|  |  |
| --- | --- |
| **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 ***Архитектура автоматизированных систем обработки информации и управления***

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

(подпись) (Ф.И.О.)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Смирнов М.Е. \_\_\_\_\_\_\_\_)

(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 ***Архитектура автоматизированных систем обработки информации и управления***

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

(фамилия, инициалы, индекс группы)

Руководитель Смирнов М.Е.

(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.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ………………. 4](#_Toc89302374)

[1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ 5](#_Toc89302375)

[1.1. Общие сведения 5](#_Toc89302376)

[1.1.1. Полное наименование системы 5](#_Toc89302377)

[1.1.2. Плановые сроки начала и окончания работы по созданию системы 5](#_Toc89302378)

[1.2. Назначение и цели создания (развития) системы 5](#_Toc89302379)

[1.2.1. Назначение системы 5](#_Toc89302380)

[1.2.2. Цели создания системы 5](#_Toc89302381)

[1.3. Характеристики объекта автоматизации 5](#_Toc89302382)

[1.4. Требования к системе 6](#_Toc89302383)

[1.4.1. Требования к системе в целом 6](#_Toc89302384)

[1.4.2. Требования к структуре и функционированию системы 6](#_Toc89302385)

[1.4.3. Требования к надежности 6](#_Toc89302386)

[1.5. Состав и содержание работ по созданию системы 6](#_Toc89302387)

[1.6. Порядок контроля и приемки системы. 7](#_Toc89302388)

[1.6.1. Состав, объем и методы испытаний системы и ее составных частей. 7](#_Toc89302389)

[1.6.2. Общие требования к приемке работ. 8](#_Toc89302390)

[1.7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие. 8](#_Toc89302391)

[1.8. Требования к документированию 8](#_Toc89302392)

[1.9. Источники разработки 8](#_Toc89302393)

[2. ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ 10](#_Toc89302394)

[2.1. Постановка задачи проектирования. 10](#_Toc89302395)

[2.2. Описание предметной области 10](#_Toc89302396)

[2.3. Анализ аналогов 12](#_Toc89302397)

[2.4. Обоснование выбора инструментов и платформы для разработки. 14](#_Toc89302398)

[3. ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ 16](#_Toc89302399)

[3.1. Разработка структуры приложения. 16](#_Toc89302400)

[3.2. Разработка алгоритмов обработки информации 18](#_Toc89302401)

[3.3. Разработка интерфейса взаимодействия пользователя с системой 20](#_Toc89302402)

[3.4. Разработка архитектуры приложения. 26](#_Toc89302403)

[4. ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 27](#_Toc89302404)

[4.1. Технологические решения, поддерживающие эксплуатационный цикл программы. 27](#_Toc89302405)

[4.2. Порядок развёртывания системы. 28](#_Toc89302406)

[4.3. Разработка руководства пользователя и руководства администратора. 29](#_Toc89302407)

[ЗАКЛЮЧЕНИЕ…………………………… 33](#_Toc89302408)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТУРАТУРЫ 34](#_Toc89302409)

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 (сервер).

- 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.  . 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:

* Обеспечение работ по вводу данных в систему
* Развертывание системы на сервере.
* 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 UML диаграмма прецедентов администратора (Рисунок 2.1.0) и пользователя (Рисунок 2.1.1).



Рисунок 2.1.0 – UML диаграмма прецедентов администратора



Рисунок 2.1.1 – UML диаграмма прецедентов пользователя

* 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.

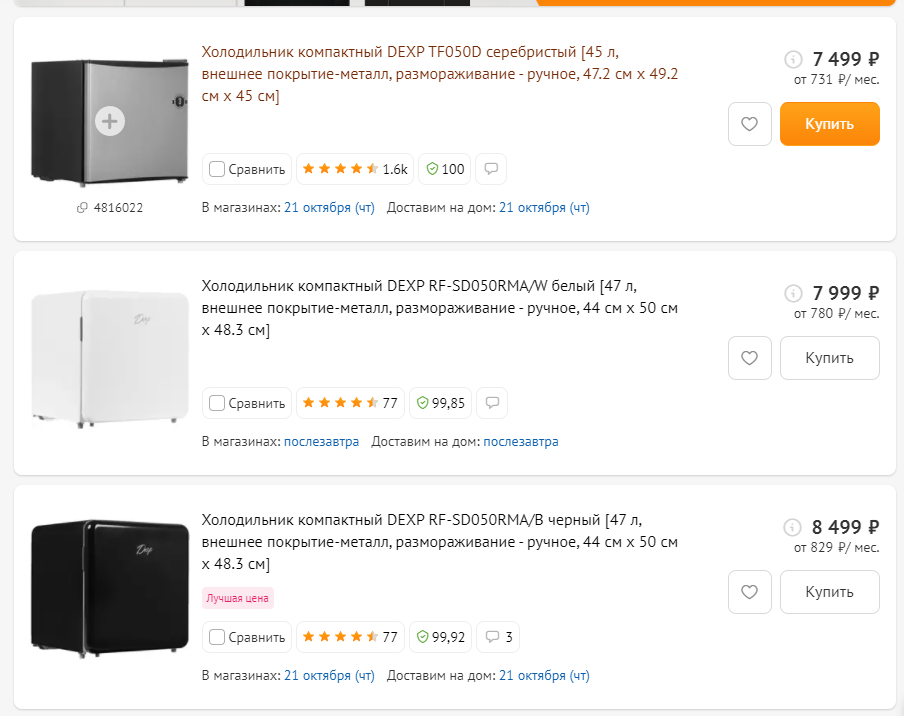


Рисунок 2.2 – Пример списка товаров в интернет-магазине DNS [[2]](#литра2)

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).

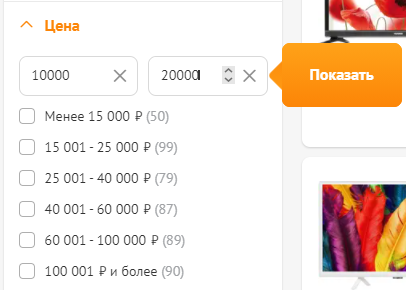


Рисунок 2.3 – Фильтрация по цене (имеется и диапазон и список значений)[[2]](#литра2)

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.

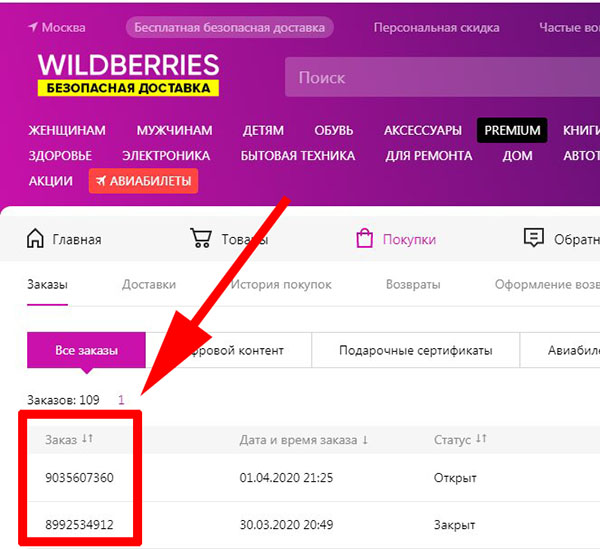


Рисунок 2.4 – Список заказов Wildberries[[4]](#литра4)

* 1. Обоснование выбора инструментов и платформы для разработки.

Для разработки системы был выбрана среда разработки Visual Studio с кросс-платформенным фреймворком ASP.NET Core с использованием Entity Framework.

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 трехзвенную клиент-серверную архитектуру с использованием MVC (Model-View-Controller) (Рисунок 3.1).



