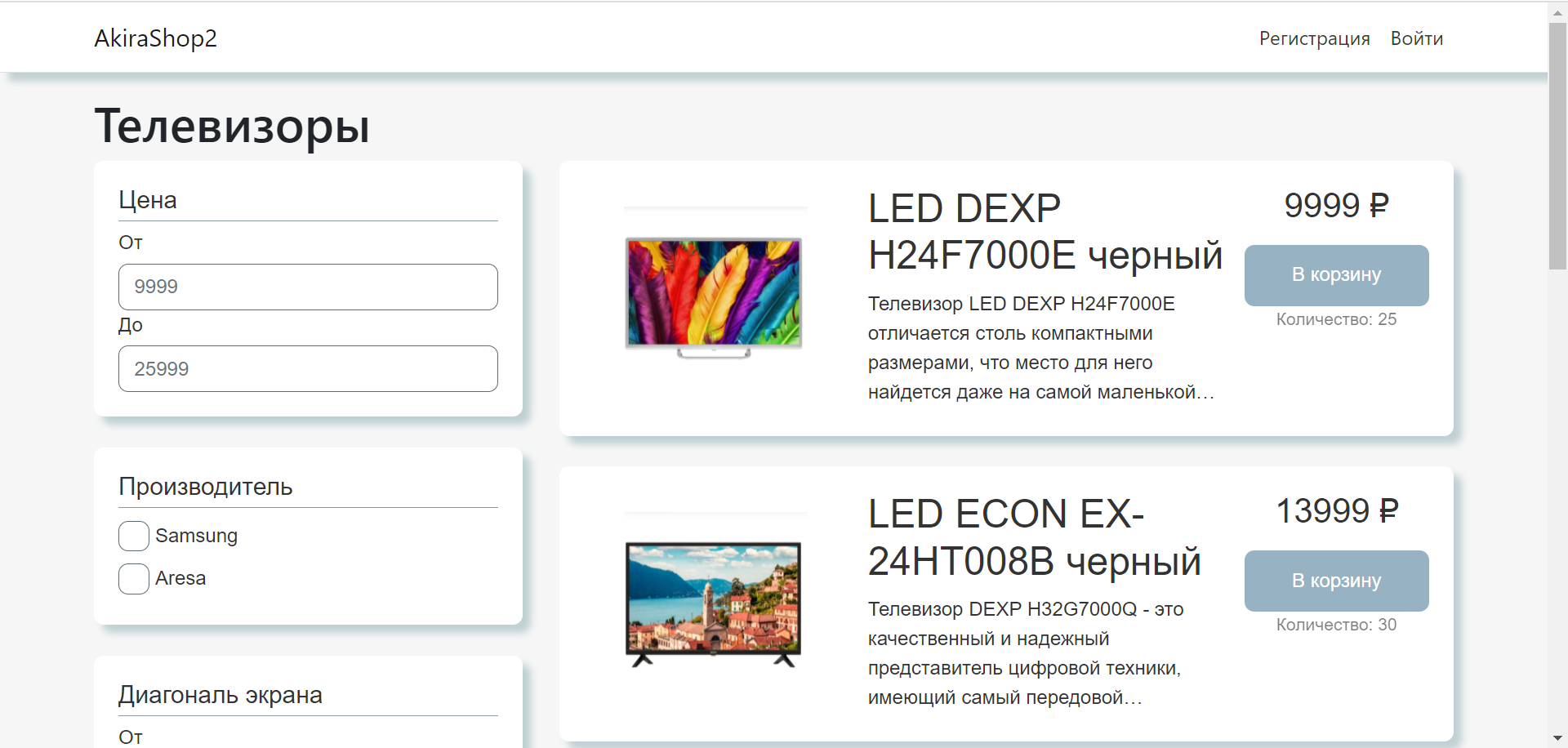


Рисунок 3.4.2 – Список товаров категории “Телевизоры”

* Страница для просмотра корзины и избранного (Рисунок 3.4.3). Аналогично выглядит страница готовых заказов. Доступно любому авторизованному пользователю.

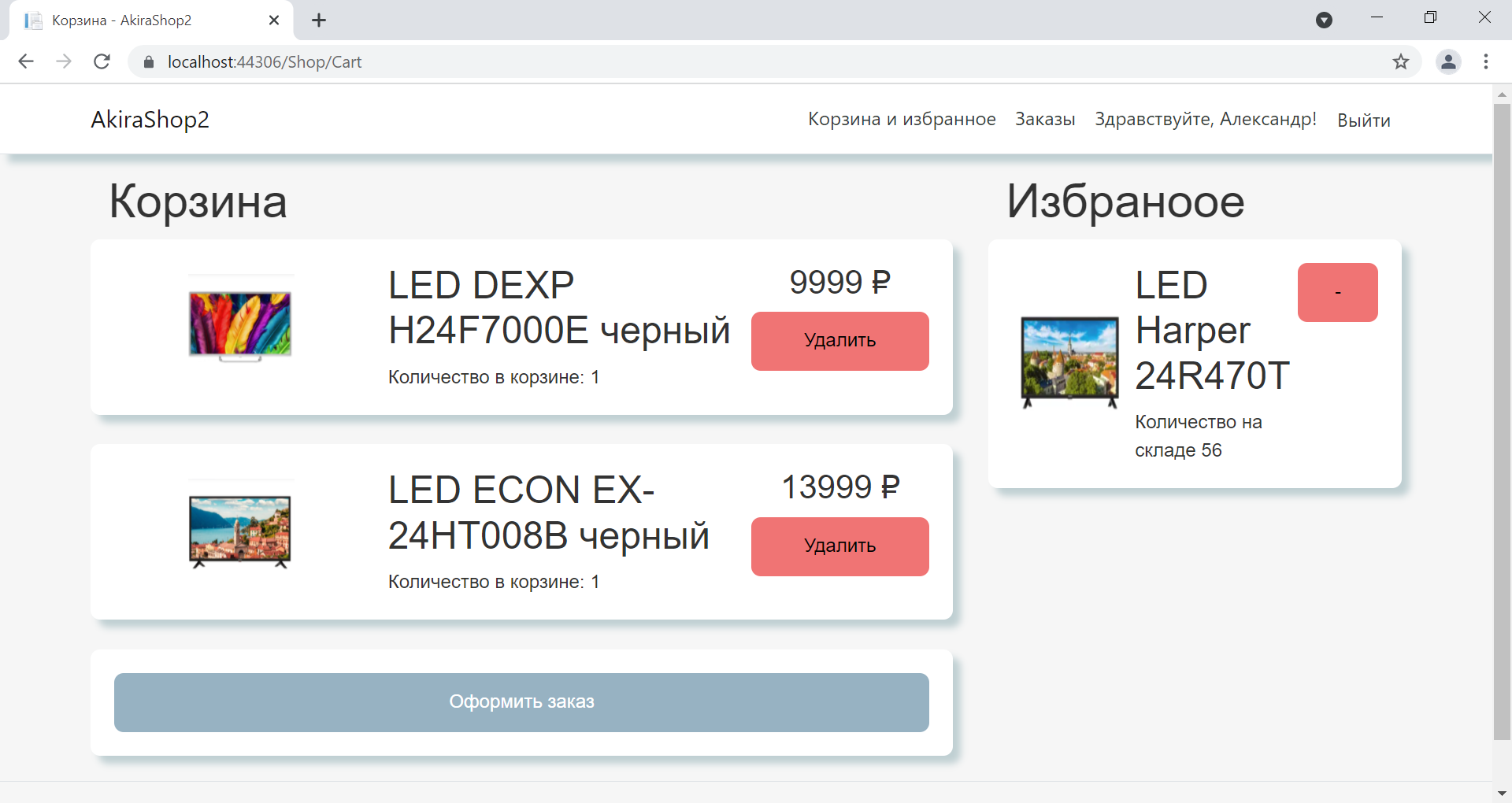


Рисунок 3.4.3 – Корзина и избранное

* Модальная страница формы авторизации пользователя (Рисунок 3.4.4) и регистрации (Рисунок 3.4.5). Данная страница появляется поверх основных страниц.

