МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ФГБОУ ВПО

Новосибирский Государственный Технический Университет

Институт Социальных Технологий

ОТЧЕТ

по дисциплине «**Технология разработки программного обеспечения**»

по теме «**Разработка ИС технической поддержки предприятия. Разработка подсистемы клиентов. Разработка подсистемы первой линии техподдержки**»

Выполнил:

Студент группы И-03

Алферов Д.С.

Проверил:

Преподаватель

Бертик А.А

Новосибирск 2023

**Оглавление**

[Введение 3](#_Toc153138260)

[Анализ предметной области 3](#_Toc153138261)

[**Обзор существующих аналогов** 4](#_Toc153138262)

[Средства разработки 5](#_Toc153138263)

[Реализация проекта 6](#_Toc153138264)

[**Страница авторизации** 7](#_Toc153138265)

[**Страница регистрации** 7](#_Toc153138266)

[**Главная страница для пользователей** 8](#_Toc153138267)

[**Страница создания заявки пользователем** 9](#_Toc153138268)

[**Страница «Мои заявки»** 10](#_Toc153138269)

[**Главная страница для администратора** 11](#_Toc153138270)

[**Страница просмотра заявки** 12](#_Toc153138271)

[**Страница «Все заявки»** 13](#_Toc153138272)

[**Страница «Пользователи»** 14](#_Toc153138273)

[**Инвентарь. Список аппаратуры** 15](#_Toc153138274)

[**Инвентарь. Страница «Добавить»** 16](#_Toc153138275)

[**Скрипты** 16](#_Toc153138276)

[Заключение 16](#_Toc153138277)

[Список источников 17](#_Toc153138278)

[Приложение 17](#_Toc153138279)

# **Введение**

Техническая поддержка включает в себя различные услуги предприятия по оказанию помощи своим клиентам, то есть пользователям услуг или продуктов данного предприятия. Техническая поддержка помогает клиенту выяснять и решать проблемы, связанные с продуктом и его использованием, но не обучает использовать или индивидуально настраивать конкретный продукт.

Техническая поддержка предоставляется, как правило, по телефону, через интернет, по электронной почте, через онлайн-сервисы поддержки на веб-сайте или через инструменты, посредством которых пользователи могут зарегистрировать заявку и затем проследить историю состояний/этапов её решения. Большие организации часто имеют собственную, внутреннюю службу технической поддержки, которая помогает персоналу решать возникающие проблемы, носящие технический характер.

# **Анализ предметной области**

Рынок порталов технической поддержки в различных отраслях и на различных сайтах является довольно разнообразным и динамичным. Анализ данного рынка включает оценку его размера, трендов, ключевых игроков и особенностей в каждой отрасли.

Размер рынка: Рынок порталов технической поддержки охватывает все отрасли, где присутствует высокий спрос на техническую поддержку. Это может быть информационные технологии, электроника, инженерные услуги, автомобильная индустрия, медицинское оборудование и многие другие. Общий объем рынка оценивается в миллиарды долларов, и он продолжает расти с увеличением числа потребителей и использования технологий.

Тренды: в последние годы наблюдается рост спроса на онлайн-порталы технической поддержки из-за увеличения числа пользователей интернета и развития онлайн-бизнеса. Компании все больше отдает предпочтение онлайн-сервисам, таким как онлайн-чаты, форумы поддержки и базы знаний, где пользователи могут найти ответы на свои вопросы без необходимости обращаться в службу поддержки по телефону или лично. Также наблюдается рост использования технологий искусственного интеллекта (ИИ) и чат-ботов для автоматизации поддержки и улучшения пользовательского опыта.

Ключевые игроки: на рынке порталов технической поддержки существует множество компаний, предлагающих свои решения. Крупные игроки в этой области включают Salesforce, Zendesk, Freshdesk, ServiceNow и JIRA. Каждая компания имеет свои особенности и целевую аудиторию, а также специализируется на конкретных отраслях или предлагает решения для различных типов клиентов.

Особенности в отраслях: Различные отрасли могут иметь уникальные потребности в технической поддержке и различные требования к порталам. Например, в сфере информационных технологий важными функциями могут быть создание тикетов для отслеживания запросов пользователей и помощь в решении сложных технических проблем. В инженерной отрасли эффективное управление проектами и обращениями с клиентами могут иметь первостепенное значение. Компании, специализирующиеся на поддержке медицинского оборудования, могут предлагать необходимые сертификации и обновленную информацию о новых системах и доступных ремонтных услугах.

## **Обзор существующих аналогов**

ZenDesk. Англоязычный хелпдеск предлагает быструю обработку заявок из разных источников (соцсети, мессенджеры, почта, телефония, лайв-чаты), создание собственной базы знаний, портала клиентского обслуживания и форумов для комьюнити. Через Zendesk можно принимать входящие звонки, анализировать KPI, формировать прейскуранты, задавать триггерные действия для рутинных задач.

Intraservice. Основной рабочий инструмент сервиса — список заявок. Он формируется через email и веб-форму на сайте, поддерживает групповые операции над задачами, позволяет сортировать запросы с помощью фильтров. Менеджер может создавать заявки от лица другого пользователя, назначать исполнителей и наблюдателей, прикреплять любые файлы и активы. Проанализировать эффективность работы техподдержки помогут отчёты по времени обработки заявок, изменениям статусов, трудозатратам и любым другим выбранным KPI.

FreeScout. Это система поддержки с открытым кодом: любой желающий может увидеть исходный код и проверить его на безопасность. Присутствует возможность отвечать на тикеты из почты, база знаний, клиентский портал самообслуживания, модуль CRM для управления информацией о клиентах, шаблоны ответов, интеграция с WhatsApp, Twitter, Telegram, API (интерфейс прикладного программирования для передачи данных), Jira, плагином WordPress Easy Digital Downloads.

# **Средства разработки**

HTML - стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц в браузере. Веб-браузеры получают HTML документ от сервера по протоколам HTTP/HTTPS или открывают с локального диска, далее интерпретируют код в интерфейс, который будет отображаться на экране монитора. Элементы HTML являются строительными блоками HTML страниц. С помощью HTML разные конструкции, изображения и другие объекты, такие как интерактивная веб-форма, могут быть встроены в отображаемую страницу. HTML предоставляет средства для создания заголовков, абзацев, списков, ссылок, цитат и других элементов. Элементы HTML выделяются тегами, записанными с использованием угловых скобок. Такие теги, как <img /> и <input />, напрямую вводят контент на страницу. Другие теги, такие как <p>, окружают и оформляют текст внутри себя и могут включать другие теги в качестве подэлементов. Браузеры не отображают HTML-теги, но используют их для интерпретации содержимого страницы.

Bootstrap — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

Visual Studio Code (VS Code) — текстовый редактор, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации.

Notepad++ — это свободный текстовый редактор с открытым исходным кодом для Windows с подсветкой синтаксиса, разметки, а также языков описания аппаратуры.