Рисунок 3.1 – MVC архитектура

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. 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).

Основные компоненты серверной части приложения (MVC):

* Модель – объектная модель предметной области (например: модель Заказа), включает в себя данные и методы работы с этими данными, реагирует на запросы из контроллера, возвращая данные и/или изменяя своё состояние. При этом модель не содержит в себе информации о способах визуализации данных или форматах их представления, а также не взаимодействует с пользователем напрямую.
* Представление – отвечает за отображение информации (визуализацию). Одни и те же данные могут представляться различными способами и в различных форматах. Например, коллекцию объектов при помощи разных представлений можно представить на уровне пользовательского интерфейса как в табличном виде, так и списком (как, например, представление товаров категории для администратора и для пользователя).
* Контроллер – обеспечивает связь между пользователем и системой, использует модель и представление для реализации необходимой реакции на действия пользователя. Как правило, на уровне контроллера осуществляется фильтрация полученных данных и авторизация — проверяются права пользователя на выполнение действий или получение информации.

Приложением-клиентом, в следствии такой архитектуры, является веб-браузер, что дает множество плюсов:

* Кроссплатформенность. Для взаимодействия с приложением не требуется дополнительное ПО, нужен только веб-браузер, который существует на всех операционных системах (в том числе и мобильных).
* Простота. Пользователям намного проще разобраться с веб-приложением, чем, например, с десктопным приложением.
* Удобная поддержка приложения. Так как по сути все приложение находится на сервере, то пользователям не нужно вручную устанавливать или ждать обновления.
  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. 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.