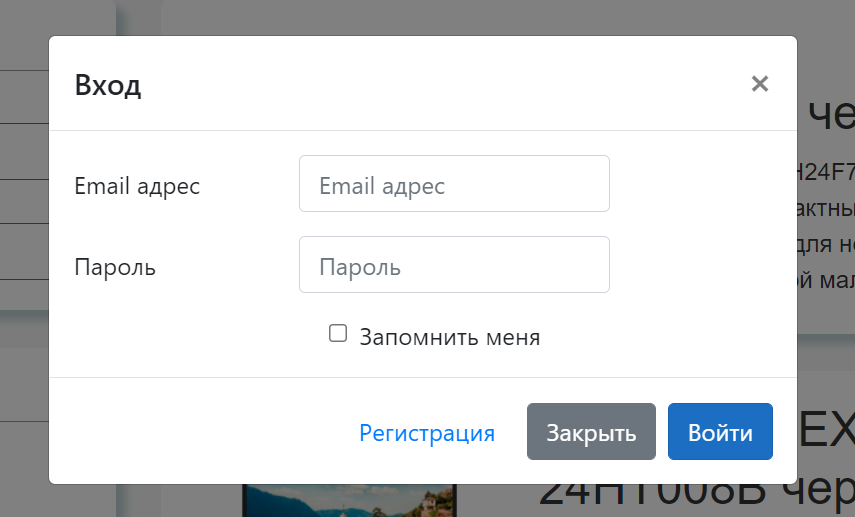


Рисунок 3.4.4 – Форма авторизации пользователя

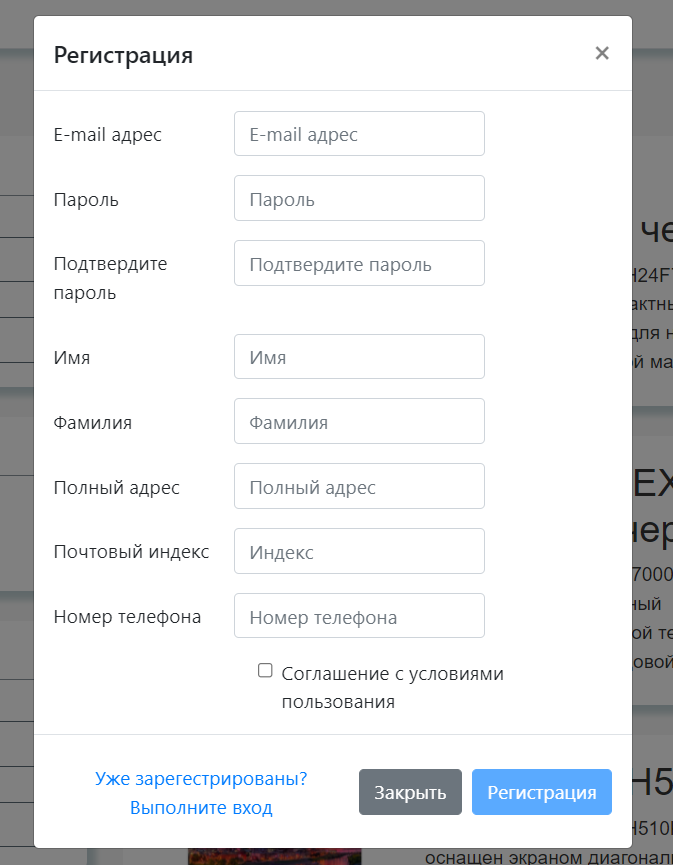


Рисунок 3.4.5 – Форма регистрации пользователя

* Страница добавления, изменения категории (Рисунок 3.4.6) и товаров (Рисунок 3.4.7). Страница производителей - аналогична предыдущим. Данные страницы доступны администратору.

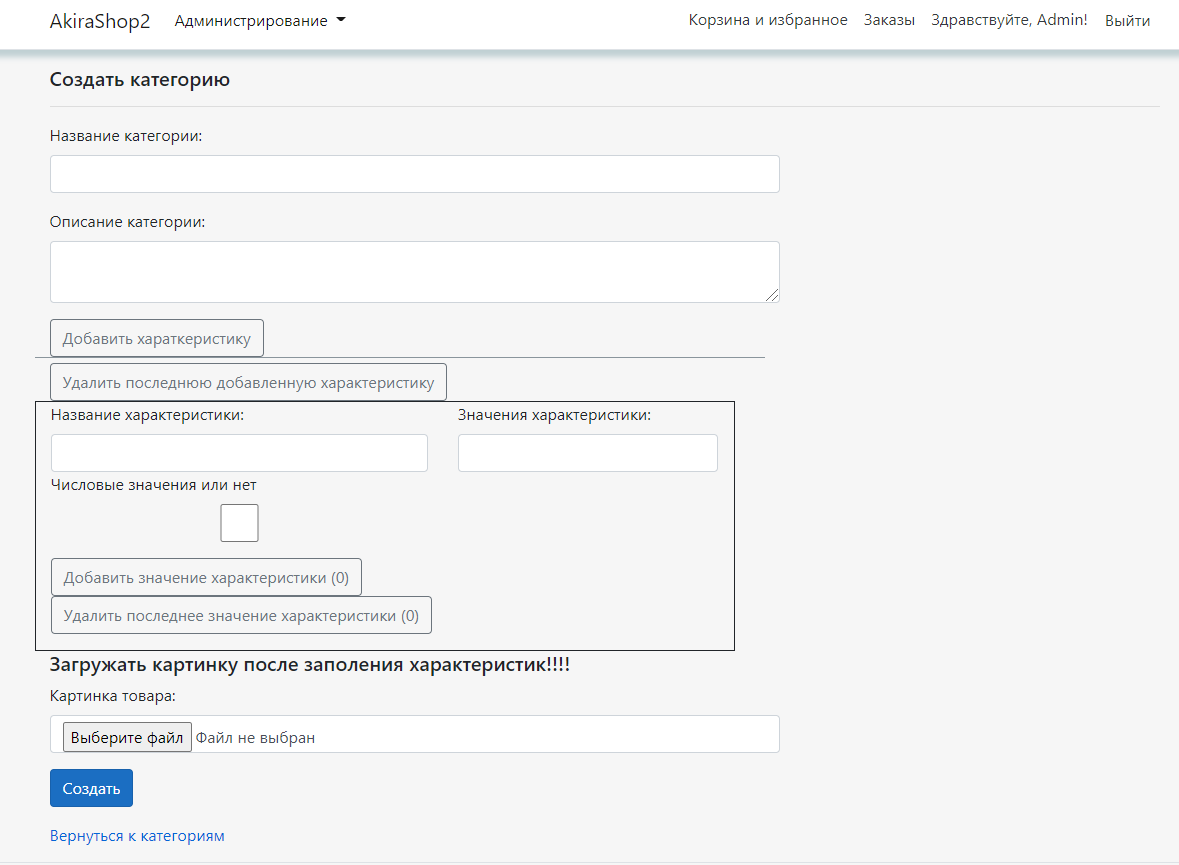


Рисунок 3.4.6 – Страница создания/изменения категории

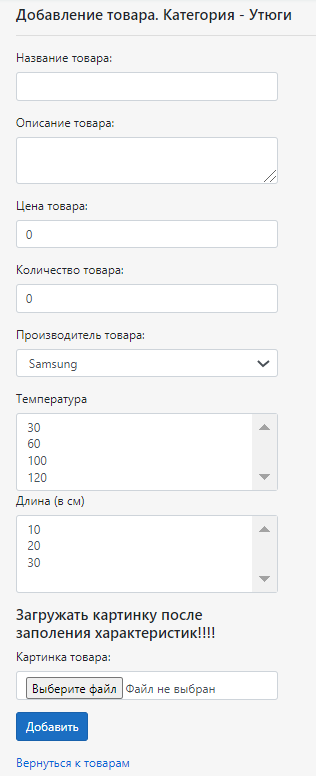


Рисунок 3.4.7 – Страница создания/изменения товара категории “Утюги”

* Страница просмотра товаров категории (Рисунок 3.4.8). Страницы просмотров категории, пользователей, производителей и заказов аналогичны. Данные страницы доступны администратору.

