МИНИСТЕРСТВО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ МОСКОВСКОЙ ОБЛАСТИ

«КРАСНОГОРСКИЙ КОЛЛЕДЖ»

**Учебная практика**

«Информационной системы для управления зимней турбазы»

специальность 09.02.06 Информационное и системное программирование

от студента 32ИС-21К Малинкин К.П.

Руководитель:

преподаватель Солдатенко Д.С.

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись преподавателя: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

г. Красногорск

2024

**Теоретические сведения.**

**Основные понятия и определения**

Информационная система (ИС) является средой, составляющими элементами которой являются компьютеры, компьютерные сети, программные продукты, базы данных, люди и т.д. Основное назначение ИС – организация хранения и передачи информации. ИС – это человеко-компьютерная система для организации хранения, обработки и выдачи информации в интересах достижения поставленной цели, использующая компьютерную информационную технологию.

Обычно в термин «информационная система» вкладывается понятие «автоматизированная система», при этом в процессе обработки информации главная роль отводится компьютеру.

Автоматизированная информационная система (АИС) – это совокупность тем или иным образом структурированных данных (базы данных) и комплекса аппаратно-програмных средств для хранения данных и манипулирования ими.

Известно несколько определений понятия «база данных». Согласно одного из них база данных (БД) – это упорядоченная информационная система, в которой информация хранится в виде таблиц и управляется специально созданной системой управления базой данных (СУБД), состоящей из таблиц, запросов, форм, макросов и отчетов.

Понятие БД можно применить к любой информации, в которой имеются связанные по определенному признаку элементы, хранимой и организованной особым образом - как правило, в виде таблиц. По сути, БД - это некоторое подобие электронной картотеки, электронного хранилища данных, которое хранится в компьютере в виде одного или нескольких файлов.

Для взаимодействия пользователя с БД используются системы управления баз данных (СУБД).

СУБД – это комплекс программ и языковых средств, предназначенных для создания, введения и использования баз данных.

Современные СУБД обеспечивают:

* набор средств для поддержки таблиц и соотношений между связанными таблицами;
* развитый пользовательский интерфейс, позволяющий вводить и модифицировать информацию, выполнять поиск и представлять информацию в текстовом или графическом виде;
* средства программирования высокого уровня, с помощью которых можно создавать собственные приложения.

БД должна опираться на одну из трех моделей данных, которые определяют связи между данными (см. рис.1). В последнее время наибольшее распространение получили реляционные БД. На идеологии реляционной модели данных построены СУБД: FoxPro, Acсess, Visual C++ и др.

**План моей работы**

1. Концепция

Представляет собой комплексную информационную систему, разработанную для управления всеми аспектами функционирования турбазы. Она объединяет в себе функциональные возможности CRM (управление взаимоотношениями с клиентами), управления персоналом, прокат инвентаря, график работы сотрудников, список и поломки инвентаря, товары магазина.

1. Цели:

* Эффективное планирование ресурсов, включая персонал, оборудование и услуги, для обеспечения высокого уровня обслуживания.
* Автоматизация процессов бронирования и учета услуг для улучшения оперативности и точности.
* Повышение операционной эффективности и сокращение издержек.

1. Основные модули:

* Модуль управления клиентами (CRM): Включает в себя хранение данных клиентов, историю взаимодействия, предпочтения, а также возможности анализа для персонализации услуг и маркетинговых кампаний.
* Модуль управления персоналом: Позволяет управлять графиком работы, создавать задачи для сотрудников.
* Модуль аналитики и отчетности: Предоставляет аналитические инструменты для выявления тенденций, анализа эффективности услуг, прогнозирования спроса.

**Практическая работа:**

1. Скачиваю Python 3.11. Потом сам фреймворк Sanic. (Рисунок 1)

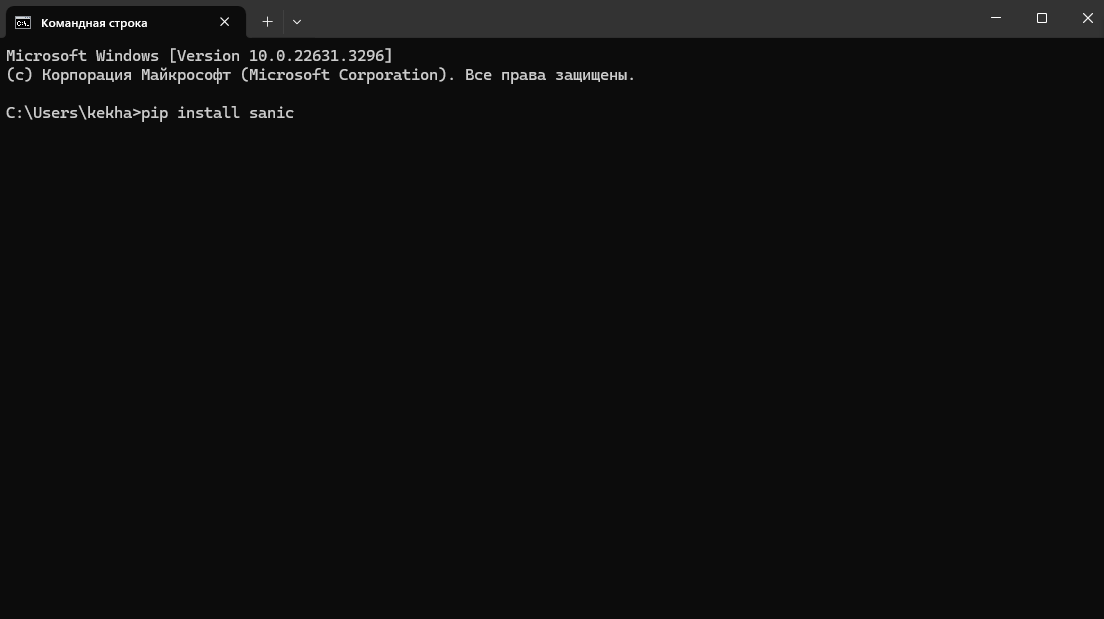


Рисунок 1. В «командной строке» скачиваю фреймворк «Sanic»

Еще скачиваю библиотеку jinja2

1. Создаю «SanicApp», который будет прописано. Первое добавляем навигацию сервера. Чтоб это сделать, мы добавляем некую маршрутизацию. Маршрутизация в веб-приложениях относится к процессу определения того, какой код должен быть выполнен для обработки конкретного HTTP-запроса, основываясь на пути (URL) запроса. (Рисунок 2)

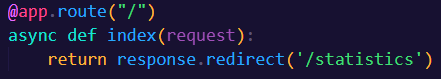


Рисунок 2. Маршрутизация на страницу статистику

Таким образом, когда кто-то обращается к вашему веб-приложению по корневому URL (например, http://example.com/), он будет перенаправлен на /statistics. Еще создаем для нашего сайта еще такие страницы: clients, shop, service, statistics, tasks, employees, inventory, rents, schedule.

1. Создаю архитектуру моего сайта (Рисунок 3)

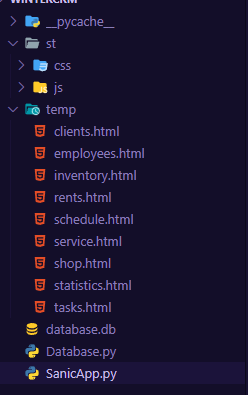


Рисунок 3. Архитектура сайта

В папке temp – будут храниться страницы; папка st (static) – будут хранить стили (css) и логика сайта (js). Database.py – нужна для создания базы данных. В нем будут храниться запросы, а также как будут выглядеть таблицы в этой БД. database.db – будет хранить все данные, которые ведет на своем сайте. В папке \_\_pycache\_\_ будет хранить модули. Он нужен, чтоб быстрее загружал сайт. А точнее – это созданный файл запуска сайта, будет создаваться интерпретатором Python при импорте модуля и содержит скомпилированный байт-код Python для данного модуля.

1. Добавляю библиотеку для Python «sqlite3». Он нужен чтоб, я мог Database.py использовать как базу данных SQLite (Рисунок 4). И создаю огромную базу данных для этого приложения.

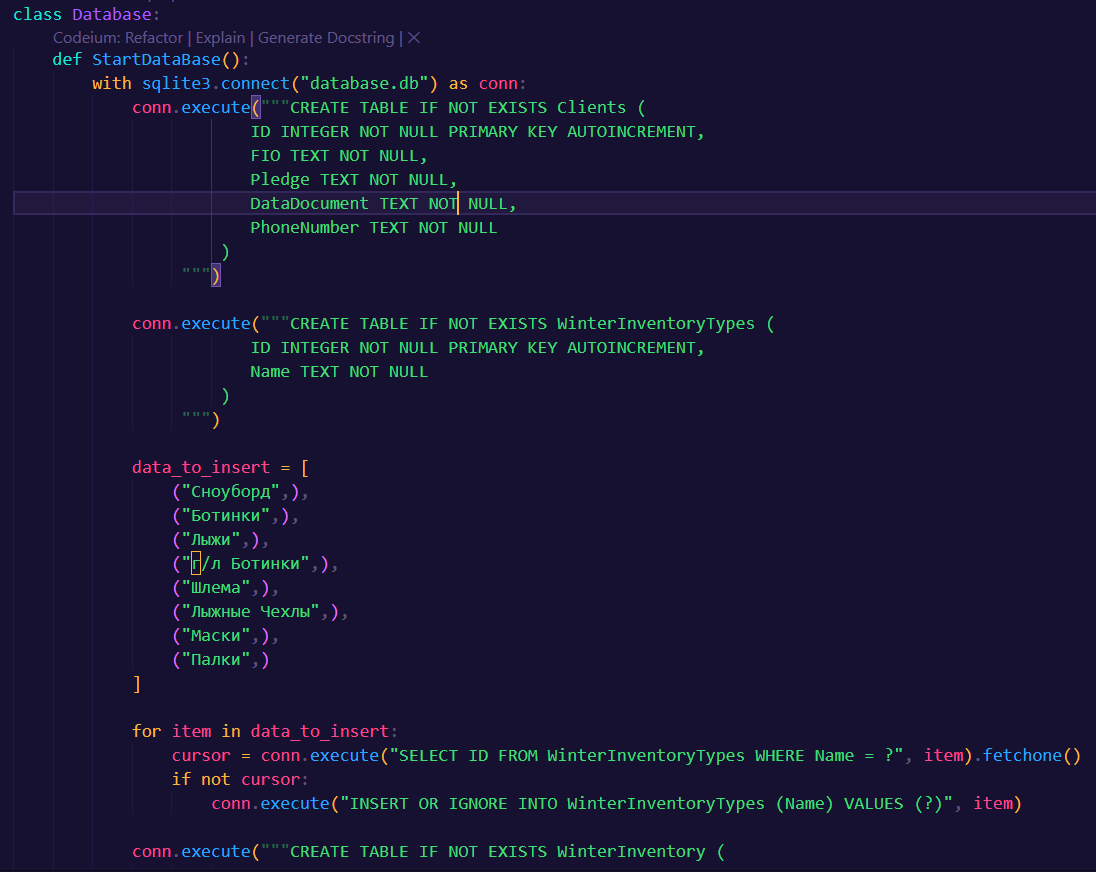


Рисунок 4. База данных сайта

1. Добавляю функционал, чтоб если не было своей database.db, то он создавался на основе Database.py. И запросы, который будет брать наш сервер с БД (Рисунок 5).

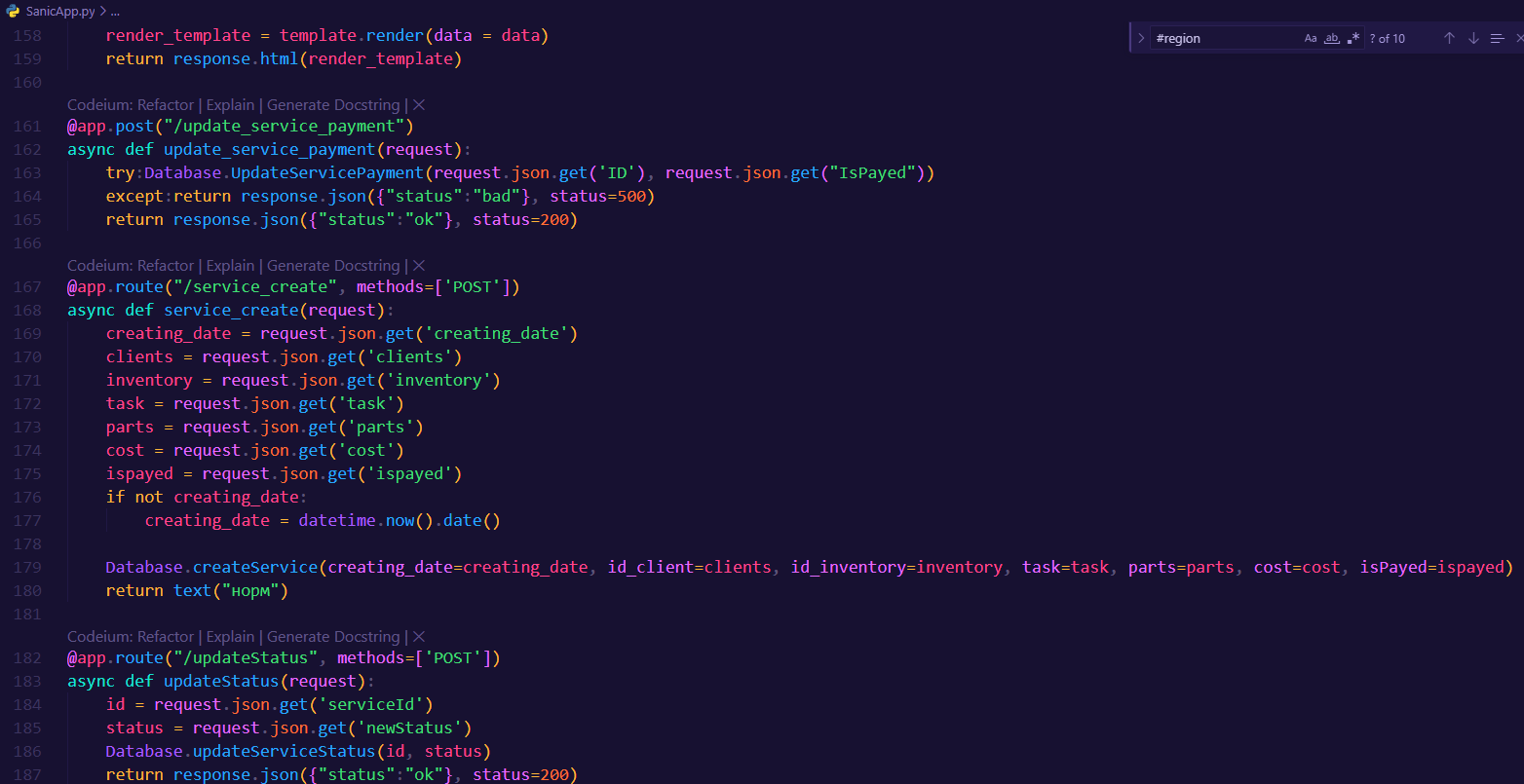


Рисунок 5. Запросы сервера на database.py

1. Создаю структуру html, для каждой страницы и ее внешность с помощью css. Вот несколько шаблонов моих сайтов.