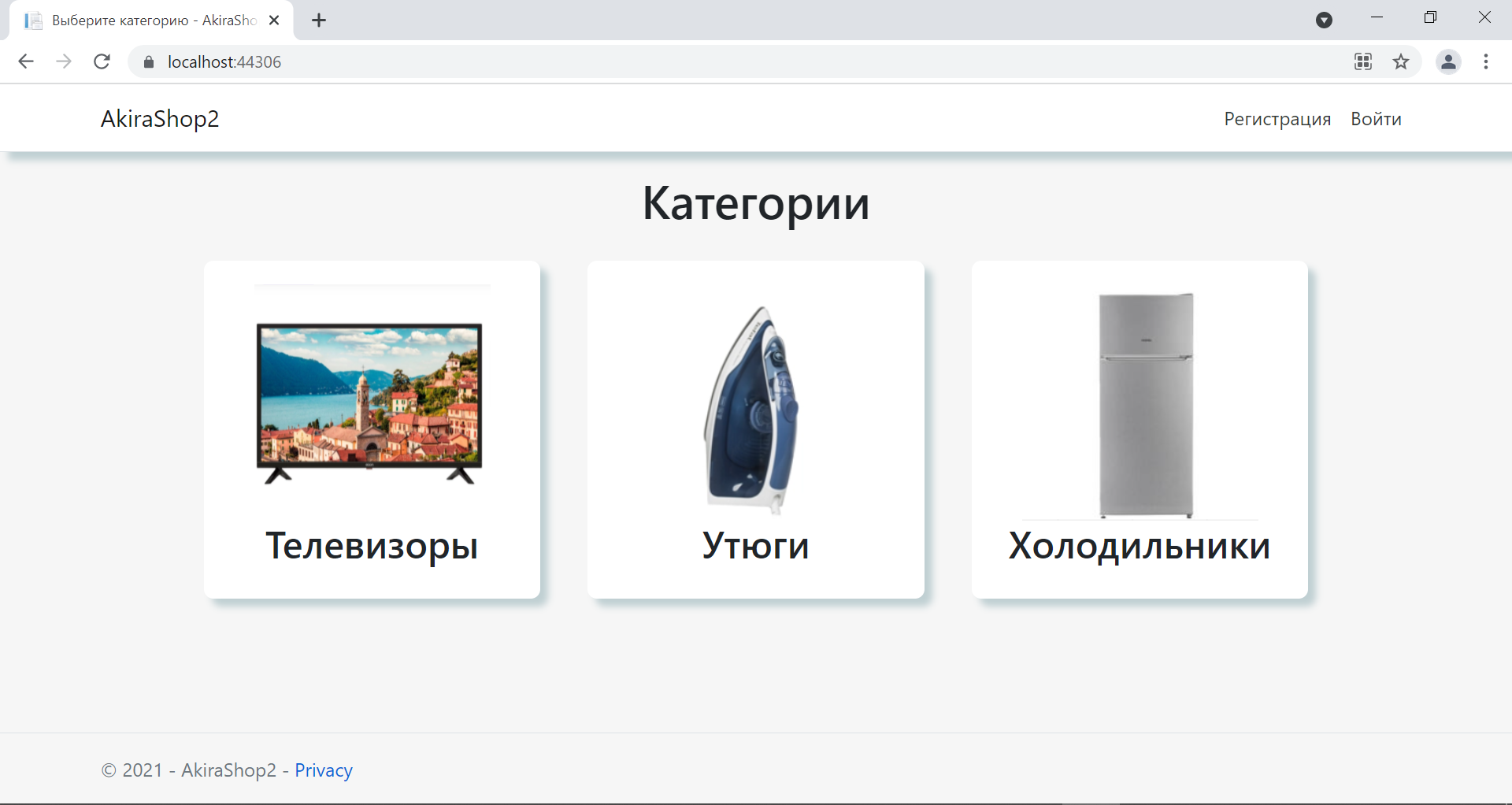


Рисунок 3.4.1 – Категории товаров

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

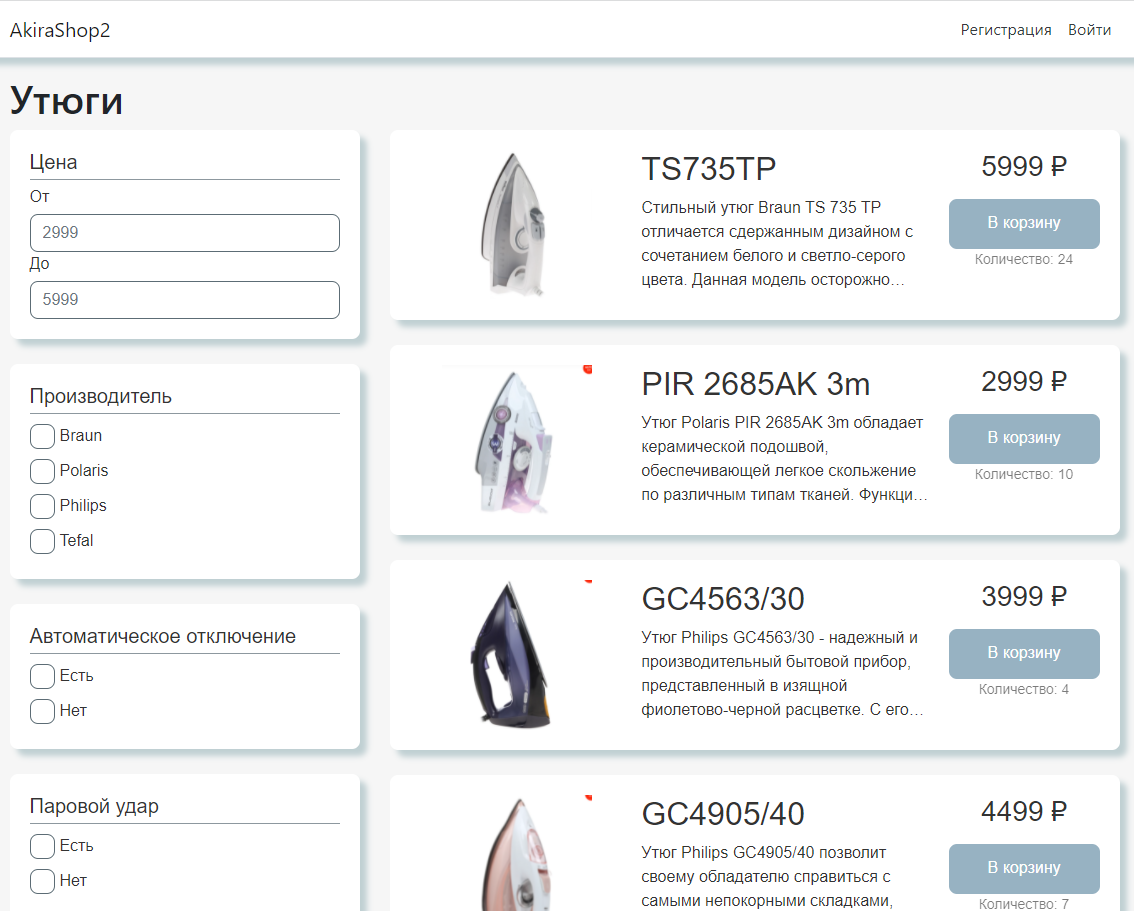


Рисунок 3.4.2 – Список товаров категории “Утюги”

* Страница отображения информации о товаре (Рисунок 3.4.3)

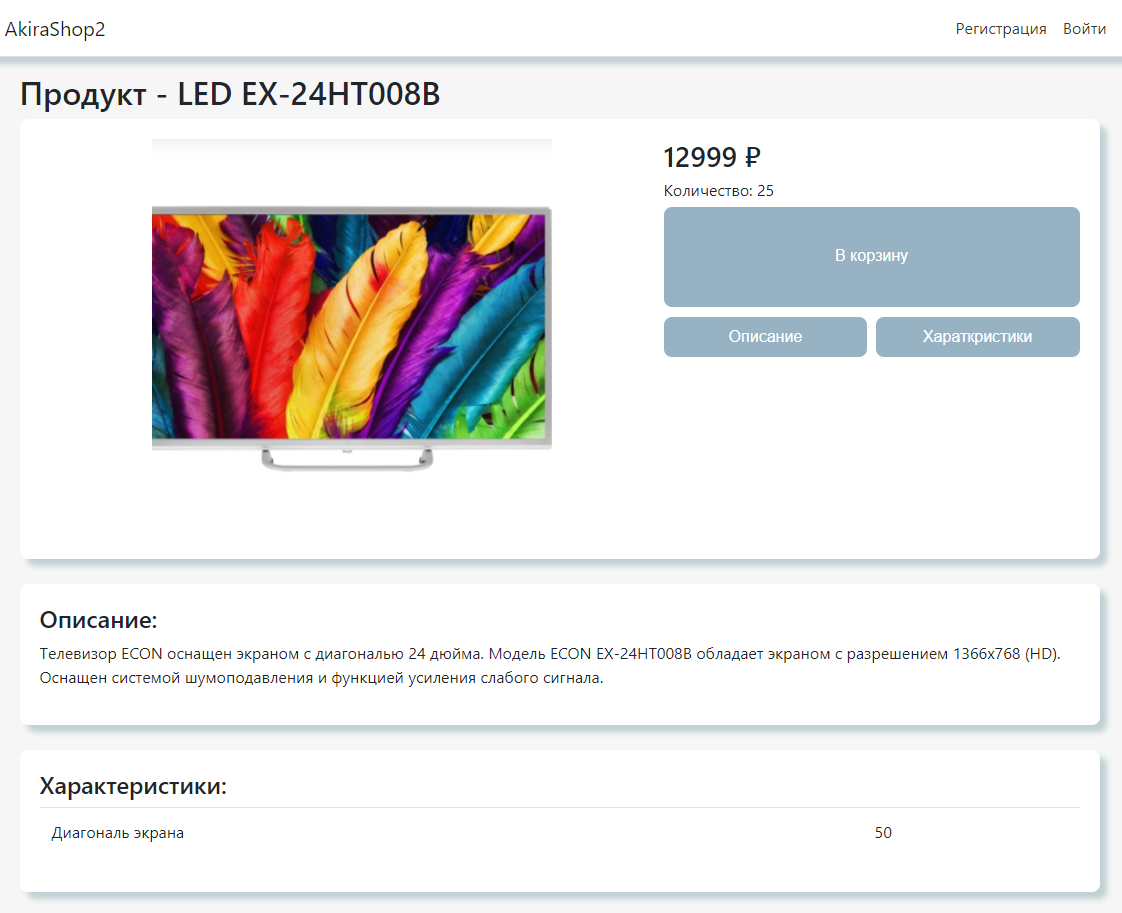


Рисунок 3.4.3 – Информацией о товаре

* Страница для просмотра корзины и избранного (Рисунок 3.4.4). Аналогично выглядит страница готовых заказов. Доступно любому авторизованному пользователю.