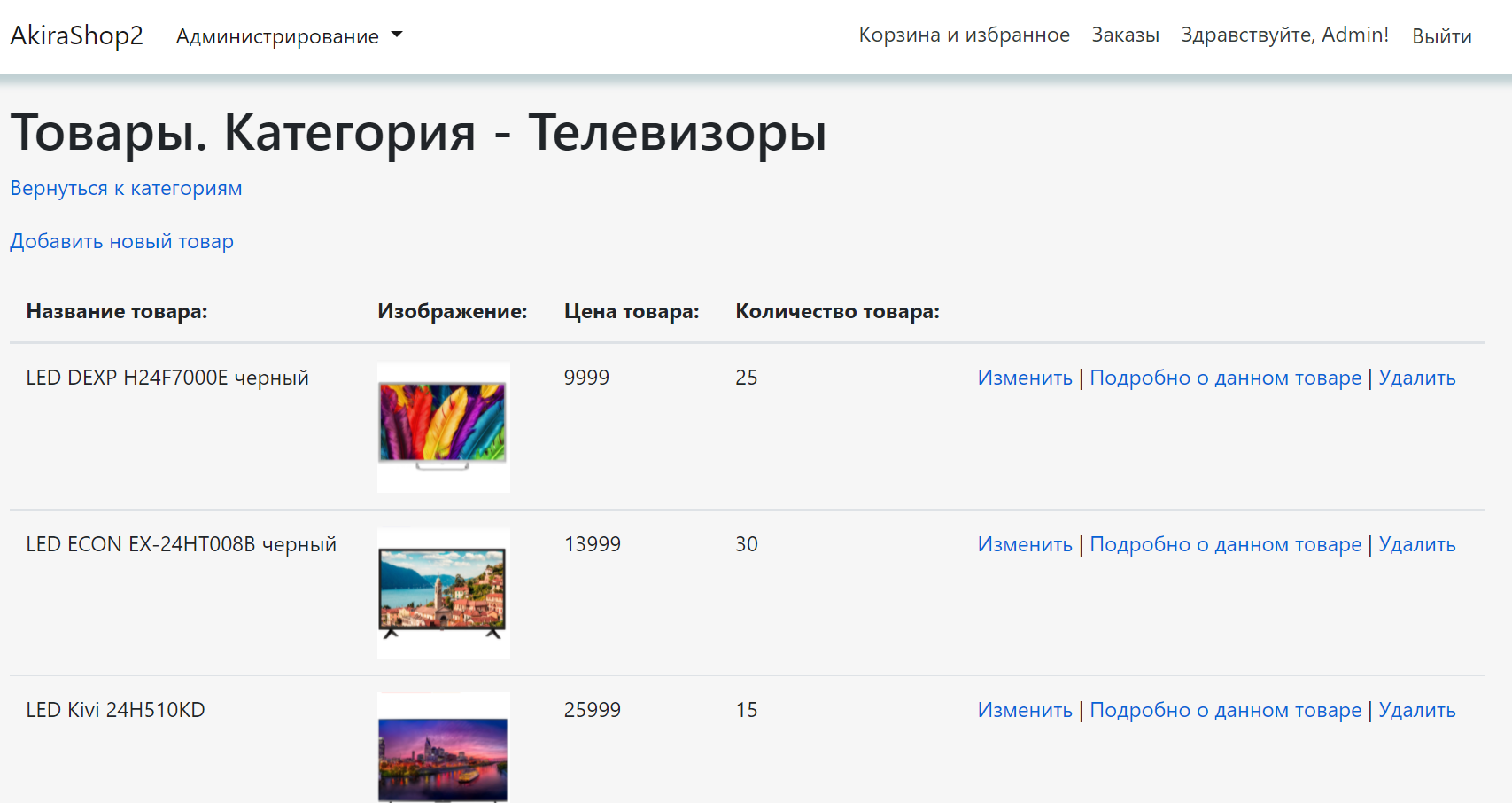


Рисунок 3.4.8 – Страница просмотра товаров категории “Телевизоры”

* Страница поиска по конкретным значениям (Рисунок 3.4.9). Доступна администратору.

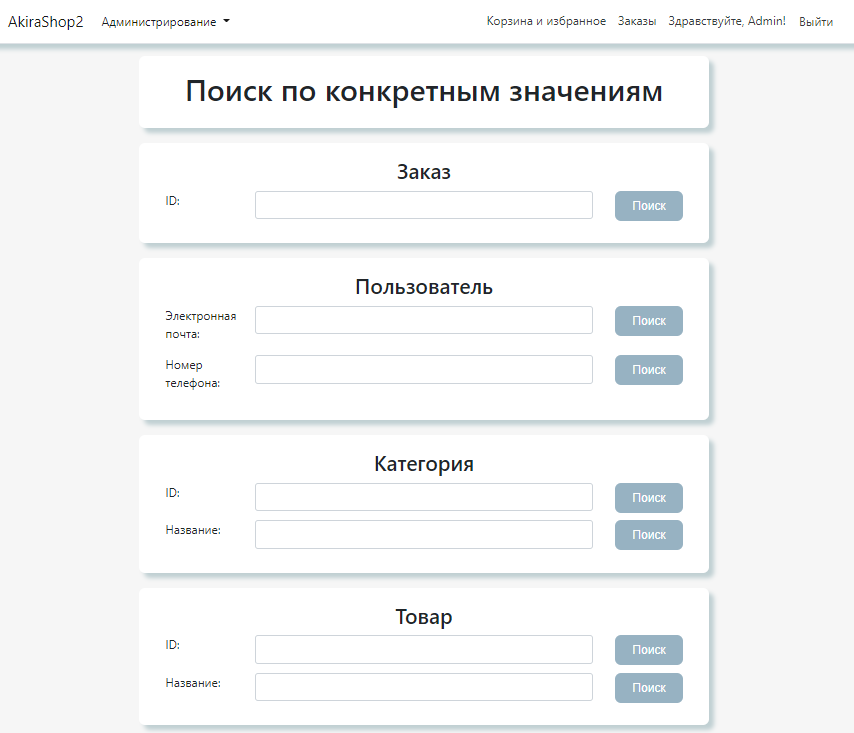


Рисунок 3.4.9 – Поиск по конкретным значениям

* Страница изменения пользовательской информации (Рисунок 3.4.10). Доступна любому авторизованному пользователю.