Рисунок 6. Как выглядит html код страницы schedule

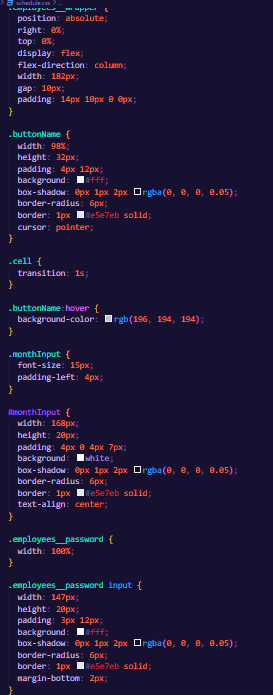


Рисунок 7. CSS стили. Для страницы schedule

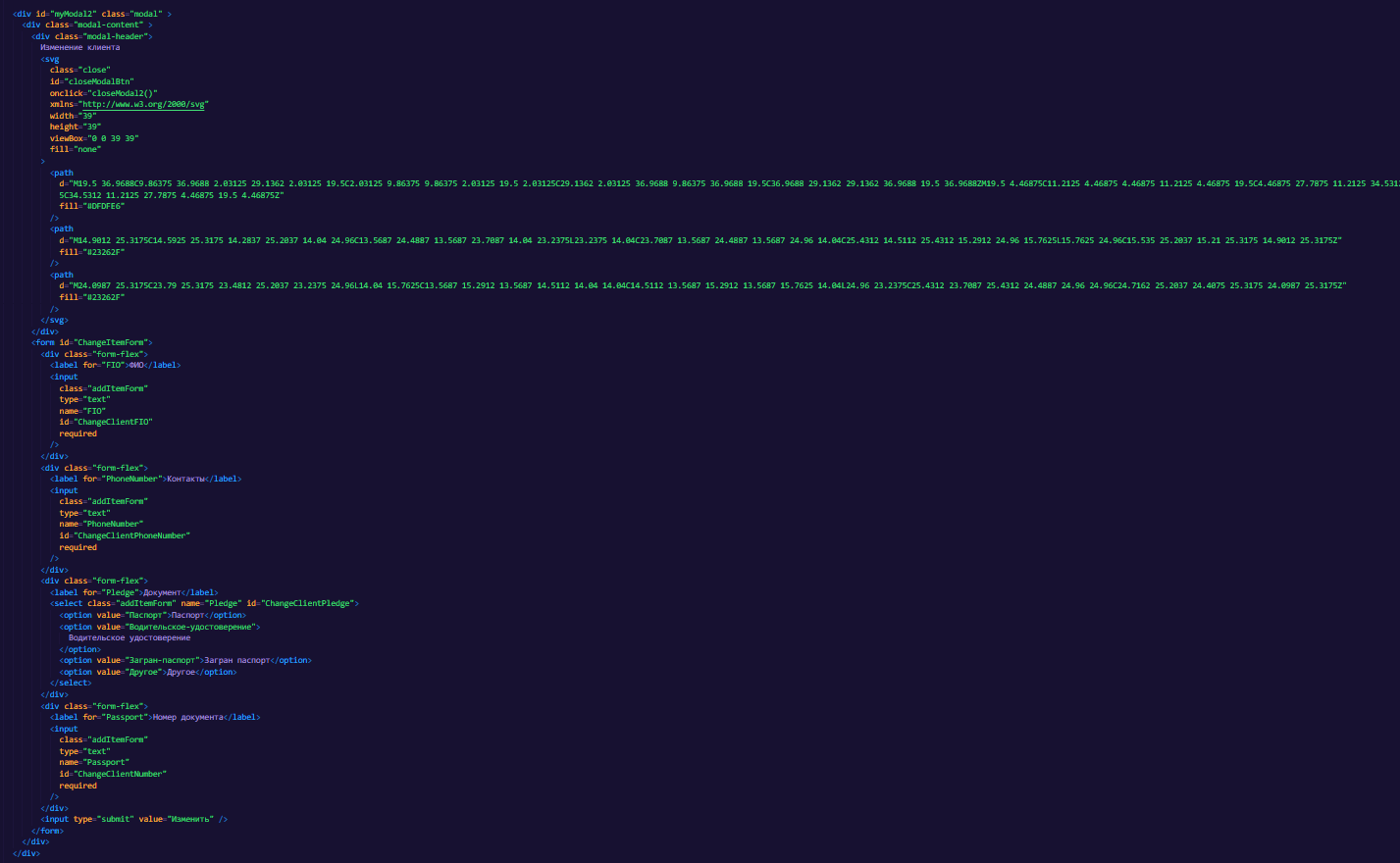


Рисунок 7. HTML-код для страницы clients

1. Добавляю логику моих страничек

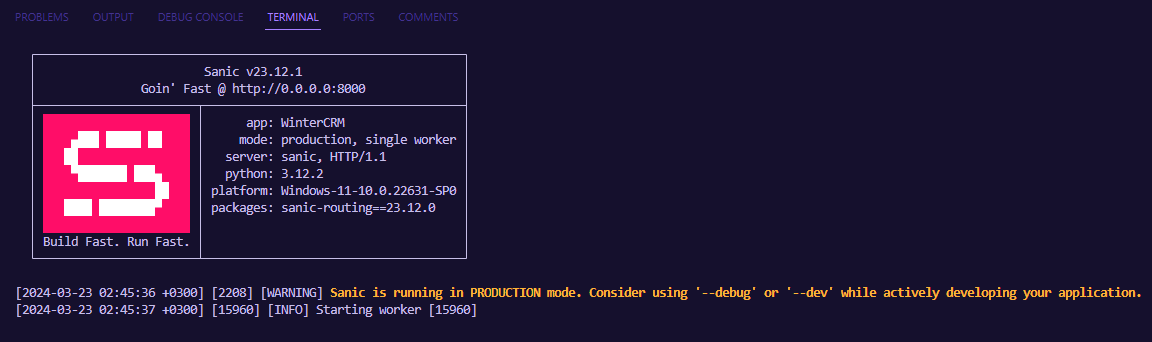


Рисунок 8. Здесь в JavaScript прописано для календаря, чтоб показывали месяца, дни.



Рисунок 9. Здесь в JavaScript прописано для аренды, что показывало таблицу, которые хранит «Предметы», «Имя клиента» и т.д. И фильтрация списка.

1. Запускаю сервер (Рисунок 10).



Появилась начальная страница, но пока нет никаких данных (Рисунок 11)

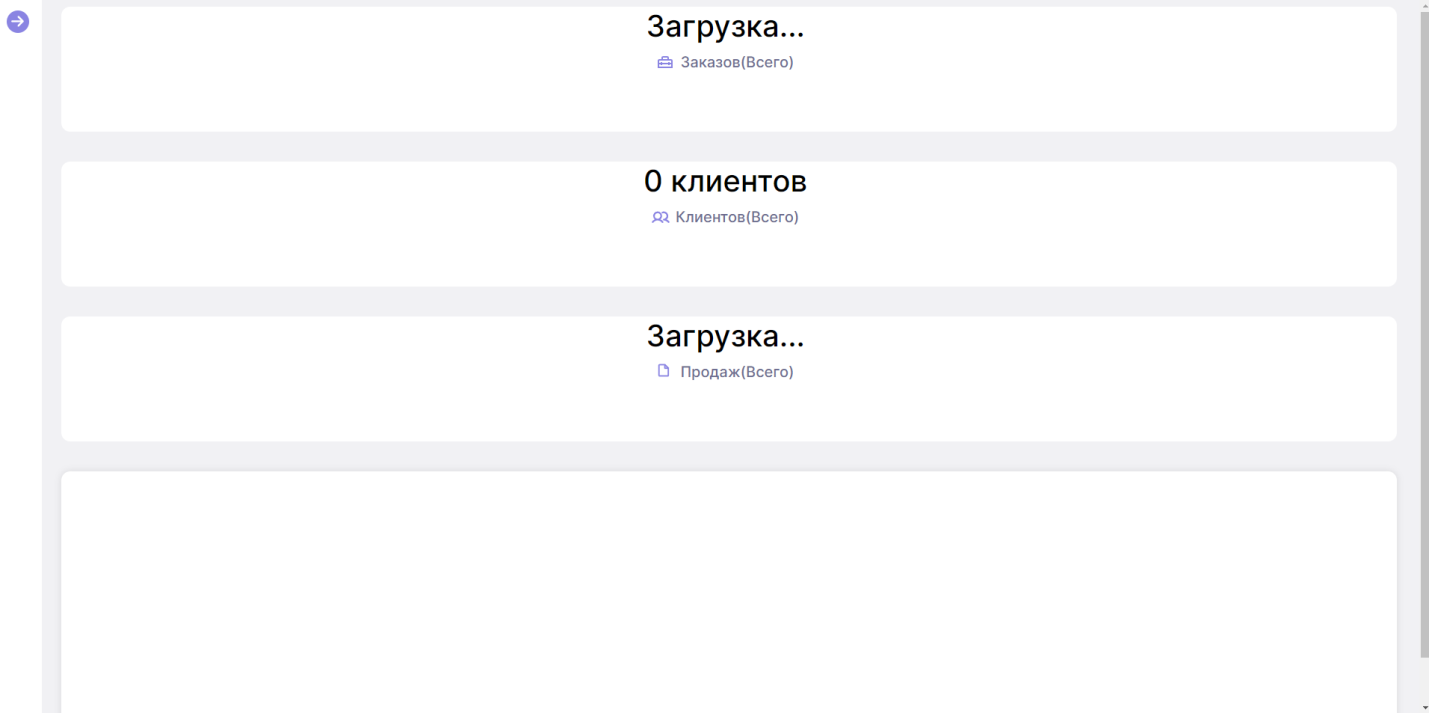


Рисунок 11. Главная страница. Это статистика

1. Создаю клиента

Чтоб создать внутри сайта клиента, я перехожу на страницу «Клиенты» (Рисунок 12).

Рисунок 10. Показывает, что сервер был успешно запущен.



Рисунок 12. Переход с сайта через slider menu.

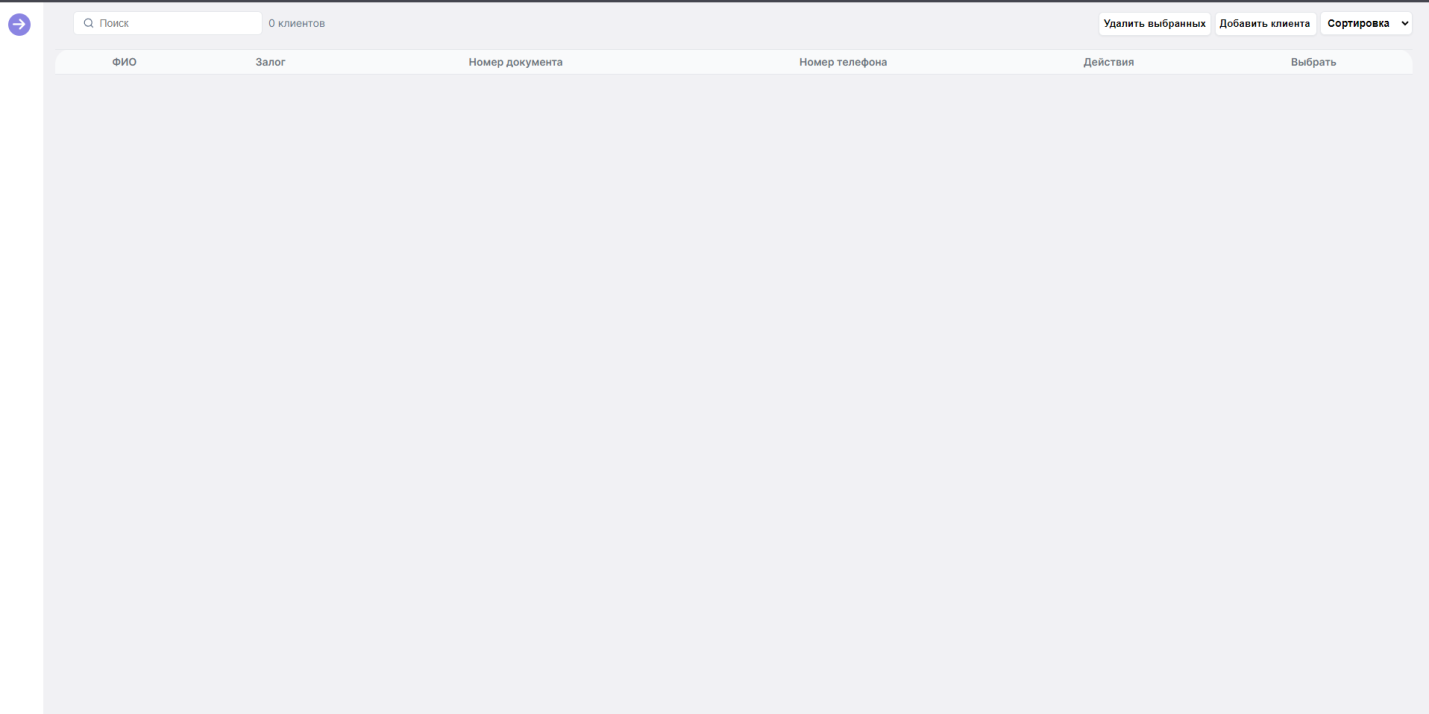


Рисунок 13. Страница «Клиенты»

Сейчас она пустая (Рисунок 13). Я нажимаю «Добавить клиента» и выскакивает модальное окно (Рисунок 14).

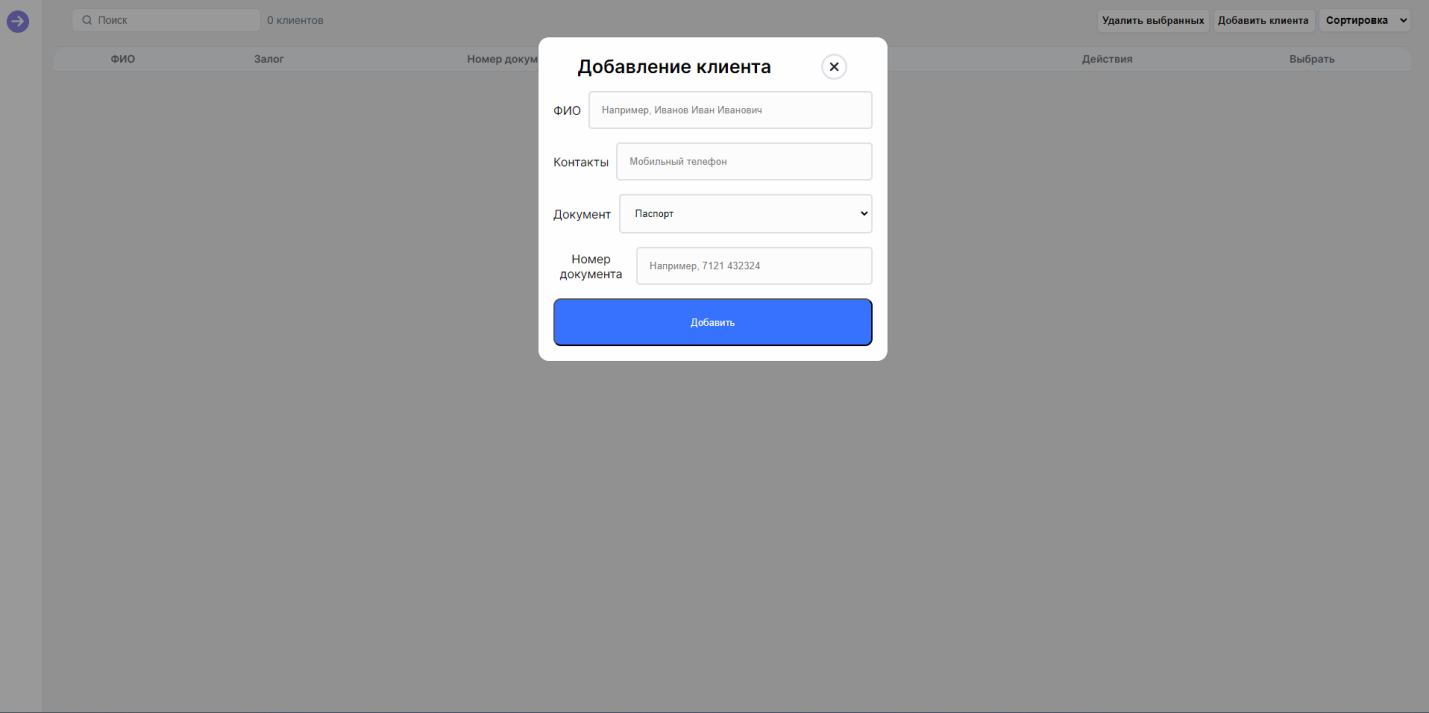


Рисунок 14. Модальное окно «Добавление клиента»