PhpMyAdmin — веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL. PhpMyAdmin позволяет через браузер и не только осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных. Приложение пользуется большой популярностью у веб-разработчиков, так как позволяет управлять СУБД MySQL без непосредственного ввода SQL команд.

Flask — фреймворк для создания веб-приложений на языке программирования Python, использующий набор инструментов Werkzeug, а также шаблонизатор Jinja2. Относится к категории так называемых микрофреймворков — минималистичных каркасов веб-приложений, сознательно предоставляющих лишь самые базовые возможности.

Git — распределённая система управления версиями.

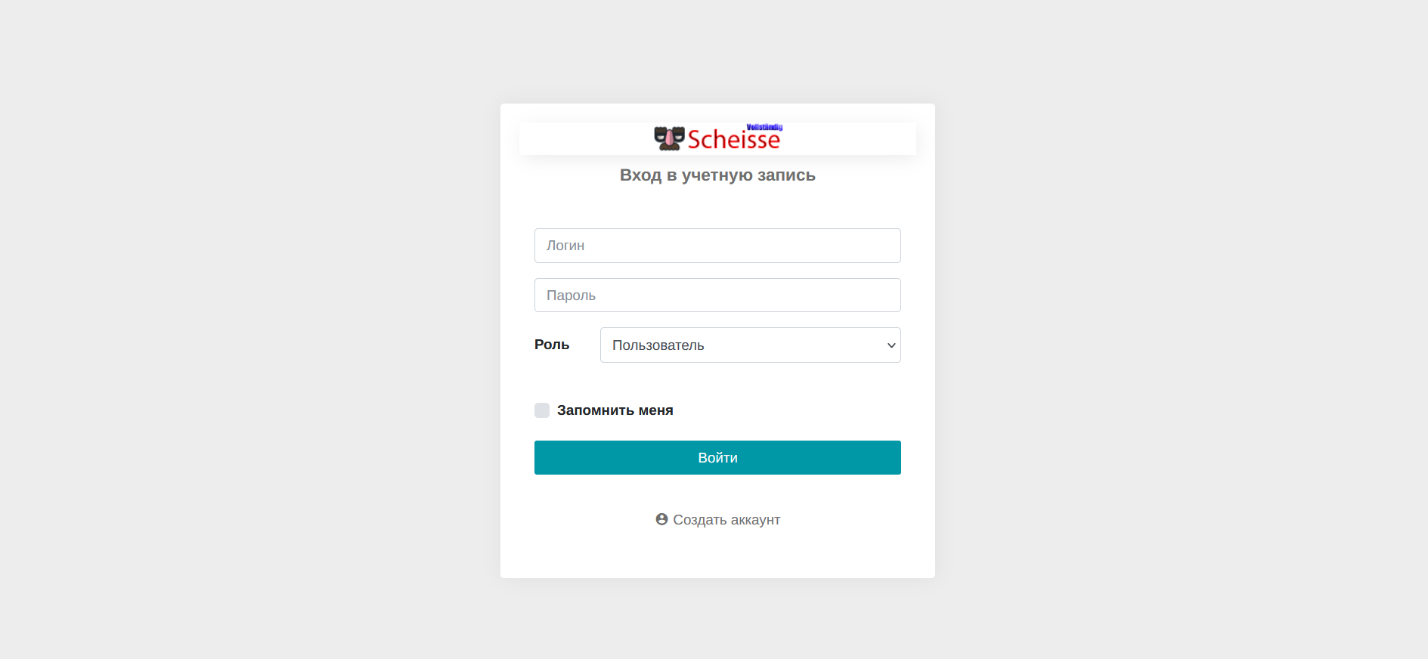
# **Реализация проекта**

Передо мной стояла задача реализовать ИС технической поддержки предприятия, включая подсистему работы с клиентами и первую линию техподдержки. Верстка осуществлялась при помощи фреймворка Bootstrap.

Были созданы все необходимые веб-страницы, такие как:

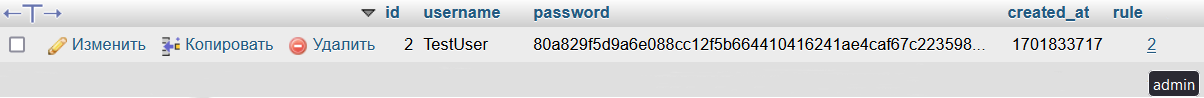
## **Страница авторизации**

Здесь реализована авторизация с возможностью выбрать свою роль в системе – Пользователь / Сотрудник / Администратор (рис.1)

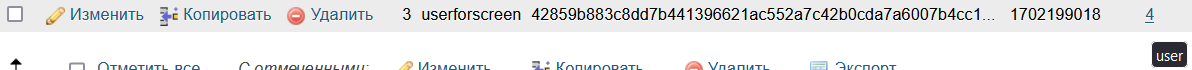


**Рисунок 1. Страница авторизации**

Изменять роль юзера соответственно способен администратор базы данных (рис. 2-3)



**Рисунок 2. Отображение пользователя со статусом admin в БД**

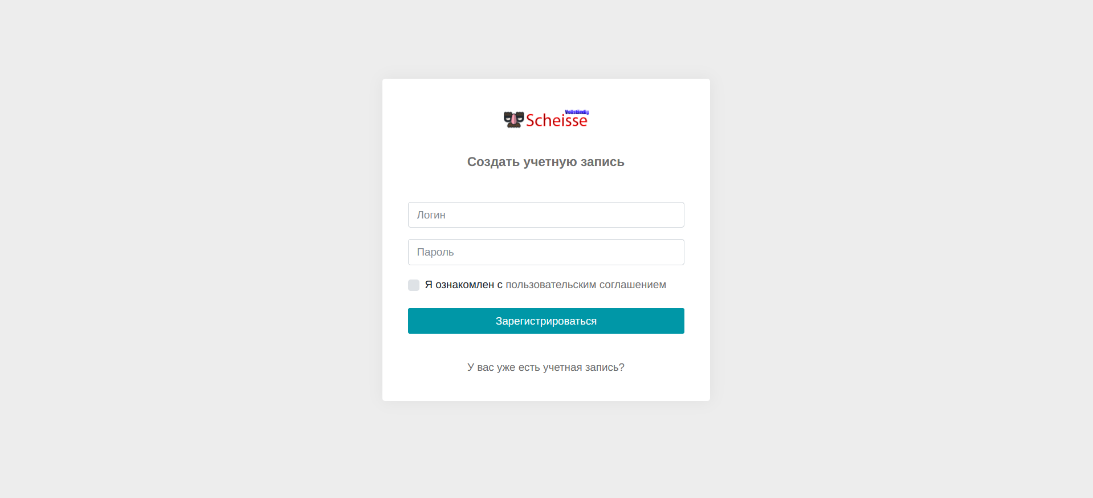


**Рисунок 3. Отображение пользователя со статусом user в БД**

([см. приложение 1](#Прил1))

## **Страница регистрации**

Страница регистрации изображена на рисунке 4.

