

Информационная система сервиса аренды оборудования

Соломенников Николай 22204

Введение

Сервис аренды оборудования позволяет клиентам временно пользоваться инструментами, техникой или устройствами без необходимости их покупки. Это удобно для тех, кому оборудование нужно на короткий срок.

Клиент ищет оборудование, бронирует его на нужные даты и забирает в пункте выдачи. Менеджер оформляет аренду, выдает оборудование, контролирует его состояние при возврате и взаимодействует с клиентами. Администратор управляет каталогом оборудования, настраивает тарифы и штрафы, а также анализирует отчеты по доходам и популярности оборудования.

Назначение приложения

Приложение нужно для автоматизации процессов управления арендой оборудования. Оно поможет компании, занимающейся арендой, эффективно управлять запасами оборудования, оформлять аренду, отслеживать возвраты и состояние оборудования, а также вести финансовую отчетность.

Целевые пользователи:

- **Администраторы:** управление каталогом оборудования, регулировка цен.
- **Менеджеры по аренде:** взаимодействие с клиентами, оформление аренды, выдача оборудования, контроль сроков возврата.
- **Клиенты:** получение информации о наличии и стоимости оборудования, бронирование.

Полный цикл аренды оборудования

1. Бронирование оборудования:

- a. Клиент выбирает оборудование на сайте.
- b. Указывает: ФИО, контактный телефон и email, желаемые даты аренды (начало и конец).
- c. Система резервирует оборудование на указанный срок.

2. Подготовка к выдаче оборудования:

- a. Менеджер в пункте выдачи видит информацию о новом бронировании.
- b. Оборудование подготавливается к выдаче (проверяется его состояние, комплектность).
- c. Если клиент не появляется в назначенное время, менеджер связывается с ним для уточнения:
 - i. Если клиент отказывается от аренды, бронь отменяется, оборудование снова становится доступным.
 - ii. Если клиент подтверждает задержку, бронь переоформляется на новые даты.

3. Оформление аренды:

- a. Клиент приходит в пункт выдачи.
- b. Менеджер проверяет документы клиента.
- c. Клиент подписывает договор аренды, в котором указаны: сроки аренды, стоимость аренды, размер залога, условия возврата и штрафы за возможные повреждения.
- d. Клиент вносит залог.
- e. Менеджер выдает оборудование клиенту.

4. Использование оборудования

- a. Клиент использует оборудование в течение оговоренного срока.
- b. Если клиенту требуется продлить аренду, он может связаться с менеджером для продления сроков возврата.

5. Возврат оборудования:

- a. Клиент возвращает оборудование в пункт выдачи и оплачивает аренду за столько дней, сколько он пользовался оборудованием.
- b. Менеджер осматривает оборудование на предмет повреждений:
 - i. Если повреждений нет, залог возвращается клиенту.
 - ii. Если есть повреждения, менеджер оценивает их стоимость и вычитает ее из залога. Остаток залога возвращается клиенту.
- c. Если клиент не вернул оборудование в срок, менеджер связывается с ним для напоминания.

6. Завершение аренды:

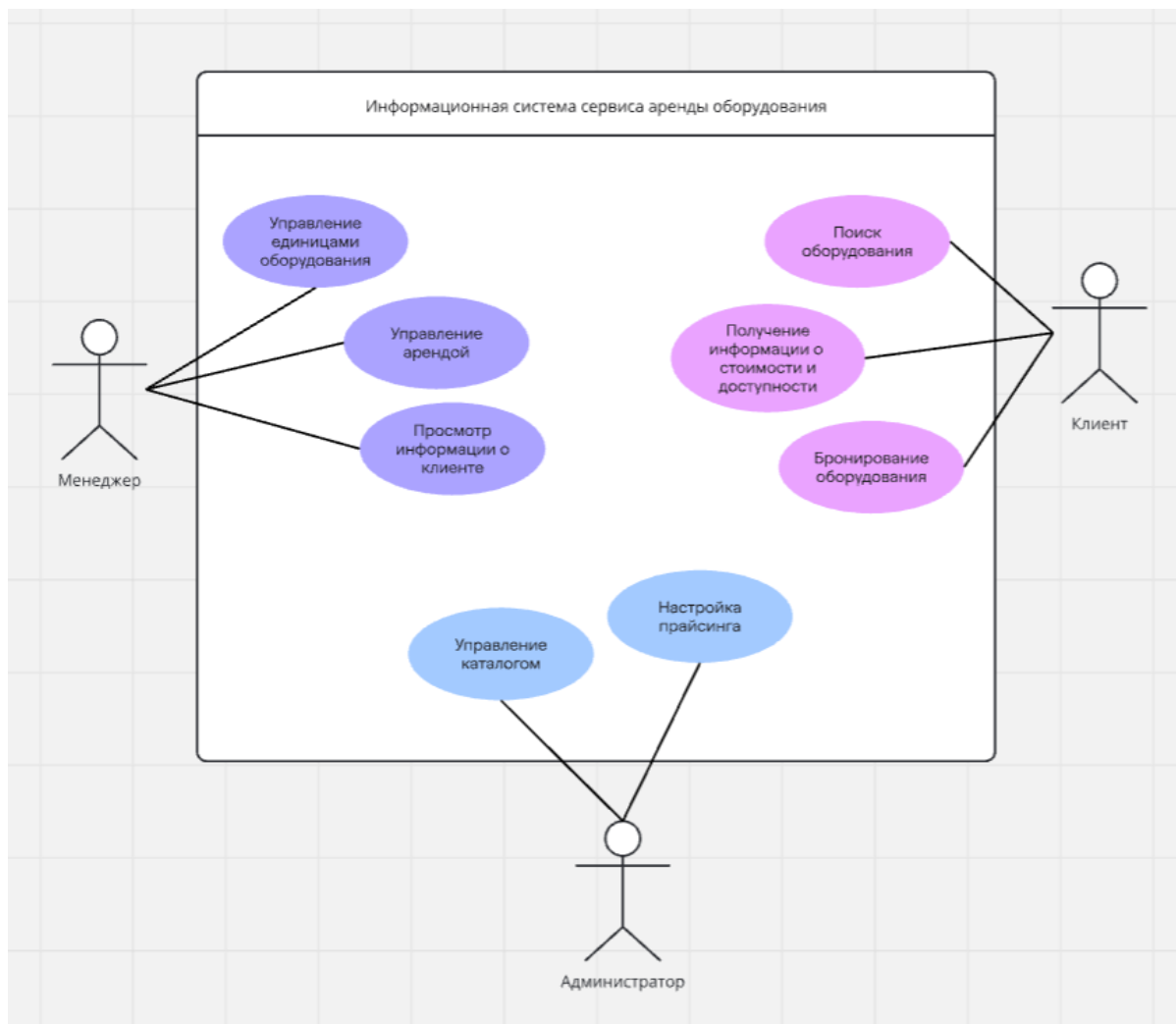
- a. Менеджер закрывает аренду в системе:
 - i. Фиксирует штрафы, если они были.

- ii. Обновляет состояние оборудования (например, отправляет его на обслуживание, если это необходимо).
- b. Клиенту предоставляется чек или отчет о завершении аренды.

Бизнес-процессы системы

- **Оформление аренды:**
 - Поиск доступного оборудования.
 - Оформление договора аренды.
 - Оплата и внесение залога.
 - Получение оборудования.
- **Возврат оборудования:**
 - Проверка состояния оборудования.
 - Начисление штрафов за повреждения.
 - Возврат залога.
 - Обновление статуса оборудования.
- **Взаимодействие с клиентами:**
 - Ответ на вопросы клиентов.
 - Оформление аренды менеджером.
 - Напоминание о сроках возврата.
- **Учет оборудования:**
 - Добавление, обновление и удаление информации об оборудовании.
 - Учет состояния оборудования (доступно, забронировано, в аренде, на обслуживании, списано).
 - Учет технического обслуживания и ремонта.

Бизнес-процессы для каждого типа пользователей



Администраторы:

- Управление каталогом оборудования (добавление / удаление карточек оборудования, редактирование описания).
- Настройка прайсинга (тарифы, залог).

Менеджеры по аренде:

- Управление конкретными единицами оборудования (добавление / удаление, изменение статуса (доступно, забронировано, в аренде, на обслуживании, списано)).
- Управление арендой (снятие брони, подтверждение аренды, продление аренды, начисление штрафов)
- Просмотр информации о клиенте, его истории аренд.

Клиенты:

- Поиск оборудования.
- Получение информации о его стоимости и доступности.

- Бронирование оборудования.