Вводим данные и добавляем клиента (Рисунок 15)

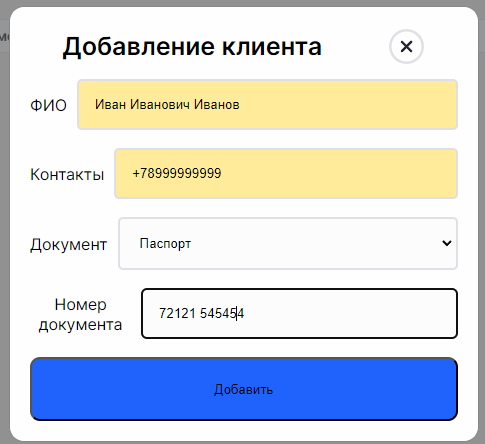


Рисунок 15. Ввел данные

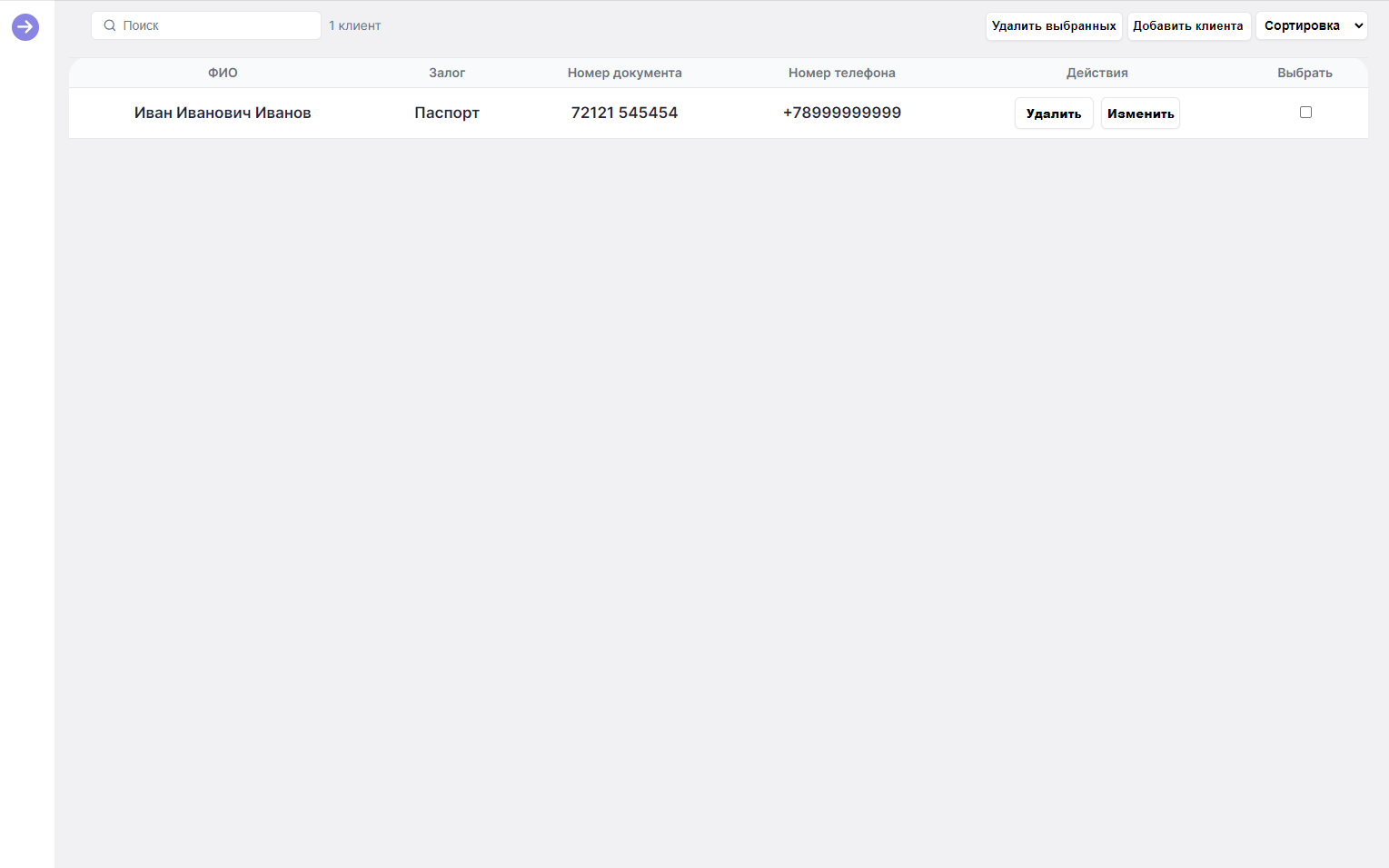


Рисунок 16. Появление первой записи в «Клиенты»

1. Создаем инвентарь для нашего проката. Захожу в страницу «Инвентарь» (Рисунок 17).

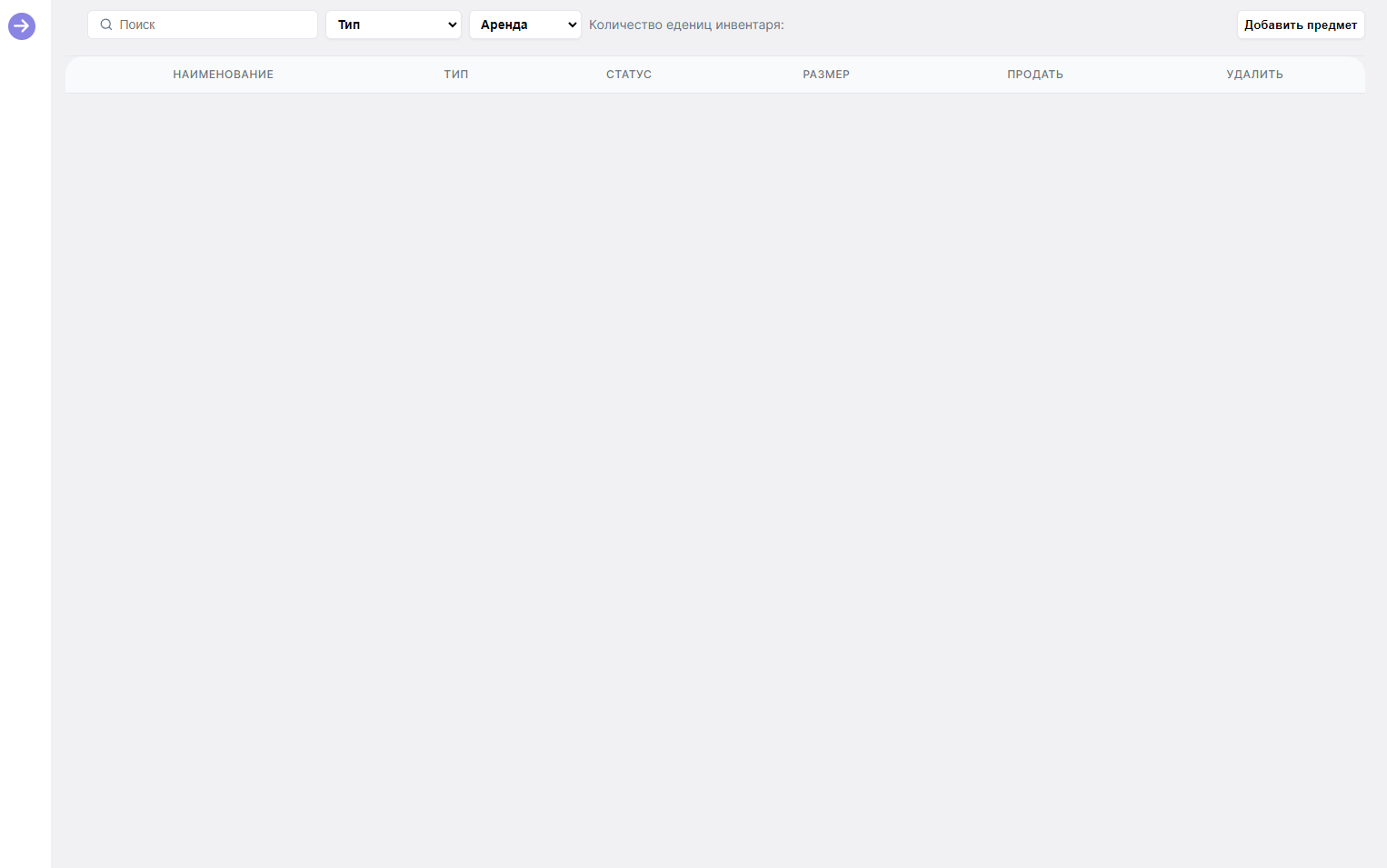


Рисунок 17. Страница «Инвентаря»

Нажимаю «Добавить предмет» (Рисунок 18).

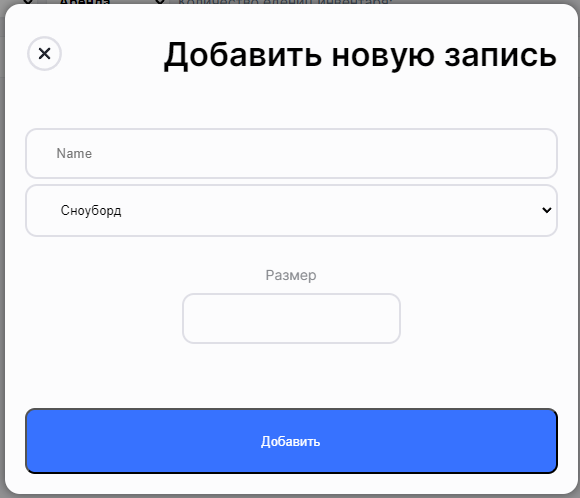


Рисунок 18. Модальное окно «Добавить новую запись»

Вписываю название и выбираю предмет, который мне нужен. Допустим, лыжи (Рисунок 19).

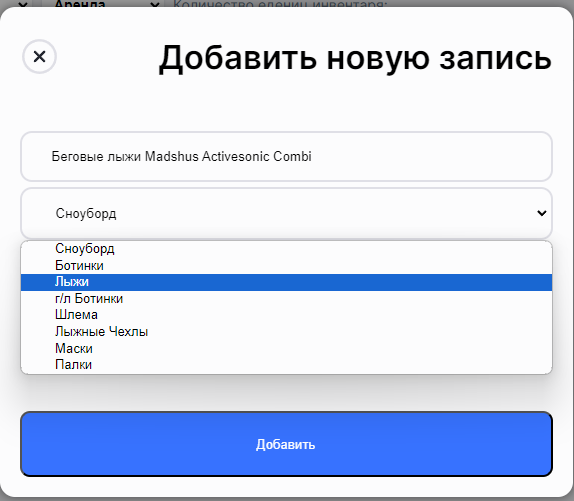


Рисунок 19. Список, который можно выбрать

И добавляю запись (Рисунок 20).

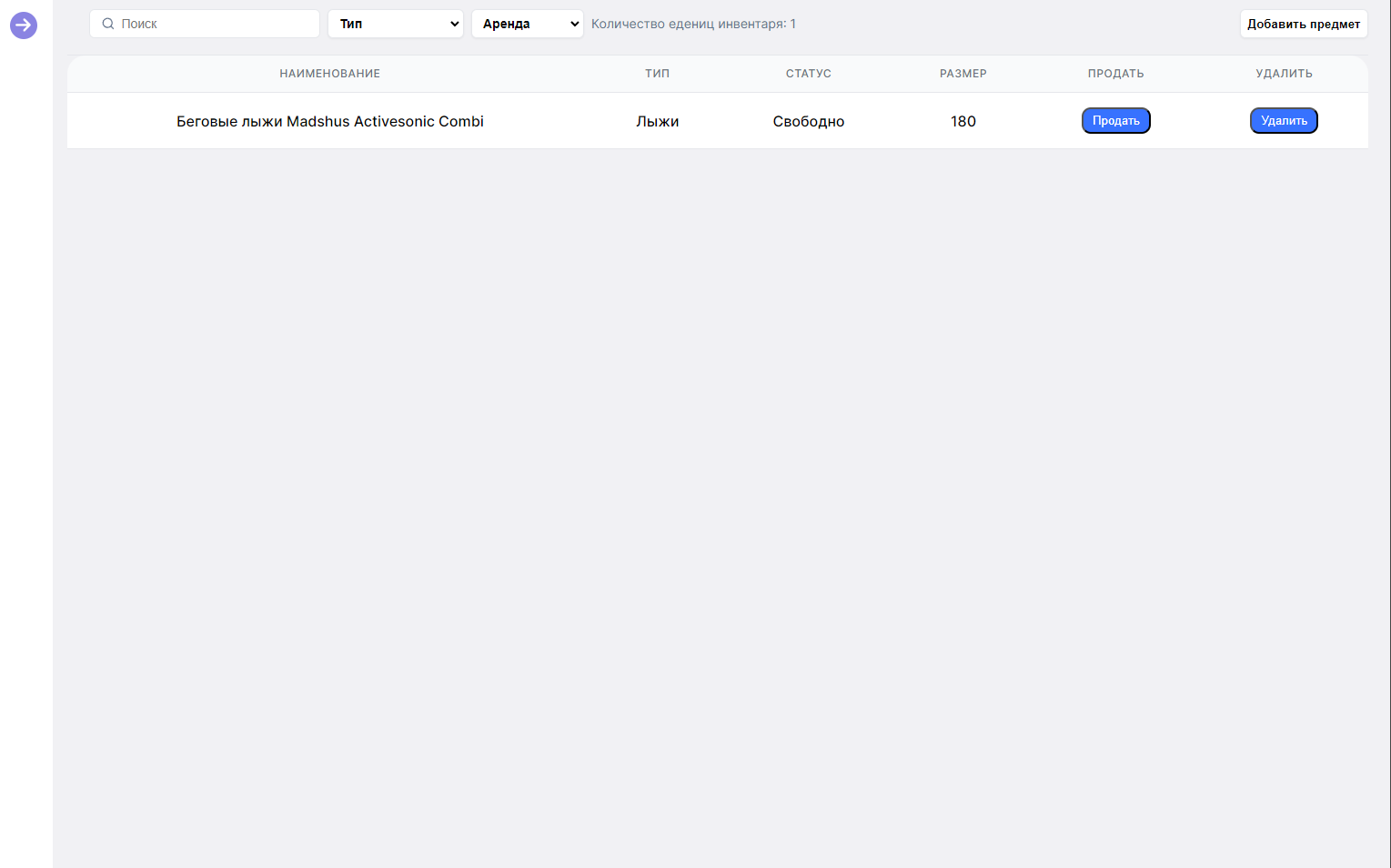


Рисунок 20. Появление первой записи в «Клиенты»

Теперь, БД хранит данные лыж, который имеет название, тип, размер. Сейчас он находиться в статусе «Свободно», как его никто не забронировал. Допустим, у нас пришел клиент и забронировал эти лыжи.

1. Создаем аренду лыж. Захожу на страницу «Заказы» (Рисунок 21).