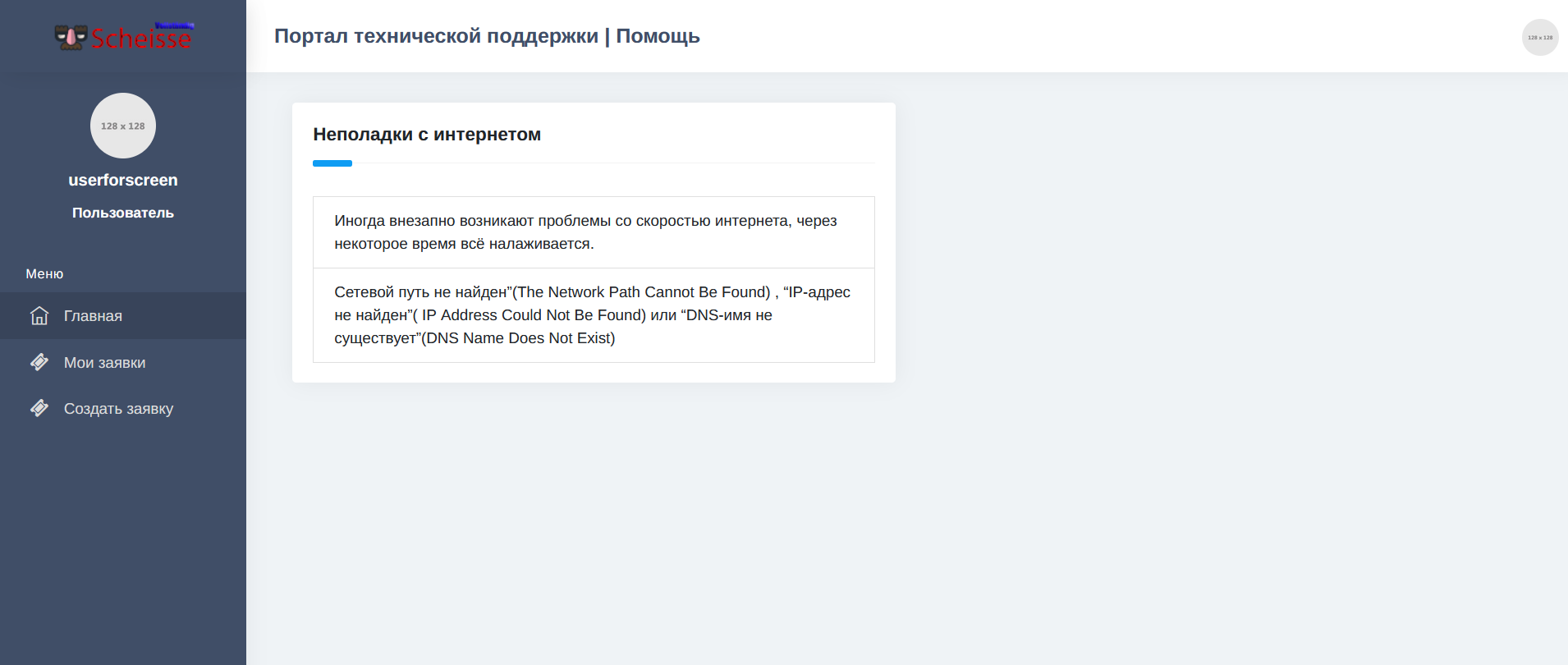


**Рисунок 4. Страница регистрации**

Присутствует возможность зарегистрировать аккаунт и принять «пользовательское соглашение» ([см. приложение 2](#Прил2))

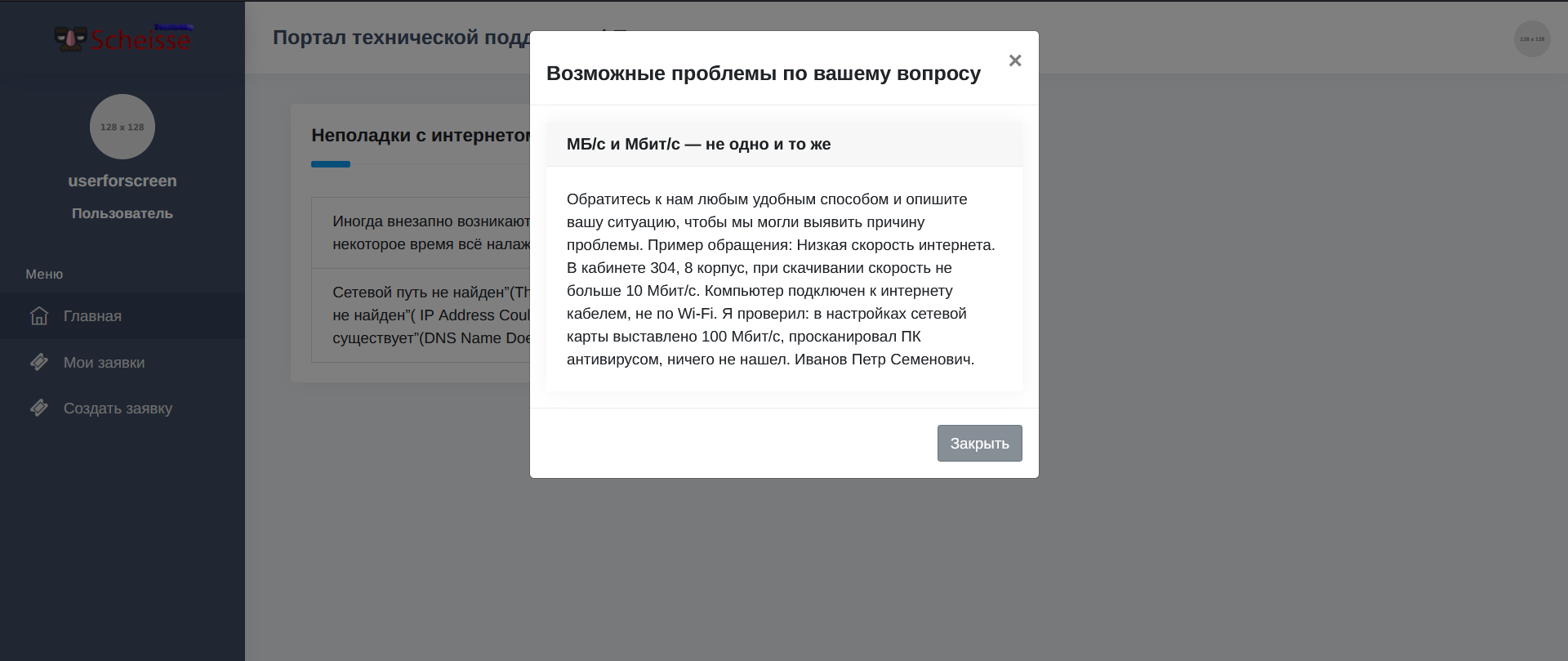
## **Главная страница для пользователей**

Главная страница для пользователей изображена на рисунке 5.