Информационные потоки для каждого типа пользователей

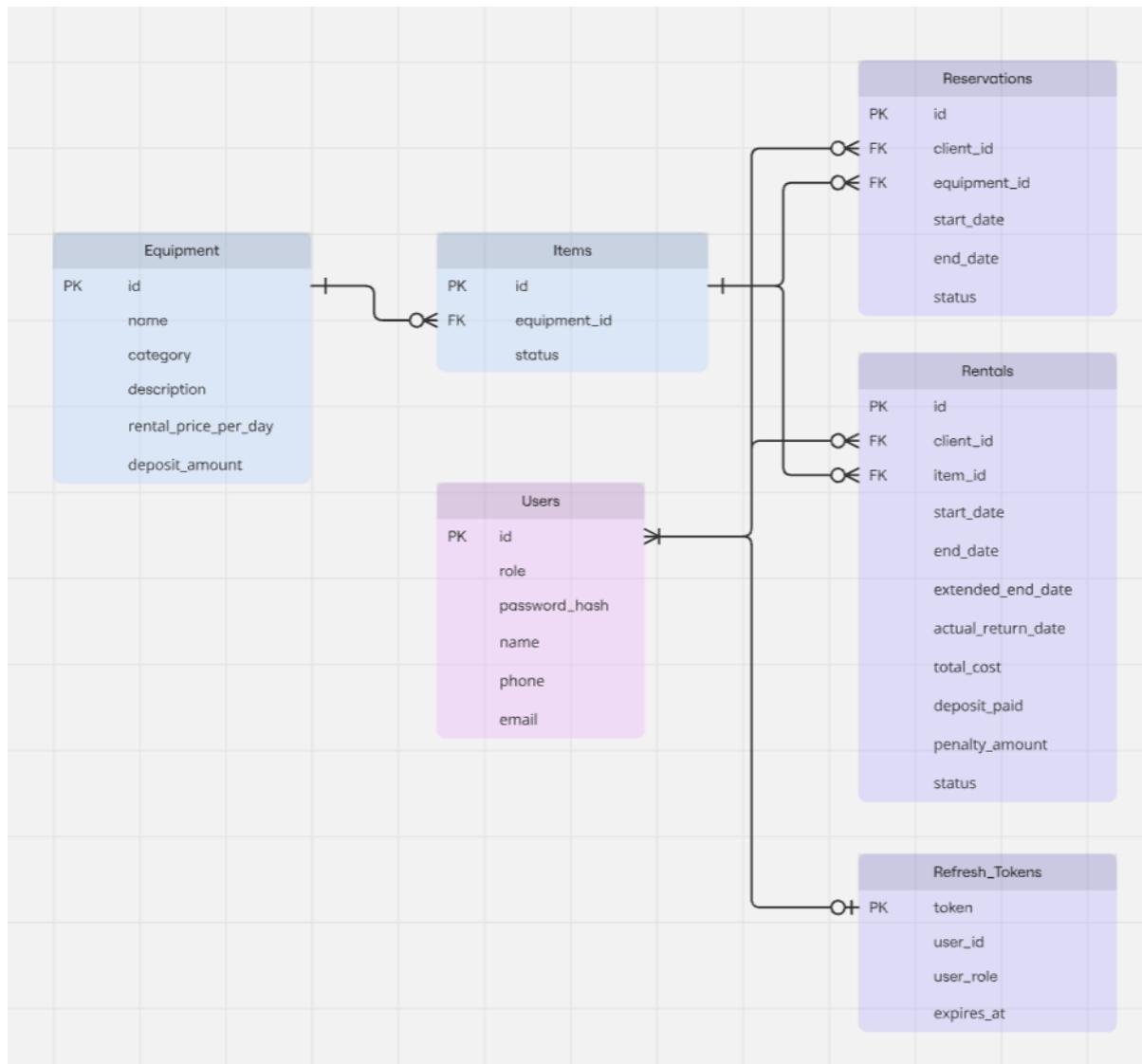


Система пользователей иерархическая:

Клиент < Менеджер < Администратор < Разработчик

То есть, например, всё, что видит менеджер на сайте, видит и администратор с разработчиком; всё, что может менеджер может и администратор с разработчиком. В контекстной диаграмме это явно не указано, но стоит это учитывать.

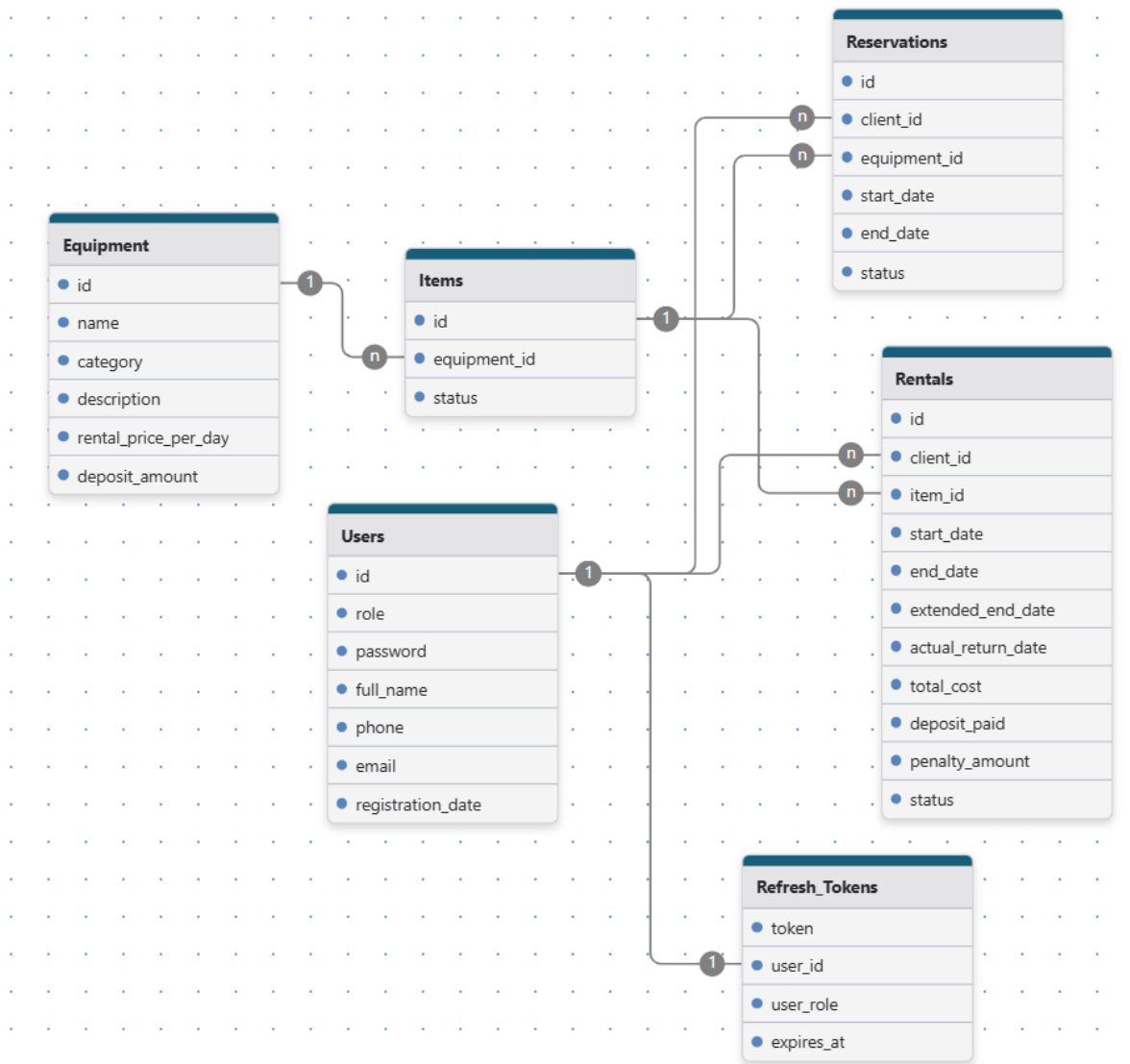
Основные информационные объекты и связи



Сущности:

- Equipment - оборудование (например, “Перфоратор Bosch”)
- Items - единицы обрудования. Представляют собой конкретные физические вещи определённого оборудования (например, перфораторов Bosch может быть несколько в обороте, и каждый из них можно отдельно арендовать)
- Users - пользователи (клиенты, менеджеры, админы)
- Reservations - брони
- Rentals - аренды
- Refresh_Token – Refresh токены (нужны для авторизации)

Схема базы данных



1. Оборудование (Equipment):

- id (уникальный идентификатор)
- name (название оборудования)
- category (категория оборудования)
- description (описание)
- rental_price_per_day (стоимость аренды за день)
- deposit_amount (сумма залога)

2. Единицы оборудования (Items):

- id (уникальный идентификатор среди всех оборудований)
- equipment_id (внешний ключ на таблицу Equipment)
- status (статус: доступно, забронировано, в аренде, на обслуживании, списано) - данное поле меняется автоматически с помощью триггеров

3. Пользователи (Users):

- id (уникальный идентификатор)
- user_role (роль: администратор, менеджер, клиент)
- password_hash (хэш пароля)
- name (ФИО или просто имя)
- phone (контактный телефон)
- email (электронная почта)

4. Бронирования (Reservations):

- id (уникальный идентификатор)
- client_id (внешний ключ на таблицу Users)
- equipment_id (внешний ключ на таблицу Equipment)
- start_date (планируемая дата начала аренды)
- end_date (планируемая дата окончания аренды)
- status (статус: активна, отменена, завершена)

5. Аренда (Rentals):

- id (уникальный идентификатор)
- client_id (внешний ключ на таблицу Users)
- item_id (внешний ключ на таблицу Items)
- start_date (дата начала аренды)
- end_date (дата окончания аренды)
- extended_end_date (продлённая дата окончания)
- actual_return_date (фактическая дата возврата)
- total_cost (сумма, заплаченная за аренду)
- deposit_paid (сумма залога, внесенная клиентом)
- penalty_amount (штраф за повреждения)
- status (статус: активна, завершена)

6. Refresh Токены (Refresh_Tokens):

- token (токен)

- user_id (внешний ключ на таблицу Users)
- user_role (клиент, менеджер или админ)
- expires_at (дата истечения токена)

Скрипт schema.sql для создания схемы БД

```
1 CREATE TABLE IF NOT EXISTS Equipment (  
2     id INT AUTO_INCREMENT PRIMARY KEY,  
3     name VARCHAR(255) NOT NULL,  
4     category VARCHAR(255) NOT NULL,  
5     description TEXT,  
6     rental_price_per_day INT NOT NULL,  
7     deposit_amount INT NOT NULL  
8 );  
9  
10 CREATE TABLE IF NOT EXISTS Items (  
11     id INT AUTO_INCREMENT PRIMARY KEY,  
12     equipment_id INT NOT NULL,  
13     status ENUM('available', 'booked', 'rented', 'serviced', 'decommissioned') NOT NULL,  
14     FOREIGN KEY (equipment_id) REFERENCES Equipment(id)  
15 );  
16  
17 CREATE TABLE IF NOT EXISTS Users (  
18     id INT AUTO_INCREMENT PRIMARY KEY,  
19     user_role ENUM('client', 'manager', 'admin') NOT NULL,  
20     password_hash VARCHAR(255),  
21     name VARCHAR(255) NOT NULL,  
22     phone VARCHAR(50),  
23     email VARCHAR(255)  
24 );  
25  
26 CREATE TABLE IF NOT EXISTS Reservations (  
27     id INT AUTO_INCREMENT PRIMARY KEY,  
28     client_id INT NOT NULL,  
29     equipment_id INT NOT NULL,  
30     start_date DATE NOT NULL,  
31     end_date DATE NOT NULL,  
32     status ENUM('active', 'cancelled', 'completed') NOT NULL,  
33     FOREIGN KEY (client_id) REFERENCES Users(id),  
34     FOREIGN KEY (equipment_id) REFERENCES Equipment(id)  
35 );  
36  
37 CREATE TABLE IF NOT EXISTS Rentals (  
38     id INT AUTO_INCREMENT PRIMARY KEY,  
39     client_id INT NOT NULL,  
40     item_id INT NOT NULL,  
41     start_date DATE NOT NULL,  
42     end_date DATE NOT NULL,  
43     extended_end_date DATE,  
44     actual_return_date DATE,  
45     deposit_paid INT,  
46     penalty_amount INT DEFAULT 0,  
47     total_cost INT DEFAULT 0,  
48     status ENUM('active', 'completed') NOT NULL,  
49     FOREIGN KEY (client_id) REFERENCES Users(id),  
50     FOREIGN KEY (item_id) REFERENCES Items(id)  
51 );  
52  
53 CREATE TABLE IF NOT EXISTS Refresh_Tokens (  
54     token CHAR(36) PRIMARY KEY,  
55     user_id INT NOT NULL,  
56     user_role ENUM('client', 'manager', 'admin') NOT NULL,  
57     expires_at TIMESTAMP NOT NULL,  
58     INDEX ix_refresh_tokens_expires (token),  
59     CONSTRAINT fk_refresh_user FOREIGN KEY (user_id) REFERENCES Users(id)  
60 );
```

Ограничения

1. Целостность данных:

- a. Каждая единица оборудования должна быть связана с оборудованием.
- b. Каждая бронь и аренда должна быть связана с существующим клиентом и оборудованием.
- c. Каждый Refresh token должен быть связан с пользователем.