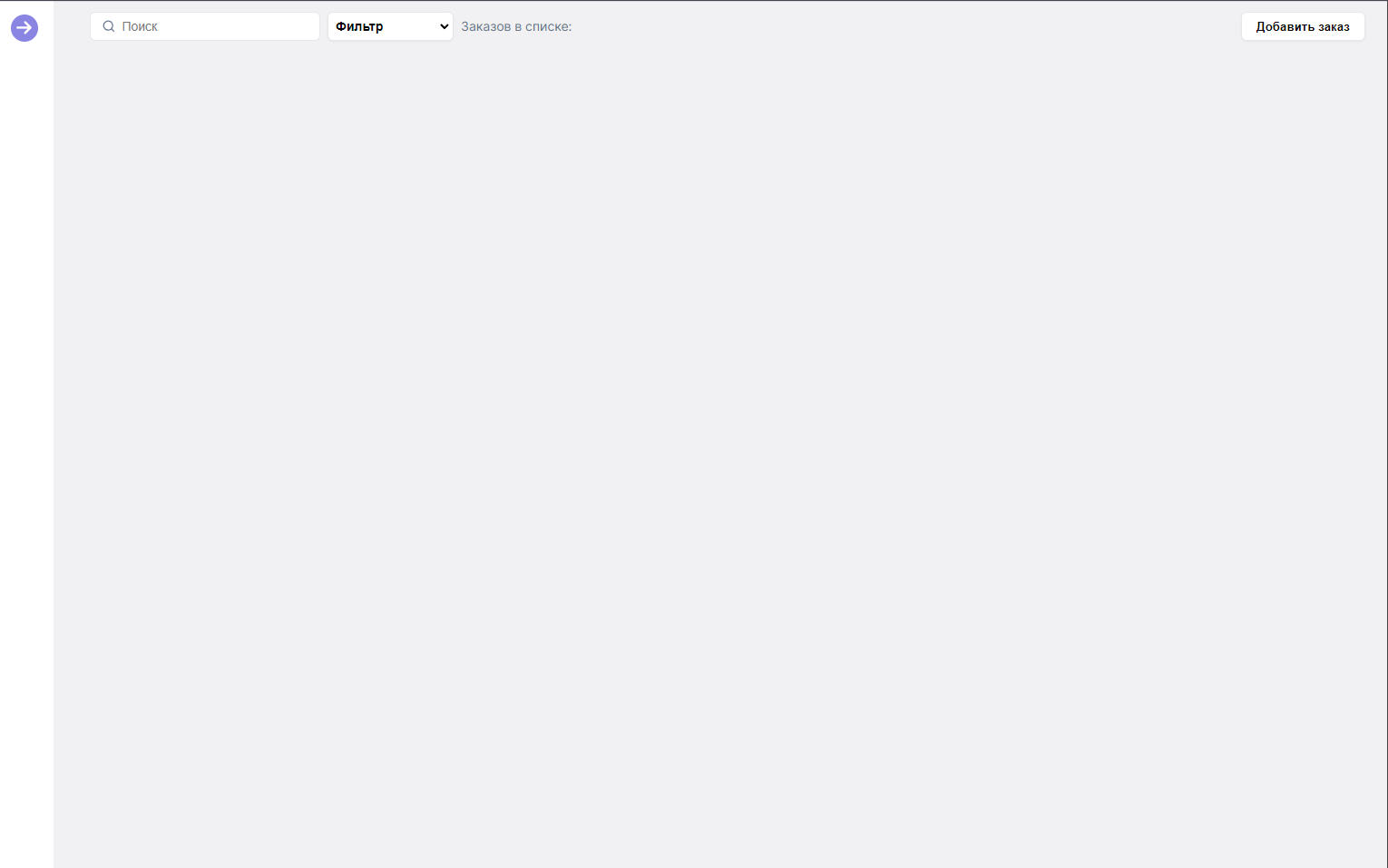


Рисунок 21. Страница «Заказы»

Нажимаю «Добавить заказ». Выскакивает модальное окно (Рисунок 22).

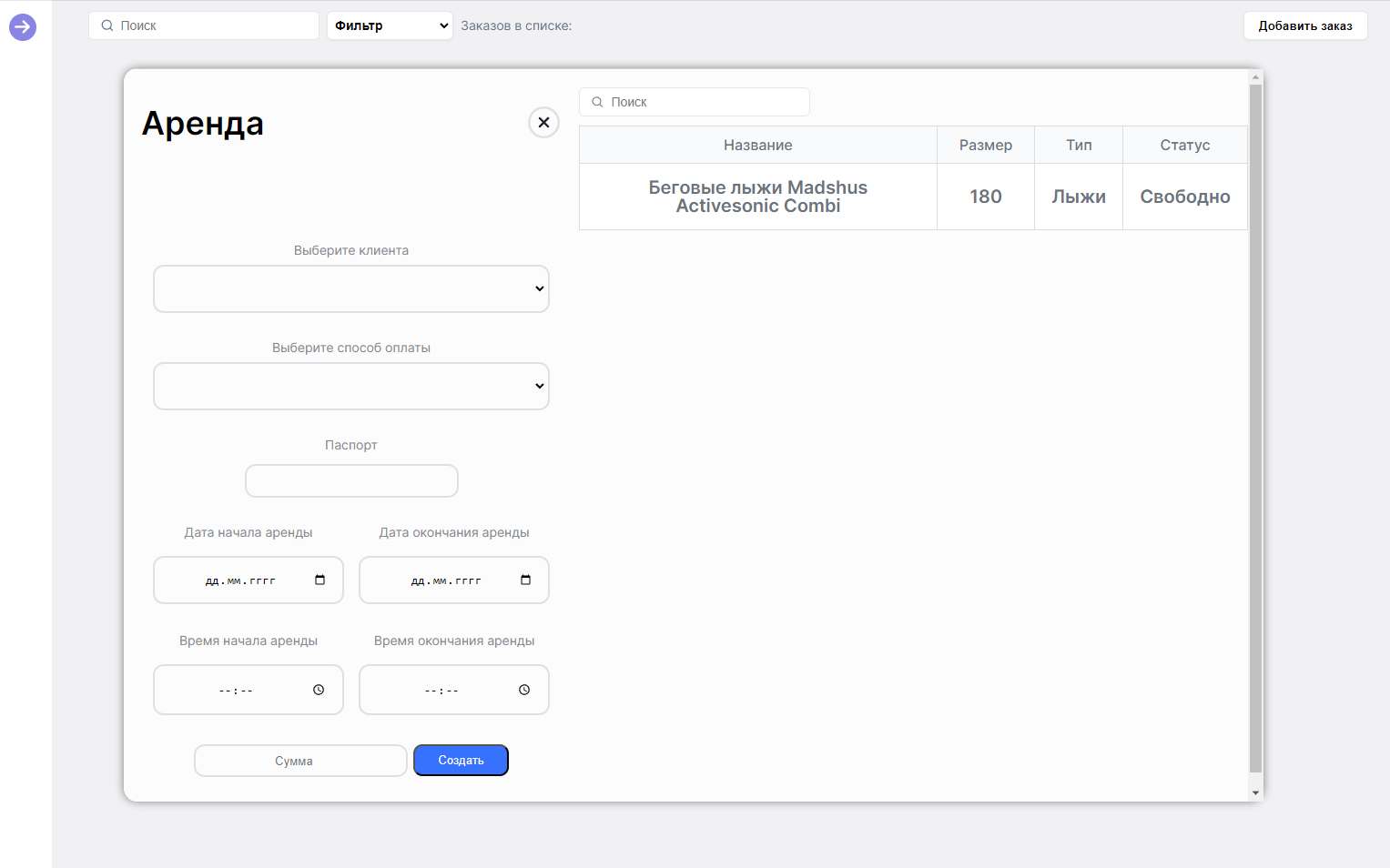


Рисунок 22. Модальное окно «Аренда»

Слева находиться данные по бронированию предметов. Справа – список предметов, который сейчас в статусе «Свободно». Выбираем предмет из инвентаря, то есть «Беговые лыжи» и теперь окно выглядит так (Рисунок 23).

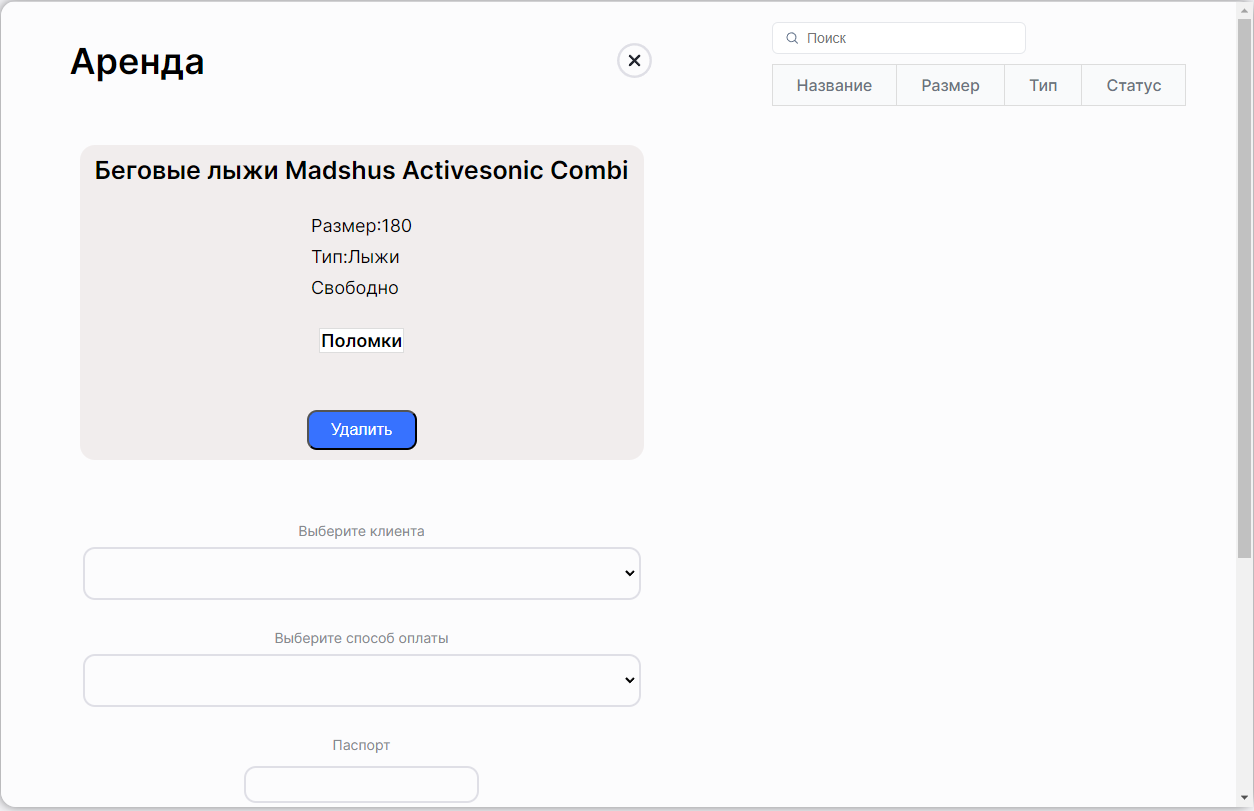


Рисунок 23. Предмет добавился в аренду

У нас пропал, справа предмет, как мы его выбрали, и переместился он налево. У нас появляется вся информация об этом предмете: название, размер, тип, статус, убрать его с списка – также о «Поломки». «Поломка» нужна для информации, если был сломан предмет. Дальше ниже, мы выбираем клиента из списка. Я записал, как Иван Иванович Иванов. И сразу вписываются его данные паспорта (Рисунок 24).

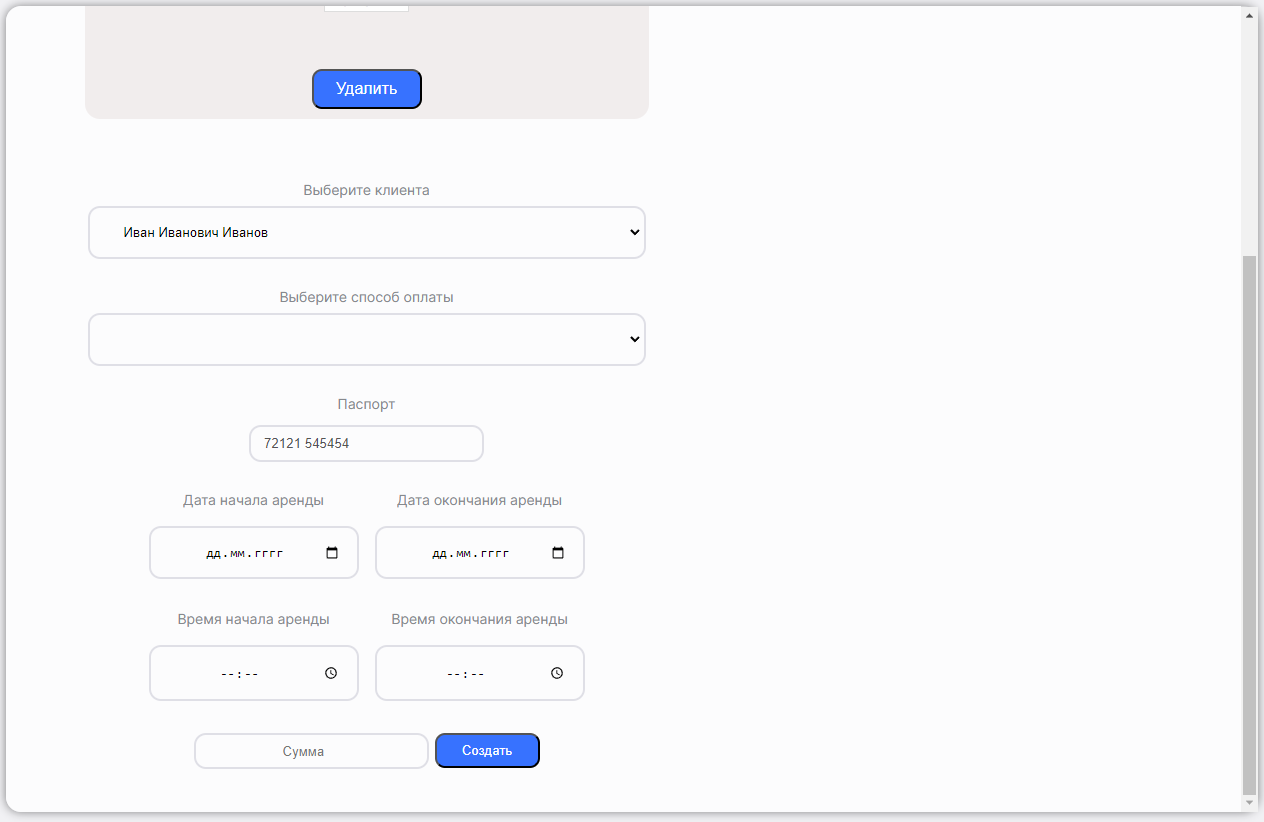


Рисунок 24. Автоматизация данных

Способ оплаты выбираю. Например, возьму «Банковский перевод» (Рисунок 25).