**Рисунок 5. Главная страница (пользователь)**

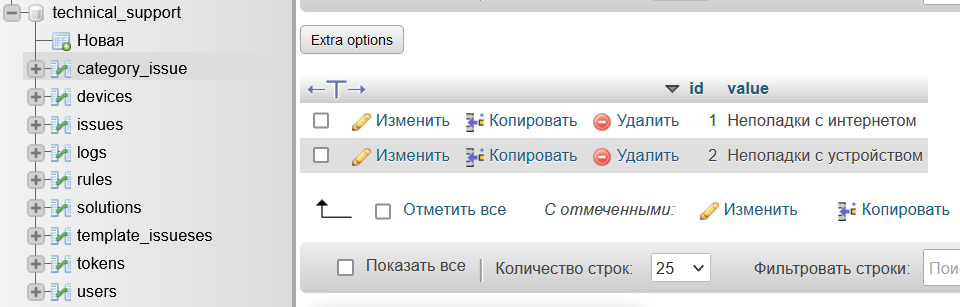
На главной странице для обычных пользователей реализован раздел с наиболее популярными проблемами и ответами на них. При нажатии на проблему появляется окошко с советом по данному вопросу (рис.6)



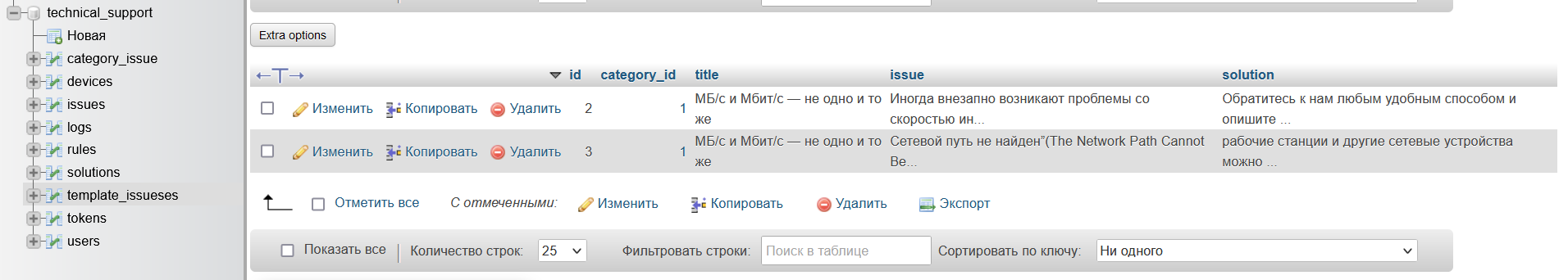
**Рисунок 6. Ответ на типовую проблему**

Наиболее популярные категории проблем, проблемы, и советы для их решения, отображаемые на главной странице, также хранятся в БД (рис.7-8)

([см. приложение 3](#Прил3))



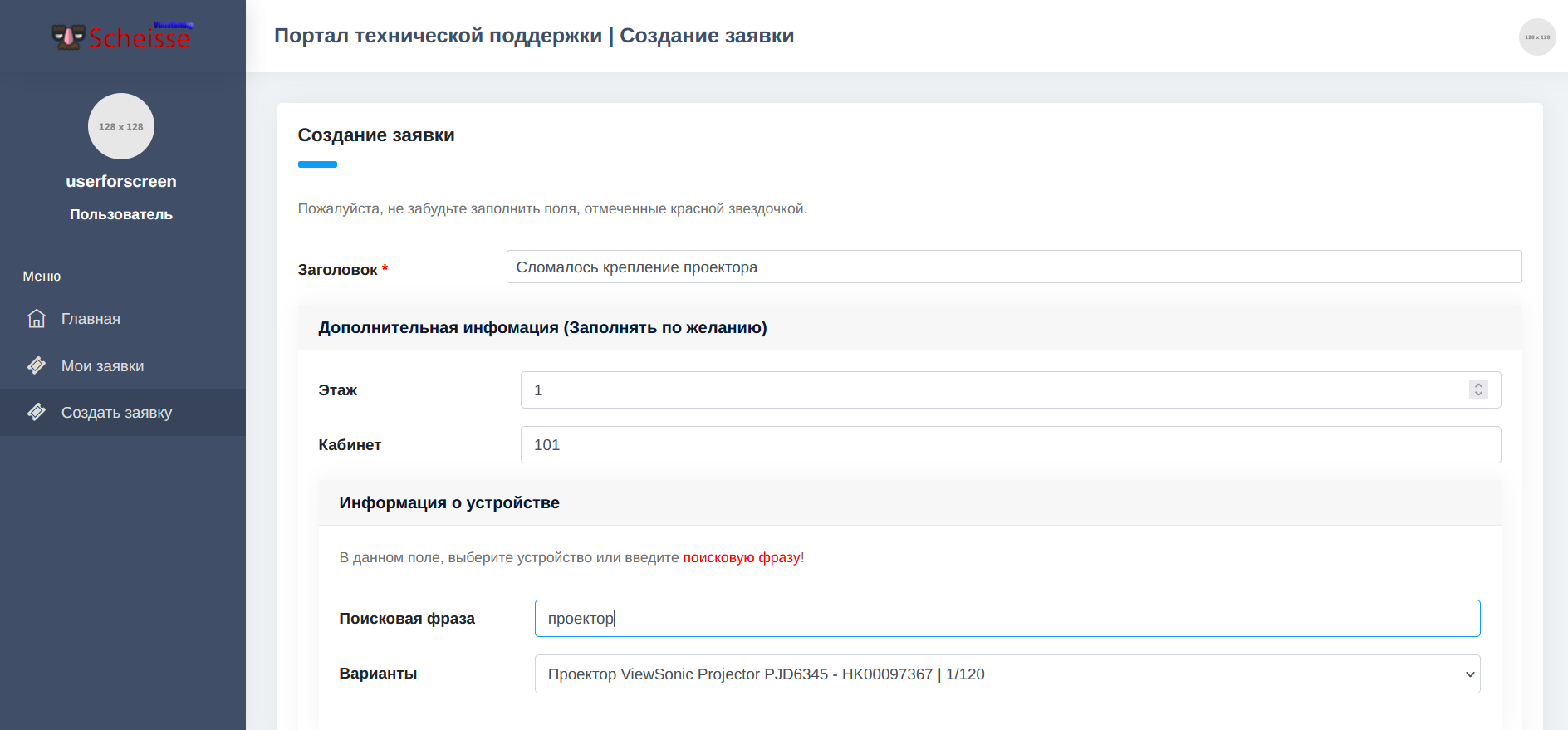
**Рисунок 7. Категории типовых проблем в БД**