2. Уникальность данных:

- a. Каждая единица оборудования должна имеет уникальный идентификатор (не только среди единиц определённого оборудования, но и среди всех единиц оборудования).
- b. Пользователи должны быть уникально идентифицированы (по имени, email или телефону).

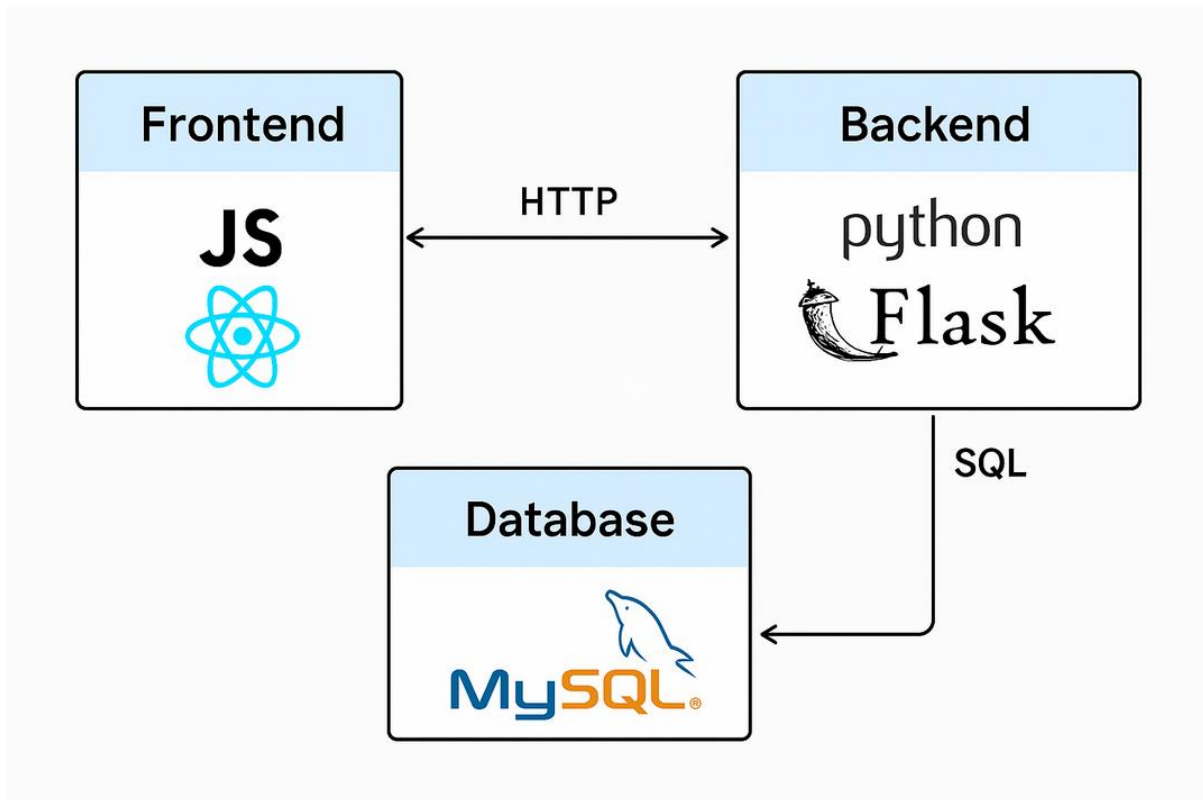
3. Ограничения на ввод данных:

- a. При бронировании дата возврата аренды не может быть раньше даты начала, а также дата начала аренды не может быть раньше сегодняшнего дня.
- b. Стоимость аренды и залог не могут быть отрицательными.

4. Бизнес-правила:

- a. Оборудование может быть забронировано только если есть хотя бы одна единица оборудования со статусом "доступно".
- b. Стоимость аренды рассчитываются автоматически на основе тарифов, даты начала аренды и текущей даты (даты завершения аренды).
- c. Нельзя удалить и изменить статус забронированной или арендованной единицы оборудования.
- d. Нельзя убрать из каталога оборудование, некоторые единицы которого находятся в брони или аренде.

Стек технологий



Фронтенд (Клиентская часть)

- **Языки:** JavaScript (де-факто лучший язык для фронтенда. Работает во всех браузерах без плагинов и компиляции, поддерживает асинхронность, удобен для разработки), CSS (отвечает за внешний вид сайта), HTML (каркас для всего веб приложения).
- **Библиотека:** React — используется для построения интерфейса на основе компонентов. React работает с Virtual DOM, что обеспечивает эффективность, поддерживает хуки (управление состоянием, отклик на изменения в компоненте, эффекты)
- **Сборщик и дев-сервер:** Vite — обеспечивает быструю сборку и горячую перезагрузку во время разработки.

Бэкенд (Серверная часть)

- **Язык программирования:** Python - удобный язык для быстрой разработки.
- **Фреймворк:** Flask — лёгкий веб-фреймворк, предназначенный для создания REST API и быстрой обработки HTTP-запросов.

База данных

- **СУБД:** MySQL
- **Связь с бэкендом:** Через Python-библиотеку mysql-connector.

Взаимодействие компонентов

- **Фронтенд ↔ Бэкенд:**
 - Используются HTTP-запросы к REST API, реализованному во Flask.
 - Данные передаются в формате JSON.
- **Бэкенд ↔ База данных:**
 - Взаимодействие через SQL-запросы.

Дополнительно

- Возможность контейнеризации с использованием Docker.

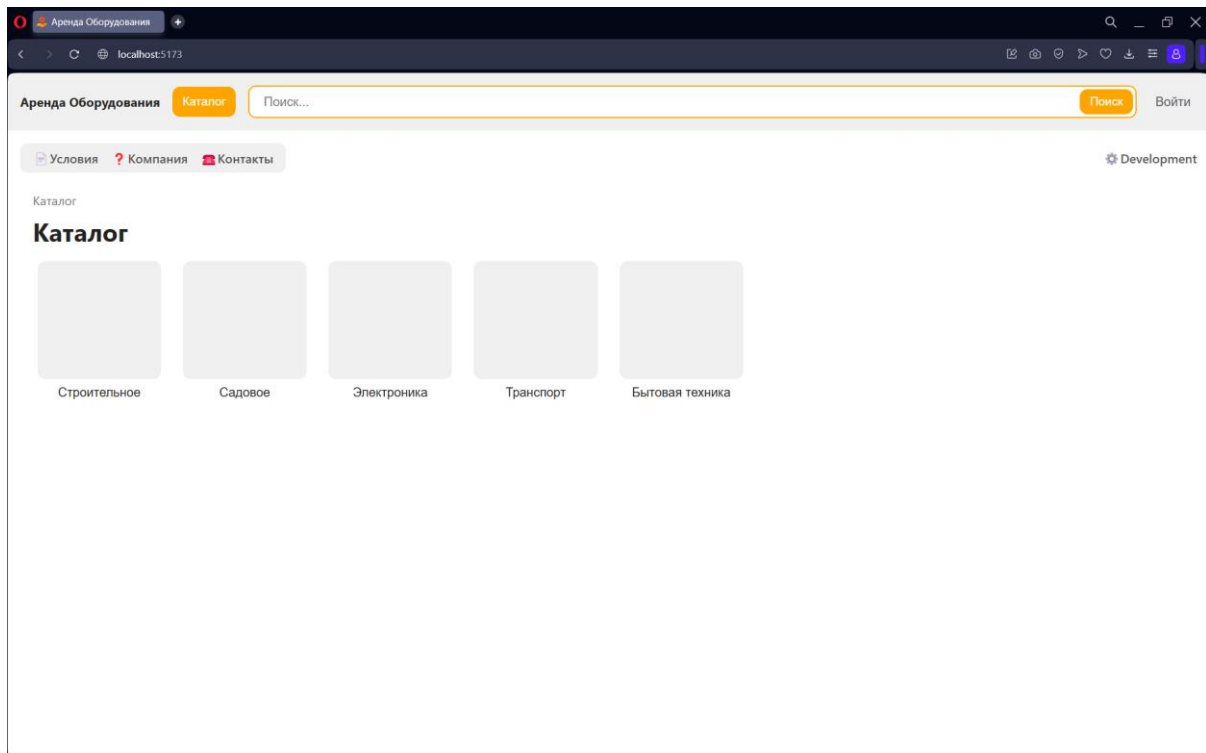
Реализация

[Репозиторий с исходниками](#) - здесь же описано, как развернуть приложение, а также объяснено назначение всех файлов (для понимания).