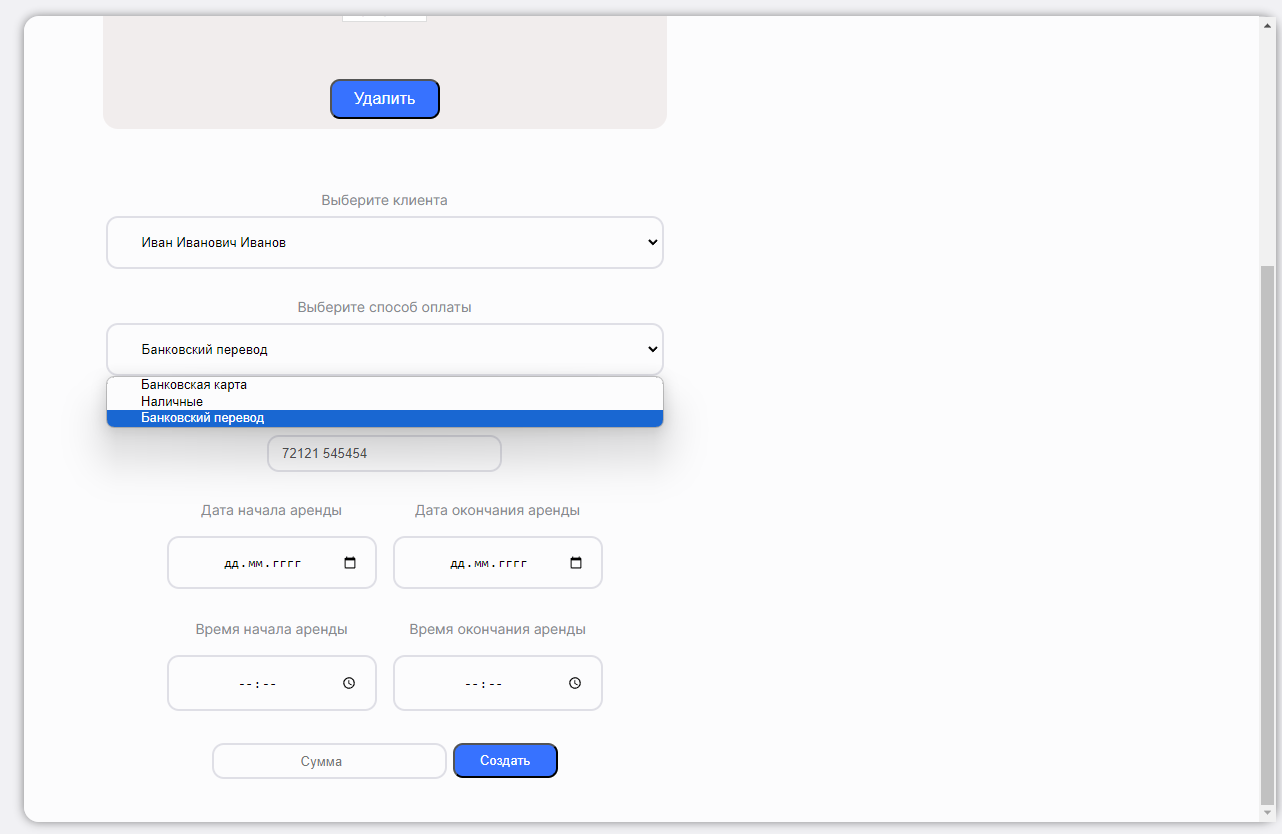


Рисунок 25. Список «Выберите способ оплаты»

Вписываю даты начала и окончания аренды, время начала аренды и окончания. И сумма оплаты аренды (Рисунок 26).



Рисунок 26. Вписываю данные «Дата начала», «Дата окончания аренды» и также со временем

Создаем, и смотрю результат (Рисунок 27).

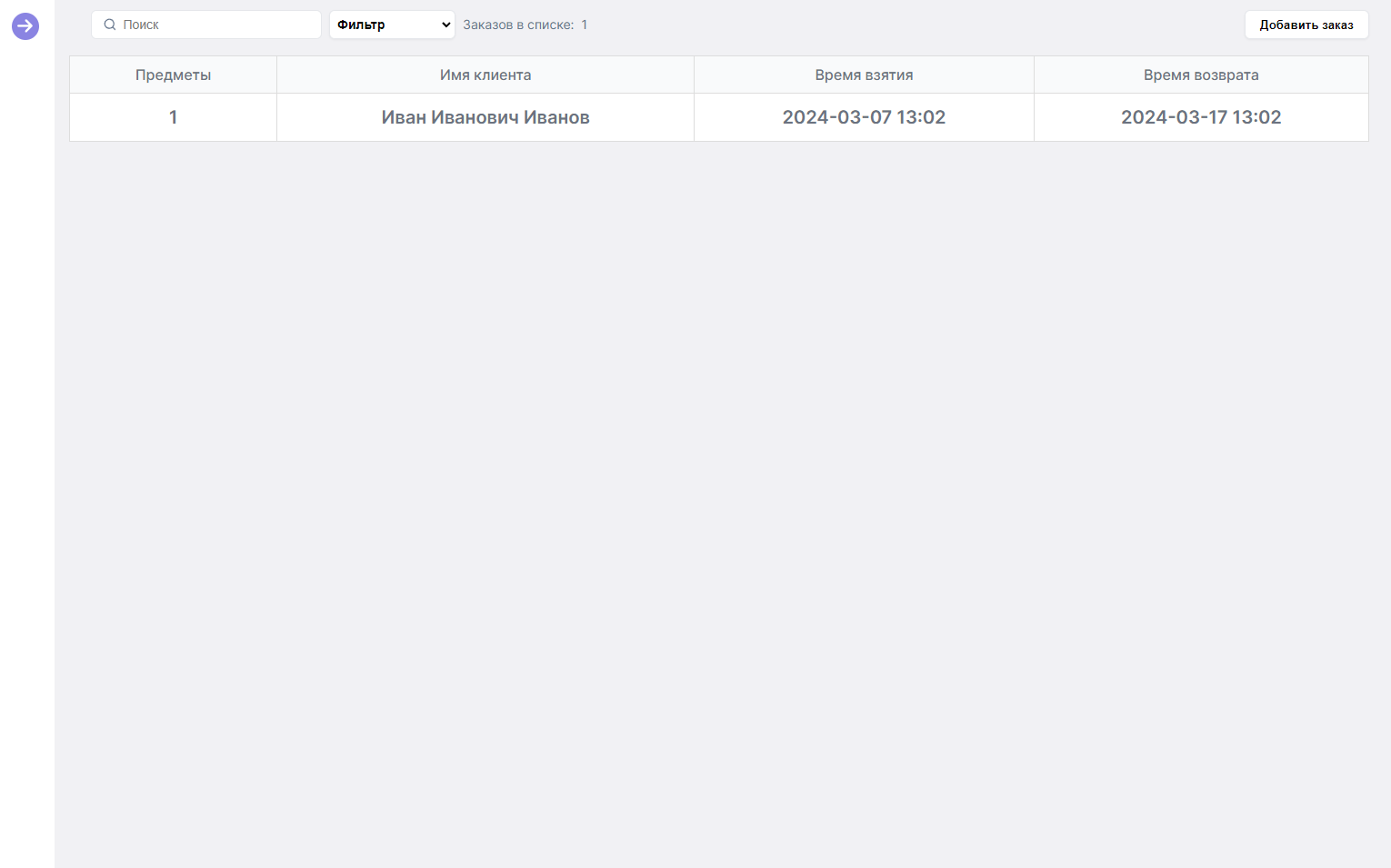


Рисунок 27. Появление записи в «Аренде»

Появилось, если я нажму на эту аренду, то мне покажется вся информация этого заказа. И изменим до другого дня (Рисунок 28).

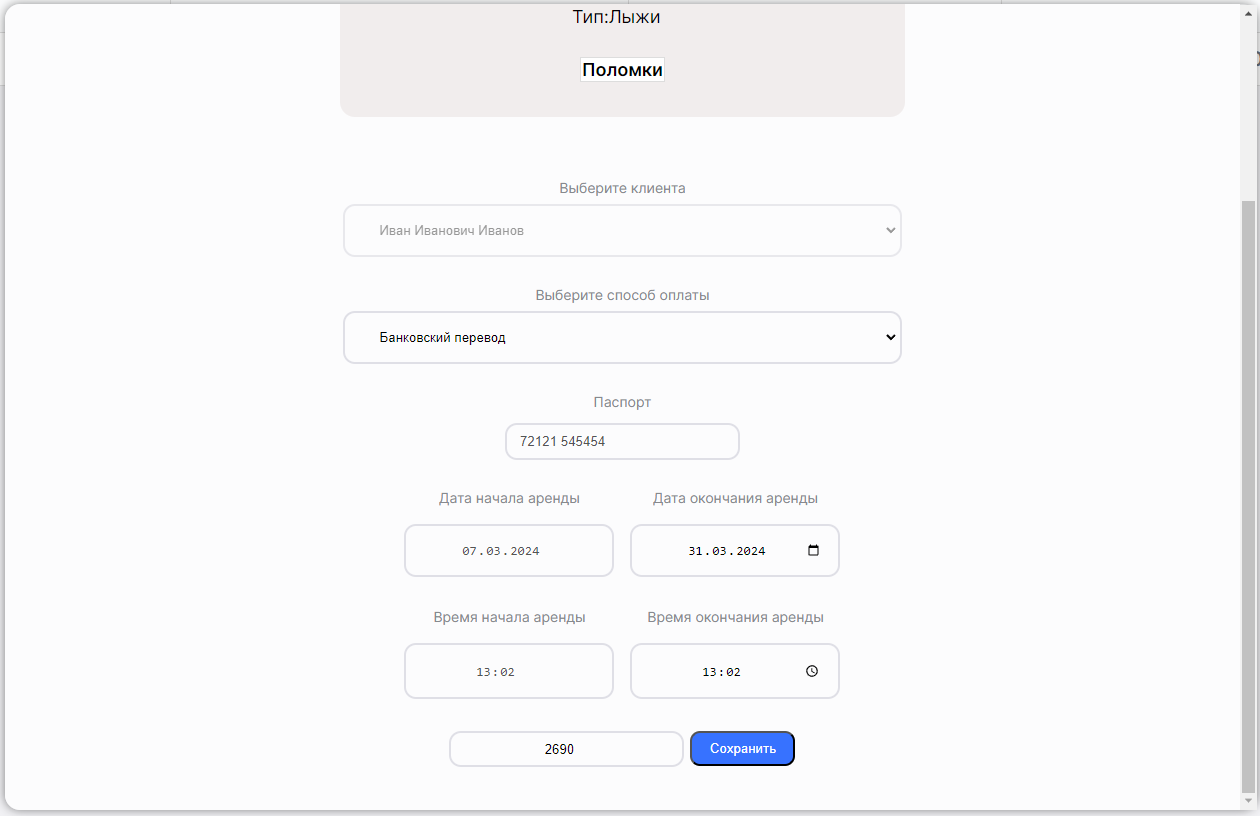


Рисунок 28. Изменение данных в заказе

Проверяем, показывает ли статус «В аренде». Заходим на страницу «Инвентарь» (Рисунок 29).



Рисунок 29. Появилось статус «В аренде»

Да и показывают, до какого числа.

1. Создаем поломку предмета. Добавляю новый предмет, а после перехожу на страницу «Сервис» (Рисунок 30).

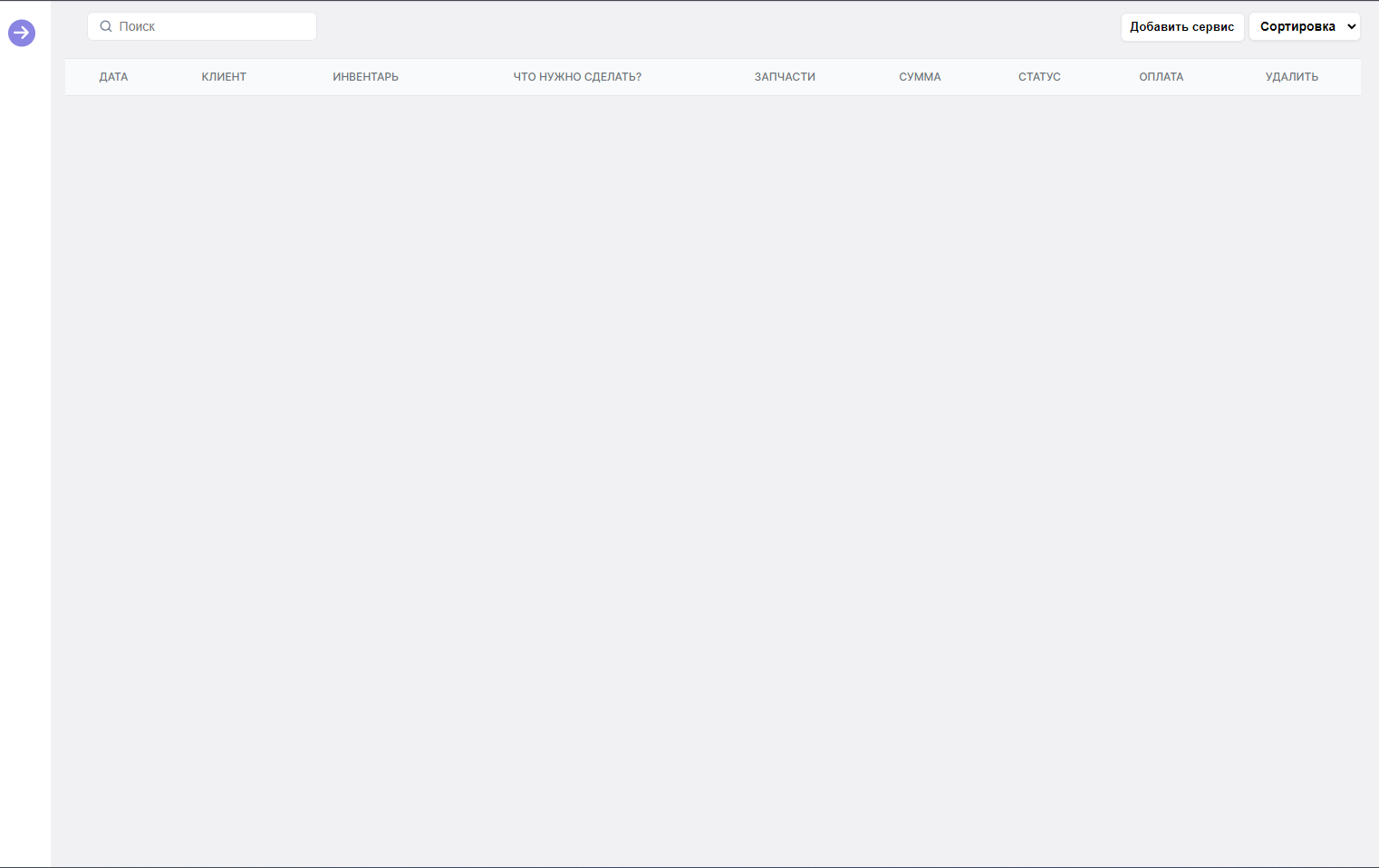


Рисунок 30. Страница «Сервис»

«Добавить сервис» и показывается окошко (Рисунок 31).