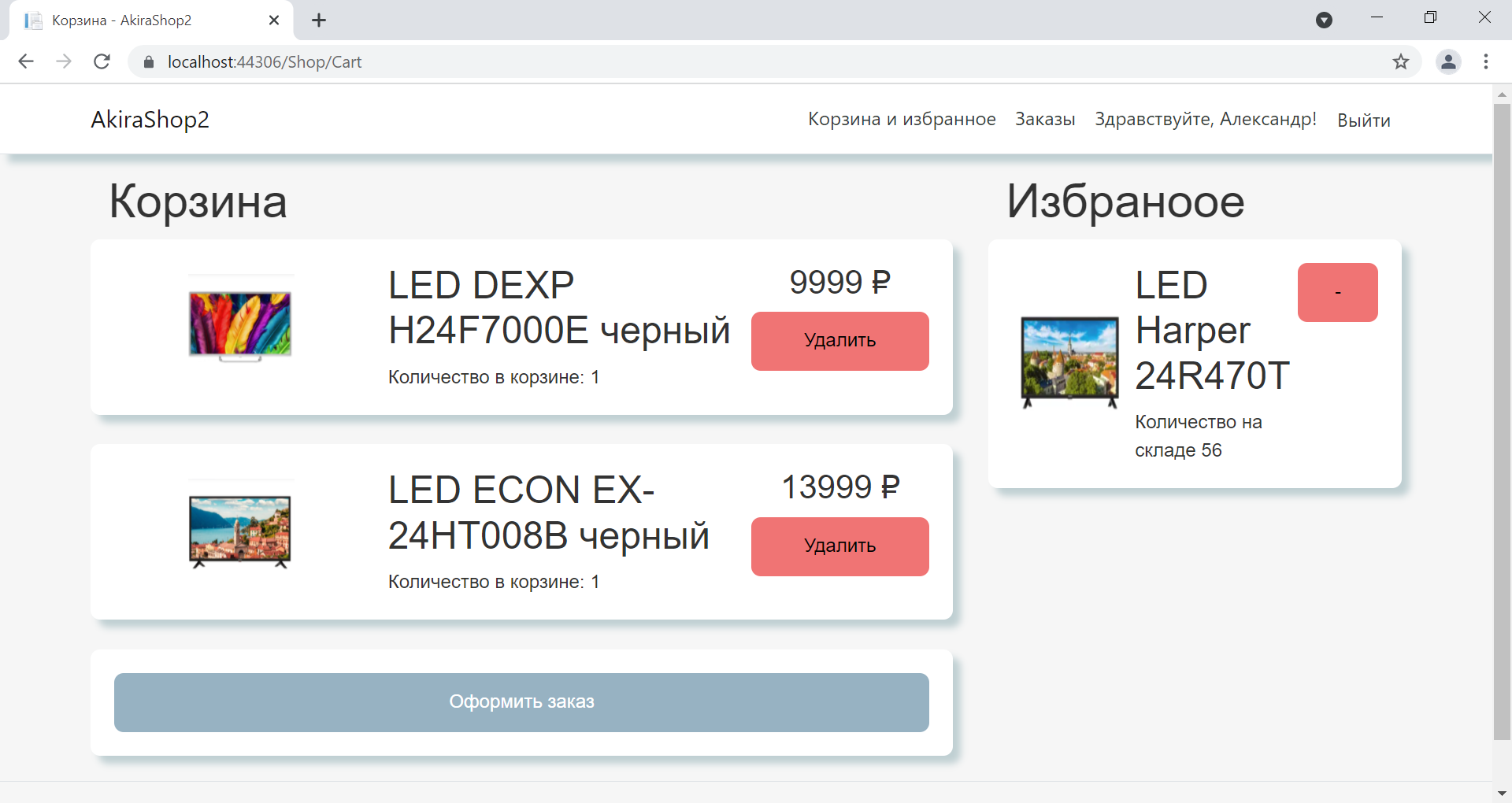


Рисунок 3.4.4 – Корзина и избранное

* Модальная страница формы авторизации пользователя (Рисунок 3.4.5) и регистрации (Рисунок 3.4.6). Данная страница появляется поверх основных страниц.

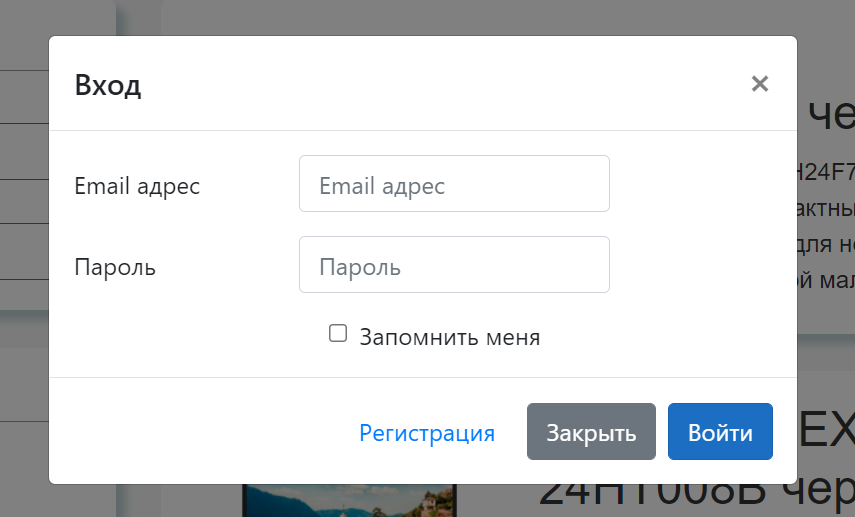


Рисунок 3.4.5 – Форма авторизации пользователя

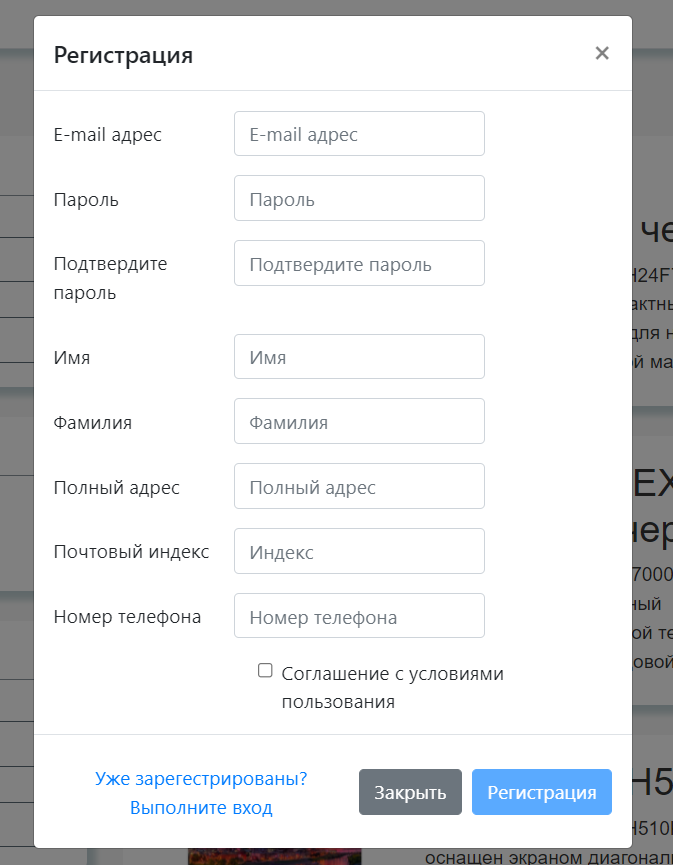


Рисунок 3.4.6 – Форма регистрации пользователя

* Страница добавления, изменения категории (Рисунок 3.4.7) и товаров (Рисунок 3.4.8). Страница производителей - аналогична предыдущим. Данные страницы доступны администратору.

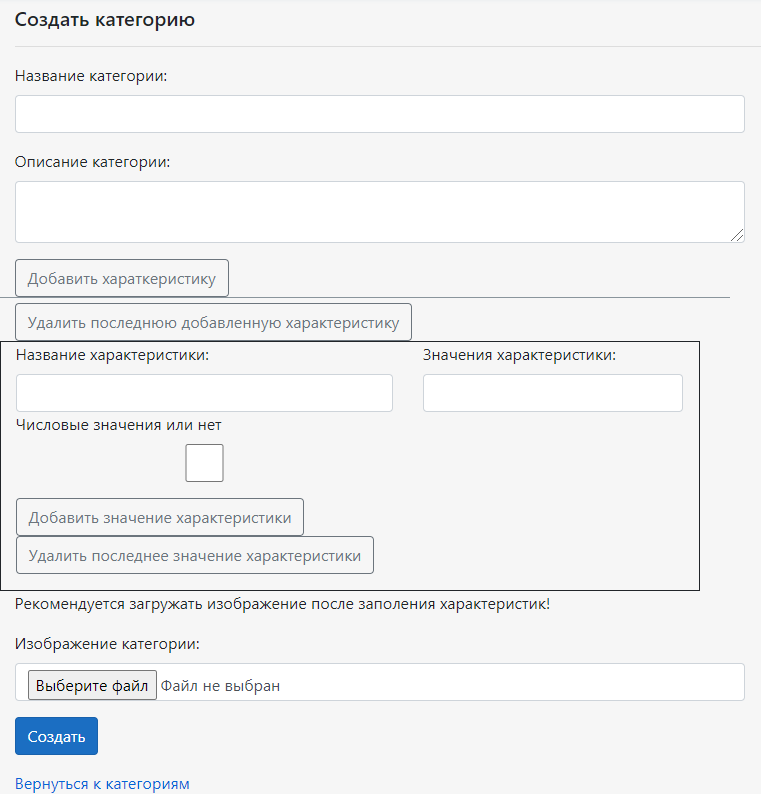


Рисунок 3.4.7 – Страница создания/изменения категории

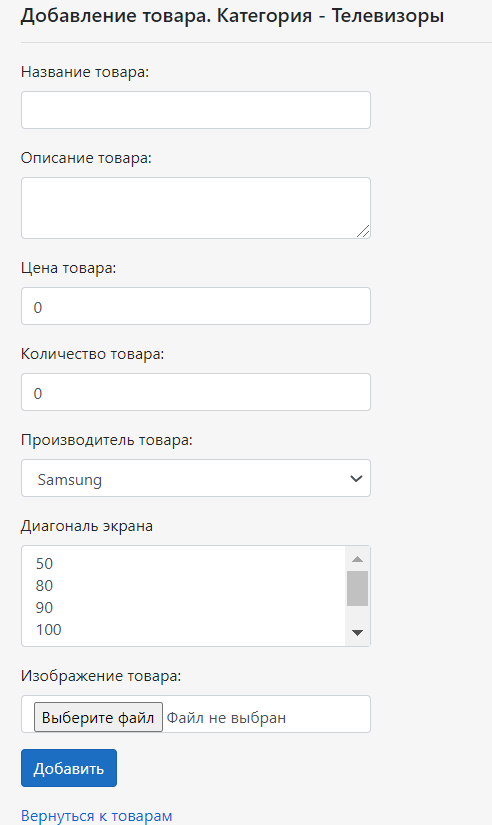


Рисунок 3.4.8 – Страница создания/изменения товара категории “Телевизоры”