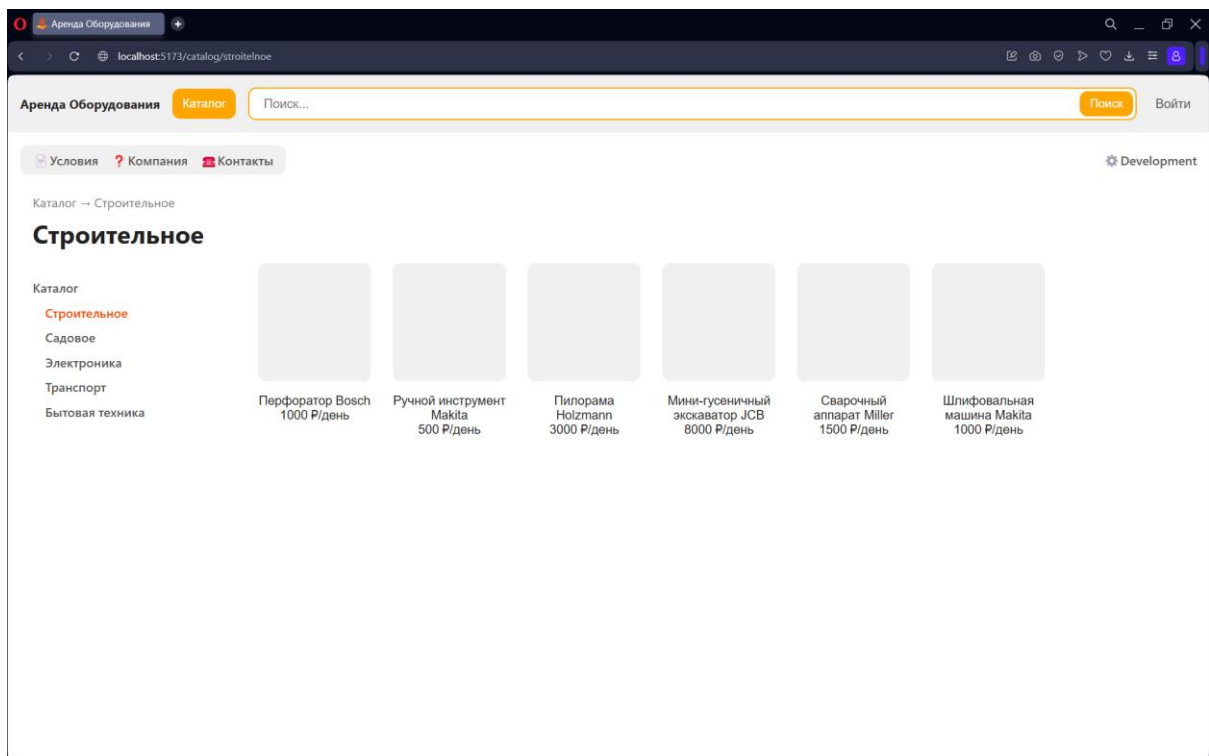
Бэкенд реализован на языке Python с библиотекой Flask. Запросы, относящиеся к разным ролям пользователя (клиент, менеджер, администратор), вынесены в разные модули. Также реализована своя обёртка для удобного взаимодействия с базой данных. Вынесены специальные команды для разработки (создание, очистка, заполнение, удаление, сохранение БД), которые доступны в терминале и на сайте в пользовательском интерфейсе. Доступ к API, предназначенным только для менеджеров или админов, проверяется через JWT токены (то есть для доступа, пользователь должен быть авторизован с соответствующей ролью). Дополнительную безопасность на уровне СУБД, обеспечивают роли БД (при выполнении команд, предназначенных для определённого пользователя, подключение к БД происходит с соответствующей ролью).

Фронтенд написан на JavaScript (библиотека React). Всё разбито на страницы, доступ к которым проверяется с помощью данных авторизации. В интерфейсе отображаются только компоненты, доступные текущему пользователю (зависит от роли пользователя). На каждой странице есть верхняя панель для перехода на другие страницы, что обеспечивает удобные переходы. Реализовано это с помощью компоненты Layout, который содержит компонент NavBarTop (верхняя панель) и дочерний элемент (вложенные компоненты). Для стилизации используется CSS.

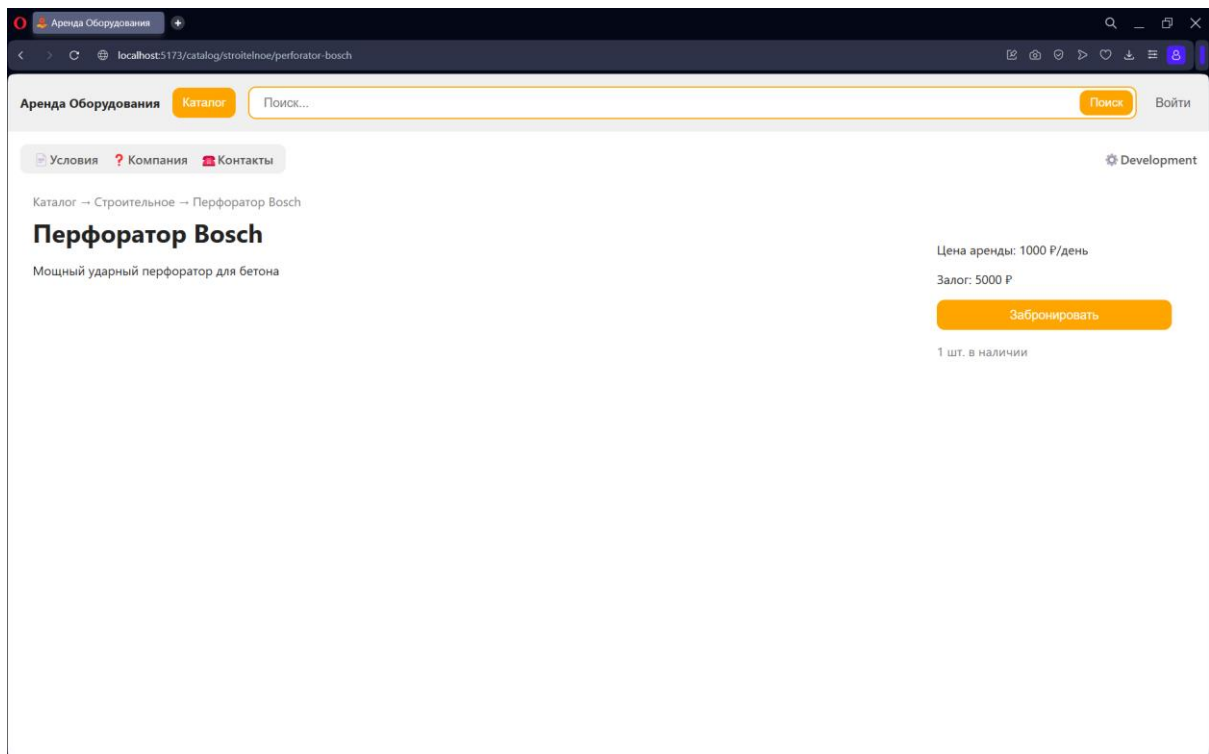
Клиентский интерфейс



При заходе на сайт, клиент сразу попадает на страницу “Каталог” (потому что маршрутизация настроена таким образом, что каталог отображается и по пути “/catalog” и просто “/”). Сверху - верхняя панель, которая содержит: название компании, кнопку “Каталог”, строку поиска (исключительно косметическую), кнопку “Войти”, ссылки на другие страницы (“Условия”, “Компания”, “Контакты”) и кнопку “Development” (нужна только при разработке). На странице каталога отображаются все категории оборудования (серые квадраты - якобы картинки).



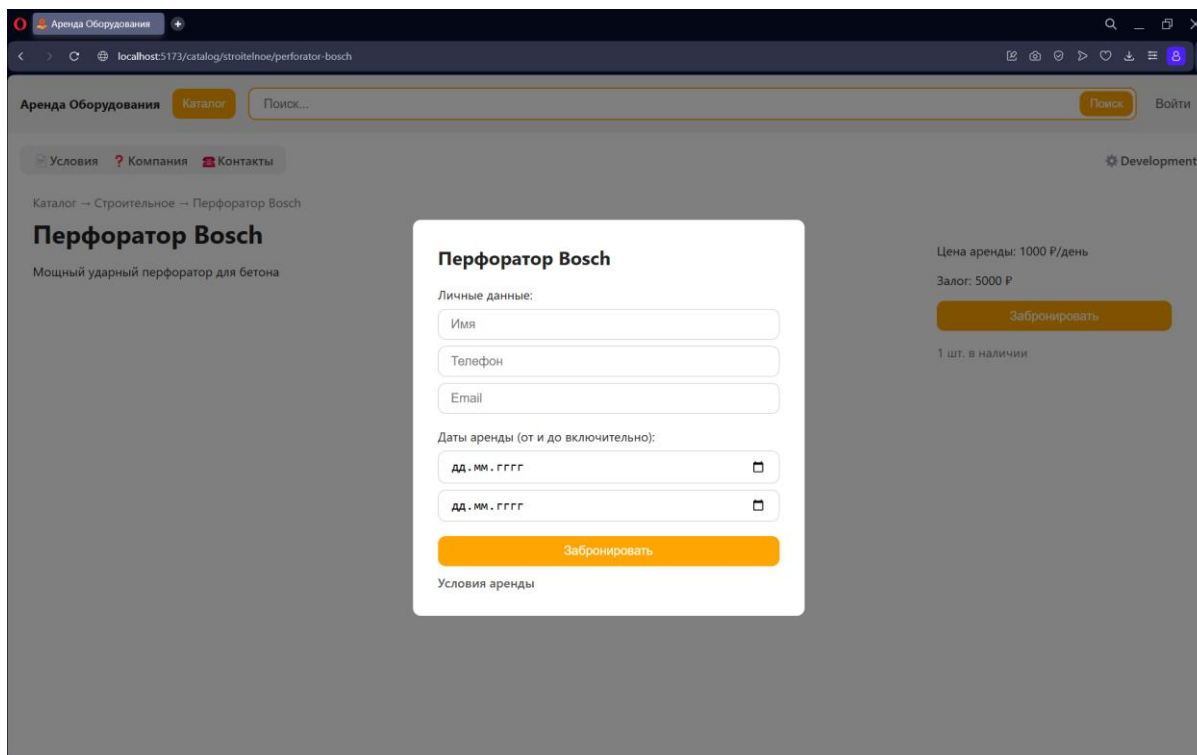
На странице определённой категории показывается оборудование соответствующей категории. Для удобства здесь же показывается цена аренды. Также слева есть навигационная панель, чтобы удобно перемещаться между категориями.