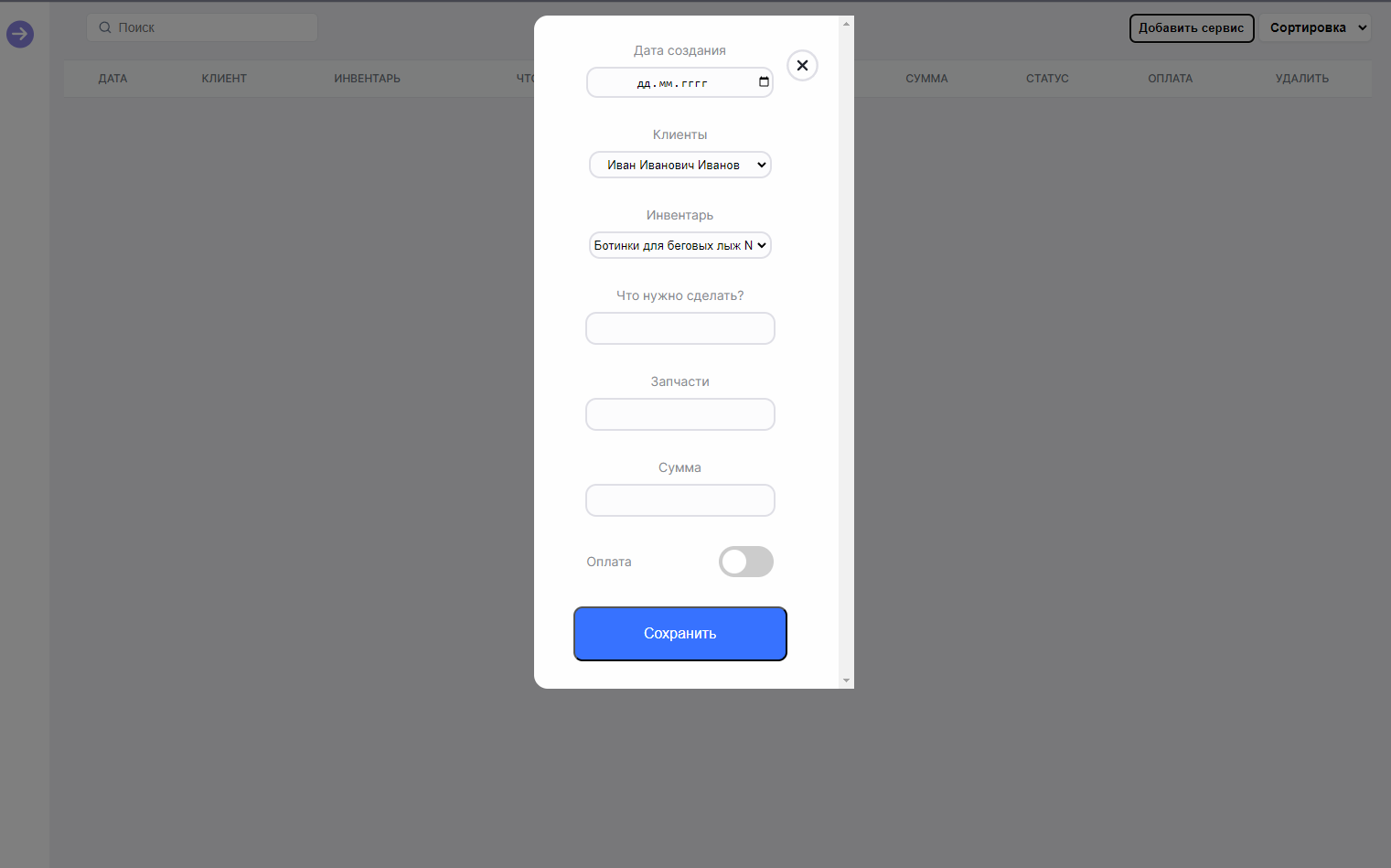


Рисунок 31. Модальное окно «Добавить сервис»

Выбираем, когда произошла поломка, какой клиент отвечал за предмет, что за предмет, что нужно починить, стоимость покупки запчасти, сумма оплаты, и оплачено или нет. Ввожу данные для ботинок, которые создал недавно (Рисунок 32).



Рисунок 32. Введение данные

Сохраняем и смотрим результат (Рисунок 33).

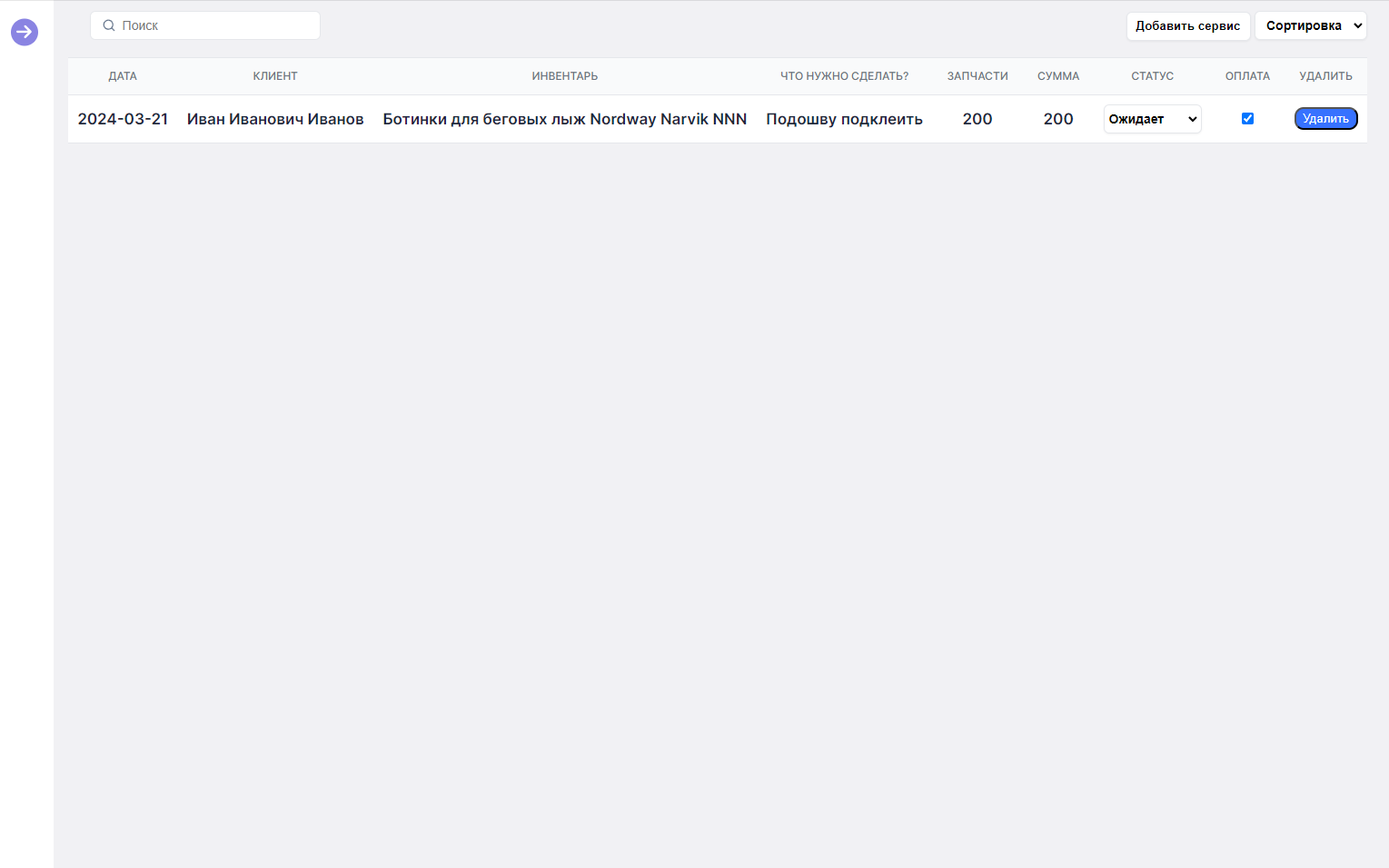
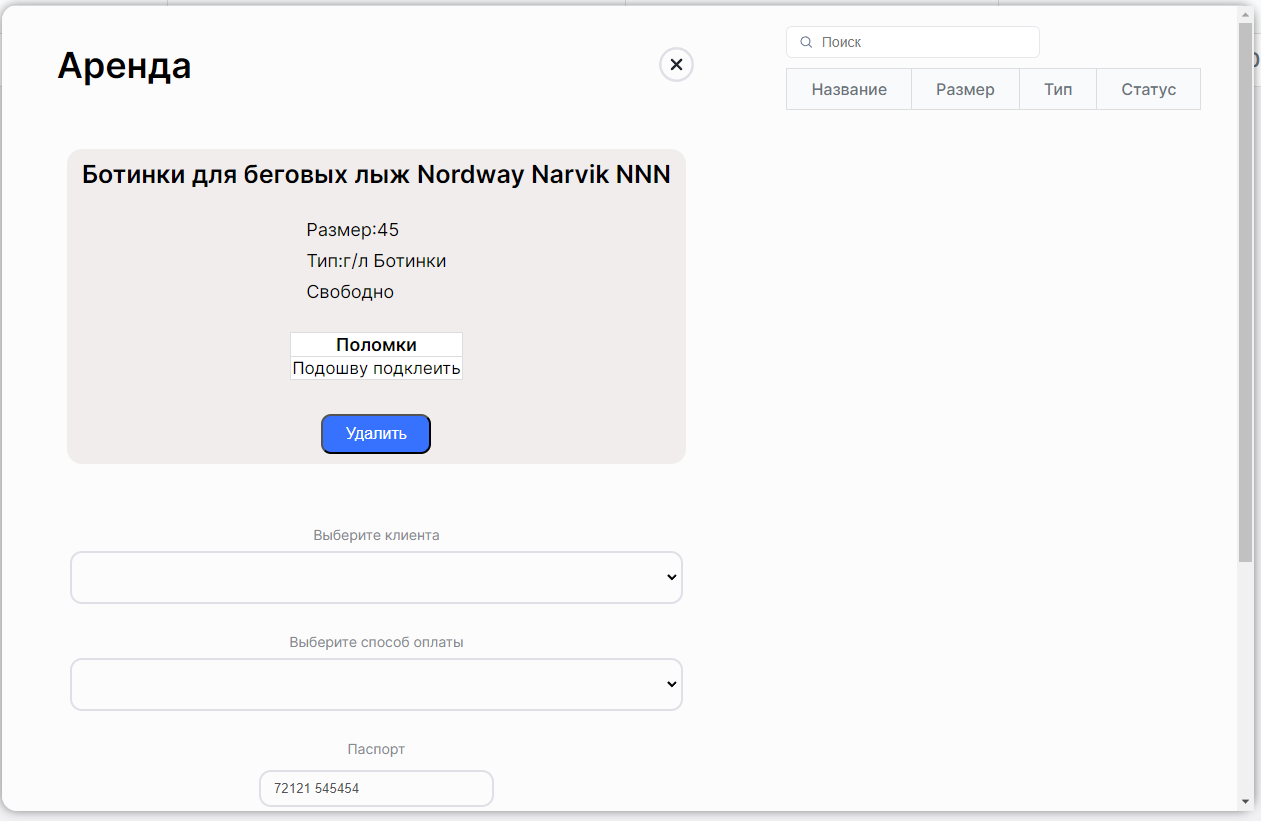


Рисунок 33. Появление первой записи

Расписывается полностью данные о поломки предмета и сейчас статус находиться в «Ожидает». Существуют три статуса: ожидает, в работе, выполнено.

Теперь, если мы создадим новую аренду с этим предметом, еще не починив. То будет выводиться такая строчка (Рисунок 34).



Нам пишет, что этот предмет в состоянии поломки, то есть «Подошву подклеить».

1. Захожу на страницу «Магазин». Тут храниться БД о продуктов магазина (Рисунок 35).

Рисунок 34. Появление информации о том, что предмет сломан.

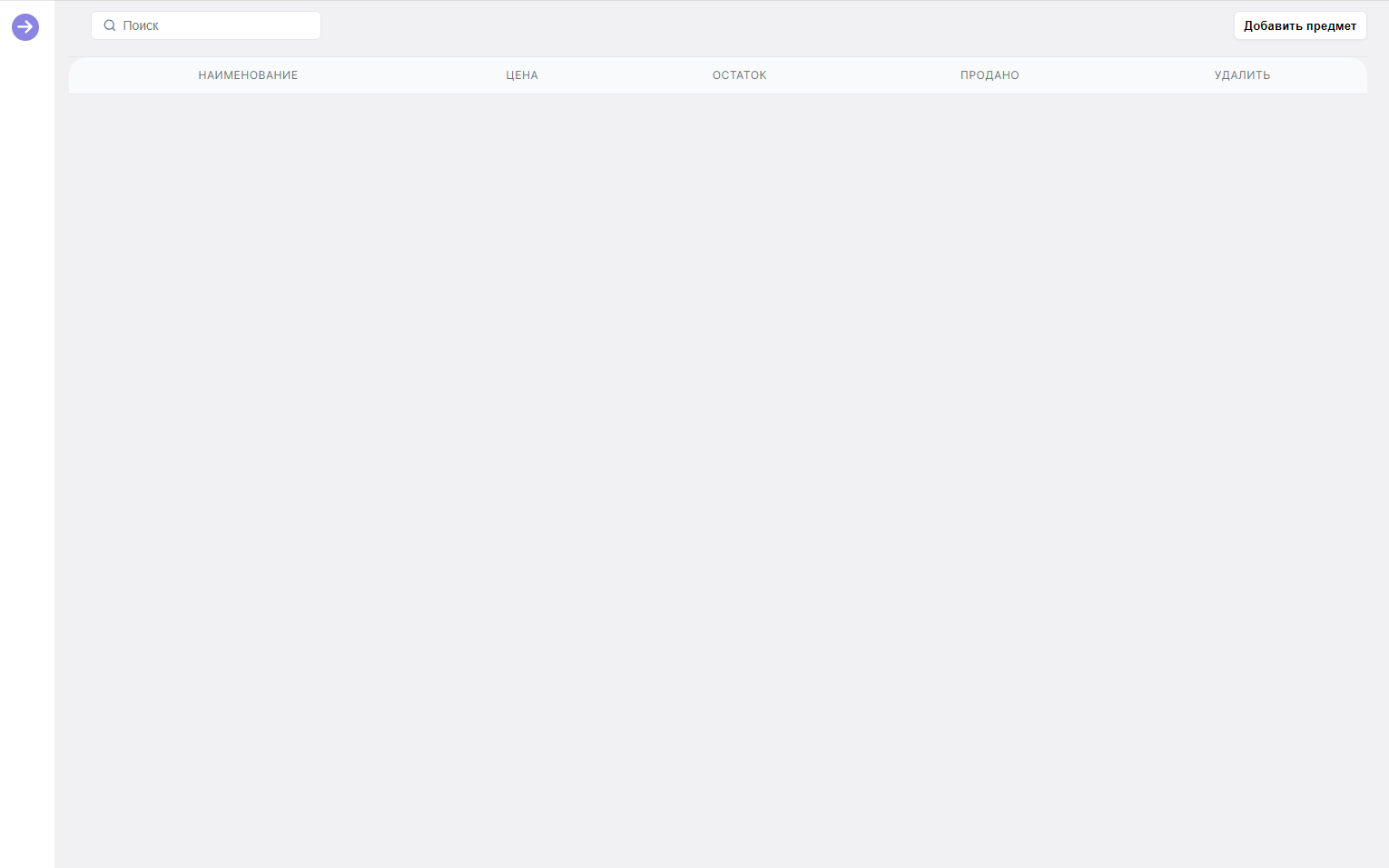


Рисунок 35. Страница «Магазин»

Добавляем, допустим, «Сок Сады Придонья Яблоко и Черная смородина». Нажимаю на «Добавить предмет» и ввожу данные (Рисунок 36).

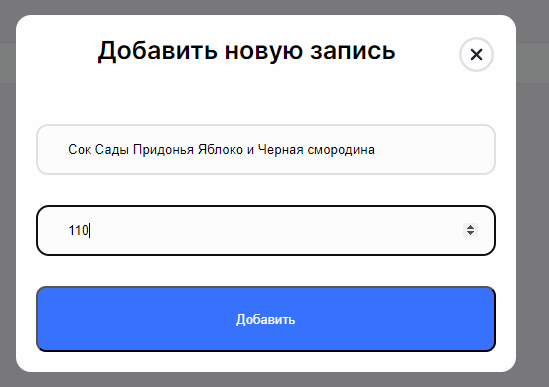


Рисунок 36. Модальное окно «Добавить новую запись» и ввел данные

Я вписал название и цена его.