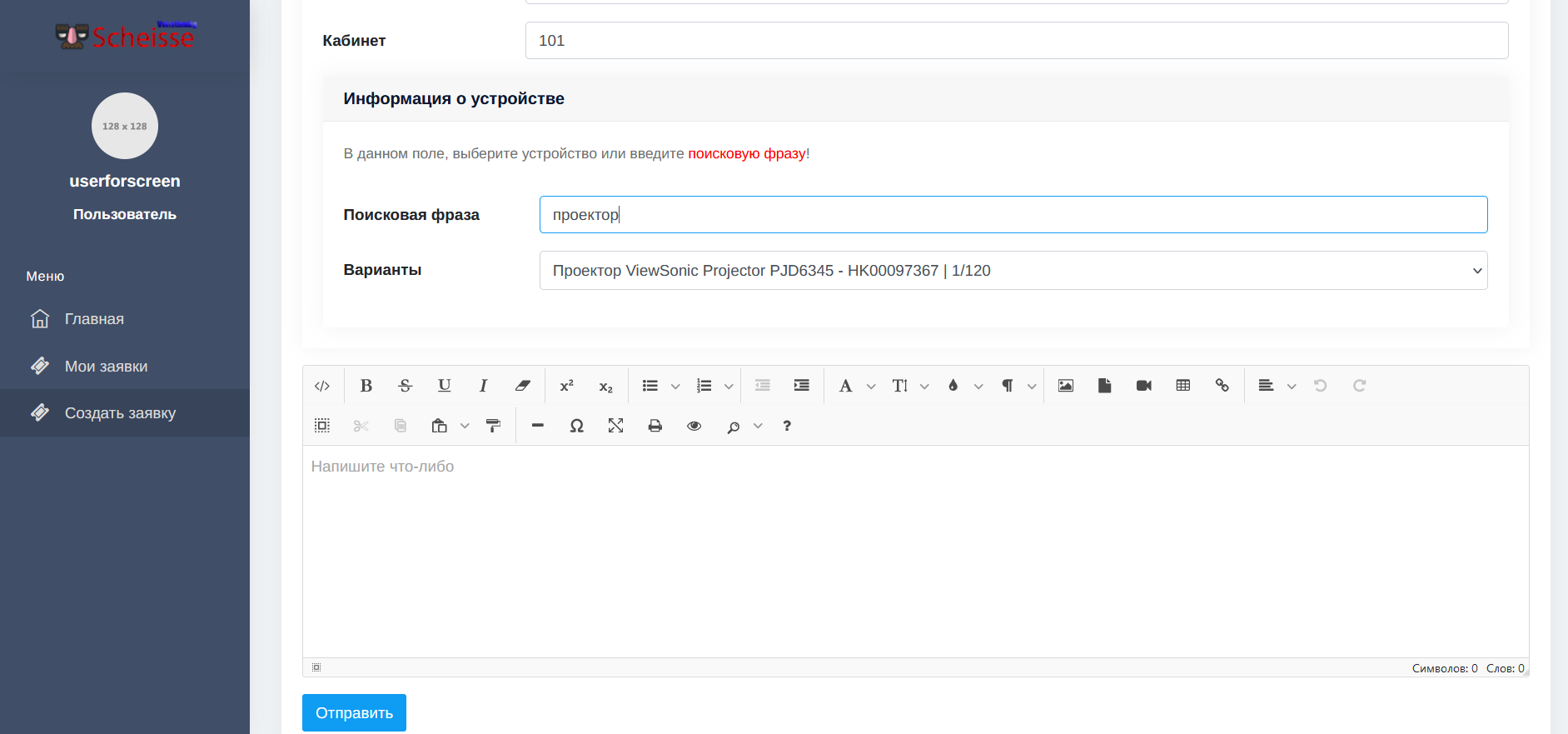
**Рисунок 8. Возможные решения типовых проблем в БД**

## **Страница создания заявки пользователем**

Страница создания заявки пользователем, собственно, позволяет пользователю создать заявку, подробно заполнив информацию о проблеме. Можно воспользоваться поисковой фразой и соответствующим ей выпадающим списком для ускорения поиска техники в инвентарной базе предприятия. Также возможно в произвольной форме описать проблему, используя поле для ввода текста (рис. 9-10)



**Рисунок 9. Страница создания заявки**



**Рисунок 10. Страница создания заявки**

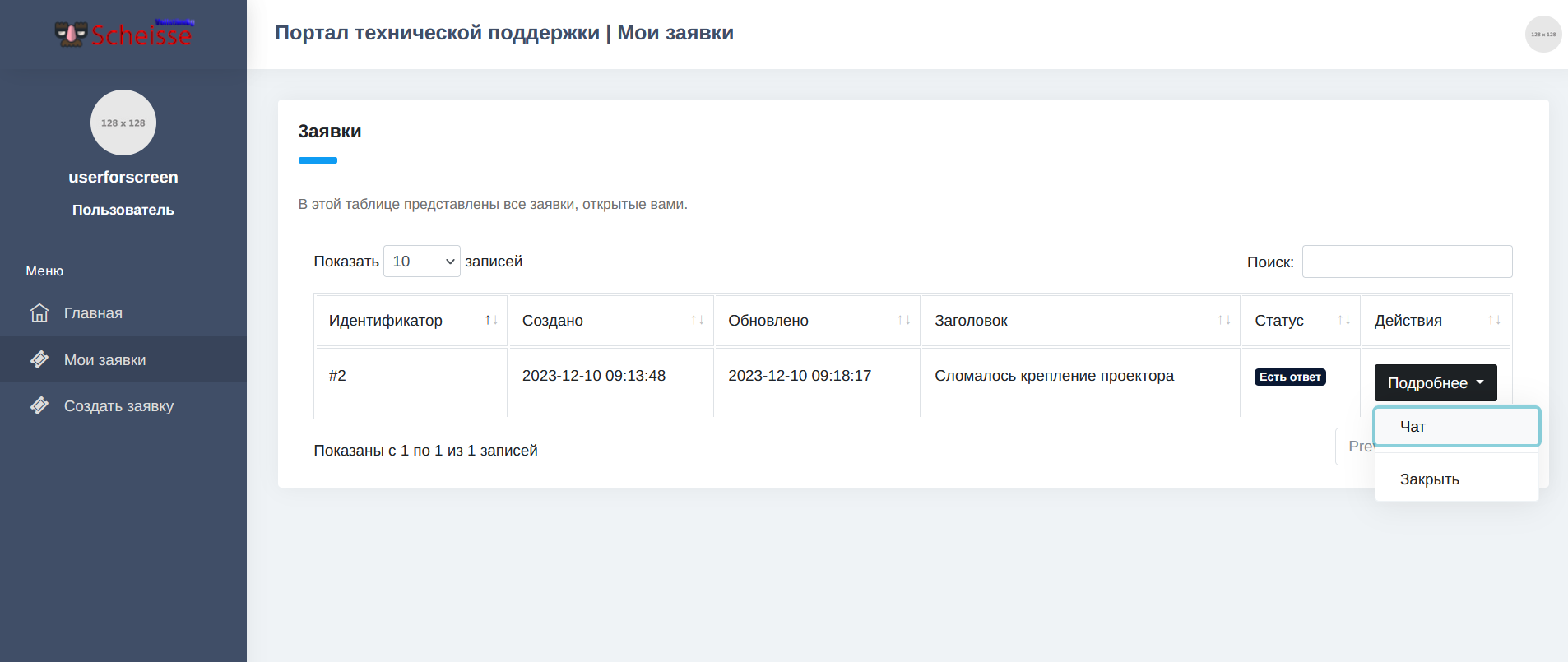
([см. приложение 4](#Прил4))