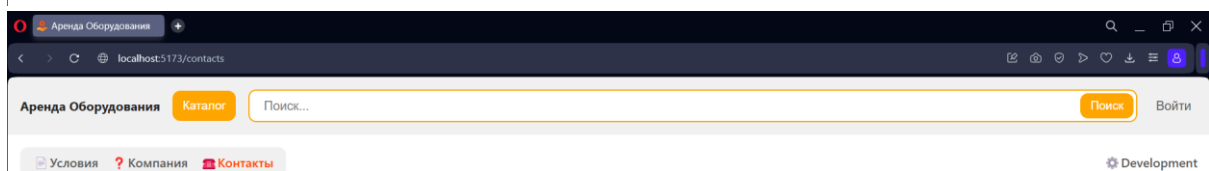
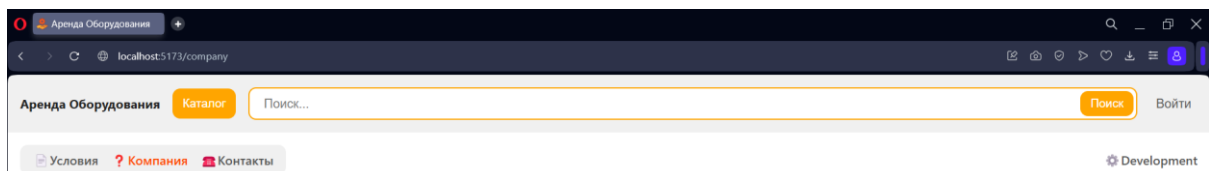
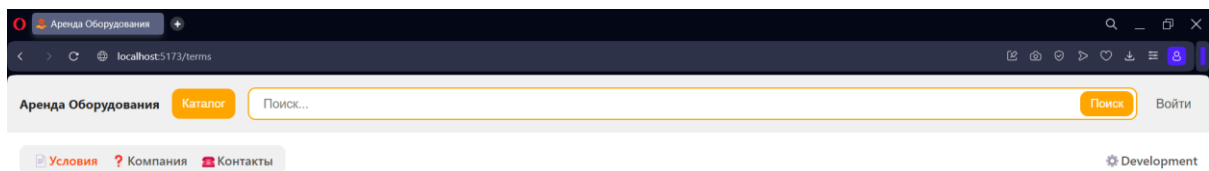
При переходе на страницу определённого оборудования показывается карточка, содержащая название оборудования, описание, цену аренды, сумму залога, количество доступных единиц оборудования и кнопку "Забронировать". Также

над названием оборудования можно заметить светло серым текстом путь по каталоги (Каталог -> Строительное -> Перфоратор Bosch). А в URL написан тот же путь только английским транслитом. Это тонкий момент, потому что в URL нельзя писать русский текст и другие спец символы. Реализовано это с помощью преобразования в SLUG формат. Маршрутизация устроена так:

```
<Route path="/catalog" element={<Catalog />} />
<Route path="/catalog/:categoryNameSlug" element={<Category />} />
<Route path="/catalog/:categoryNameSlug/:equipmentNameSlug" element={<EquipmentCard />} />
```

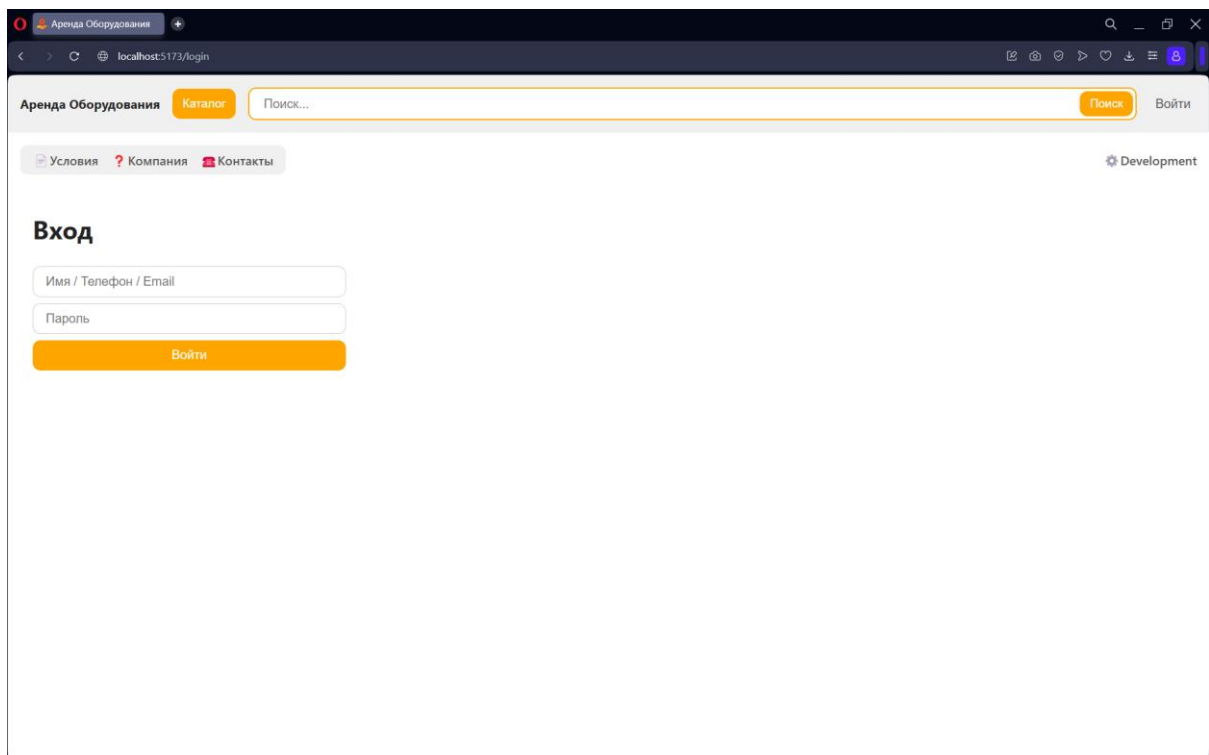


При нажатии на “Забронировать” появляется модальное окно. Здесь указывается имя, телефон, email клиента, даты аренды (когда клиент собирается использовать оборудование). Если данный пользователь ещё не делал заказов, в базе появляется новая запись с данными этого клиента. Дату начала брони нельзя указать раньше сегодняшнего дня, а дату окончания аренды нельзя указать раньше даты начала. Также есть ссылка на условия аренды, чтобы клиент точно не забыл с ними ознакомиться.



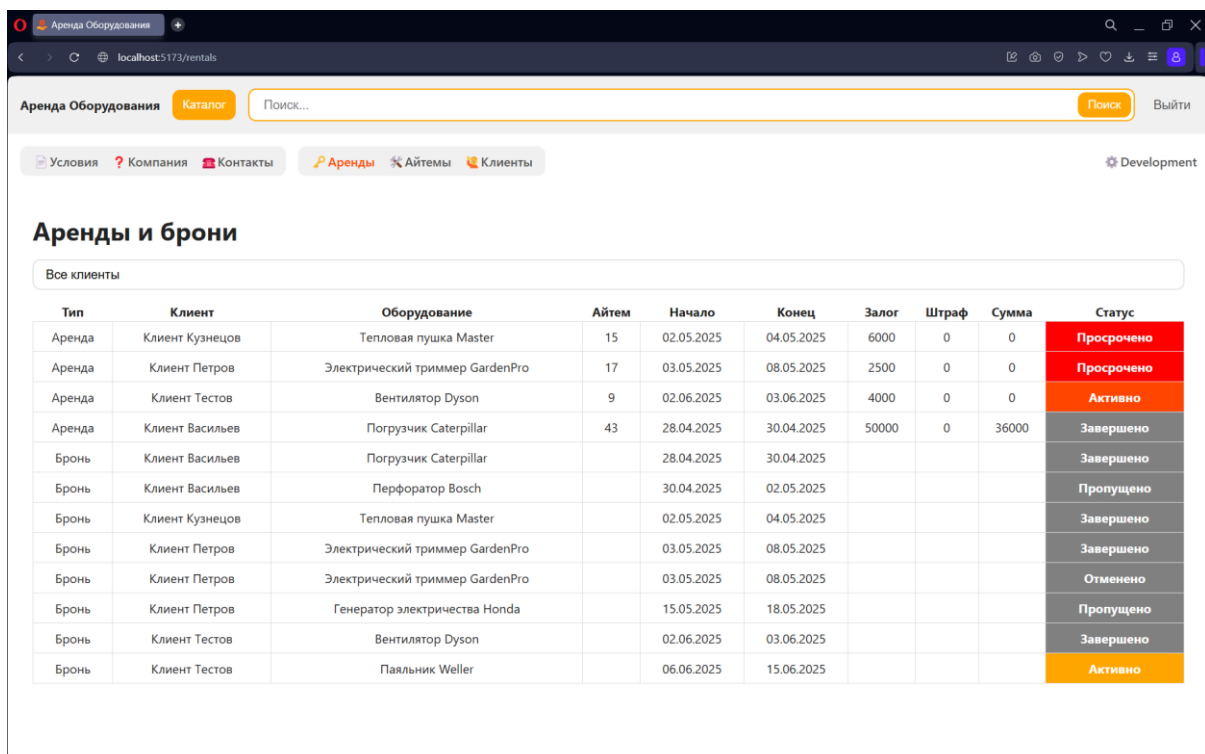
Условия, Компания и Контакты - простые информационные страницы с текстом.

Менеджерский интерфейс



Для использования прав менеджера надо авторизоваться. Можно войти по имени,

телефону или email и паролю, если в базе есть соответствующие данные (предполагается, что при добавлении нового сотрудника в штат, админ должен создать учётную запись для него). Та же логика работает при авторизации с ролью администратора. После успешного входа сервер присылает JWT (токен), который хранится в браузере и отправляется с каждым запросом на сервер, чтобы подтвердить аутентификацию. На сервере JWT декодируется с помощью ключа, который был использован при его создании, после чего проверяются роль (она вложена в JWT) и даётся доступ. Но срок действия JWT мал (около 15 минут), и поэтому существует Refresh токен. Он также отправляется сервером при авторизации. Клиент должен каждый раз отправлять его вместе с куки, чтобы в случае истечения JWT, выдался новый токен. Если и Refresh токен истёк, клиенту придётся войти заново.



Тип	Клиент	Оборудование	Айтем	Начало	Конец	Залог	Штраф	Сумма	Статус
Аренда	Клиент Кузнецов	Тепловая пушка Master	15	02.05.2025	04.05.2025	6000	0	0	Просрочено
Аренда	Клиент Петров	Электрический триммер GardenPro	17	03.05.2025	08.05.2025	2500	0	0	Просрочено
Аренда	Клиент Тестов	Вентилятор Dyson	9	02.06.2025	03.06.2025	4000	0	0	Активно
Аренда	Клиент Васильев	Погрузчик Caterpillar	43	28.04.2025	30.04.2025	50000	0	36000	Завершено
Бронь	Клиент Васильев	Погрузчик Caterpillar		28.04.2025	30.04.2025				Завершено
Бронь	Клиент Васильев	Перфоратор Bosch		30.04.2025	02.05.2025				Пропущено
Бронь	Клиент Кузнецов	Тепловая пушка Master		02.05.2025	04.05.2025				Завершено
Бронь	Клиент Петров	Электрический триммер GardenPro		03.05.2025	08.05.2025				Завершено
Бронь	Клиент Петров	Электрический триммер GardenPro		03.05.2025	08.05.2025				Отменено
Бронь	Клиент Петров	Генератор электричества Honda		15.05.2025	18.05.2025				Пропущено
Бронь	Клиент Тестов	Вентилятор Dyson		02.06.2025	03.06.2025				Завершено
Бронь	Клиент Тестов	Паяльник Weller		06.06.2025	15.06.2025				Активно

Менеджеру доступна страница со всеми бронями и арендами. Здесь показывается, кто что заказал / забронировал, даты начала, конца аренды, а также залог, штраф и суммарная заплаченная цена аренды, статус. Сумма считается автоматически при завершении аренды (считается, что клиент оплачивает аренду после того, как он вернул оборудование, поэтому пока аренда активна, сумма = 0). Статус аренды “Просрочено” явно не хранится в БД. Он учитывается как статус “Активно”, но, если дата конца аренды меньше текущей даты. Это нужно, чтобы менеджер видел, каких клиентов предупредить о необходимости вернуть оборудование. Если клиент вернул товар позже, чем надо, то он просто должен будет оплатить аренду за все дни, пока оборудование было у него на руках. Айтем (единица оборудования) указывается только для аренд, так как бронируется оборудование, а не конкретный айтем. Для броней предусмотрены статусы:

“Активно”, “Завершено”, “Пропущено”, “Отменено”. Пропущено — значит, что бронь активна, но дата начала предполагаемой аренды больше текущей даты. Есть возможность отсортировать таблицу по каждому столбцу.