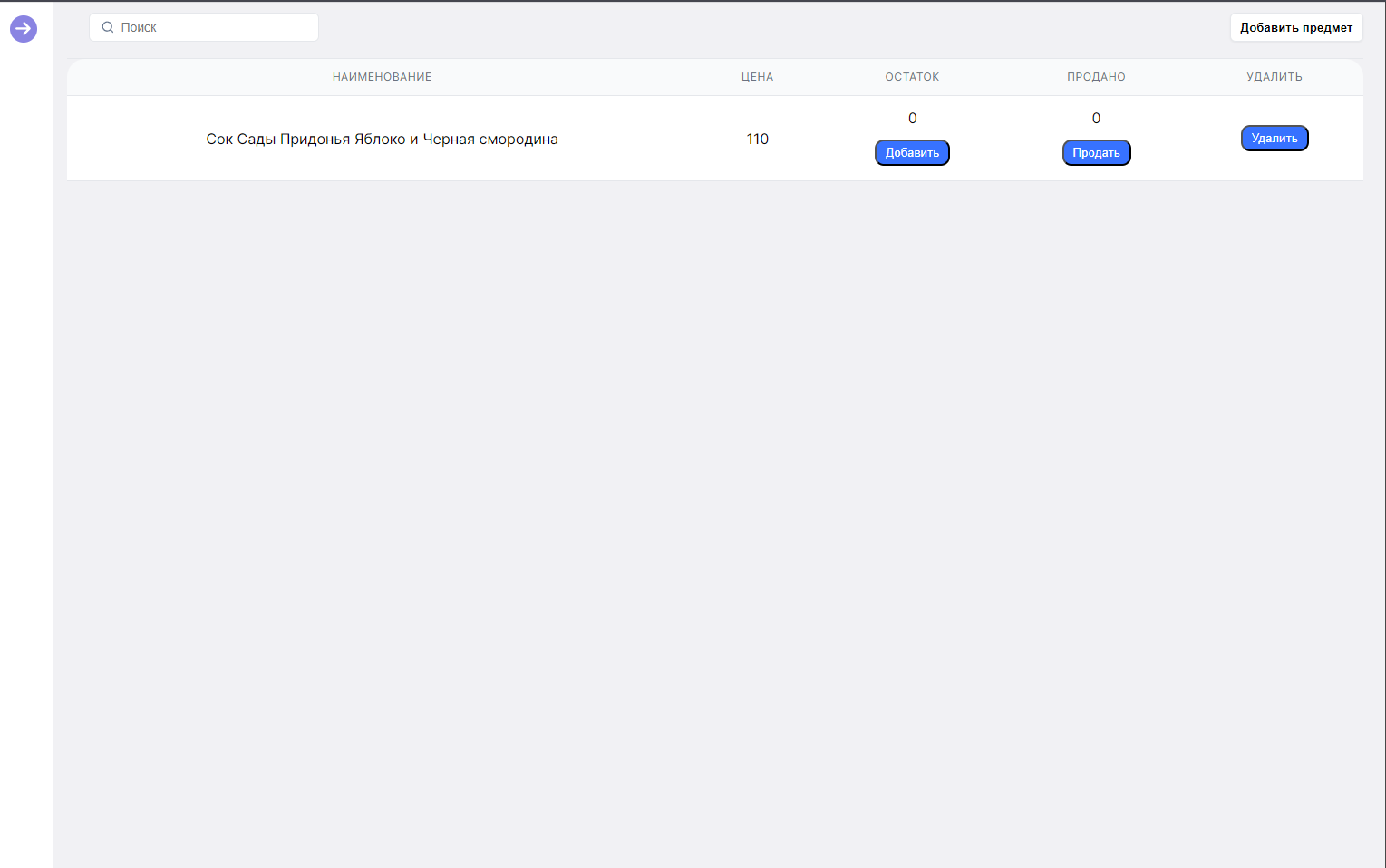


Рисунок 37. Появление первых данных

Я могу вписать, сколько всего у меня и по одному разу нажимать кнопку «Продать», когда клиент купил сок (Рисунок 38).



Рисунок 38. Количество товара

Вписываем количество сока. И продаем 3 сока. Смотрю на результат (Рисунок 39).

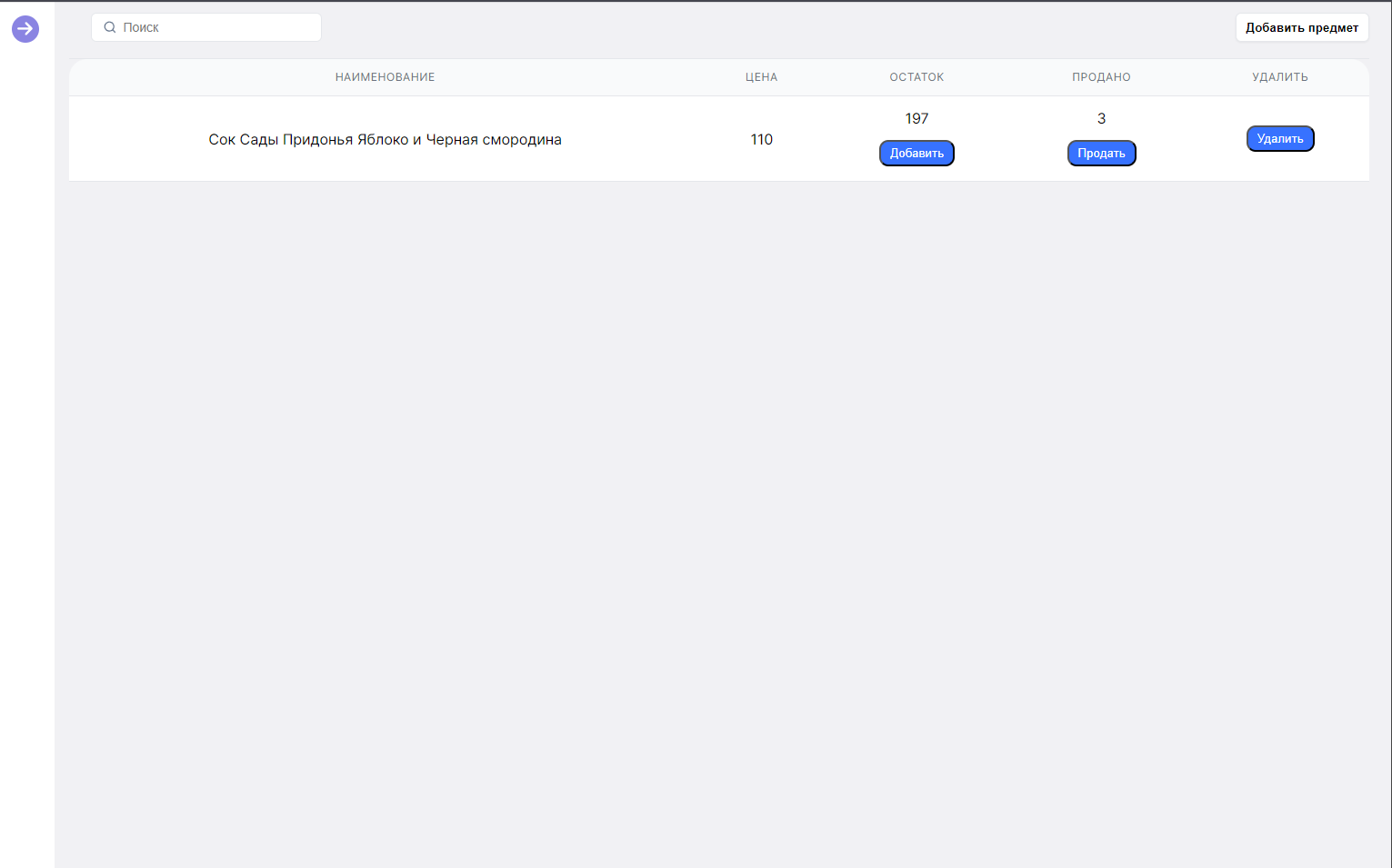


Рисунок 39. Изменение количество товара

1. Теперь захожу на страницу «Задачи». Здесь будут храниться задачи для сотрудников (Рисунок 40).

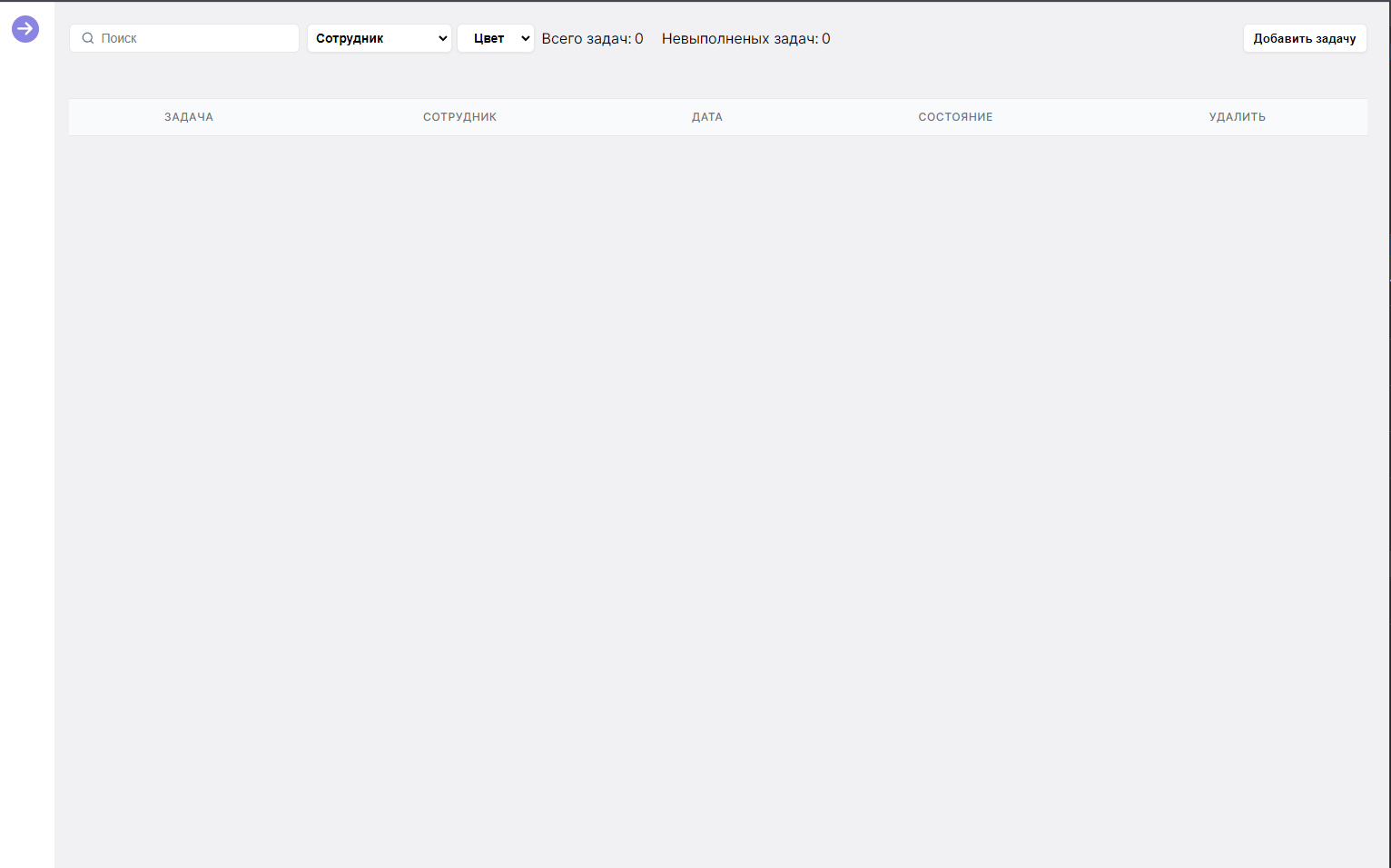


Рисунок 40. Страница «Задачи»

Чтоб создать задачу, я создам сотрудника. Перехожу на страницу «Сотрудники» и добавляю сотрудника (Рисунок 41).

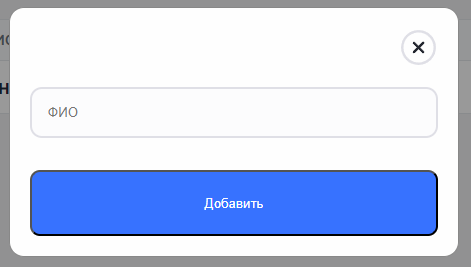


Рисунок 41. Модальное окно «Добавить соотрудника»

Вписываю ФИ, можно и отчество и добавляю (Рисунок 42).

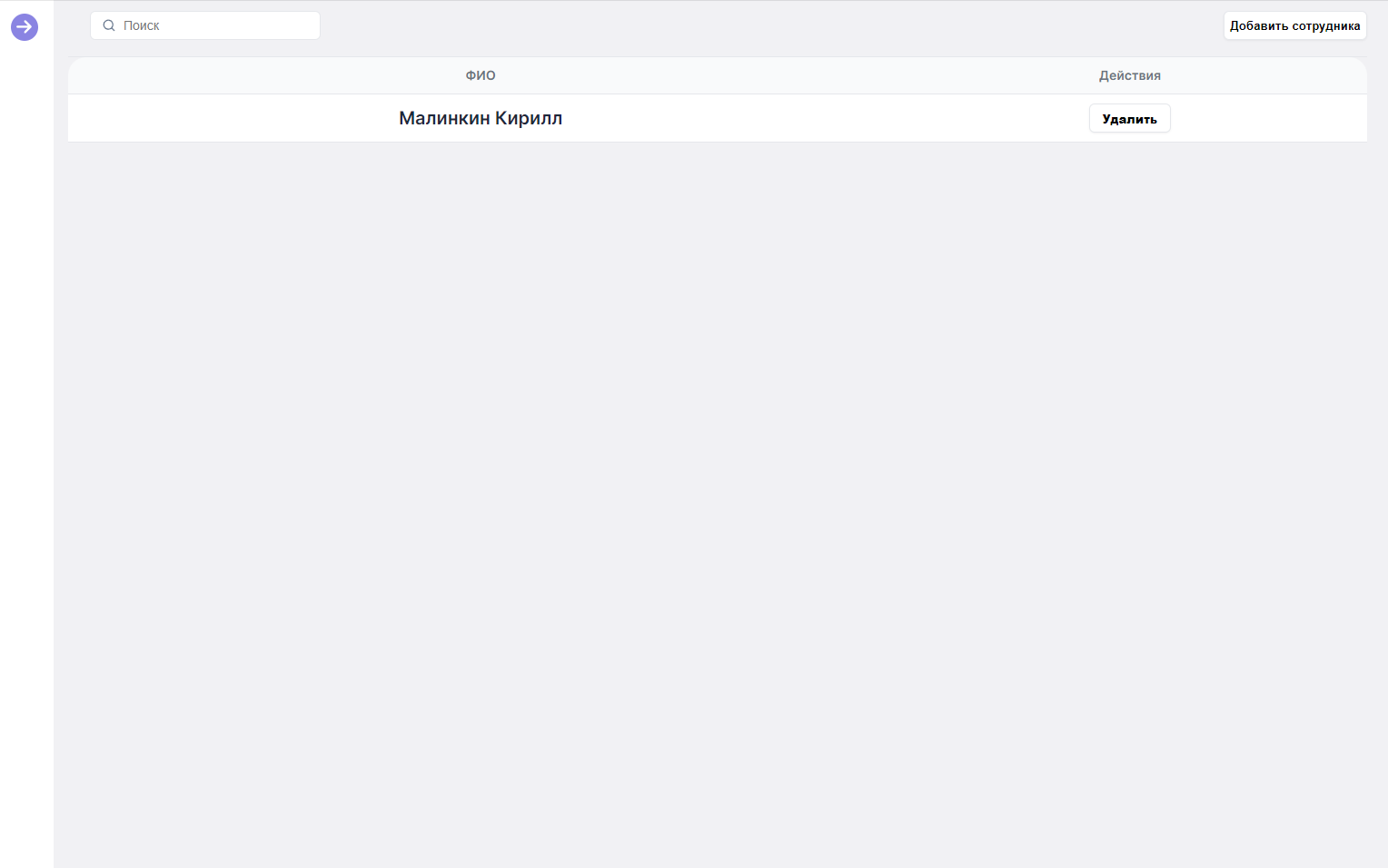


Рисунок 42. Страница «Сотрудники» и первый сотрудник

Теперь создаю задачу в «Добавить задачи» (Рисунок 43).

