Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский политехнический университет»

Кафедра ««Инфокогнитивные технологии»

Образовательная программа «Веб-технологии»

Отчет по курсовому проекту

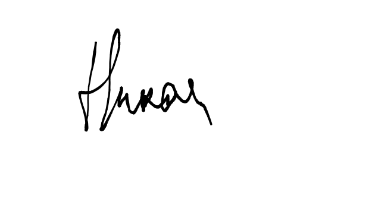
по дисциплине «Основы серверной веб-разработки»

Тема: «Сервисный центр по ремонту техники»

Выполнил:

Студент группы 221-321

Николаева А.И.



03.07.2023

Принял:

Старший преподаватель

Никишина И.Н.

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

03.07.2023

Москва 2023

Содержание

[**Введение** 2](#_Toc139220170)

[**1. Анализ аналогов** 4](#_Toc139220171)

[**1.1 Отечественные и зарубежные сайты** 4](#_Toc139220172)

[**1.2 Анализ структуры сайтов** 6](#_Toc139220173)

[**1.3 Юзабилити и функциональность сайтов** 9](#_Toc139220174)

[**2. Проектирование** 11](#_Toc139220175)

[**2.1 Обзор кейсов использования** 11](#_Toc139220176)

[**2.2 Выбор функционала** 13](#_Toc139220177)

[**3. Разработка и тестирование** 14](#_Toc139220178)

[**3.1 Архитектура проекта** 14](#_Toc139220179)

[**3.2 Структура базы данных** 20](#_Toc139220180)

[**3.3 Типовые запросы СУБД** 23](#_Toc139220181)

[**Заключение** 24](#_Toc139220182)

[**Список использованных источников** 25](#_Toc139220183)

[**Приложение 1** 26](#_Toc139220184)

[**Приложение 2** 26](#_Toc139220185)

[**Приложение 3** 27](#_Toc139220186)

[**Приложение 4** 27](#_Toc139220187)

[**Приложение 5** 28](#_Toc139220188)

# **Введение**

В современном обществе техника и технологии стали неотъемлемой частью повседневной жизни людей. От смартфонов и компьютеров до бытовой техники и автомобилей, люди зависят от множества устройств. Однако ни одно техническое устройство не застраховано от возможных поломок или сбоев в работе. В таких случаях сервисные центры по ремонту техники играют важную роль, предоставляя необходимую помощь и поддержку.

Сервисные центры по ремонту техники выполняют важные функции в обеспечении технического обслуживания и восстановления работоспособности различных устройств. Они предлагают услуги по диагностике, ремонту и замене неисправных компонентов, а также предоставляют гарантийное и послегарантийное обслуживание. Качество и эффективность работы сервисного центра непосредственно зависят от системы управления данными и процессами, используемых в его деятельности.

Целью данного проекта является разработка базы данных и веб-приложения для сервисного центра по ремонту техники, которые позволят оптимизировать и улучшить работу центра. База данных будет служить основой для хранения и управления информацией о клиентах, устройствах, заказах, сотрудниках и запчастях. Это позволит сервисному центру эффективно отслеживать и обрабатывать данные, управлять запасами и обеспечивать высокий уровень обслуживания.

Для разработки веб-приложения будет использоваться фреймворк Django, который является одним из наиболее популярных инструментов для создания веб-приложений на языке программирования Python. Django предоставляет широкие возможности для разработки функциональности веб-приложений, включая создание моделей данных, управление маршрутизацией, реализацию аутентификации и авторизации пользователей. Кроме того, Django предлагает готовую административную панель, которая упрощает работу с базой данных и управлением данными.

Анализ существующих сервисных центров и их подходов к управлению данными позволит выявить сильные и слабые стороны этих систем. Используя лучшие практики и учитывая опыт других центров, разработанный веб-проект будет нацелен на повышение эффективности работы сервисного центра и улучшение качества предоставляемых услуг.

В данной курсовой работе будет выполнен проект базы данных и веб-приложения для сервисного центра по ремонту техники с использованием фреймворка Django. База данных будет состоять из шести таблиц, содержащих не менее 10 записей в каждой, что обеспечит достаточный объем данных для анализа и проверки функциональности разработанного веб-приложения. Проект будет предоставлять возможности управления клиентами, устройствами, заказами и другой информацией, необходимой для эффективной работы сервисного центра.

Разработка проекта базы данных и веб-приложения для сервисного центра по ремонту техники имеет большое значение для оптимизации работы сервисных центров, улучшения качества обслуживания клиентов и повышения их удовлетворенности.

# **1. Анализ аналогов**

## **1.1 Отечественные и зарубежные сайты**

В настоящее время все больше людей обращаются к сервисным центрам для ремонта и обслуживания своей техники. Интернет-платформы сервисных центров становятся неотъемлемой частью этого процесса, предоставляя пользователям возможность быстро и удобно получить необходимые услуги. В рамках данного анализа были рассмотрены два сайта: отечественный <https://service-engineering.ru/> и зарубежный <https://theservicecentreuk.com/>.

Сайт <https://service-engineering.ru/> - отечественный ресурс, предоставляющий информацию о сервисном центре, специализирующемся на ремонте и обслуживании различных устройств. На этом сайте пользователи могут ознакомиться с перечнем услуг, предлагаемых сервисным центром, а также получить информацию о доступных запчастях для ремонта. Он ориентирован на клиентов в Москве, предлагая им удобный способ найти надежный сервисный центр и получить качественные услуги.