## **Страница «Мои заявки»**

После того, как пользователь создал заявку, на странице «Мои заявки» он сможет увидеть список своих заявок и подробную информацию о них:

* дата создания,
* дата обновления,
* заголовок,
* статус заявки: «открыта», «есть ответ», «в работе», «закрыта»

При нажатии на кнопку «Подробнее» в колонке «Действия», пользователь сможет открыть чат со специалистом поддержки (рис.11)



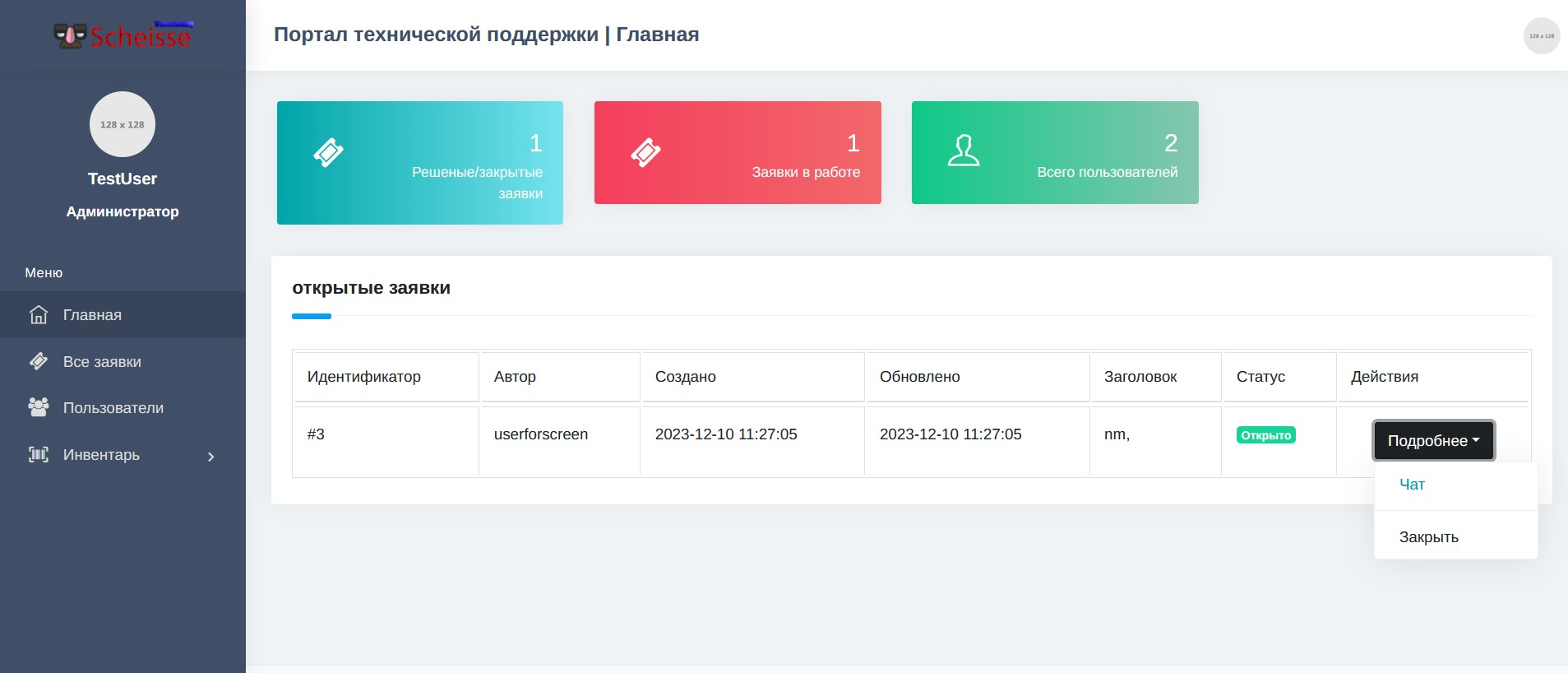
**Рисунок 11. Страница «Мои заявки»**

([см. приложение 5](#Прил5))

## **Главная страница для администратора**

На главной странице администратор может видеть список открытых заявок, а также статистику: количество решенных/закрытых заявок, количество заявок в работе, количество зарегистрированных пользователей.

При нажатии на кнопку «Подробнее» в колонке «Действия», администратор может открыть чат с пользователем или же закрыть заявку (рис.12)