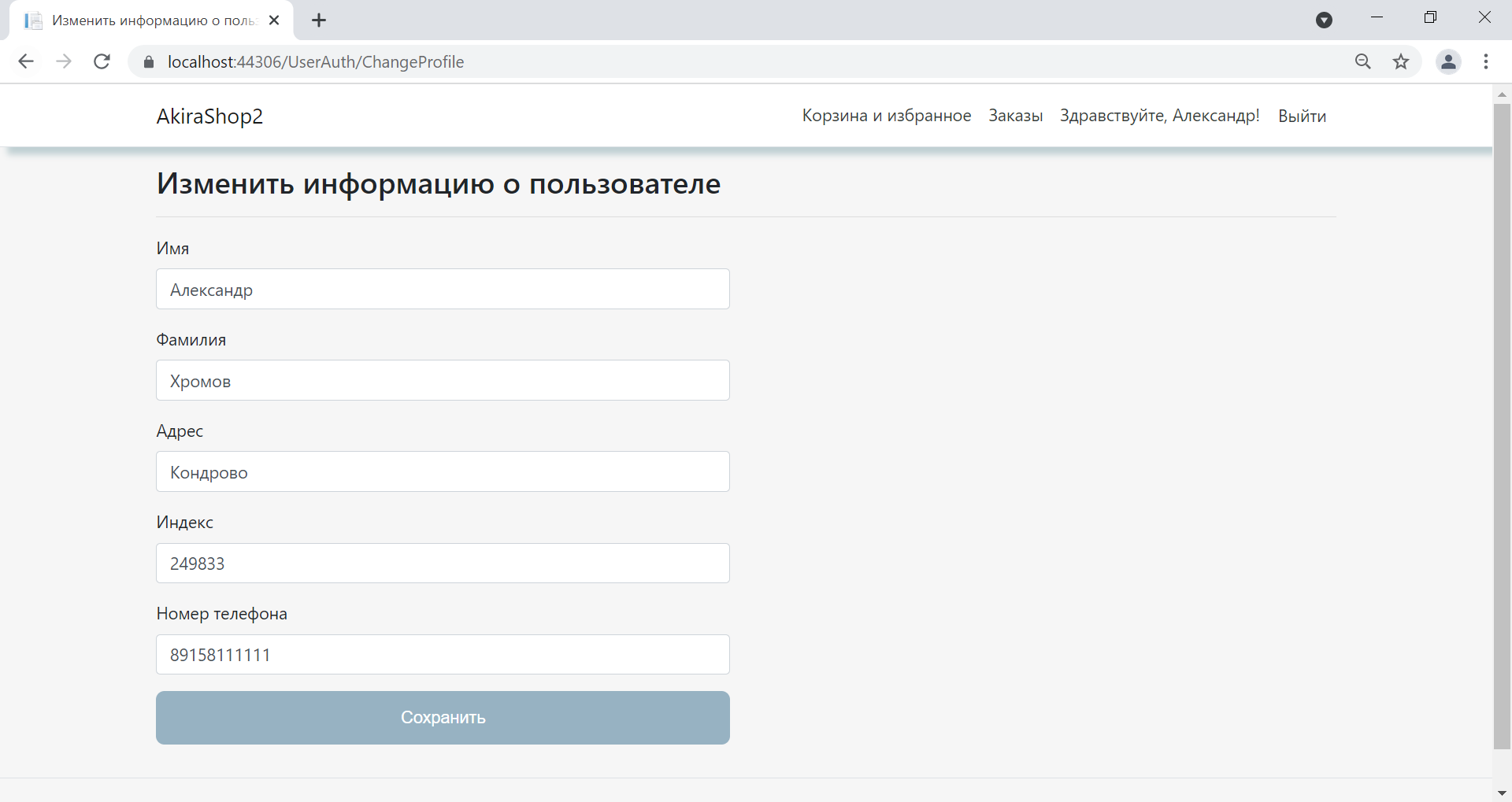


Рисунок 3.4.10 – Изменение пользовательской информации

* 1. Разработка архитектуры приложения.

Было решено создавать именно веб-приложение, так как оно более универсально и практично для конечного пользователя. Веб-приложением можно пользоваться из любого браузера на любой операционной системы.

Для данного проекта подошла бы любая платформа разработки веб-приложения, была выбрана, как уже указывалось в исследовательской части, ASP.NET Core. Данная платформа выбрана из-за того что язык C# является уже хорошо знакомым. В следствии этого во время написания кода почти не возникло проблем.

Частью платформы .NET является Entity Framework. Данный ORM-фреймворк тоже имеет ряд преимуществ. Он позволяет создавать таблицы базы данных и саму базу данных напрямую из кода проекта. Entity Framework выполняет соответствующий запрос в базу данных, а затем предоставляет результаты в экземплярах объектов веб-приложения, для того чтобы далее с ним было проще работать. Данный фреймворк позволяет выполнять запросы LINQ к базе данных.

В данном проекте используется СУБД Microsoft SQL Server. Данная СУБД не особо отличается от уже изученной PostgreSQL. Так же данная СУБД уже является частью ASP .NET и не требуется устанавливать дополнительные библиотеки для другой СУБД.

Передача информации между компонентами внутри приложения осуществляется, как уже говорилось в исследовательской части, с использование MVC (Model-view-controller). Контроллеры используют REST архитектуру. В некоторых случаях (например при фильтрации) используется модель поведения PRG (Post-Redirect-Get), которая защищает пользователя от повторной отправки данных веб-форм, и значительно улучшает восприятие приложения пользователем.

* 1. Реализация функционирующего приложения

Описание сущностей базы данных:

‘Category’ – сущность для категорий. Имеет поля:

* Id – PRIMARY KEY, int, Не NULL
* Title – nvarchar(200), Не NULL
* Description – nvarchar(200), Не NULL
* Image – nvarchar(max), NULL
* Json\_Characteristics\_model – nvarchar(max), NULL

Последнее поле используется для фильтрации и имеет вид JSON: {“НазваниеХарактеристики” : { “Значения”: [“значение1”, “значение2”, “значение3”, …], “ДиапазонИлиНет”: true/false } }

Далее данное поле используется вместе с сущностью ‘Item’ для заполнения характеристик товара и фильтрации по ним.

‘Item’ – сущность для товара. Имеет поля:

* Id – PRIMARY KEY, int, Не NULL
* Title – nvarchar(200), Не NULL
* Description – nvarchar(1000), Не NULL
* Image – nvarchar(max), NULL
* Price – bigint, Не NULL
* Json\_Characteristics\_exact – nvarchar(max), NULL
* Amount – bigint, Не NULL
* CategoryId – FOREIGN KEY => Category.Id, int, Не NULL
* ManufacturerId – FOREIGN KEY => Manufacturer.Id, int, Не NULL

Поле Json\_Characteristics\_exact используется, как уже говорилось выше, для фильтрации товара. Имеет вид JSON: { “Характеристика1” : ”Значение1”, “Характеристика2” : ”Значение2”, “Характеристика3” : ”Значение3”, …}

Значения и характеристики соответствуют полю Json\_Characteristics\_model в сущности ‘Category’.

‘Manufacturer’ – сущность для производителя. Имеет поля:

* Id – PRIMARY KEY, int, Не NULL
* Title – nvarchar(200), Не NULL
* Description – nvarchar(200), Не NULL

‘Order’ – сущность для заказа. Имеет поля:

* Id – PRIMARY KEY, int, Не NULL
* UserOrderId – FOREIGN KEY => AspNetUsers.Id, navchar(450), Не NULL
* Status – nvarchar(max), Не NULL
* AdditionalInfo – nvarchar(max), NULL
* TotalPrice – bigint, Не NULL

‘OrderItem’ – дополнительная сущность для осуществления связи “многое-ко-многим” сущностей Order и Item и для хранения количества каждого товара в заказе. Имеет поля:

* Id – PRIMARY KEY, int, Не NULL
* OrderItem\_OrderId – FOREIGN KEY => Order.Id, int, Не NULL
* OrderItem\_ItemId – FOREIGN KEY=> Item.Id, int, Не NULL
* OrderItem\_Amount – int, Не NULL

‘AspNetUsers’ – сущность пользователя. В сущности содержатся стандартные поля ASP .NET фреймворка. К ним были добавлены дополнительные поля. Перечислим только значимые поля данной таблицы:

* Id – PRIMARY KEY, nvarchar(450), Не NULL
* Email – nvarchar(256), NULL
* PhoneNumber – nvarchar(max), NULL
* Address – nvarchar(250), NULL
* FirstName – nvarchar(250), NULL
* LastName – nvarchar(250), NULL
* PostCode – nvarchar(50), NULL

‘AspNetRoles’ – сущность ролей пользователей. В сущности содержатся стандартные поля ASP .NET фреймворка. Перечислим только значимые поля данной таблицы:

* Id – PRIMARY KEY, nvarchar(450), Не NULL
* Name – nvarchar(256), NULL

‘AspNetUserRoles’ – дополнительная сущность для осуществления связи “многое-ко-многим” сущностей AspNetUsers и AspNetRoles. Имеет поля:

* UserId – PRIMARY KEY, FOREIGN KEY => User.Id, nvarchar(450), Не NULL
* RoleId – PRIMARY KEY, FOREIGN KEY => Role.Id, nvarchar(450), Не NULL

1. ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ
   1. Проектирование начального и тестового наполнения базы данных. Процедуры автоматизации.
   2. Технологические решения, поддерживающие эксплуатационный цикл программы.

Данное приложение можно улучшить в будущем многими способами. Так как разработка приложения была ограничена небольшим количеством времени, то большинство алгоритмов были реализованы не идеально. Большинство из них можно оптимизировать. Алгоритмы фильтрации и сериализации JSON характеристик (описанные в конструкторской части) возможно оптимизировать или выбрать другое решение, более оптимальное. Так же оптимизации подлежит визуальная составляющая. Многие страницы приложения не адаптивны, и неверно отображаются на мобильных устройствах.

В данное приложение можно добавить систему отзывов покупателей. Можно разделять отзывы от тех, кто купил товар, и тех кто не покупал на данном сайте (осуществлять проверку на то, что имеется ли товар, к которому пишется отзыв, в доставленных заказах пользователя).

Полезным нововведением будет роль доставщика в системе. Администратор может выдавать доставщикам заказы, которые они могут просмотреть на сайте, узнать подробно информацию о заказе, адрес и контактные данные пользователя. Но данная функция будет уместна при расширении самого магазина бытовой техники, так как для некрупного магазина это не дает особых преимуществ.

* 1. Порядок развёртывания системы.