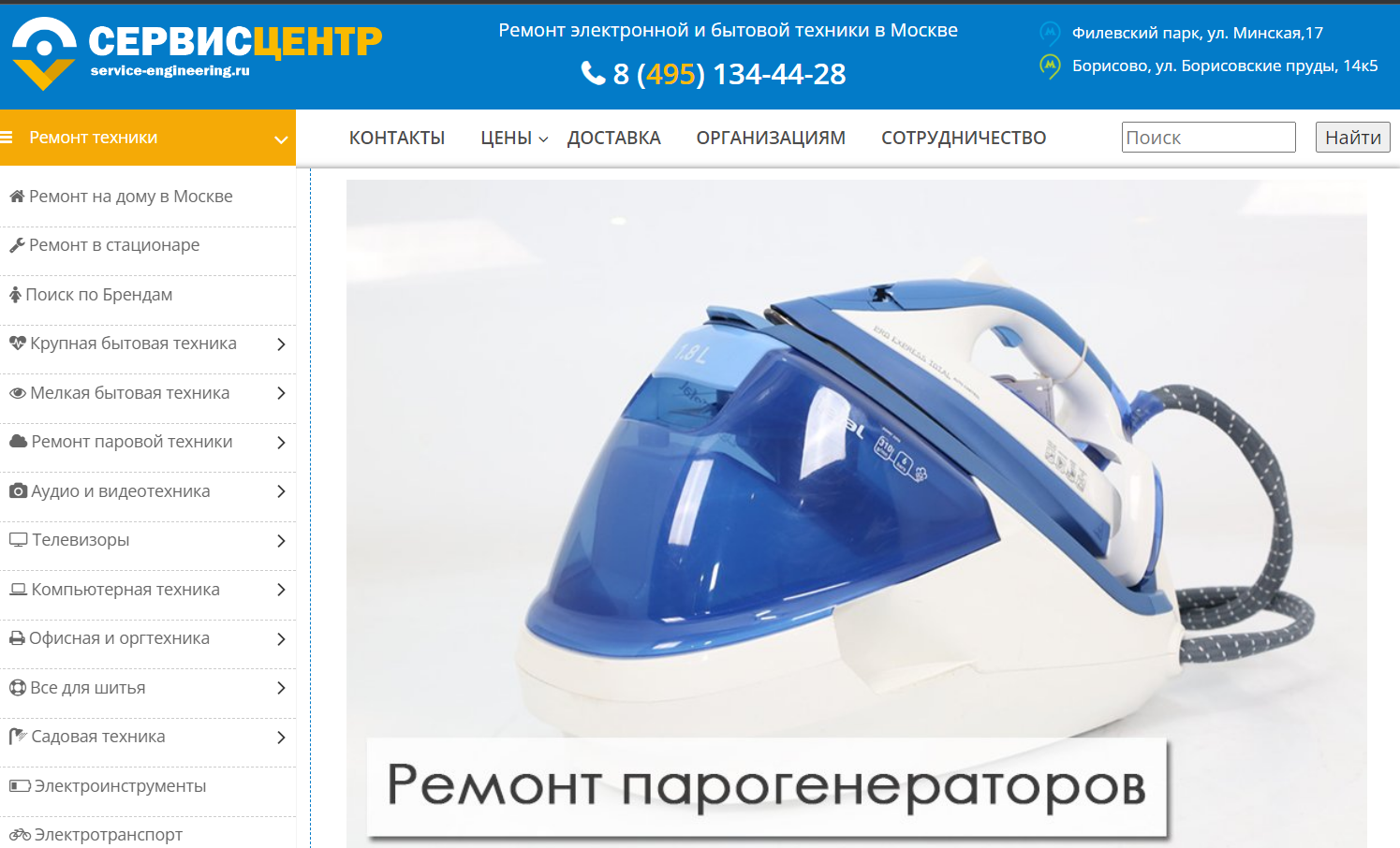


Рисунок 1 – Аналог отечественного сайта “СервисЦентр”

Сайт <https://theservicecentreuk.com/> - зарубежная онлайн-платформа, предоставляющая услуги по ремонту и обслуживанию техники в Великобритании. Он создан для удобства пользователей, которые ищут надежный сервисный центр на территории страны. На этом сайте можно оформить заявку на ремонт, а также отслеживать статус выполнения заказа. Помимо этого, сайт предлагает подробную информацию о сервисных центрах, их расположении и контактных данных, что обеспечивает прозрачность и удобство взаимодействия с клиентами.

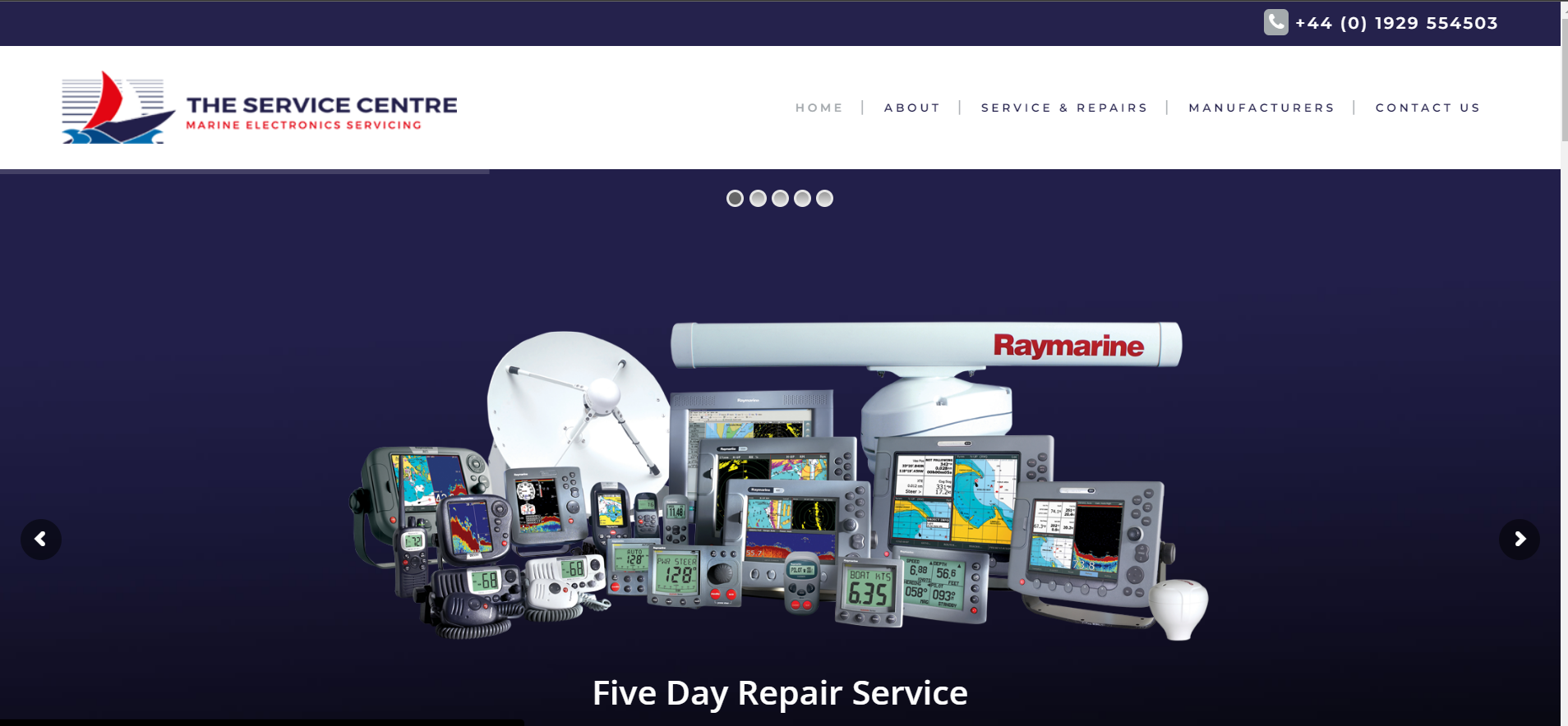


Рисунок 2 – Аналог зарубежного сайта “The Service Centre ”

Оба сайта предлагают схожие функциональные возможности, такие как описание услуг, информацию о запчастях и контактную информацию сервисных центров. Однако, сайт <https://theservicecentreuk.com/> уделяет особое внимание пользовательскому опыту, предоставляя возможность отслеживания статуса ремонта и раздел FAQ с ответами на наиболее часто задаваемые вопросы. В свою очередь, сайт <https://service-engineering.ru/> фокусируется на информировании о доступных услугах и запчастях для клиентов в России.