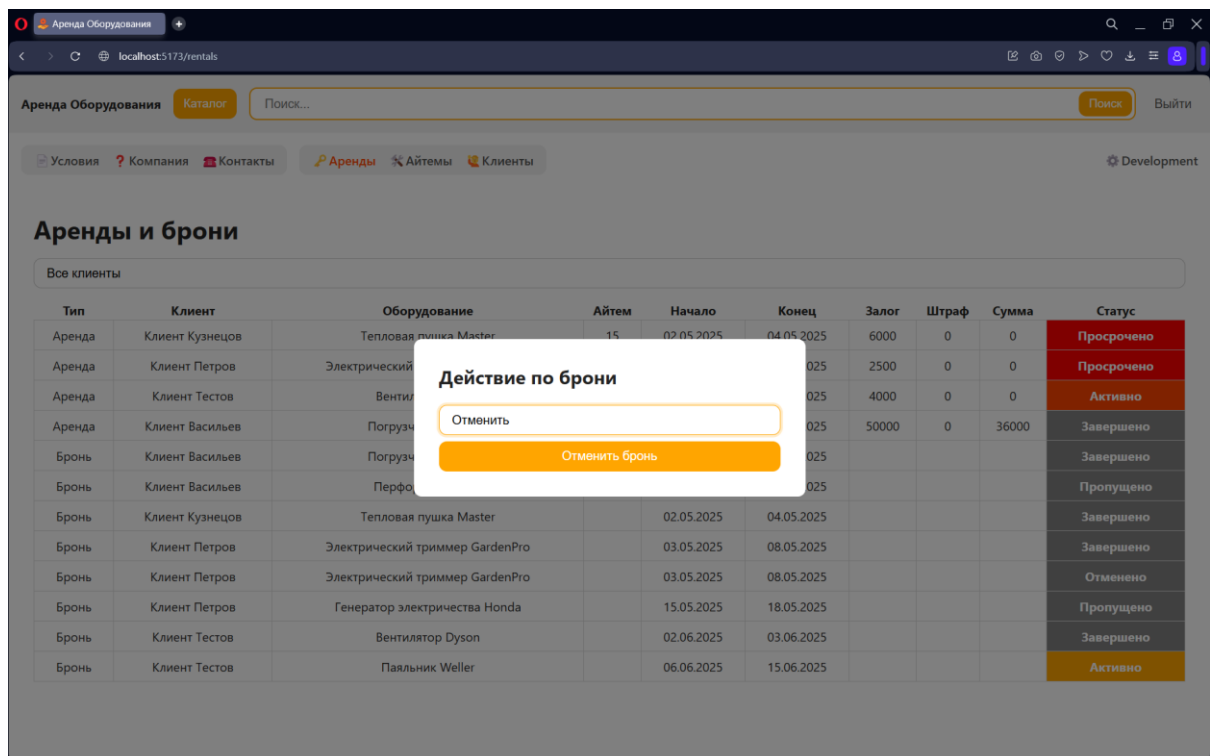
The screenshot shows a web application titled "Аренда Оборудования". At the top, there is a navigation bar with a search bar and a "Выйти" button. Below the navigation bar, there are tabs for "Условия", "Компания", "Контакты", "Аренды", "Айтемы", and "Клиенты". The "Аренды" tab is selected. The main section is titled "Аренды и брони" and features a dropdown menu with "Клиент Тестов" selected. Below this, there is a table with the following columns: Тип, Клиент, Оборудование, Айтем, Начало, Конец, Залог, Штраф, Сумма, and Статус. The table contains three rows:

Тип	Клиент	Оборудование	Айтем	Начало	Конец	Залог	Штраф	Сумма	Статус
Аренда	Клиент Тестов	Вентилятор Dyson	9	02.06.2025	03.06.2025	4000	0	0	Активно
Бронь	Клиент Тестов	Вентилятор Dyson		02.06.2025	03.06.2025				Завершено
Бронь	Клиент Тестов	Паяльник Weller		06.06.2025	15.06.2025				Активно

Можно выбрать аренды/брони только одного определённого клиента.

The screenshot shows the same web application, but with a modal window titled "Действие по брони" (Action on reservation) open. The modal contains two buttons: "Оформить аренду" (Form reservation) and "Оформить аренду" (Form reservation). The background table is dimmed, showing a list of rentals and reservations for various clients. The table has the same columns as the previous screenshot. The status of the reservations is "Просрочено" (Overdue) for the first two rows and "Завершено" (Completed) for the others.

Если кликнуть по строчке с бронью, появится модальное окно действий по брони. Бронь можно перевести в аренду. В этом случае бронь станет со статусом “Завершено” и создастся новая аренда с данными из брони.



Также бронь можно просто отменить.

Действие по аренде

Завершить аренду

Завершить аренду

Действие по аренде

Продлить аренду

Новая дата окончания аренды:

дд . мм . гggг



Продлить аренду

Действие по аренде

Добавить штраф

Сумма штрафа

Начислить штраф

Если кликнуть по строчке с записью аренды, то откроется модельное окно с действиями по аренде. Можно завершить аренду, продлить её или начислить штраф.

Аренда Оборудования **Каталог** Поиск... Поиск Выйти

Условия Компания Контакты Аренды **Айтемы** Клиенты Development

Единицы оборудования

Всё оборудование

ID	Оборудование	Статус	
1	Генератор электричества Honda	Забронировано	X
2	Скруббер Karcher	Доступно	X
3	Бензопила Stihl	Доступно	X
4	Бензопила Stihl	На обслуживании	X
5	Бензопила Stihl	Списано	X
6	Сварочный аппарат Miller	Доступно	X
7	Паяльник Weller	Забронировано	X
8	Шлифовальная машина Makita	Доступно	X
9	Вентилятор Dyson	Арендовано	X
10	Перфоратор Bosch	Забронировано	X
11	Мотоцикл Yamaha	Доступно	X
12	Компьютерная мышь Logitech	Доступно	X
13	Газонокосилка Husqvarna	Доступно	X
14	Паяльник Weller	Доступно	X
15	Тепловая пушка Master	Арендовано	X

Также менеджеру доступна страница управления айтемами.

Аренда Оборудования **Каталог** Поиск... Поиск Выйти

Условия Компания Контакты Аренды **Айтемы** Клиенты Development

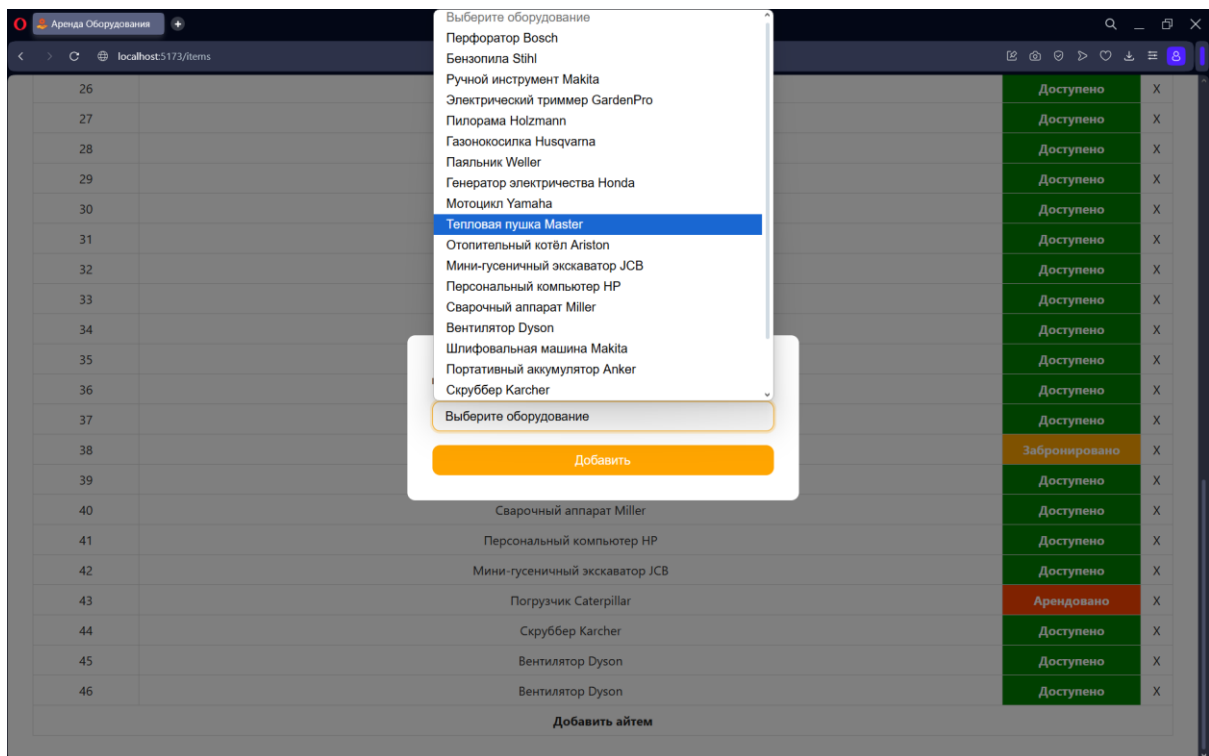
Единицы оборудования

Всё оборудование

Всё оборудование
Перфоратор Bosch
Бензопила Stihl
Ручной инструмент Makita
Электрический триммер GardenPro
Пилорама Holzmann
Газонокосилка Husqvarna
Паяльник Weller
Генератор электричества Honda
Мотоцикл Yamaha
Тепловая пушка Master
Отопительный котёл Ariston
Мини-гусеничный экскаватор JCB
Персональный компьютер HP
Сварочный аппарат Miller
Вентилятор Dyson
Шлифовальная машина Makita
Портативный аккумулятор Anker
Скруббер Karcher
Компьютерная мышь Logitech