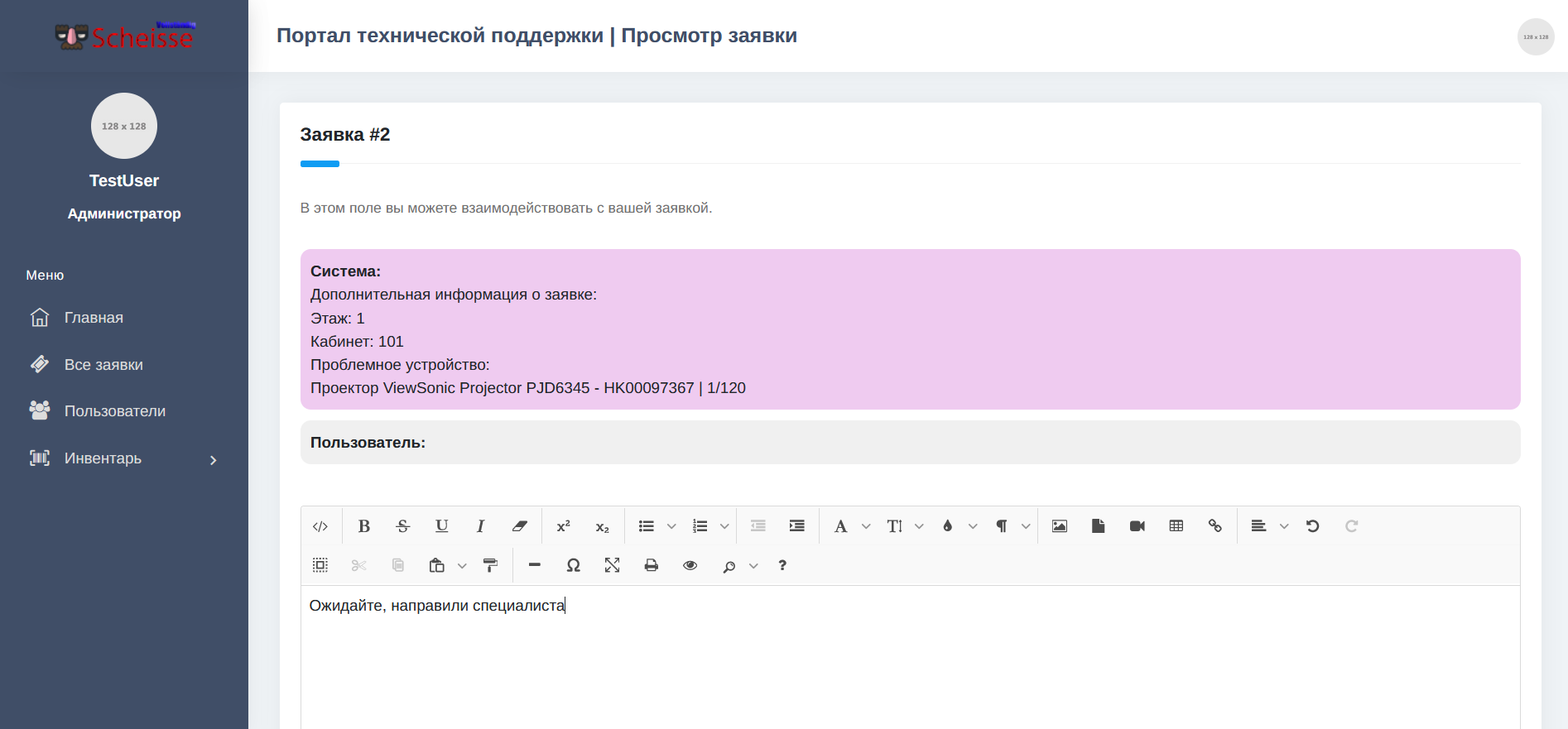


**Рисунок 12. Главная страница администратора**

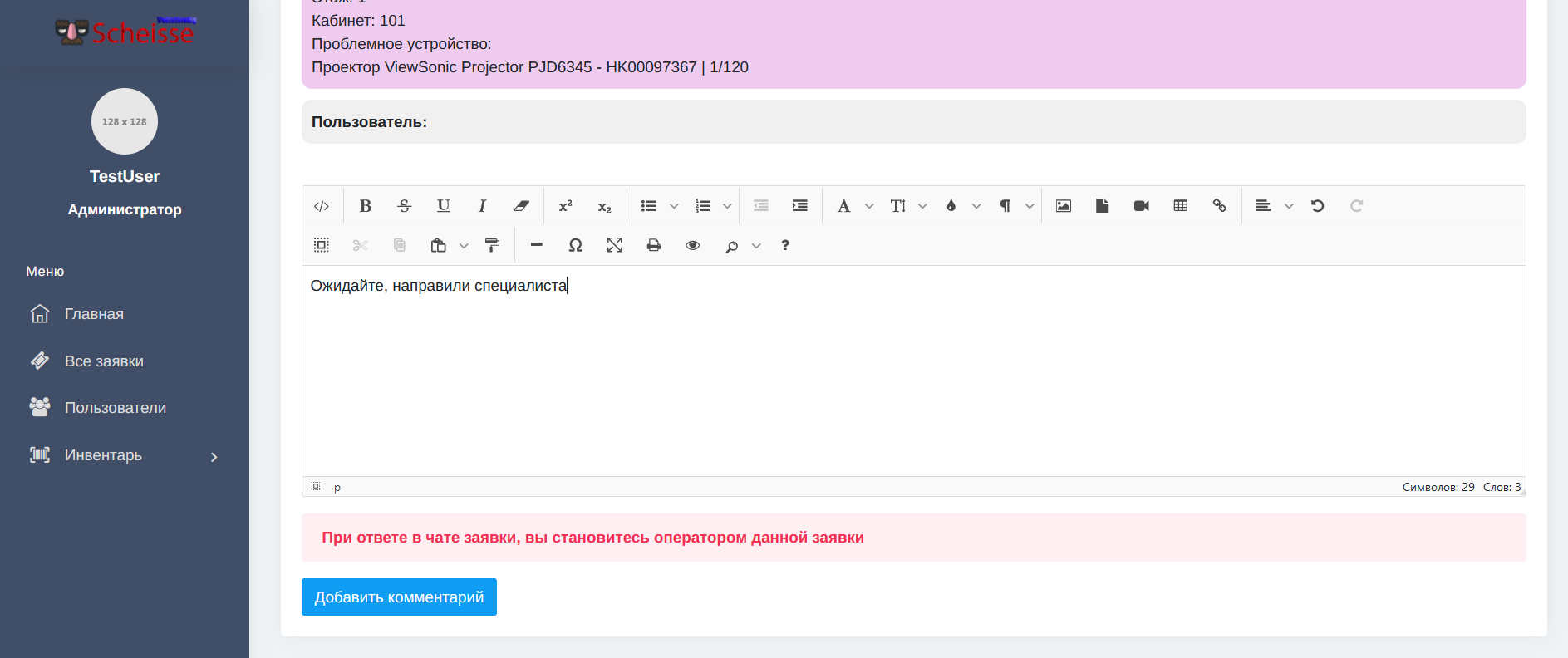
([см. приложение 6](#Прил6))