Оба этих сайта представляют собой надежные платформы для обслуживания и ремонта техники, сочетая в себе удобство использования и информационную полноту. Независимо от места нахождения пользователя, эти ресурсы помогут им найти подходящий сервисный центр и получить качественные услуги для своей техники.

## **1.2 Анализ структуры сайтов**

Анализ структуры сайта <https://service-engineering.ru/> позволяет выявить следующую организацию его разделов:

1. Главная страница: На главной странице сайта представлена краткая информация о компании и ее деятельности. Здесь приведено краткое описание услуг, которые предлагает сервисный центр, а также основные преимущества и ценности компании.
2. Услуги: в разделе "Услуги" представлены подробные описания услуг, предоставляемых сервисным центром. Здесь перечислены различные виды ремонта техники, такие как ремонт смартфонов, ноутбуков, планшетов и других устройств. Каждая услуга сопровождается описанием проблем, с которыми центр может помочь, а также указанием возможных сроков и стоимости ремонта.
3. Запчасти: в этом разделе представлена информация о доступных запчастях и комплектующих для ремонта техники. Здесь можно найти перечень запчастей, их описания, совместимость с различными устройствами, а также информацию о наличии и ценах.
4. О нас: в разделе "О нас" представлена информация о самой компании. Здесь можно узнать об истории сервисного центра, его миссии, ценностях и достижениях. Также предоставляется информация о квалификации специалистов и сертификатах, подтверждающих их компетентность.
5. Контакты: Раздел "Контакты" содержит контактную информацию сервисного центра, включая адрес, телефоны, электронную почту и форму обратной связи. Это позволяет пользователям связаться с центром для получения дополнительной информации, записи на ремонт или задания вопросов.
6. Статьи: в данном разделе представлены статьи и новости, связанные с ремонтом техники и сферой сервисного обслуживания. Здесь можно найти полезную информацию, советы по эксплуатации техники и новости из мира технологий.

Структура сайта <https://service-engineering.ru/> является логичной и интуитивно понятной. Она обеспечивает удобное и быстрое нахождение необходимой информации для клиентов, которые ищут ремонт техники.

Анализ структуры сайта <https://theservicecentreuk.com/> позволяет выявить следующую организацию его разделов:

1. Home: На главной странице сайта представлена краткая информация о компании и ее основных услугах. Здесь можно найти актуальные новости и специальные предложения, а также ссылки на другие разделы сайта.
2. Repairs: в разделе "Repairs" представлена информация о ремонте различных типов устройств. Здесь перечислены категории устройств, такие как смартфоны, планшеты, ноутбуки, игровые приставки и другая техника. Каждая категория сопровождается описанием возможных проблем и услуг, предлагаемых для ремонта данных устройств.
3. How It Works: В этом разделе представлена информация о процессе ремонта и взаимодействия с сервисным центром. Здесь объясняется, как оформить заявку на ремонт, какие шаги пройти и что ожидать во время процесса ремонта. Также предоставляются сведения о гарантии и возможностях отслеживания статуса ремонта.
4. Locations: Раздел "Locations" содержит информацию о местонахождении сервисных центров. Здесь представлены адреса и контактные данные различных филиалов компании, что позволяет клиентам быстро найти ближайший сервисный центр.
5. About Us: в данном разделе представлена информация о самой компании. Здесь можно узнать об истории сервисного центра, его ценностях, миссии и команде специалистов. Также предоставляется информация о сертификациях и опыте работы компании.
6. FAQ: Раздел "FAQ" содержит ответы на часто задаваемые вопросы. Здесь клиенты могут найти полезную информацию о процессе ремонта, стоимости, сроках и других вопросах, связанных с обслуживанием и ремонтом техники.
7. Contact: в разделе "Contact" представлена контактная информация сервисного центра, включая адрес, телефоны и электронную почту. Клиенты могут связаться с центром для записи на ремонт, получения дополнительной информации или задания вопросов.

Структура сайта <https://theservicecentreuk.com/> логична и удобна для пользователя. Она позволяет легко найти необходимую информацию о ремонте техники, оформить заявку и связаться с сервисным центром.

## **1.3 Юзабилити и функциональность сайтов**

Юзабилити и функциональность играют ключевую роль в опыте пользователей при использовании веб-сайтов. Будет рассмотрено юзабилити и функциональность сайтов <https://service-engineering.ru/> и <https://theservicecentreuk.com/>.

На сайте <https://service-engineering.ru/>, отечественном ресурсе, предлагается простой и интуитивно понятный интерфейс. Главное меню содержит основные разделы сайта, такие как услуги, запчасти, о компании и контакты. Это обеспечивает удобную навигацию и быстрый доступ к нужной информации. Верхняя панель содержит ссылки на дополнительные разделы и функциональности, такие как поиск и вход в аккаунт. Пользователям предоставляется возможность легко находить нужную информацию и взаимодействовать с сайтом.

Что касается функциональности, сайт <https://service-engineering.ru/> предоставляет подробную информацию о предоставляемых услугах и доступных запчастях. На странице услуг представлены различные виды ремонта и обслуживания, с описанием каждого из них. Пользователи также могут ознакомиться с информацией о запчастях, их наличии и стоимости. Это позволяет клиентам получить всю необходимую информацию о сервисном центре и принять решение о ремонте своего устройства.

Перейдем к сайту <https://theservicecentreuk.com/>, зарубежному ресурсу, который также обладает хорошей юзабилити и функциональностью. Сайт предлагает интуитивно понятный интерфейс с ясной навигацией. Главное меню содержит основные разделы, такие как услуги, о нас, FAQ и контакты. Пользователи могут быстро найти нужную информацию и перейти к интересующим их разделам.

Важной функциональностью сайта <https://theservicecentreuk.com/> является возможность оформления заявки на ремонт онлайн. Пользователи могут заполнить необходимую информацию о своем устройстве и описать проблему, после чего отправить заявку. Кроме того, сайт предоставляет функцию отслеживания статуса ремонта, что позволяет пользователям быть в курсе процесса обработки и ремонта своего устройства. Раздел FAQ также обеспечивает ответы на наиболее часто задаваемые вопросы, что упрощает взаимодействие пользователей с сайтом и получение нужной информации.

В обоих случаях, и на сайте <https://service-engineering.ru/>, и на <https://theservicecentreuk.com/>, уделяется внимание юзабилити и функциональности, чтобы предоставить пользователям удобство в использовании и необходимые функции для взаимодействия с сервисными центрами.

# **2. Проектирование**

## **2.1 Обзор кейсов использования**

В рамках данного обзора представляется обзор кейсов использования проекта "Сервисный центр по ремонту техники ". Рассмотрим различные сценарии использования сайта и функционал, доступный пользователям в различных ролях.