* Страница просмотра товаров категории (Рисунок 3.4.9). Страницы просмотров категории, пользователей, производителей и заказов аналогичны. Данные страницы доступны администратору.

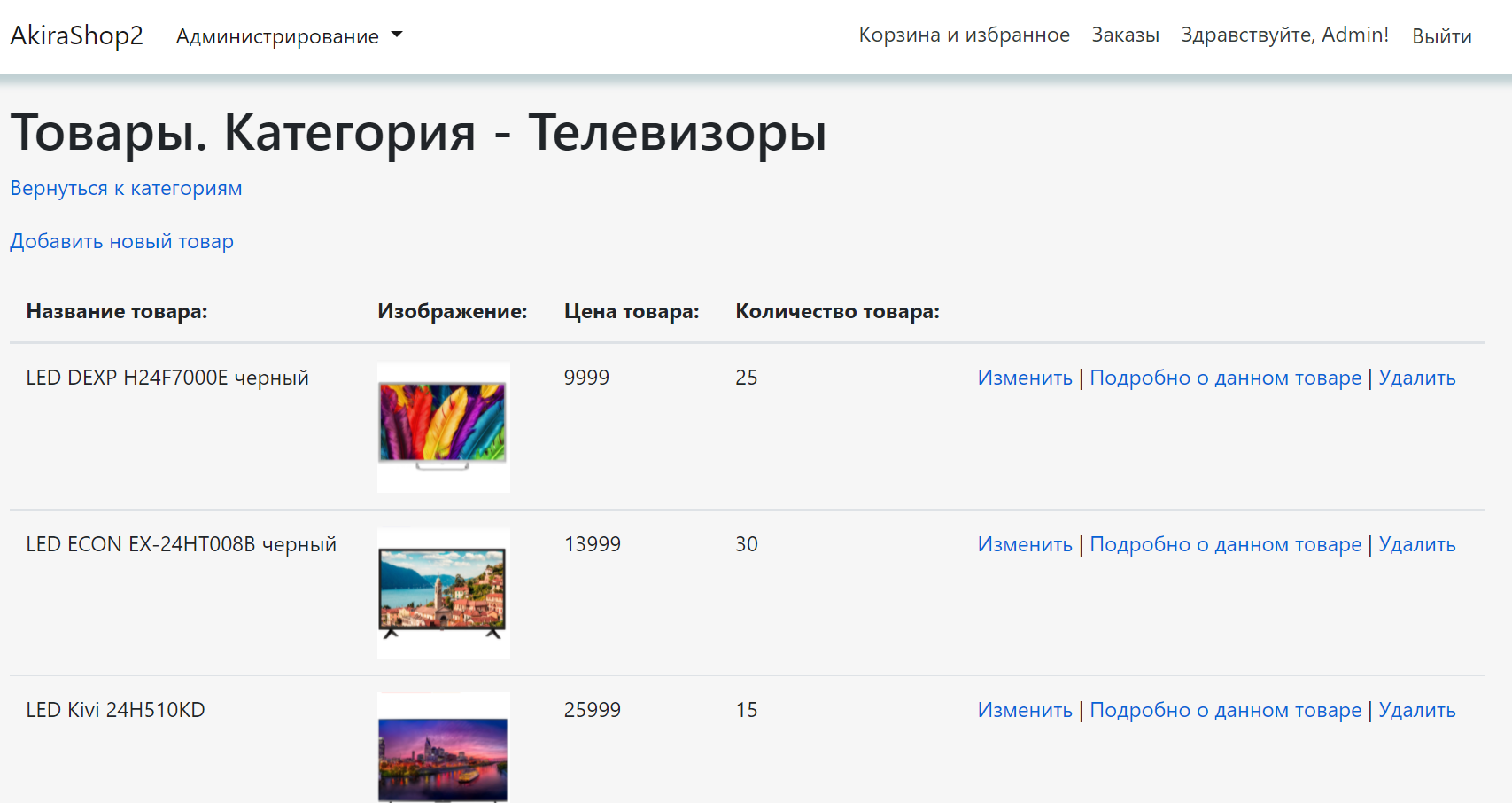


Рисунок 3.4.9 – Страница просмотра товаров категории “Телевизоры”

* Страница поиска по конкретным значениям (Рисунок 3.4.10). Доступна администратору.

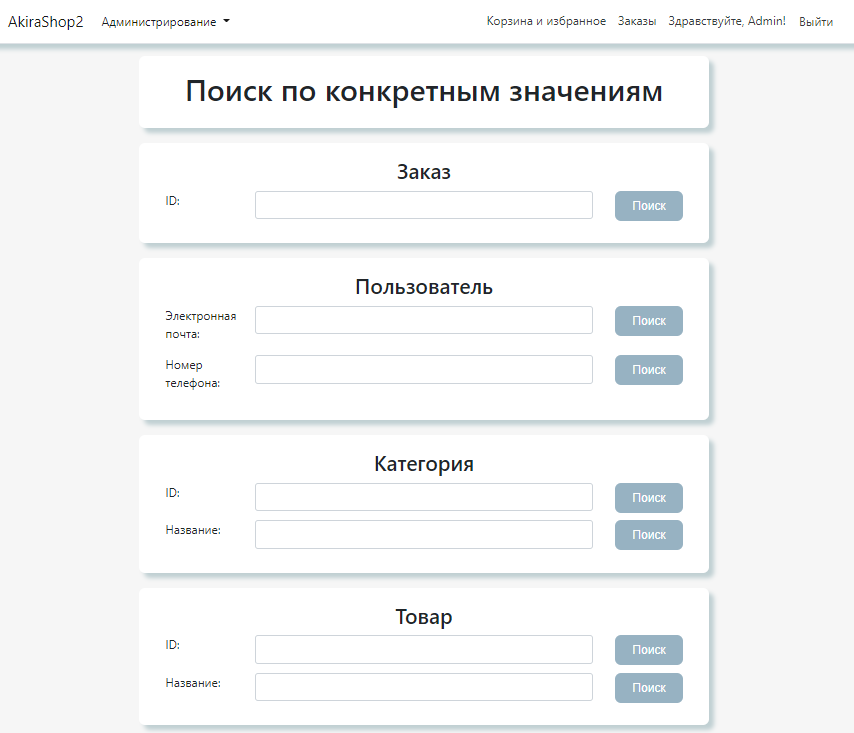


Рисунок 3.4.10 – Поиск по конкретным значениям

* Страница изменения пользовательской информации (Рисунок 3.4.11). Доступна любому авторизованному пользователю.