Данное веб приложение было развернуто в облачной платформе Azure компании Microsoft. Данное развертывание является весьма простым и быстрым в реализации, из-за использования Visual Studio (Рисунок 4.3.1).

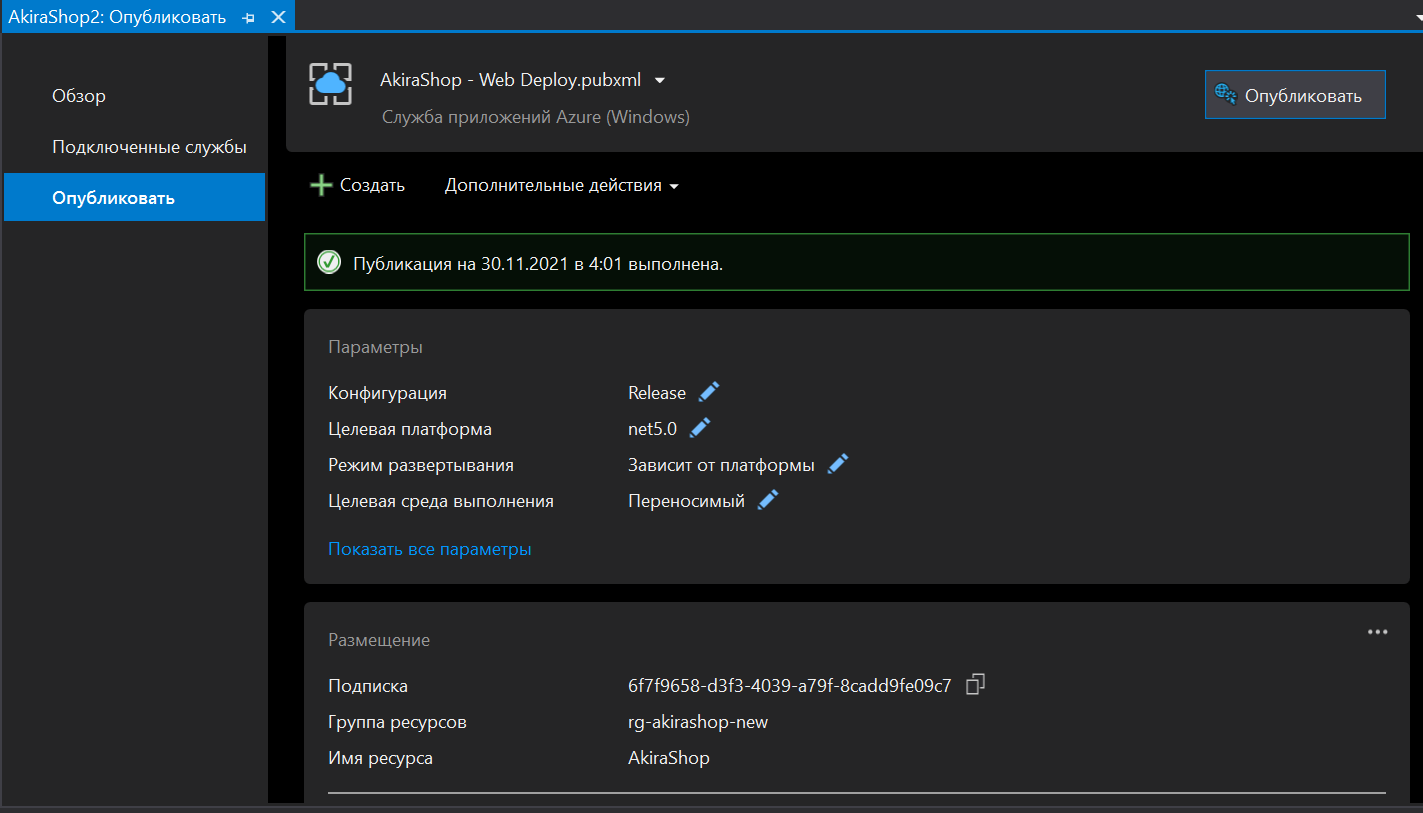


Рисунок 4.3.1 – Профиль опубликованного приложения

Для корректной публикации в Azure был создана группа ресурсов для приложения и базы данных. База данных была так же настроена на связь с Entity Framework для корректного применения миграций.

* 1. Разработка руководства пользователя и руководства администратора.

Было составлено руководство пользования веб-приложением только администратора, так как именно администратор будет иметь трудности работы с данным приложением из-за большого количества административных функций. Со стороны пользователя данное приложение выглядит и работает как уже знакомый многим интернет-магазин. По этой причине руководство для обычного пользователя не было составлено.

**Руководство администратора:**

1. Для возможности администрирования веб-приложения требуется выполнить авторизацию с логином и паролем пользователя, имеющего доступ к администрированию сайта (Рисунок 4.4.0). Если нужно установить данную роль пользователю, то требуется сделать это в СУБД.

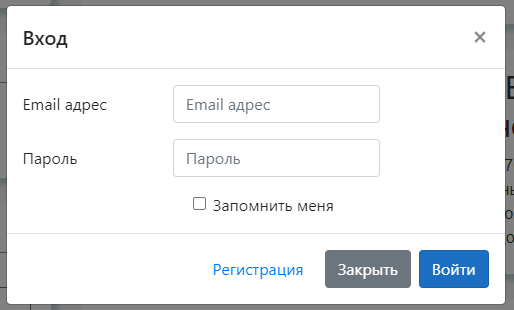


Рисунок 4.4.0 – Форма авторизации

1. После авторизации в верхней части страницы появится вкладка “Администрирование” (Рисунок 4.4.1). При нажатии на вкладку появляются несколько опций выбора. Подробнее о них далее.

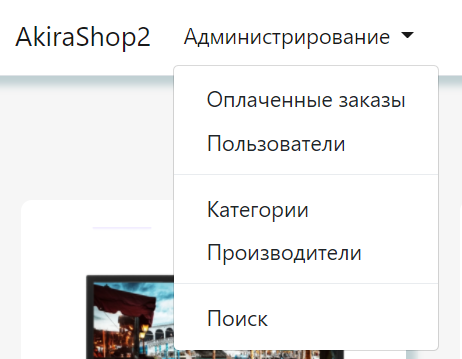


Рисунок 4.4.1 – Вкладка администрирования

1. Оплаченные заказы (Рисунок 4.4.2) – заказы, у которых находится статус “Оплачено”. Данная страница нужна для того, чтобы при доставке оплаченного заказа изменить статус данного заказа на “Доставлено”. У каждого оплаченного заказа на данной странице есть кнопка “Отметить заказ доставленным”, которая и меняет статус заказа на “Доставлено”. Так-же есть кнопка удаления заказа. Для просмотра дополнительной информации пользователя есть ссылка “Подробнее о пользователе” (Рисунок 4.4.4).