1. Кейс использования для клиента:

* Регистрация и создание профиля: Пользователь может зарегистрироваться на сайте, создавая персональный профиль. Это позволяет получить доступ к дополнительным функциям, сохранить информацию о заказах и использовать персонализированные настройки.
* Просмотр услуг и запчастей: Пользователь может просматривать доступные услуги и запчасти. Информация о предлагаемых услугах, их стоимости, а также о наличии и цене запчастей доступна для ознакомления.
* Размещение заказа: Пользователь может разместить заказ на ремонт или обслуживание. При размещении заказа пользователь указывает детали проблемы или запрашиваемую услугу.
* Отслеживание статуса заказа: После размещения заказа, пользователь может отслеживать его статус через свой профиль. Статусы заказа, такие как "выполнен" или "в процессе выполнения", отображаются для информирования пользователя о текущем состоянии заказа.

1. Кейс использования для администратора:

* Управление услугами и запчастями: Администратор имеет доступ к панели управления, где может добавлять, редактировать или удалять предлагаемые услуги и запчасти.
* Управление пользователями: Администратор может управлять пользователями, имеющими доступ к сайту. Это включает возможность добавления новых пользователей, изменения их ролей и настройку прав доступа.
* Мониторинг заказов: Администратор может отслеживать и управлять заказами, полученными через сайт. Это включает просмотр информации о заказе, обновление статусов и взаимодействие с пользователями в случае необходимости.

1. Кейс использования для модератора:

* Модерация заказов: Модератор имеет доступ к функционалу модерации заказов и ользователей. Он может проверять и модерировать размещенные заказы, убеждаясь в их соответствии правилам и стандартам проекта.
* Управление контентом: Модератор может управлять контентом, размещаемым на сайте, включая тексты, изображения и другие элементы. Он может редактировать, удалить или скрыть контент, несоответствующий правилам и требованиям проекта.

## **2.2 Выбор функционала**

В данном проекте представлены следующие функции, доступные пользователям:

1. Регистрация и авторизация:

* Пользователи могут зарегистрироваться на сайте, создавая персональные аккаунты.
* Авторизованные пользователи имеют возможность входа в свои аккаунты с помощью логина и пароля.

1. Профиль пользователя:

* Пользователи могут создать и настроить свой профиль, включая указание ФИО, контактной информации и адреса проживания.
* В профиле пользователи могут просматривать и редактировать информацию о себе.

1. Просмотр услуг и запчастей:

* Пользователи могут просматривать список доступных услуг, предлагаемых на сайте, с описанием и стоимостью каждой услуги.
* Также пользователи могут ознакомиться с ассортиментом доступных запчастей, их стоимостью и сроками доставки.

1. Размещение заказа:

* Пользователи могут разместить заказ на ремонт или обслуживание своего устройства, указывая необходимые детали и проблему.
* После размещения заказа, пользователь получает уникальный номер заказа для отслеживания его статуса.

1. Отслеживание статуса заказа:

* Пользователи могут отслеживать статус своих заказов через свой профиль или по уникальному номеру заказа.
* Статусы заказов, такие как "выполнен", "в процессе выполнения" или "ожидает подтверждения", отображаются для информирования пользователей о текущем состоянии заказа.

1. Контактная информация и обратная связь:

* Пользователи имеют возможность получить контактные данные сервисного центра для связи и консультации.
* Также предоставляется возможность отправки обратной связи, вопросов или запросов через специальную форму обратной связи.

# **3. Разработка и тестирование**

## **3.1 Архитектура проекта**

Для разработки бэкенда были выбраны различные технологии и инструменты.

Для создания Web API был выбран Django REST Framework (DRF), поскольку он обеспечивает удобство и быстроту создания операций CRUD (создание, чтение, обновление, удаление). DRF является мощным и гибким инструментом для проектирования Web API.

Архитектура проекта имеет следующую структуру:

serviceCenter: главная директория проекта, содержит основные настройки и список URL-адресов; customers: приложение, отвечающее за функционал, связанный с клиентами, включая их данные и заказы; devices: приложение, отвечающее за работу с устройствами, содержащее информацию о моделях, серийных номерах и других характеристиках; services: приложение, обеспечивающее функционал по работе с услугами, включая их название и стоимость; parts: приложение, занимающееся работой с запчастями, содержащее информацию о названии, стоимости и сроках доставки; employees: приложение, отвечающее за управление данными о сотрудниках, включая их личные данные, должности и зарплату.

В проекте была настроена административная панель, которая обеспечивает возможность просмотра, создания, удаления и редактирования всех моделей данных.

Customer: клиенты, содержащие уникальный идентификатор, ФИО, адрес электронной почты, телефон и адрес проживания; Order: заказы, включающие номер заказа, ссылку на соответствующего клиента, дату заказа, стоимость и статус выполнения заказа; Device: устройства, содержащие уникальный идентификатор, наименование, модель и серийный номер; Service: услуги, включающие уникальный идентификатор, название и стоимость; Part: запчасти, содержащие уникальный идентификатор, название, стоимость и сроки доставки; Employee: сотрудники, включающие уникальный идентификатор, ФИО, дату рождения, должность и зарплату.

При настройке панели администратора использовались специальные атрибуты классов, позволяющие отображать связи между таблицами в удобной форме. Например, ManyToMany связь между услугами и устройствами может быть отображена с помощью filter\_horizontal.

В результате разработки было создано функциональное и эффективное приложение для управления клиентами, заказами, устройствами, услугами, запчастями и сотрудниками в сервисном центре. Проект предоставляет надежное хранение и обработку информации.

Дальнейшее развитие проекта может включать добавление новых функциональных возможностей и улучшение пользовательского опыта.

Таким образом, выбор использованных технологий и архитектуры проекта был обусловлен необходимостью обеспечения эффективной и гибкой разработки, а также достижения высокого качества и удобства использования в конечном продукте.



Рисунок 3 – Администрирование сайта

Архитектура фронтенда проекта разделена на четыре основные страницы: "Вход", "Профиль", "Регистрация", "Профиль пользователя" и "Список пользователей". Каждая страница разработана на языке разметки HTML с использованием минимального дизайна.

Структура проекта включает следующие ключевые элементы:

1. **Шаблон base.html:** Данный шаблон является базовым для всех страниц и определяет общую структуру и стилизацию проекта. Он содержит основные блоки и подключает файлы стилей для обеспечения единообразного внешнего вида всех страниц.
2. **Страница "Вход" (login.html):** На этой странице пользователь может войти в систему, предоставив свои учетные данные. Форма входа реализована с помощью HTML-элементов и стилей из файла формы (form.html). После успешного входа пользователь перенаправляется на страницу "Профиль".