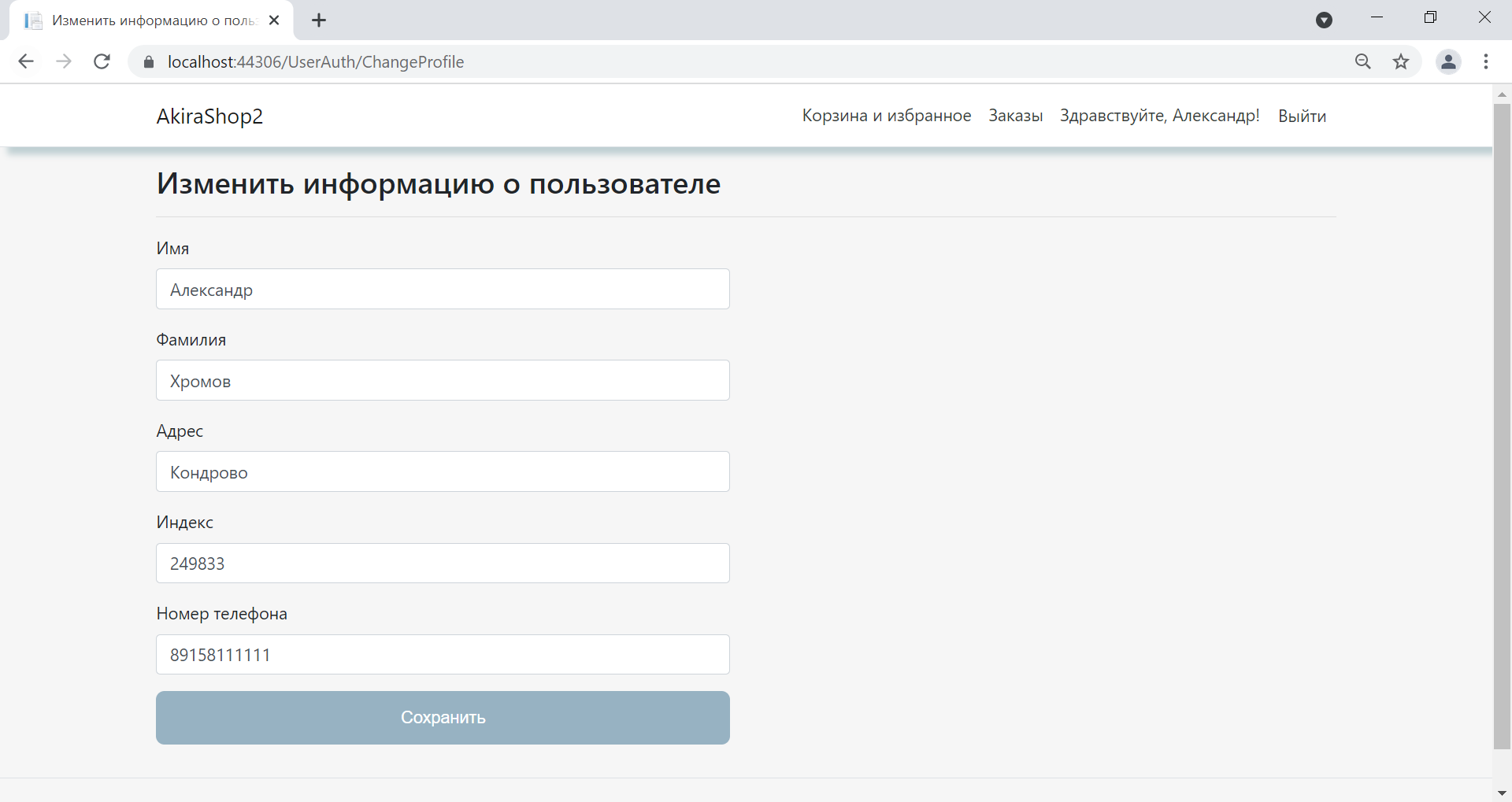


Рисунок 3.4.11 – Изменение пользовательской информации

* 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.

Передача информации между компонентами внутри приложения осуществляется, как уже говорилось, с использованием 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 (Рисунок 3.4.12).



Рисунок 3.4.13 – Модель поведения PRG

Так же стоит отметить использование модели асинхронного программирования (async await), что позволяет серверу делать асинхронные запросы к базе данных, ускоряя отклик сервера.

1. ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ
   1. 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).

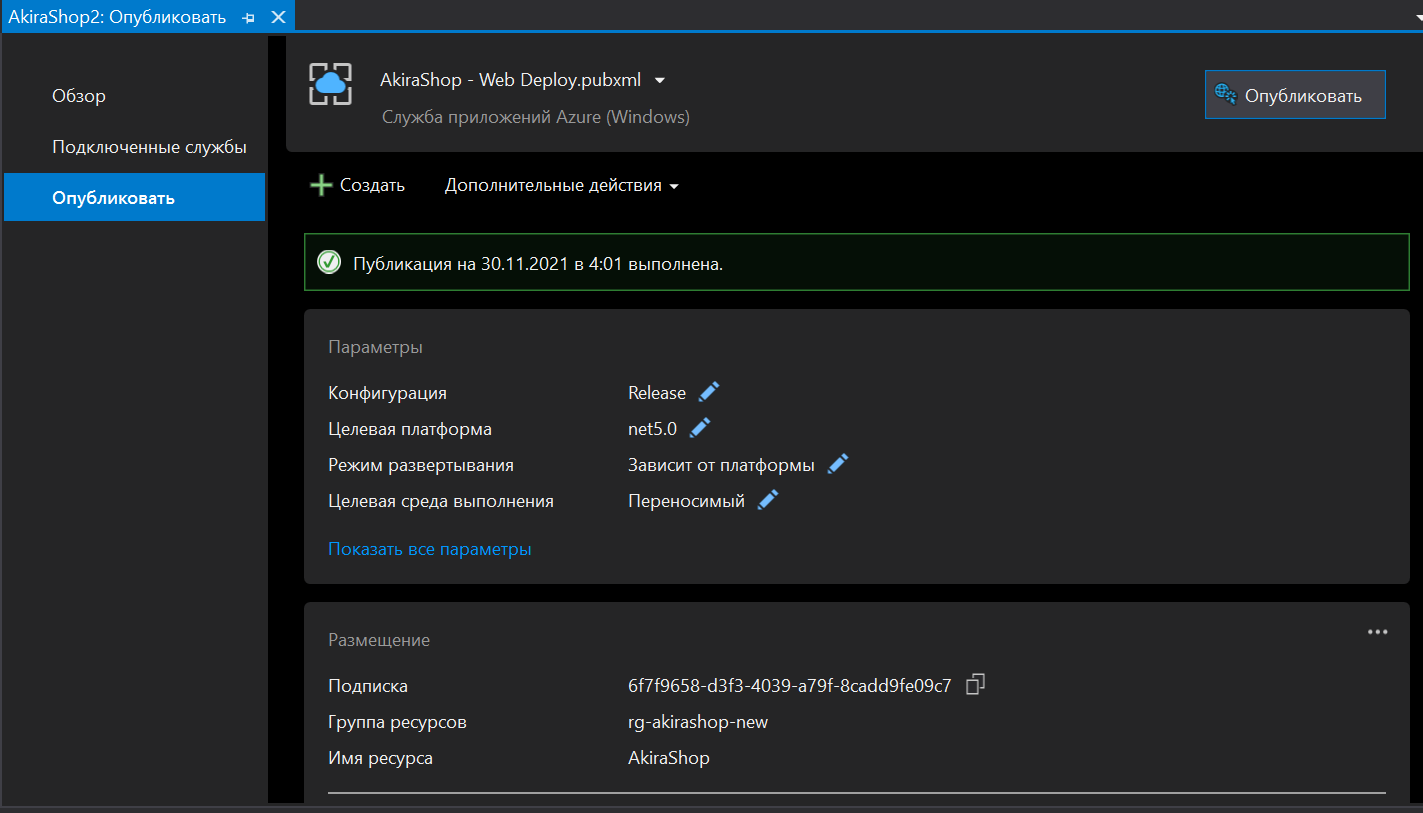


Рисунок 4.3.1 – Профиль опубликованного приложения

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.