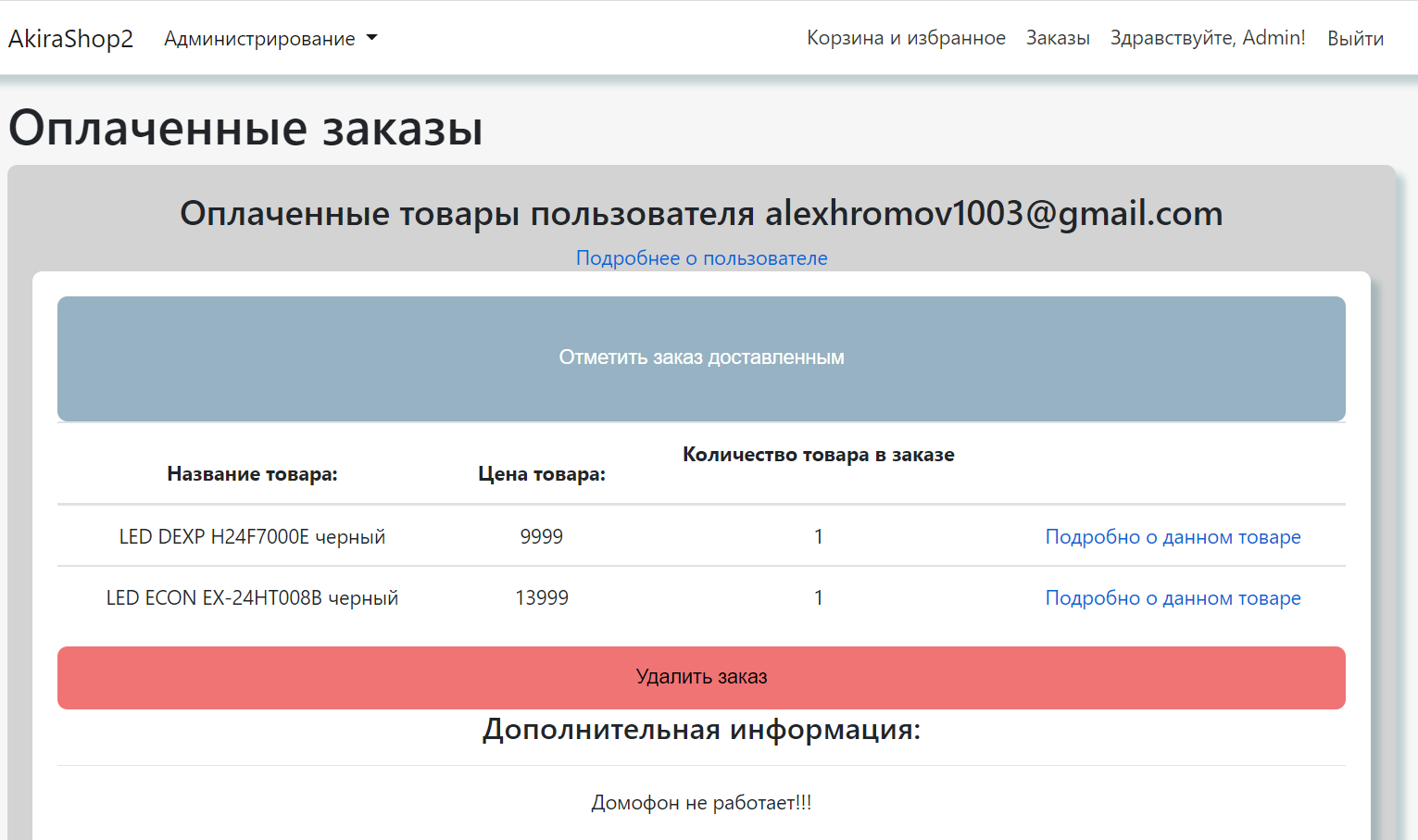


Рисунок 4.4.2 – Страница с оплаченными заказами

1. Пользователи (Рисунок 4.4.3) – список всех зарегистрированных пользователей. У каждого пользователя есть ссылка на страницу с подробной информацией (Рисунок 4.4.4) и на страницу с его заказами (Рисунок 4.4.5). На странице с заказами пользователя в разделе “Оплаченные заказы” такой же функционал как и на странице с Оплаченными заказами всех пользователей.

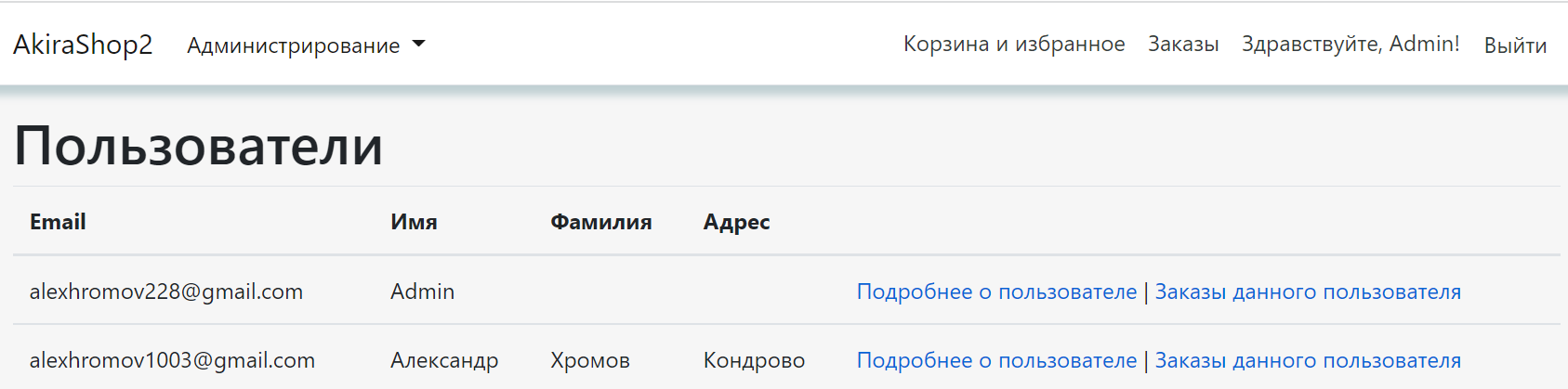


Рисунок 4.4.3 – Страница со списком пользователей

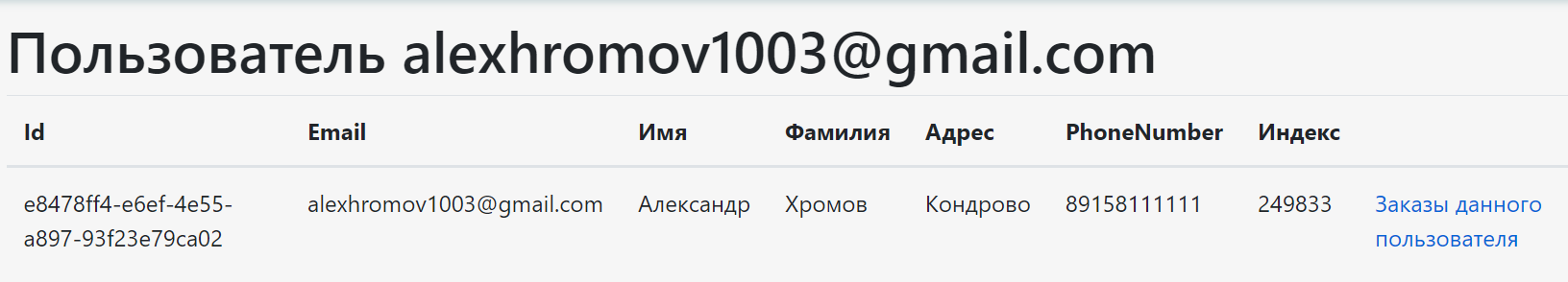


Рисунок 4.4.4 – Страница с подробной информацией о пользователе

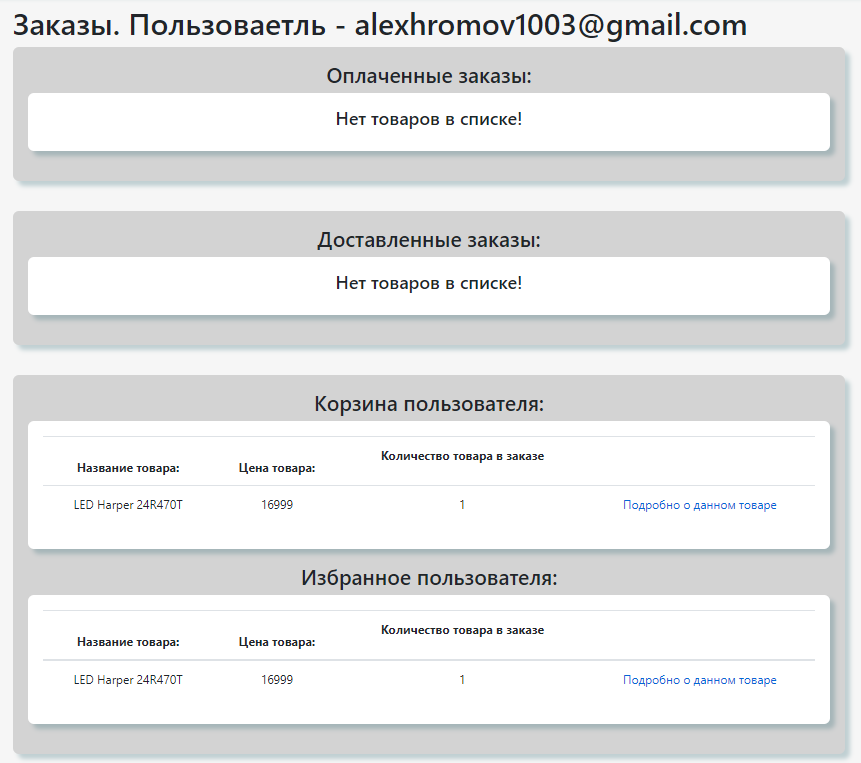


Рисунок 4.4.5 – Страница с заказами пользователя

1. Категории (Рисунок 4.4.6) – страница со всей информацией о категориях и их настройками. У каждой категории есть ссылки на изменение, детальной информации, удаление, и товары данный категории. Так же в верхней части страницы есть кнопка создания категории. Нажатие на нее открывает страницу создания категории (Рисунок 4.4.7). На данной странице при нажатии на кнопку “Добавить характеристику” происходит добавление нового блока характеристик, в котором можно добавлять и удалять значения характеристики соответствующими кнопками. Ниже есть кнопка для добавления изображения категории. После ввода всех данных следует нажать на кнопку “Создать” для сохранения введенной информации. Страница изменения категории идентична странице создания. На странице “Детали” можно посмотреть все характеристики выбранной категории.