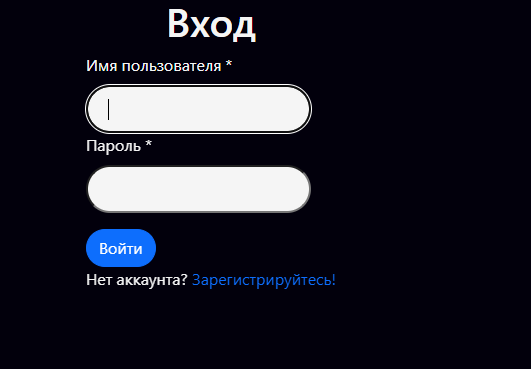


Рисунок 1 – Страница "Вход"

1. **Страница "Профиль" (profile.html):** Данная страница отображает информацию о текущем пользователе. Здесь пользователь может изменить свои данные, используя форму, которая также основана на файле формы (form.html). После сохранения изменений пользователь остается на странице "Профиль".

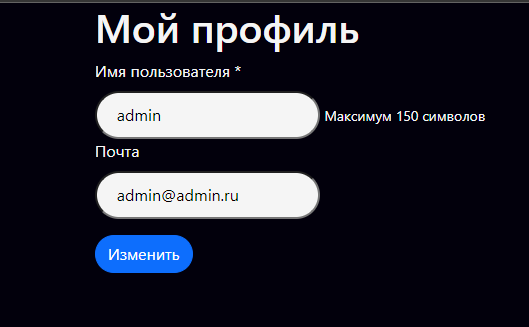


Рисунок 2 – Страница "Профиль"

1. **Страница "Регистрация" (registration.html):** На этой странице новые пользователи могут зарегистрироваться, предоставив свои данные. Форма регистрации также реализована с помощью HTML-элементов и стилей из файла формы (form.html). После успешной регистрации пользователь перенаправляется на страницу "Профиль".

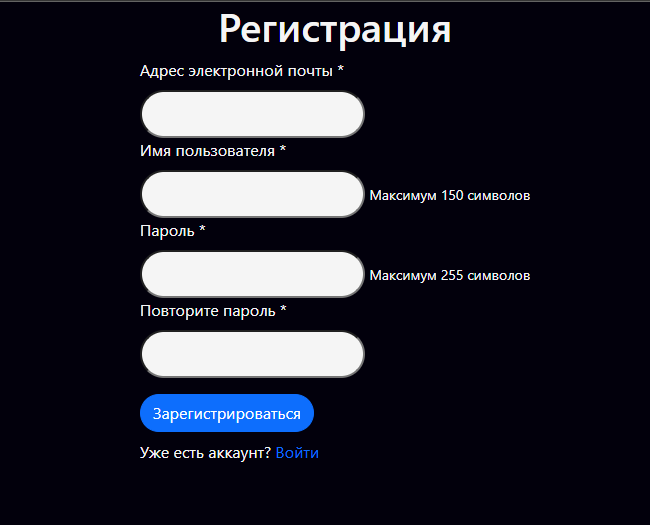


Рисунок 3 – Страница "Регистрация"

1. **Страница "Профиль пользователя" (user\_profile.html):** Данная страница отображает информацию о другом пользователе, выбранном из списка пользователей. Здесь отображаются основные данные пользователя, такие как имя и электронная почта, если они доступны.

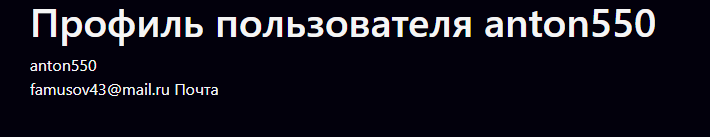


Рисунок 4 – Страница "Профиль пользователя"

1. **Страница "Список пользователей" (user\_list.html):** На этой странице выводится список всех пользователей, доступных в системе. Каждый пользователь представлен в виде ссылки, при нажатии на которую открывается страница "Профиль пользователя" для просмотра дополнительной информации.

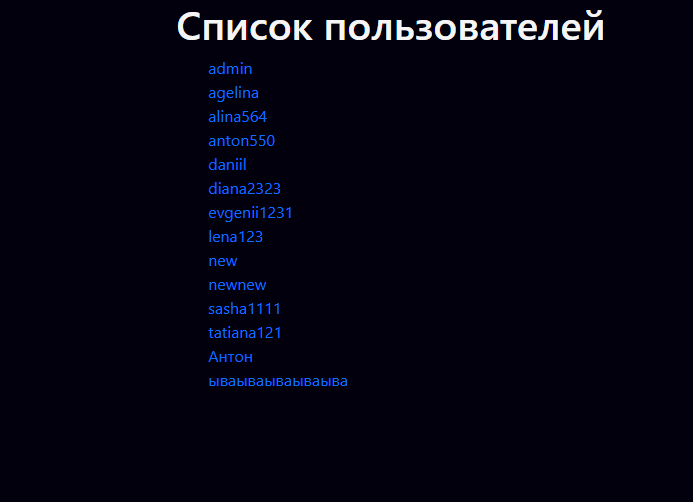


Рисунок 5 – Страница "Список пользователей"

Статические файлы (CSS-файлы, изображения) используются для стилизации и визуального оформления страниц. Файл стилей (styles.css) содержит определение классов и свойств, которые применяются к различным элементам интерфейса для создания согласованного и привлекательного вида.

Таким образом, архитектура проекта обеспечивает простоту использования и навигации для пользователей, а также позволяет легко добавлять и изменять функциональность и дизайн в дальнейшем.

## **3.2 Структура базы данных**

Для хранения данных в проекте будет использована Система Управления Базами Данных СУБД SQLite.

В таблице 1 представлены структуры таблиц базы данных:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица “Заказы” | Номер заказа (уникальный идентификатор) | ID клиента (ссылка на таблицу "Клиенты") | Дата заказа | Стоимость заказа | Статус заказа (выполнен/в процессе выполнения/готов) |
| Таблица “Устройства” | ID устройства (уникальный идентификатор) | Наименование устройства | Модель устройства | Серийный номер | Дата покупки/ гарантийный срок |
| Таблица “Услуги” | - ID услуги (уникальный идентификатор) | Название услуги | Стоимость услуги |  |  |
| Таблица “Запчасти” | - ID запчасти (уникальный идентификатор) | Наименование запчасти | Стоимость запчасти | Срок доставки |  |
| Таблица “Сотрудники” | - ID сотрудника (уникальный идентификатор) | ФИО сотрудника | Дата рождения | Должность | Зарплата |

Таблица 3.1 представляет структуру базы данных.