**Руководство пользователя:**

1. На домашней странице сайта (Рисунок 4.4.13) представлен список категорий. В верхней части сайта есть кнопка для регистрации и для авторизации.

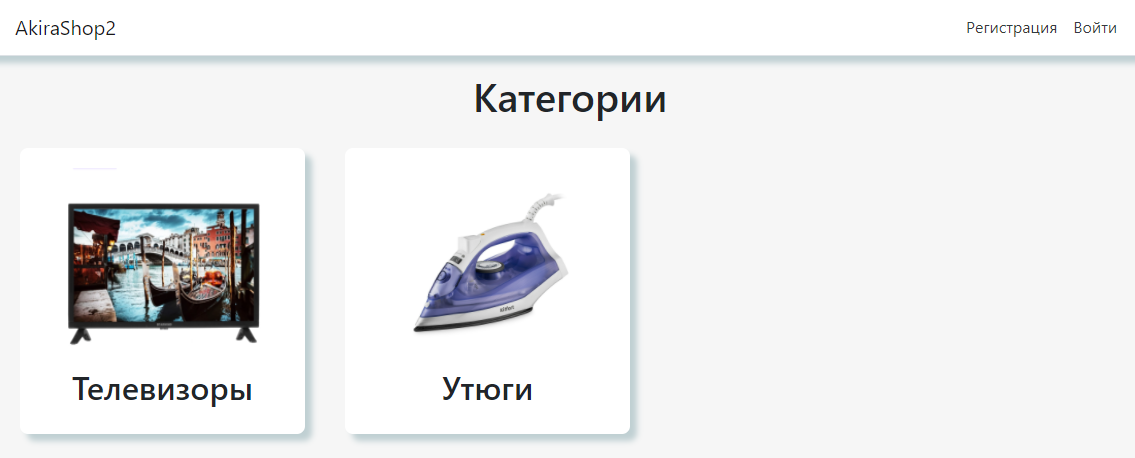


Рисунок 4.4.13 – Домашняя страница

1. Для просмотра товаров категории следует нажать на нее на домашней странице. После нажатия открывается страница с товарами и фильтрами (Рисунок 4.4.14). Фильтры бывают двух типов: числовые и фильтры с конкретными значениями. Для применения фильтра следует нажать на кнопку “Применить фильтр”, находящуюся под списком фильтров. Правее фильтров находятся товары, подходящие выбранным фильтрам (или все товары данной категории, если ни один фильтр не активирован). Чтобы просмотреть полную информацию о товаре и добавить его в корзину и/или в избранное, следует нажать на любую область товара. После этого откроется страница с подробной информацией о товаре (Рисунок 4.4.15). Для добавления товара в корзину требуется авторизация. Ее можно произвести нажатием на кнопку “Войти” в верхней части сайта, или нажатием на кнопку “В корзину” любого из товара. После этого есть возможность, нажатием на ту же кнопку, добавлять товар в корзину.

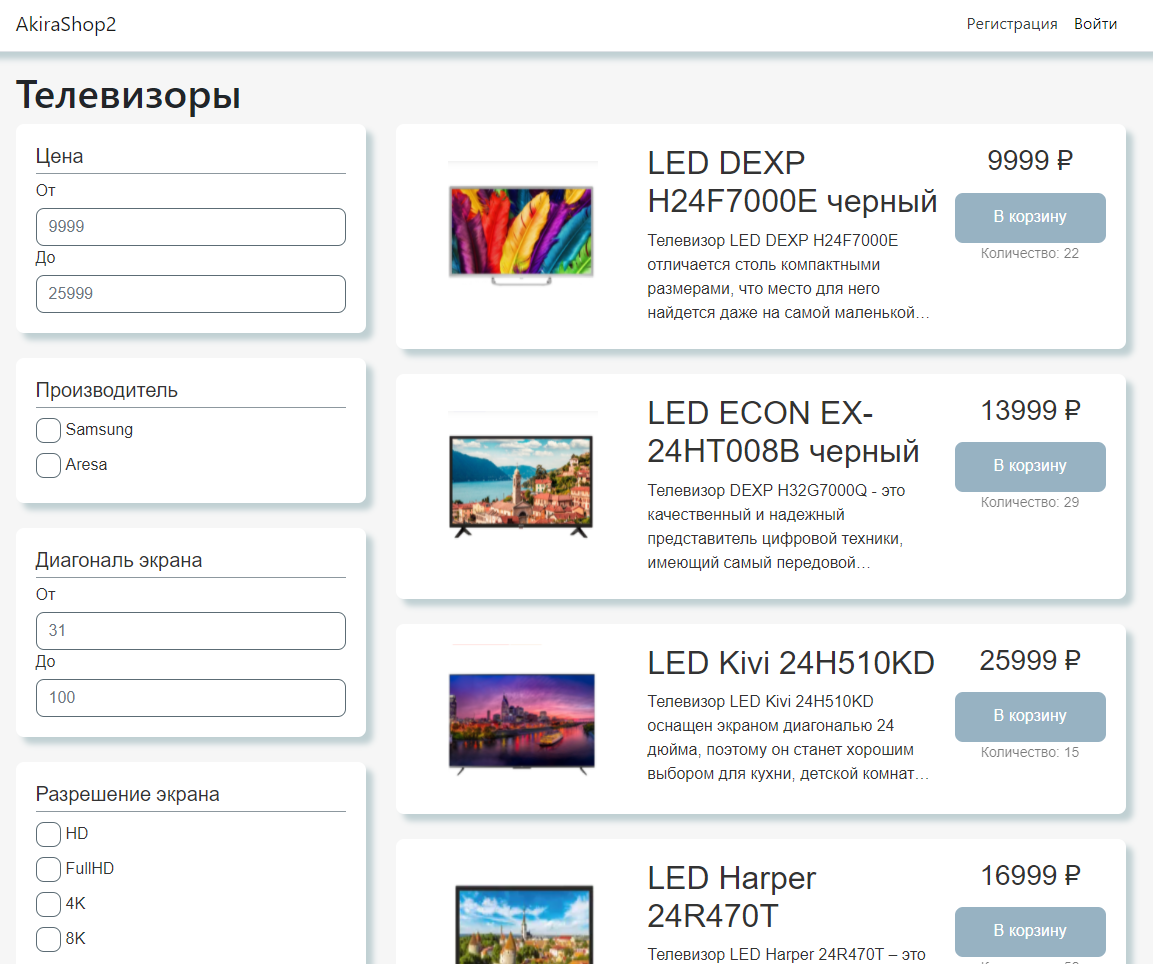


Рисунок 4.4.14 - Страница с товарами и фильтрами

1. После авторизации в верхней части сайта можно перейти на страницу с корзиной, нажатием на кнопку “Корзина и избранное” (Рисунок 4.4.14). На данной странице слева есть раздел с корзиной, в которой можно удалять товары, либо оформить заказ. Справа находится раздел с избранным, в нее может быть включен товар даже которого нет на складе.