При нажатии на ссылку “Товары данной категории” открывается страница со списком всех товаров данной категории (Рисунок 4.4.8).

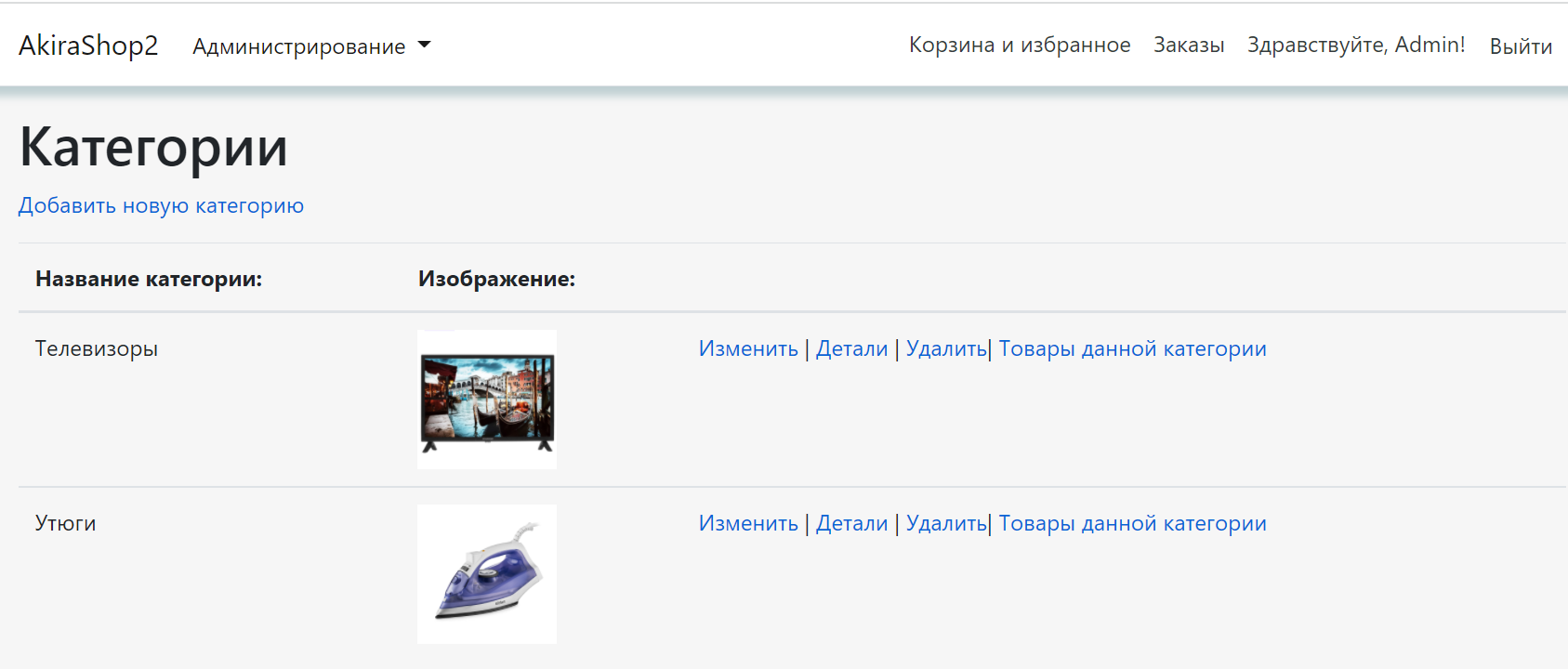


Рисунок 4.4.6 – Страница со списком категорий

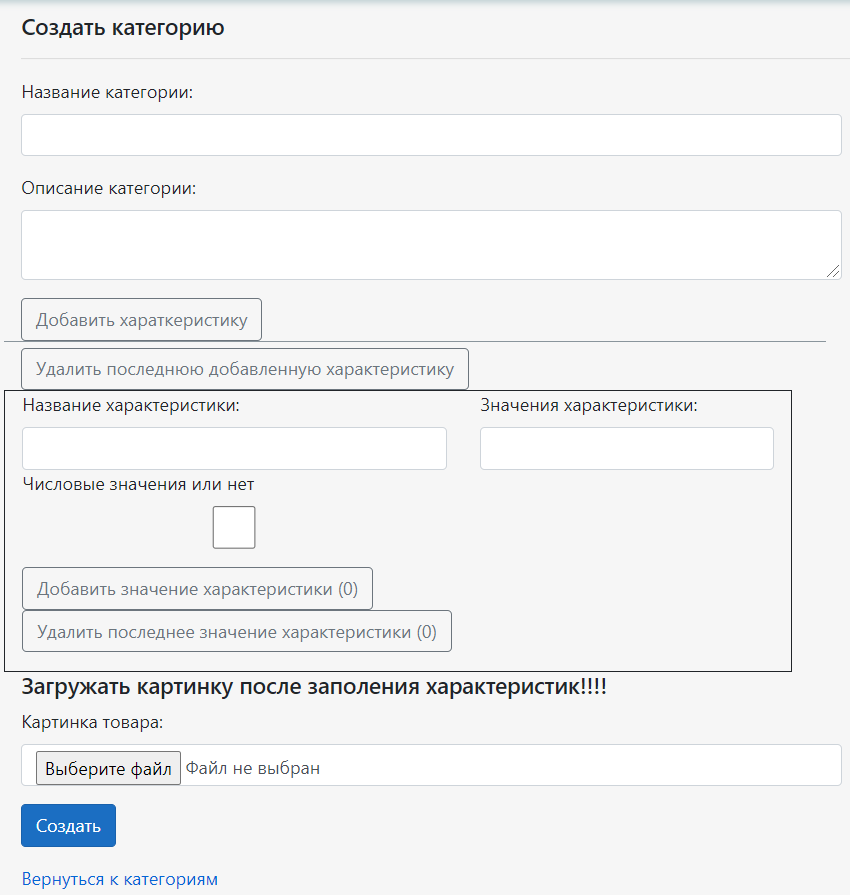


Рисунок 4.4.7 – Создание категории

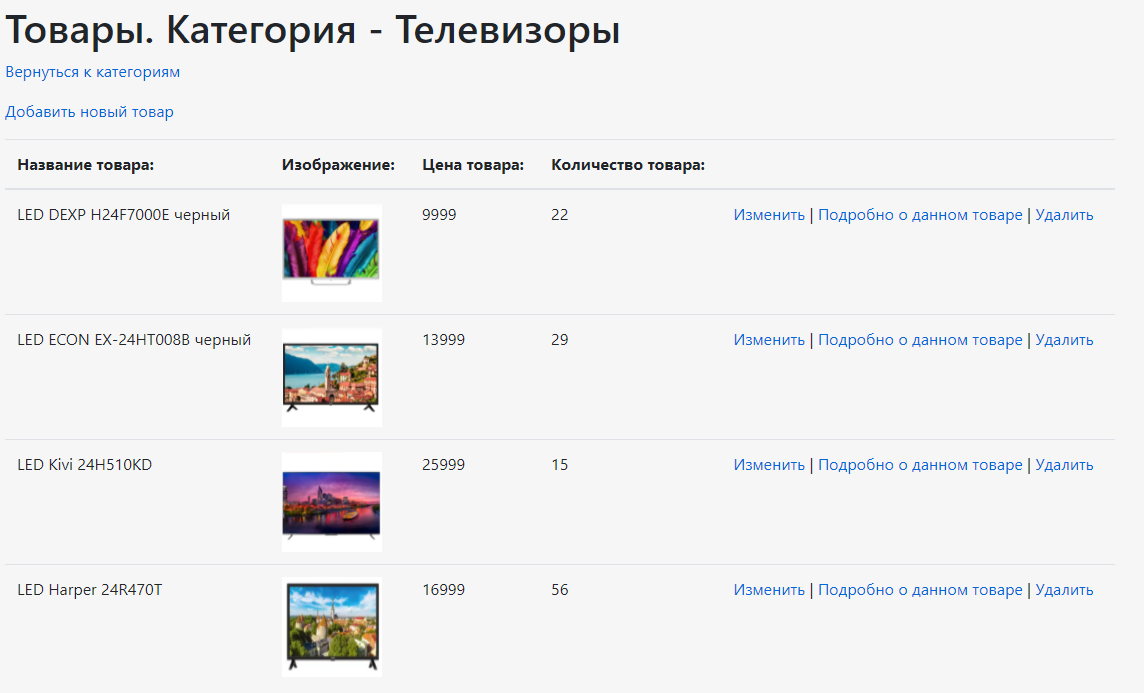


Рисунок 4.4.8 – Список товаров категории телевизоры

В верхней части страницы есть кнопка добавления нового товара. Нажатие на нее открывает страницу добавления товара (Рисунок 4.4.9). На данной странице нужно выбрать значения всех характеристик ранее добавленной категории. Ниже есть кнопка для добавления изображения товара. После ввода всех данных следует нажать на кнопку “Добавить” для сохранения введенной информации. Страница изменения товара идентична странице создания. На странице “Детали” можно посмотреть полную информацию о товаре.

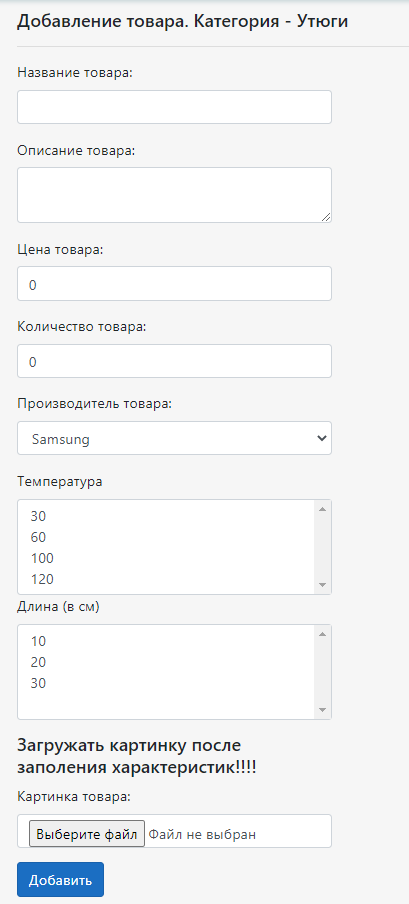


Рисунок 4.4.9 – Добавление нового товара

1. Производители (Рисунок 4.4.10) – страница со всей информацией о производителях и их настройками. В верхней части страницы есть ссылка на страницу для добавления нового производителя (Рисунок 4.4.11). У каждого производителя есть ссылки на страницы изменения и удаления.