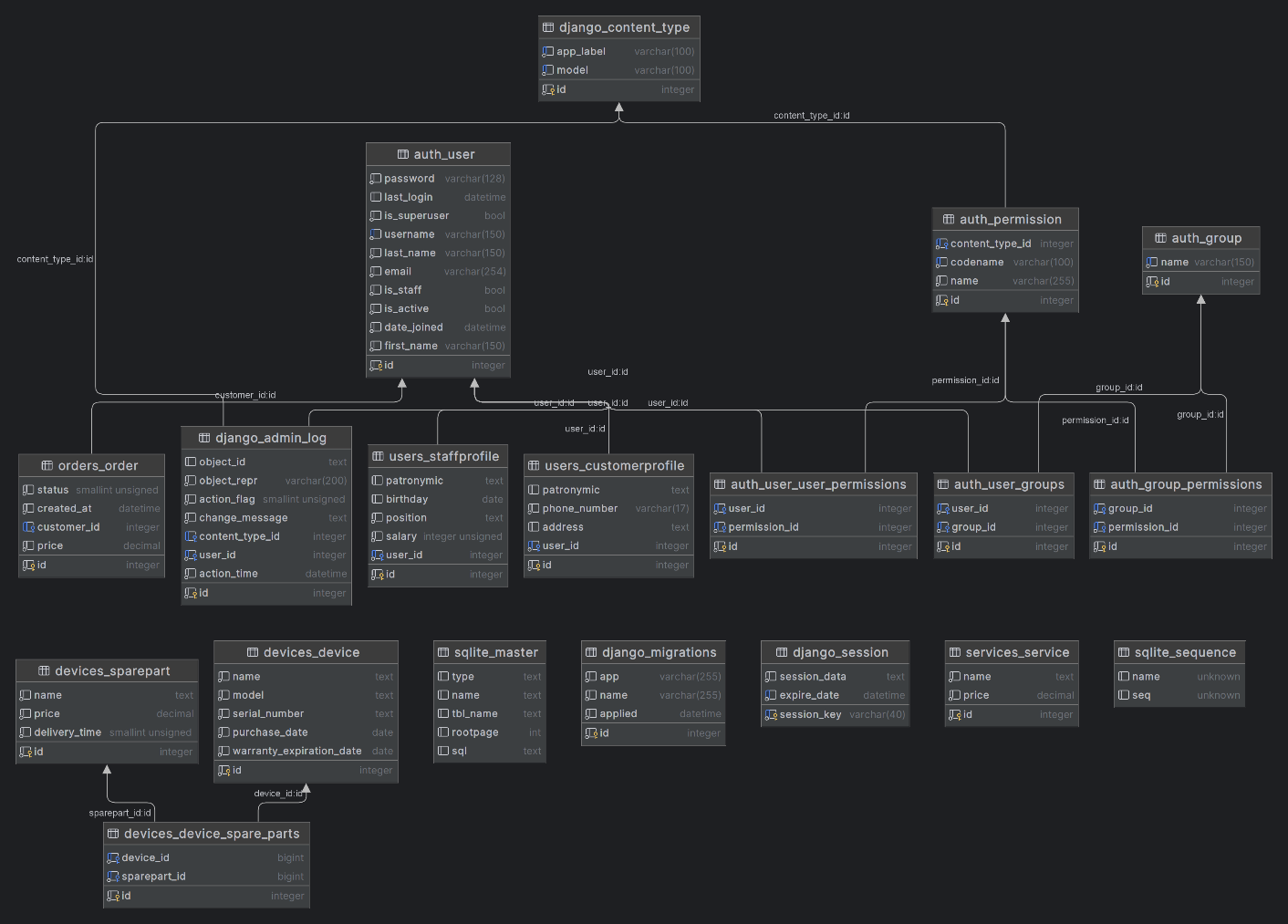


Рисунок 6 - структура базы данных.

## **3.3 Типовые запросы СУБД**

В проекте для выполнения типовых запросов к базе данных используется Django REST Framework (DRF). DRF предоставляет мощный инструментарий для создания и обработки запросов к базе данных автоматически.

В Django REST Framework запросы к базе данных генерируются автоматически благодаря наследованию от различных классов, предоставляемых фреймворком. Например, при работе с таблицей "Устройства" можно использовать класс `Device.objects.all()**`**, который позволяет получить все записи из таблицы "Устройства". Этот класс наследуется от базового класса `ModelViewSet**`**, который уже предоставляет функционал по умолчанию для выполнения типовых операций с данными, таких как создание, чтение, обновление и удаление записей.

Благодаря Django REST Framework, можно легко и удобно выполнять типовые запросы к базе данных и обрабатывать полученные данные в проекте.

# **Заключение**

В заключение данной курсовой работы необходимо подвести итоги выполненных этапов разработки и достигнутых результатов. Проект был тщательно спланирован, а его архитектура включала бэкенд сервера.

Бэкенд часть проекта была реализована с использованием Python и Django REST-framework. Django REST-framework предоставил мощный инструментарий для разработки Web API, упростив процесс. Структура базы данных была спроектирована с учетом нескольких основных таблиц, таких как "Клиенты", "Заказы", "Устройства", "Услуги", "Запчасти" и "Сотрудники". Организация данных в виде таблиц обеспечивает эффективное хранение и доступ к информации.

В рамках проекта были созданы типовые запросы, которые автоматически генерируются Django REST-framework на основе наследуемых классов. Это значительно упростило работу с базой данных и позволило легко выполнять различные операции, включая выборку, фильтрацию и обновление данных.

В итоге, проект успешно достиг своей цели, предоставив функциональное и эффективное приложение для управления клиентами, заказами, устройствами, услугами, запчастями и сотрудниками. Разработанный бэкенд взаимодействуют с базой данных, обеспечивая надежное хранение и обработку информации.

В будущем, проект может быть расширен и усовершенствован для добавления новых функциональных возможностей и повышения пользовательского опыта. Он может служить основой для создания реального веб-приложения, которое будет полезно и актуально для различных организаций и предприятий.

В целом, выполнение данной курсовой работы позволило расширить понимание процесса разработки веб-приложений и получить ценный опыт, который может быть применен в будущих проектах.

# 

# **Список использованных источников**

1. Меле, Антонио. Django 4 в примерах. 2023.
2. Django Documentation. Introduction and Overview. <https://docs.djangoproject.com/en/4.2/intro/overview/>.
3. Django Girls Tutorial. [Онлайн]. <https://tutorial.djangogirls.org/ru/>.
4. Django Documentation. Contributing to Django. Admin documentation. <https://docs.djangoproject.com/en/4.2/ref/contrib/admin/>.
5. Mozilla Developer Network. Learn Web Development. Server-side website programming. Django: Creating an Admin Site. <https://developer.mozilla.org/ru/docs/Learn/Server-side/Django/Admin_site>.