14	Паяльник Weller	Доступно	X
15	Тепловая пушка Master	Арендовано	X

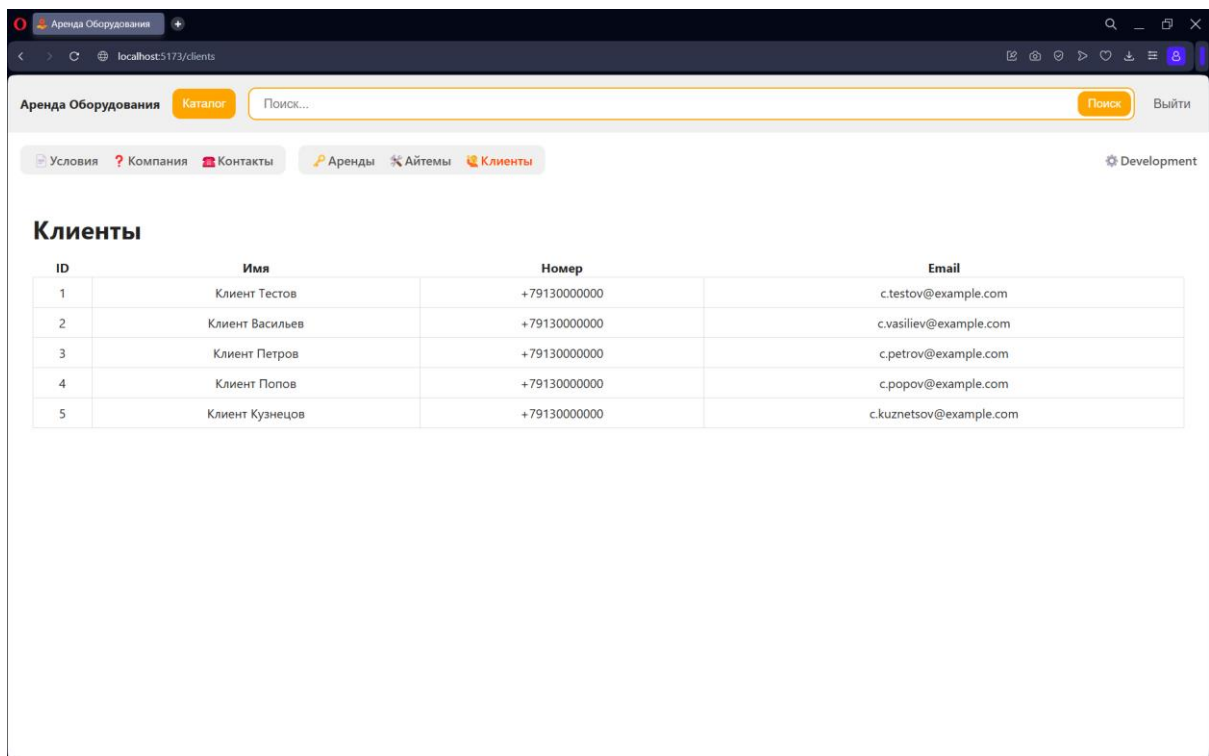
Можно выбрать оборудование, чтобы в списке показывались только айтемы этого оборудования. Также можно отсортировать таблицу по каждой колонке.



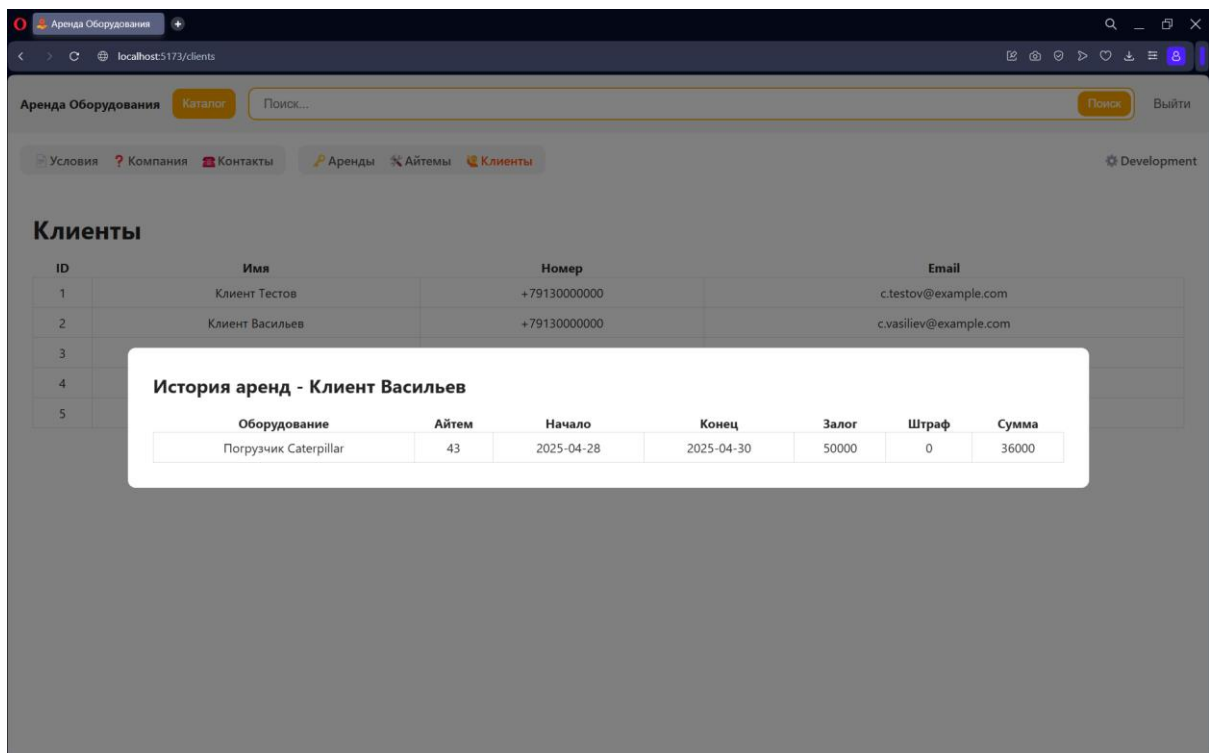
При нажатии на кнопку “Добавить айтем” снизу страницы, можно добавить новый айтем определённого оборудования.

	Списано	X
	Доступно	X
	Забронировано	X

Если айтем не забронирован и не арендован, его можно удалить, нажав на крестик или сменить статус, нажав на статус (статус меняется по кругу: Доступно - На обслуживании - Списано).



Ещё менеджеру доступна страница со всеми клиентами, которые когда-либо что-то арендовали. Можно отсортировать таблицу по любому столбцу.



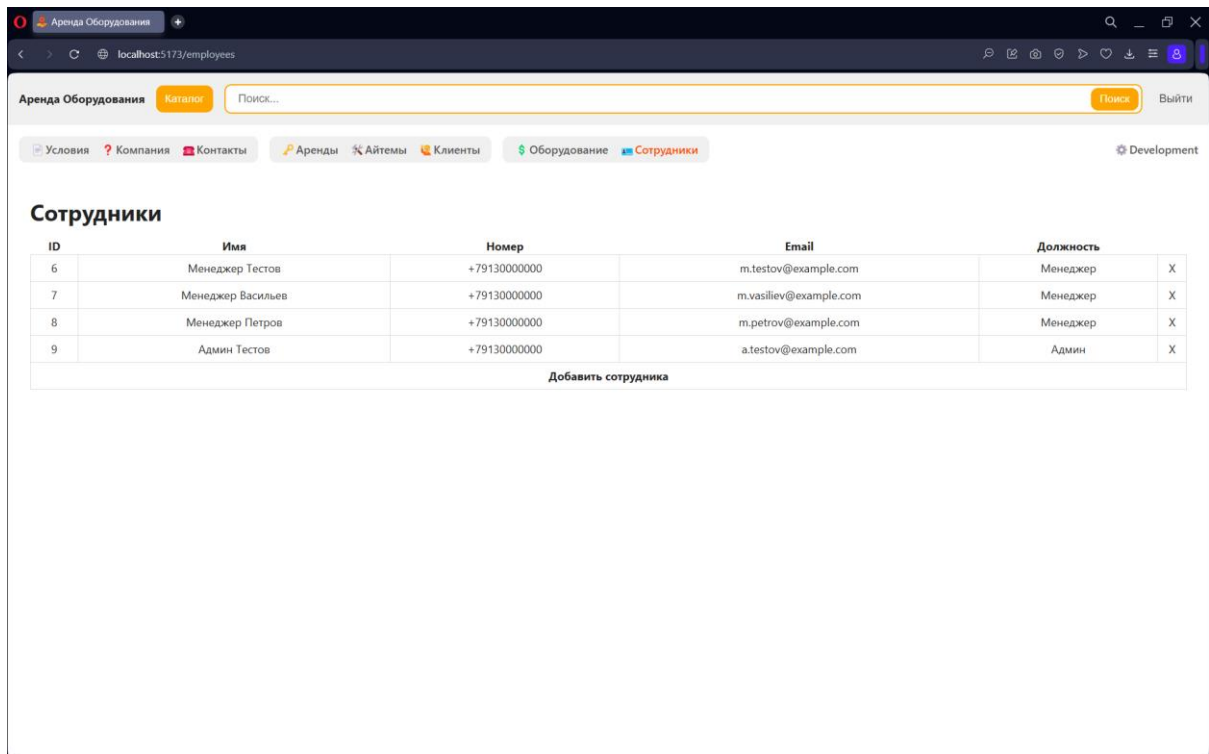
Если кликнуть по клиенту, появился история всех его завершённых аренд.

Админский интерфейс

The screenshot displays the admin interface for an equipment rental system. The main table lists various pieces of equipment, including power tools, generators, and vehicles. A modal window for adding new equipment is currently active, showing input fields for Name, Category, Description, Rental Price, and Deposit, along with a 'Добавить' (Add) button.