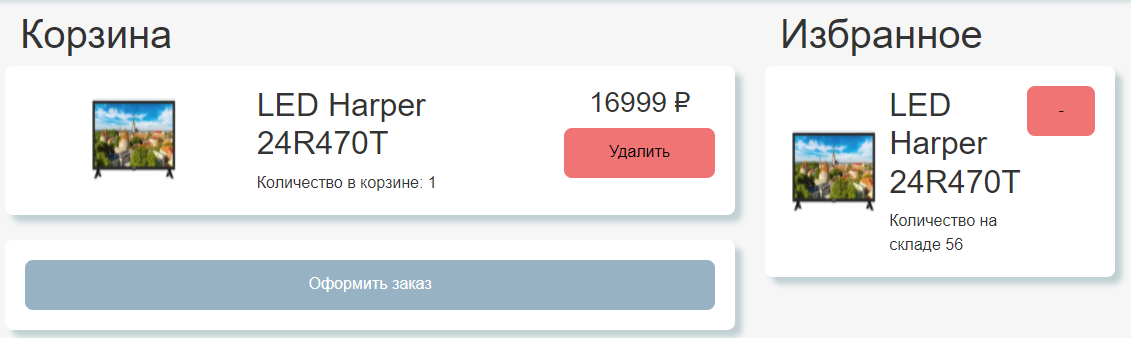


Рисунок 4.4.15 – Страница с корзиной и избранным

1. После нажатия на кнопку “Оформить заказ” откроется страница с подтверждением заказа (Рисунок 4.4.16). На данной странице можно перейти на странице изменения информации о пользователе (Рисунок 4.4.17), нажав на соответствующую кнопку, а также ввести дополнительную информацию к заказу, и оплатить заказ. Просмотреть свои заказы можно, нажав на кнопку “Заказы” в верхней части сайта (Рисунок 4.4.18).



Рисунок 4.4.15 - Страница с подробной информацией о товаре

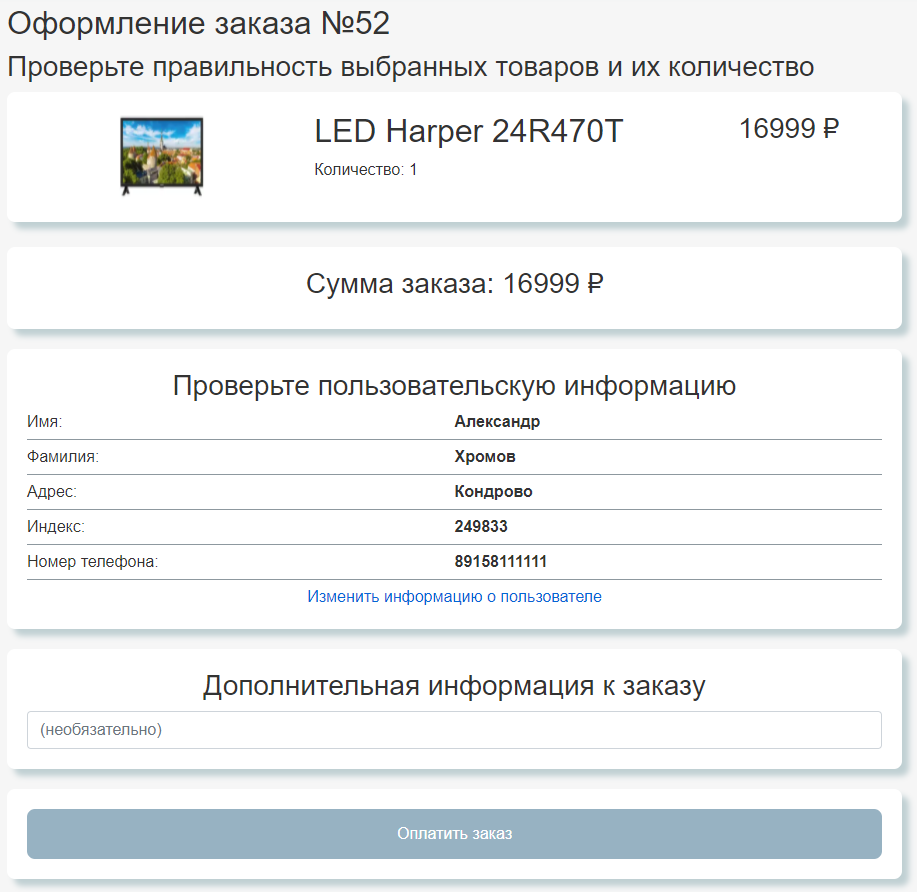


Рисунок 4.4.16 - Страница с подтверждением заказа

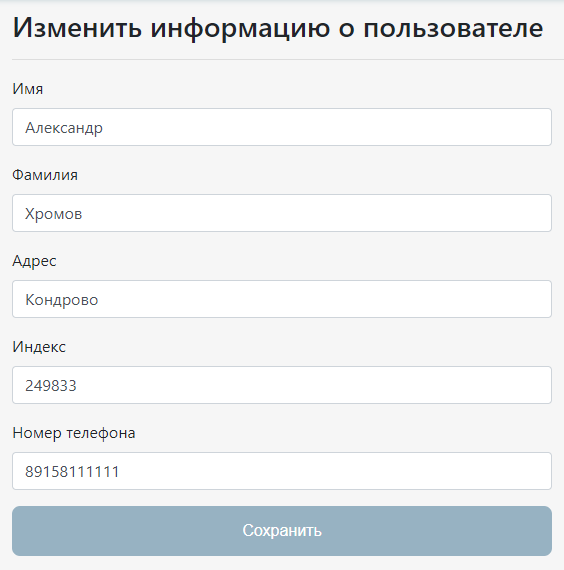


Рисунок 4.4.17 - Страница изменения информации о пользователе

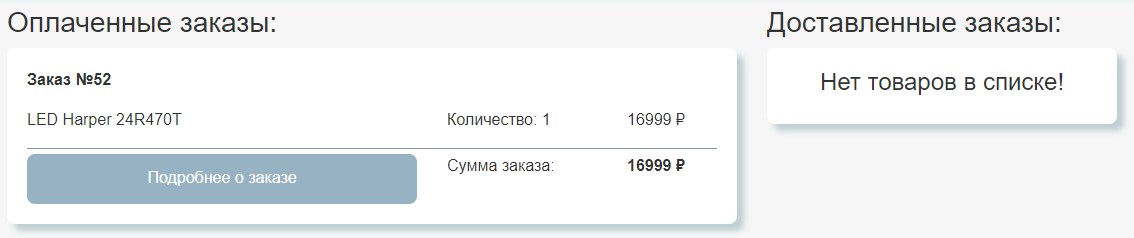


Рисунок 4.4.18 – Страница с заказами

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>
2. Шнырёв, С.Л. Базы данных: учебное пособие для вузов - М. : НИЯУ МИФИ, 2011. — 224 с. — Режим доступа: <http://e.lanbook.com/book/75809>
3. Ревунков, Г.И. Базы и банки данных - М.: МГТУ им. Н.Э. Баумана, 2011. — 68 с. — Режим доступа: <http://e.lanbook.com/book/52425>
4. Ревунков, Г.И. Проектирование баз данных - М.: МГТУ им. Н.Э. Баумана, 2009. — 20 с. — Режим доступа: <http://e.lanbook.com/book/52390>
5. Кудрявцев, К.Я. Создание баз данных: учебное пособие — М.: НИЯУ МИФИ, 2010. — 155 с. — Режим доступа: <http://e.lanbook.com/book/75822>
6. Сидоров В.Н., Сломинская Е.Н., Полникова Т.В., Макарова О.Ю. Оформление графической части выпускной квалификационной работы. Учебное пособие. М.: МГТУ им. Н.Э. Баумана, 2016.