# **Приложение 1**

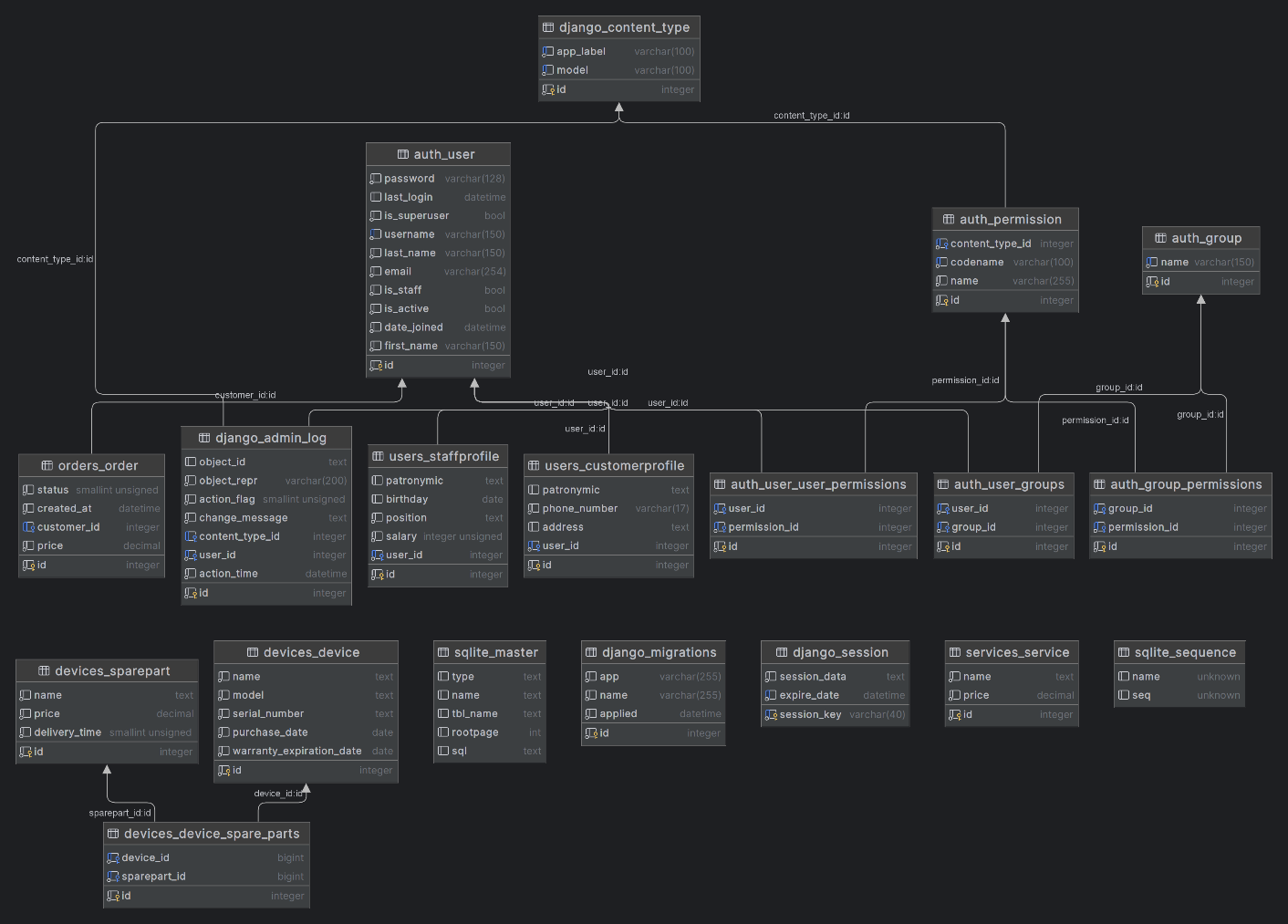


Рисунок приложения 1 –Схема бд

# **Приложение 2**

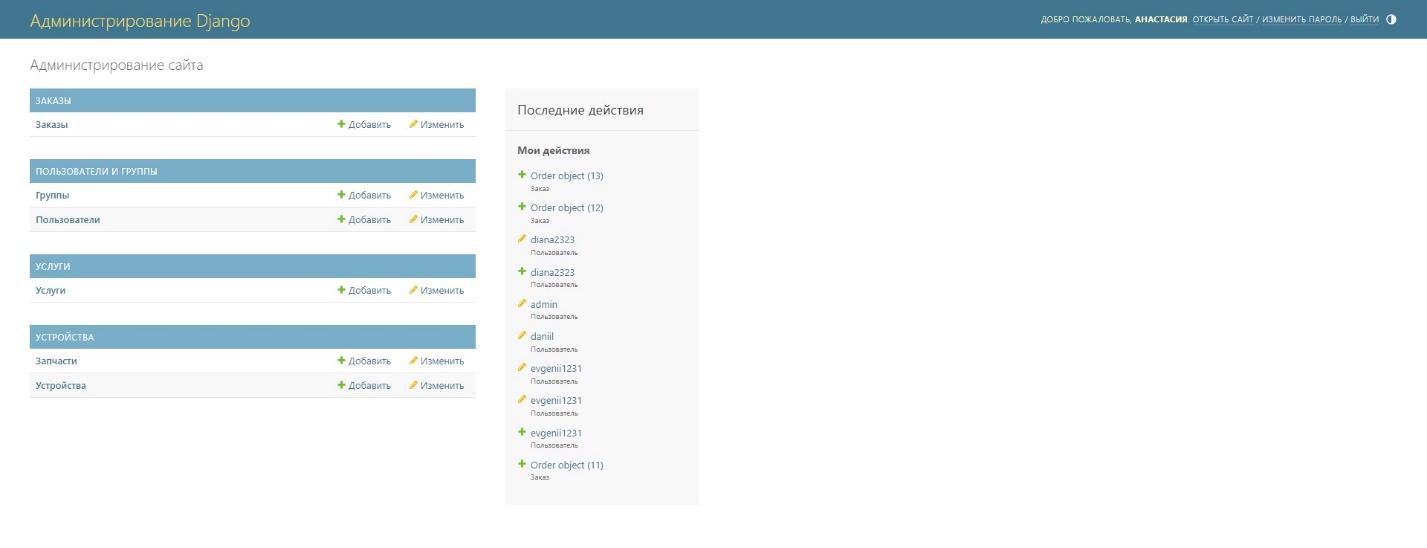


Рисунок приложения 2 –Администрирование сайта Django

# **Приложение 3**

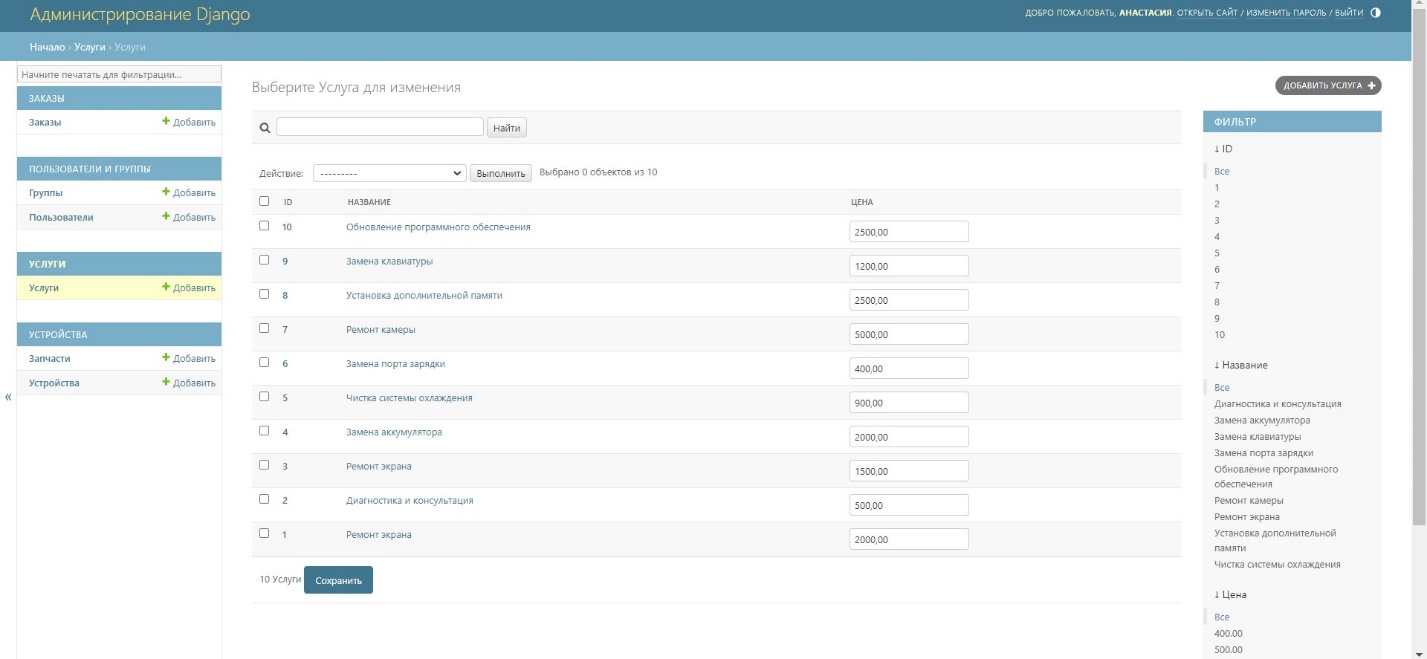