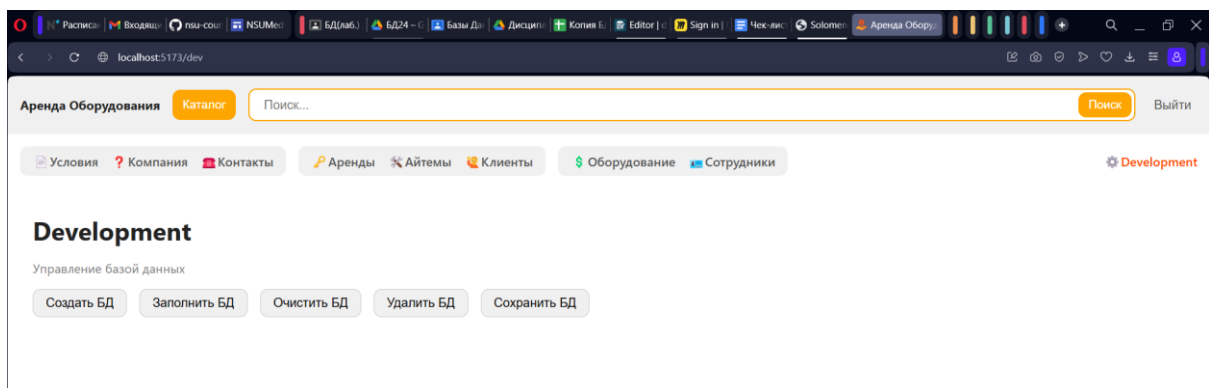
ID	Название	Категория	Описание	Цена аренды	Залог	
1	Перфоратор Bosch	Строительное	Мощный ударный перфоратор для бетона	1000	5000	X
2	Бензопила Stihl	Садовое	Для обрезки деревьев	800	4000	X
3	Ручной инструмент Makita	Строительное	Для мелких работ	500	2000	X
4	Электрический триммер GardenPro	Садовое	Триммер для газонов	600	2500	X
5	Пилорама Holzmann	Строительное	Для распиловки древесины	3000	15000	X
6	Газонокосилка Husqvarna	Садовое	Для косовых работ на частных домах	700	3000	X
7	Паяльник Weller	Электроника	Для ремонта электроники	400	2000	X
8	Генератор электричества Honda	Транспорт	Для питания строительных площадок	1500	8000	X
9	Мотоцикл Yamaha	Транспорт	Для отдыха	5000	25000	X
10	Тепловая пушка Master	Бытовая техника	Для обогрева больших помещений	1200	6000	X
11	Отопительный котёл Ariston	Бытовая техника	Для отопления в котёл для квартиры	2500	10000	X
12	Мини-гусеничный экскаватор JCB	Строительное	Для выполнения земляных строительных работ	8000	35000	X
13	Персональный компьютер HP	Электроника	Для работы на личном компьютере	500	3000	X
14	Сварочный аппарат Miller	Строительное	Сварочный аппарат для металлоконструкций	1500	7000	X
15	Вентилятор Dyson	Бытовая техника	Энергосберегающий вентилятор для охлаждения воздуха	800	4000	X
16	Шлифовальная машина Makita	Строительное	Шлифовальная машина для отделочных работ	1000	5000	X
17	Портативный аккумулятор Anker	Электроника	Мобильный аккумулятор для зарядки устройств	300	1000	X
18	Скrubber Karcher	Бытовая техника	Профессиональная уборочная техника для очистки полов	2500	12000	X
19	Компьютерная мышь Logitech	Электроника	Эргономичная компьютерная мышь для длительных рабочих сессий	250	500	X
20	Погрузчик Caterpillar	Транспорт	Погрузчик для транспортировки строительных материалов	12000	50000	X
21	Миксер Bosch	Бытовая техника	Электрический миксер для кухни	500	2000	X

Администратор может управлять каталогом и прайсингом. Он может добавить оборудование и удалить его, если оно не участвует в активных бронях или арендах. Чтобы изменить данные оборудования, надо добавить оборудование с таким же названием, но с изменёнными характеристиками. И, конечно, можно сортировать таблицу по любому столбцу.



Также администратор может управлять учётными записями сотрудником.

Интерфейс разработчика



Для удобной разработки был сделан интерфейс для управления БД. Можно создать, заполнить, очистить, удалить и сохранить БД. Сохранение БД работает так, что создаётся sql скрипт для заполнения БД по текущим данным. Это позволяет не терять данные даже при полном падении БД.

Безопасность

Разные действия, реализующие бизнес логику, выполняются из-под разных пользователей базы данных, у которых есть доступ только к необходимым возможностям. Также есть поддержка .env файлов для хранения паролей, ключей и других приватных данных. Каждому пользователю даны ровно только нужные ему права:

```

1 DROP USER 'equipment_rental_client'@'localhost';
2 CREATE USER 'equipment_rental_client'@'localhost' IDENTIFIED BY 'client';
3
4 GRANT SELECT ON equipment_rental.Equipment TO 'equipment_rental_client'@'localhost';
5 GRANT SELECT ON equipment_rental.Items TO 'equipment_rental_client'@'localhost';
6 GRANT SELECT, INSERT ON equipment_rental.Users TO 'equipment_rental_client'@'localhost';
7 GRANT INSERT ON equipment_rental.Reservations TO 'equipment_rental_client'@'localhost';
8
9 FLUSH PRIVILEGES;
10
11 DROP USER 'equipment_rental_manager'@'localhost';
12 CREATE USER 'equipment_rental_manager'@'localhost' IDENTIFIED BY 'manager';
13
14 GRANT SELECT ON equipment_rental.Equipment TO 'equipment_rental_manager'@'localhost';
15 GRANT SELECT, INSERT ON equipment_rental.Users TO 'equipment_rental_manager'@'localhost';
16 GRANT SELECT, INSERT, UPDATE, DELETE ON equipment_rental.Reservations TO 'equipment_rental_manager'@'localhost';
17 GRANT SELECT, INSERT, UPDATE, DELETE ON equipment_rental.Rentals TO 'equipment_rental_manager'@'localhost';
18 GRANT SELECT, INSERT, UPDATE, DELETE ON equipment_rental.Items TO 'equipment_rental_manager'@'localhost';

```

Триггеры

Для удобной работы с бронями, были реализованы триггеры:

```

1 -- При добавлении брони, статус первого доступного айтема (оборудования, указанного в брони) меняем на 'booked'
2 CREATE TRIGGER after_reservation_insert
3 AFTER INSERT ON Reservations
4 FOR EACH ROW
5 BEGIN
6     UPDATE Items
7     SET status = 'booked'
8     WHERE equipment_id = NEW.equipment_id
9     AND status = 'available'
10    LIMIT 1;
11 END//
12
13 -- При отмени брони, статус первого забронированного айтема (оборудования, указанного в брони) меняем на 'available'
14 CREATE TRIGGER after_reservation_cancelled
15 AFTER UPDATE ON Reservations
16 FOR EACH ROW
17 BEGIN
18     IF NEW.status = 'cancelled' THEN
19         UPDATE Items
20         SET status = 'available'
21         WHERE equipment_id = NEW.equipment_id
22         AND status = 'booked'
23         LIMIT 1;
24     END IF;
25 END//
26
27 -- При завершении брони (то есть перехода брони в аренду):
28 -- 1. Находится первый айтем оборудования, указанного в брони, со статусом 'booked'
29 -- 2. Статус найденного айтема меняется на 'rented'
30 -- 3. Добавляется соответствующая аренда
31 CREATE TRIGGER after_reservation_completed
32 AFTER UPDATE ON Reservations
33 FOR EACH ROW
34 BEGIN
35     DECLARE rented_item_id INT;
36     DECLARE deposit INT;
37
38     IF NEW.status = 'completed' THEN
39         SELECT id INTO rented_item_id
40         FROM Items
41         WHERE equipment_id = NEW.equipment_id
42         AND status = 'booked'
43         LIMIT 1;
44
45         IF rented_item_id IS NOT NULL THEN
46             UPDATE Items
47             SET status = 'rented'
48             WHERE id = rented_item_id;
49
50             SELECT deposit_amount INTO deposit
51             FROM Equipment
52             WHERE id = NEW.equipment_id;
53
54             INSERT INTO Rentals (client_id, item_id, start_date, end_date, status, deposit_paid)
55             VALUES (NEW.client_id, rented_item_id, NEW.start_date, NEW.end_date, 'active', COALESCE(deposit, 0));
56         END IF;
57     END IF;
58 END//

```

SQL запросы

Основные SQL запросы

Остальные SQL запросы представлены в коде и обернуты в функции, причём у этих функций указан собственный декоратор @with_db. Это означает, что при выполнениях каждой из этих функций будет производиться подключение к базе данных с заданными параметрами входа, а при завершении функции - автоматическое освобождение всех ресурсов и отключение от базы данных.

Решение не поддерживать соединение, а переподключаться к базе данных каждый раз заново, было принято специально, так как запросы от клиентов приходят не часто, а пользователей может быть много, поэтому поддерживать несколько соединений для каждого пользователя может быть ресурсозатратно.

Пример кода:

```
@with_db(user=DB_ROOT_NAME, password=DB_ROOT_PASSWORD, host=DB_HOST, database=DB_NAME)
def drop_db(conn, cursor):
    """Полное удаление базы данных."""
    cursor.execute(f"DROP DATABASE IF EXISTS {DB_NAME}")

@with_db(user=DB_ROOT_NAME, password=DB_ROOT_PASSWORD, host=DB_HOST, database=DB_NAME)
def get_user(identifier, conn, cursor):
    """Получение пользователя из БД по имени, телефону или емейлу."""
    cursor.execute("SELECT * FROM Users WHERE name = %s OR phone = %s OR email = %s", (identifier, identifier, identifier))
    return cursor.fetchone()

@with_db(user=DB_ROOT_NAME, password=DB_ROOT_PASSWORD, host=DB_HOST, database=DB_NAME)
def insert_refresh_token(token, user_id, role, expires, conn, cursor) -> str:
    """Добавление Refresh токена в БД."""
    cursor.execute(
        "INSERT INTO refresh_tokens(token, user_id, role, expires_at) "
        "VALUES (%s, %s, %s, %s)",
        (token, user_id, role, expires)
    )

@with_db(user=DB_ROOT_NAME, password=DB_ROOT_PASSWORD, host=DB_HOST, database=DB_NAME)
def get_refresh_token(token, conn, cursor):
    """Получение Refresh токена из БД."""
    cursor.execute(
        "SELECT user_id, role, expires_at "
        "FROM refresh_tokens WHERE token = %s",
        (token)
    )
    return cursor.fetchone()

@with_db(user=DB_ROOT_NAME, password=DB_ROOT_PASSWORD, host=DB_HOST, database=DB_NAME)
def delete_refresh_token(token, conn, cursor):
    """Удаление Refresh токена из БД."""
    cursor.execute("DELETE FROM refresh_tokens WHERE token = %s", (token,))
```

Тестирование

API тестировался с помощью заполнения БД тестовыми данными и работу через веб интерфейс.