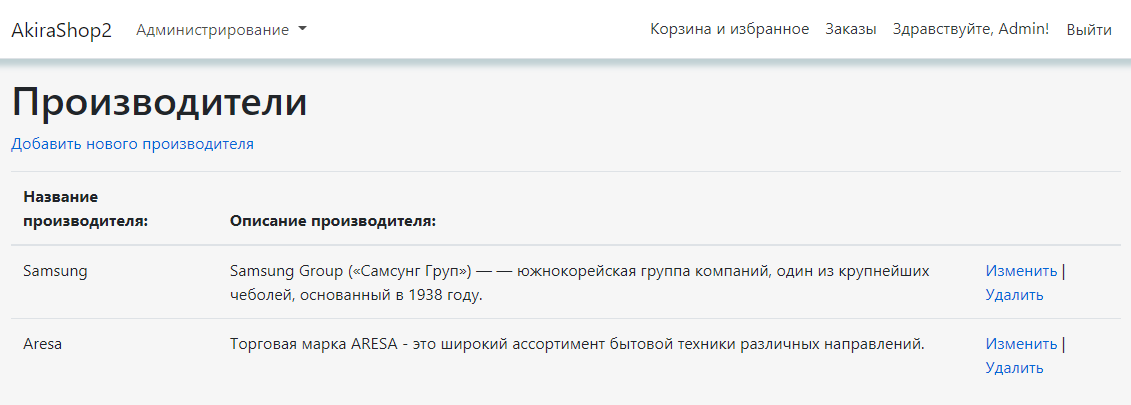


Рисунок 4.4.10 – Список производителей

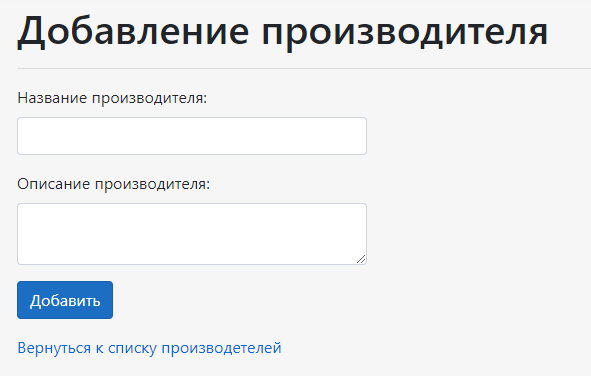


Рисунок 4.4.11 – Страница добавления нового производителя

1. Поиск (Рисунок 4.4.12) – страница с поиском по заданным критериям.

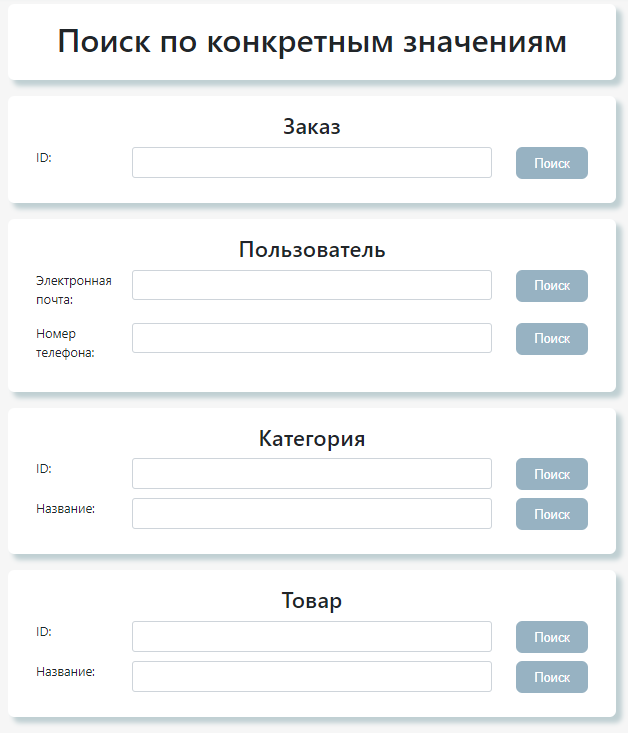


Рисунок 4.4.14 - Поиск по заданным критериям.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной курсовой работы было разработано веб-приложение для продажи бытовой техники.

Работа выполнялась в несколько этапов: была выбрана архитектура, СУБД и разработана структура системы, реализовано веб-приложение.

Были сформированы навыки по разработке и реализации программного приложения с базы данных.

В данной работе выполнены все поставленные задачи.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТУРАТУРЫ

1. <https://ru.wikipedia.org/wiki/DNS_(компания)#:~:text=DNS%20(ООО%20>

1. <https://www.dns-shop.ru>

1. <https://ru.wikipedia.org/wiki/Wildberries>

1. <https://it-doc.info/kak-uznat-nomer-zakaza-na-vajldberriz/>

1. <https://docs.microsoft.com/ru-ru/dotnet/csharp/linq/>

1. <https://tqm.com.ua/likbez/article/pochemu-net-ru>