Рисунок приложения 3 –Таблица “Услуги”

# **Приложение 4**

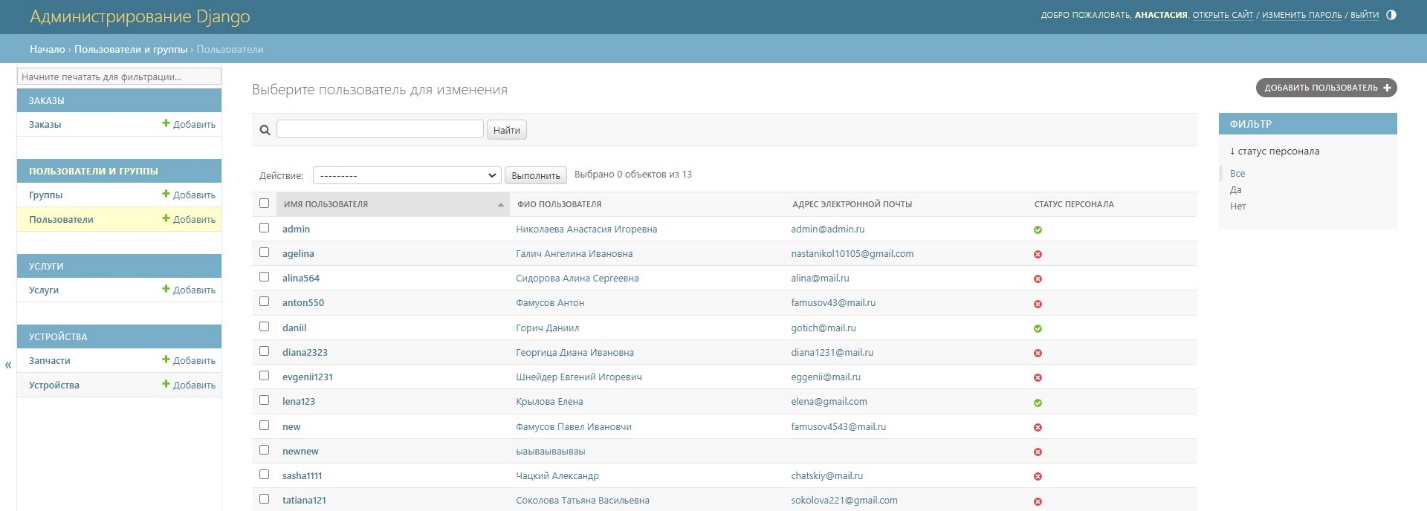


Рисунок приложения 4 –Таблица “Пользователи”

# **Приложение 5**

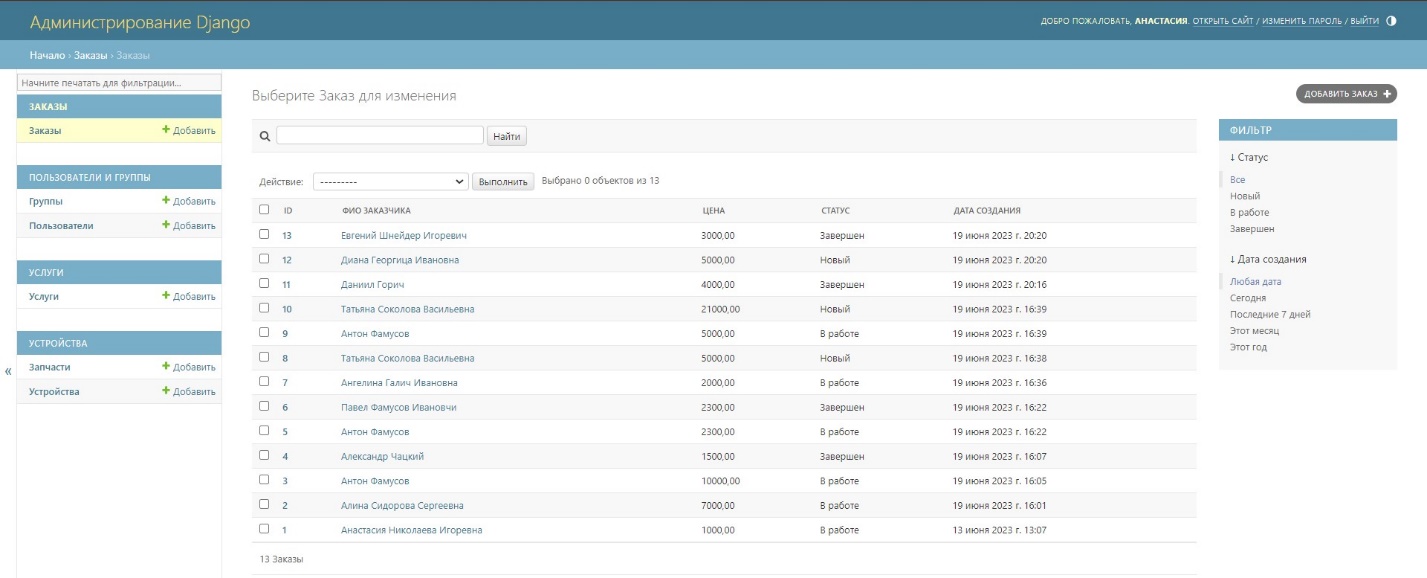


Рисунок приложения 5 –Таблица “Заказы”