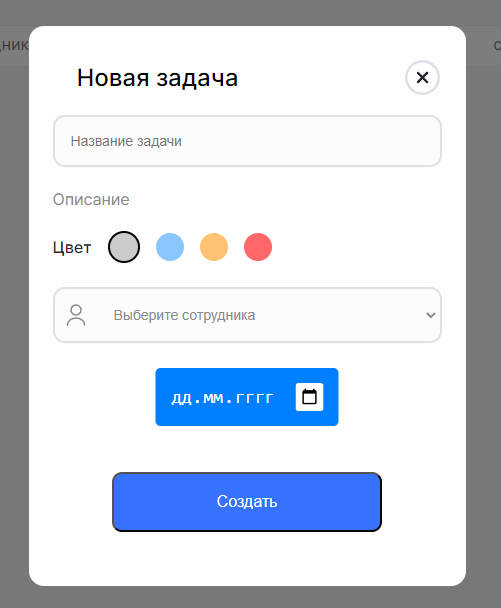


Рисунок 43. Модальное окно «Новая задача»

Вписываю данные, выбираю цвет, сотрудника выбираю «Малинкин Кирилл» и день когда появилась задача.

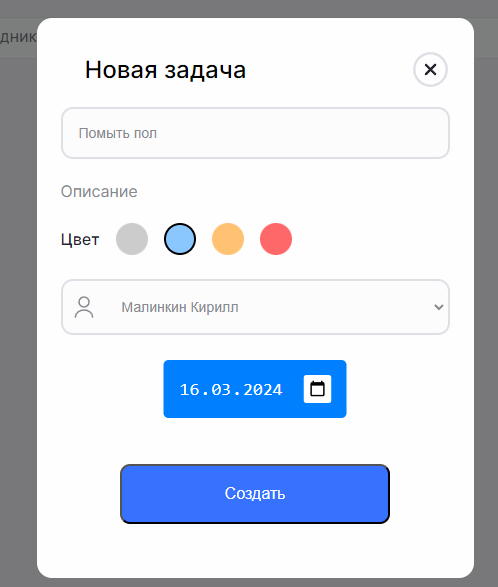


Рисунок 44. Введение данных в модальном окне



Рисунок 45. Первая запись в «Задачи»

Создалось и также есть 3 состояния: ожидает, в работе, выполнено (Рисунок 45).

1. Последняя страница «Календарь» (Рисунок 46).



Рисунок 46. Страница «Календарь»

Верхнем правом углу есть меню управление календаря. Нажимаю на него, показывается список действие странице (Рисунки 47).

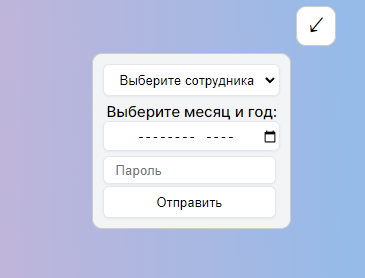


Рисунок 47. Список действие «Календаря»

Первое: мы выбираем сотрудника, который покажет его расписание на месяц. Второе: выбираем месяц и год. Третья, четвертая: нужна для администратора расписания.

Выберем сотрудника и месяц (Рисунки 48).

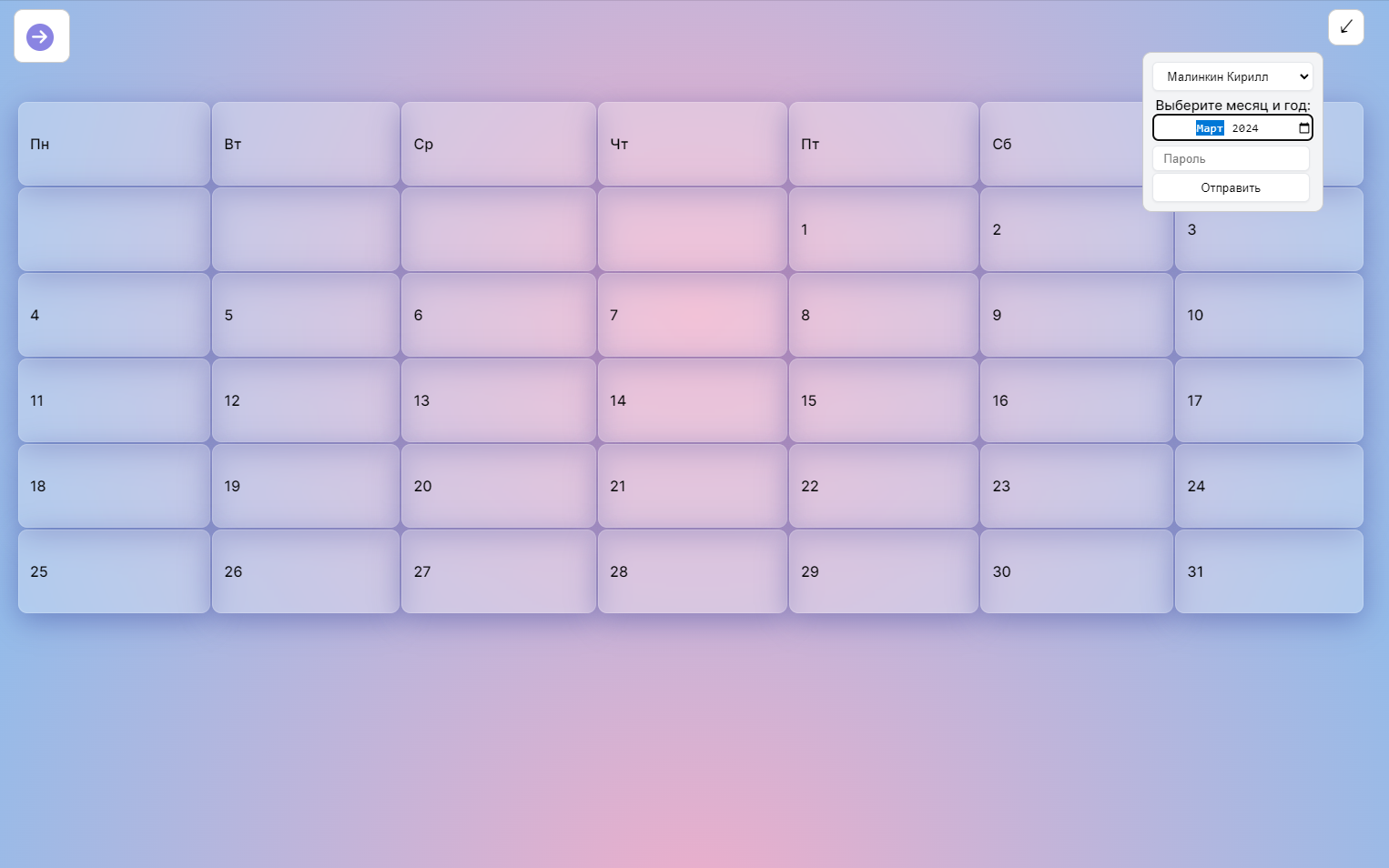


Рисунок 48. Календарь – март 2024

У меня появилось расписание. Но нет данных сотрудника, когда он работал, болел или был в отпуске или вообще не работал. Для этого нам нужен панель «Пароль» и «Отправить». Вводим пароль администратора (у меня прописано adminqwerty) и отправляем. Высвечивается панель администратора (Рисунок 49).

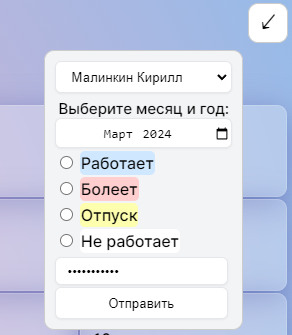


Рисунок 49. Админ панель

Теперь если мы нажмем на пункт «Работает» и на дату, то в БД заноситься данные о том, что этот сотрудник в тот день, месяц и год работал (Рисунок 50).

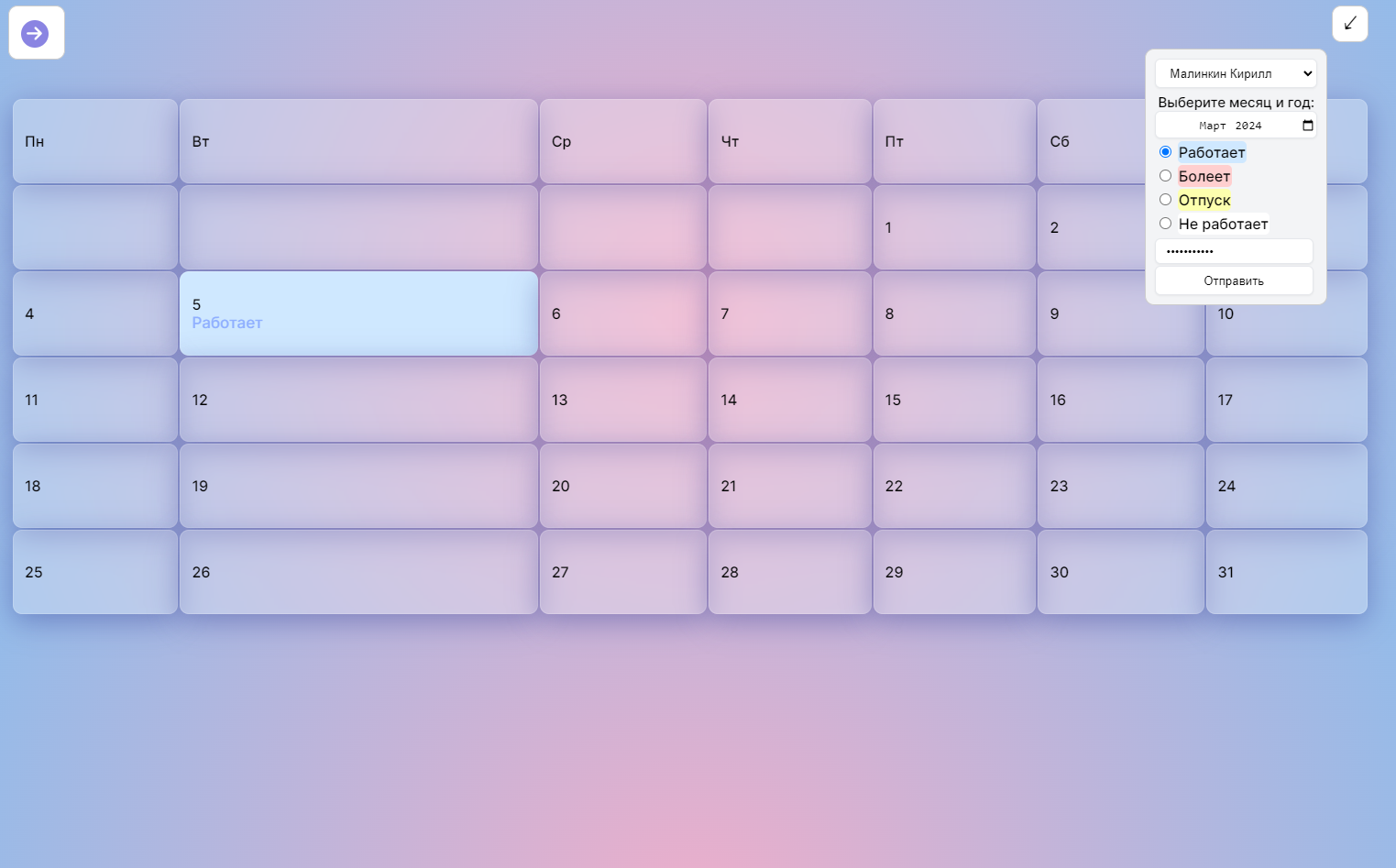


Рисунок 50. Добавил «Работает» на календарь

И так можно заполнить весь календарь, используя при этом и другие (Рисунок 51).



Рисунок 51. График работы сотрудника

1. После добавление первых записей, на сайте появилось первая статистика за день (Рисунок 52)

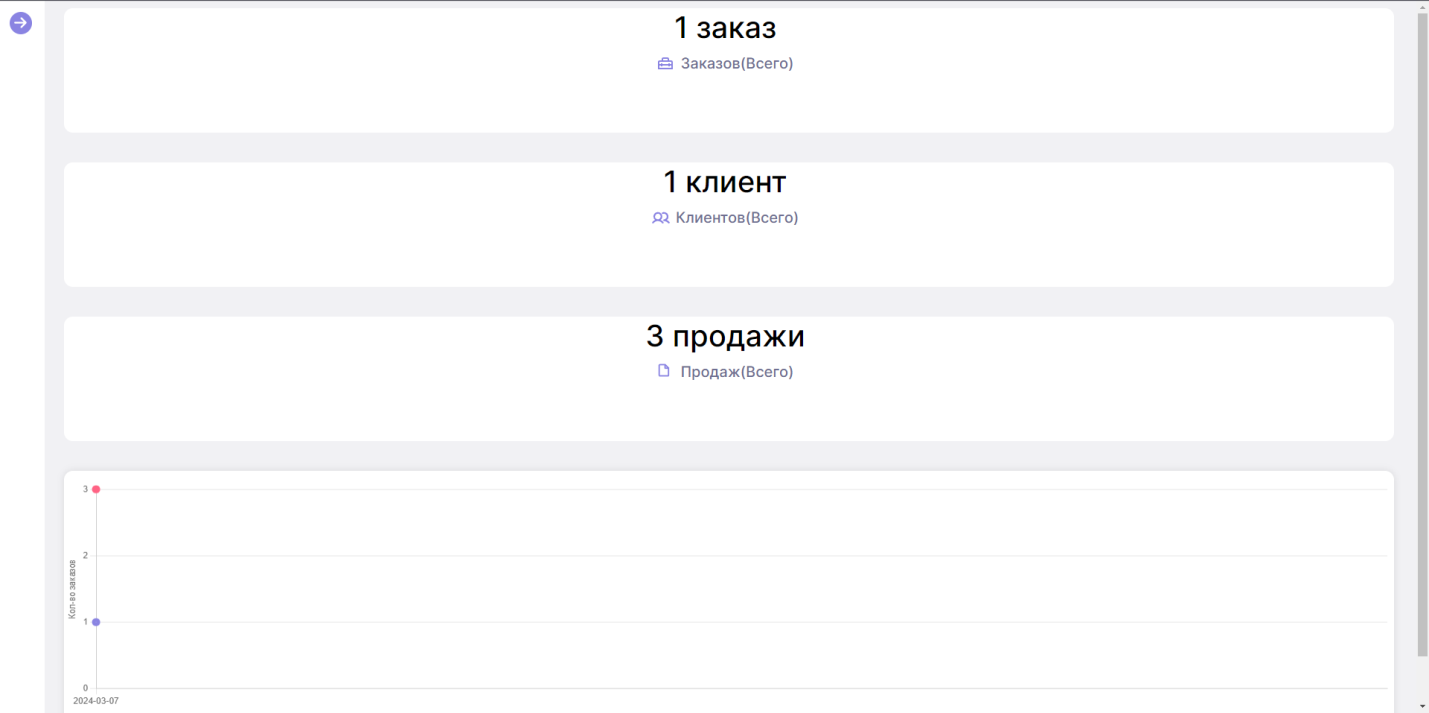


Рисунок 52. Появление первых дынных на странице «Статистика»

**Вывод**: в ходе учебной практики по созданию информационной системы для управления зимней турбазы, я получил ценный опыт и практические навыки в области информационного и системного программирования. Разработка данной системы позволила мне применить знания, полученные в процессе обучения, на практике и оценить их эффективность в реальном проекте.

В ходе работы над информационной системой я освоил различные инструменты и технологии программирования, углубил свои знания в области управления базами данных, взаимодействия с пользовательским интерфейсом и архитектурой программных систем. Кроме того, я приобрел навыки командной работы, планирования проектов и анализа требований заказчика.

Важным результатом учебной практики стало не только создание работоспособной информационной системы, но и умение адаптировать ее под изменяющиеся потребности и условия эксплуатации. Этот опыт будет полезен мне в будущем как специалисту в области информационных технологий, а также поможет мне успешно реализовывать иные проекты в этой сфере.

Таким образом, учебная практика по разработке информационной системы для управления зимней турбазы стала важным этапом моего профессионального развития, обогатив мой опыт и знания в области информационного и системного программирования.