## **Страница просмотра заявки**

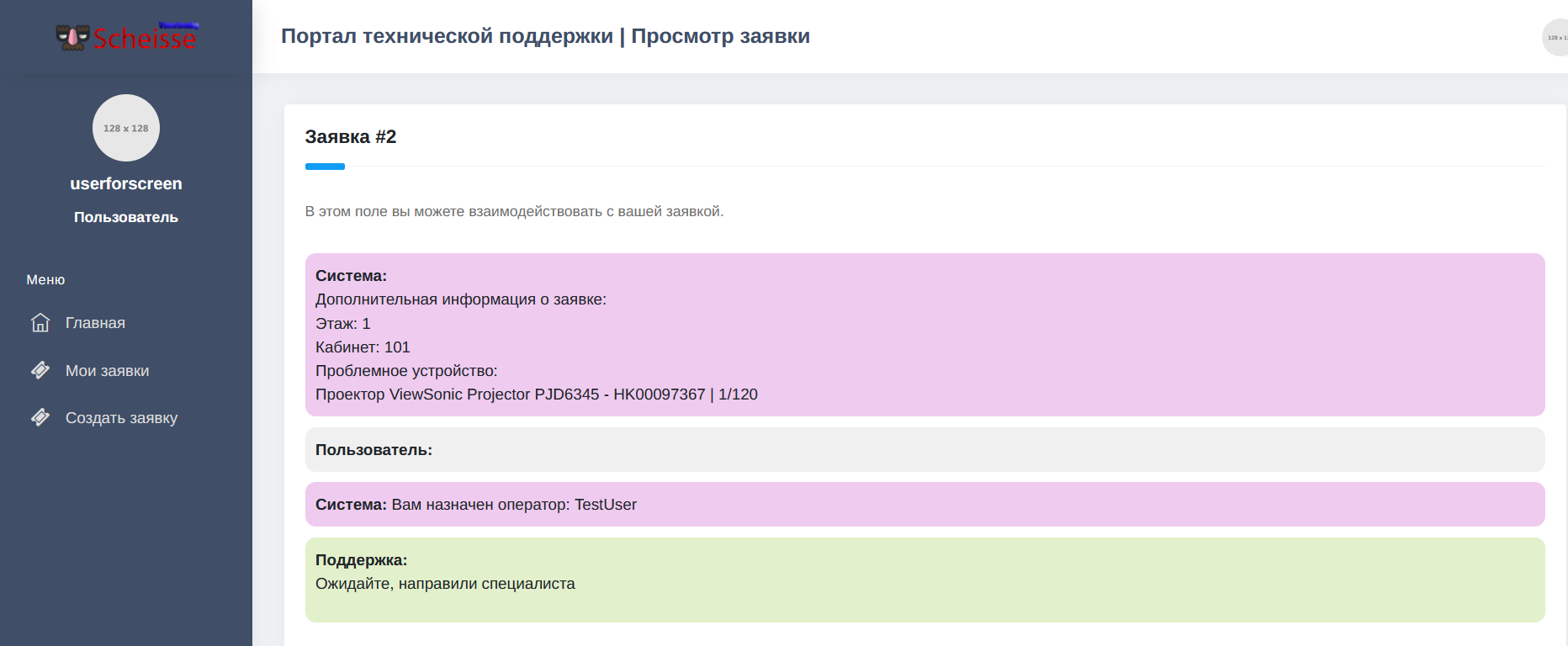
Как со стороны пользователя, так и со стороны администратора, по кнопке «Подробнее» в колонке «Действия» есть возможность открыть чат с противоположной стороной. Администратору будет подробно видна вся информация, что указал пользователь в заявке. Администратор и пользователь имеют возможность переписываться на данной странице. При ответе на заявку, данная заявка закрепляется за конкретным администратором. При смене статуса заявки, в чат пользователю будет направлено системное уведомление. Вся информация о заявке, включая закрепившегося администратора, хранится в БД (рис.13-17)



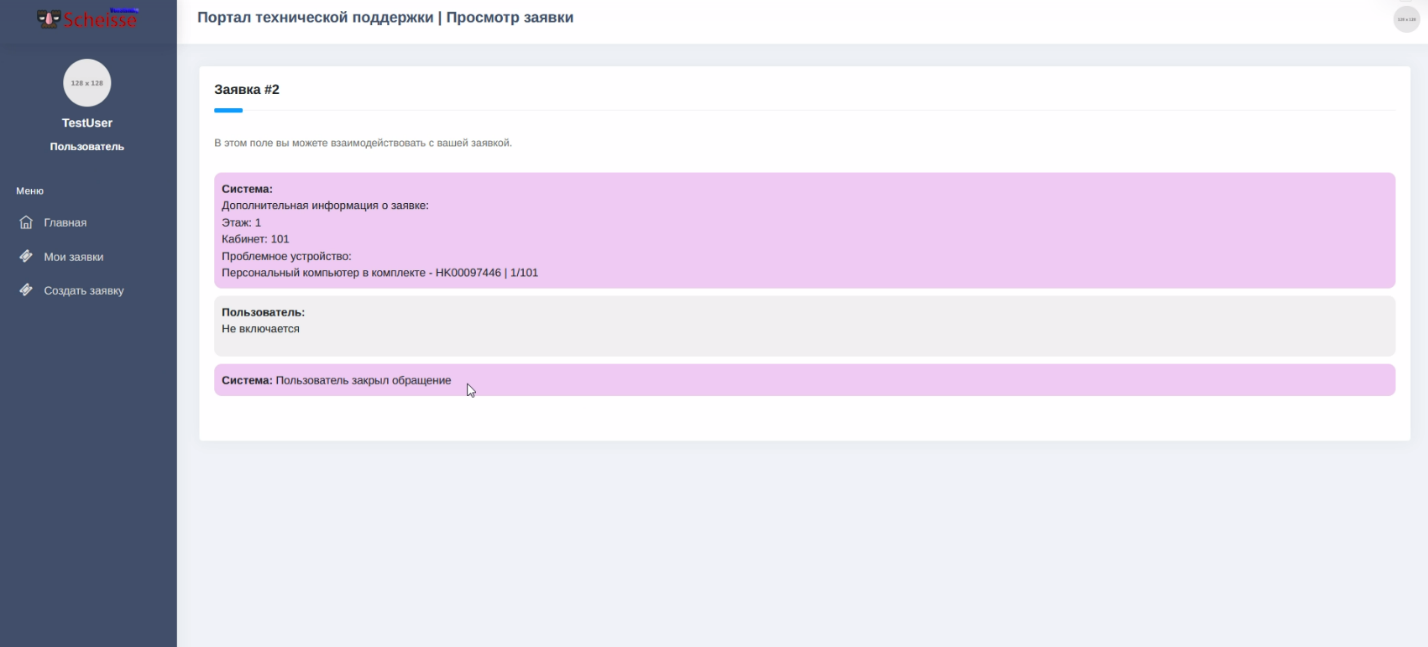
**Рисунок 13. Страница просмотра заявки (администратор)**



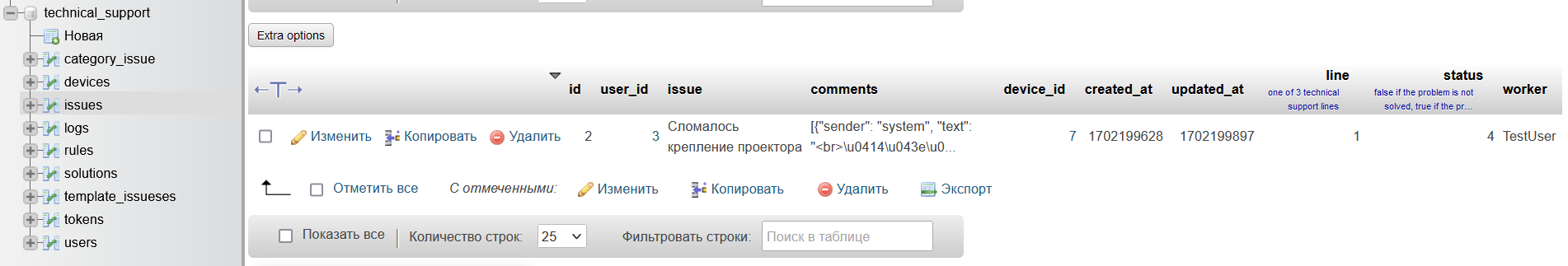
**Рисунок 14. Страница просмотра заявки (администратор)**



**Рисунок 15. Страница просмотра заявки (пользователь)**



**Рисунок 16. Страница просмотра заявки (пользователь)**

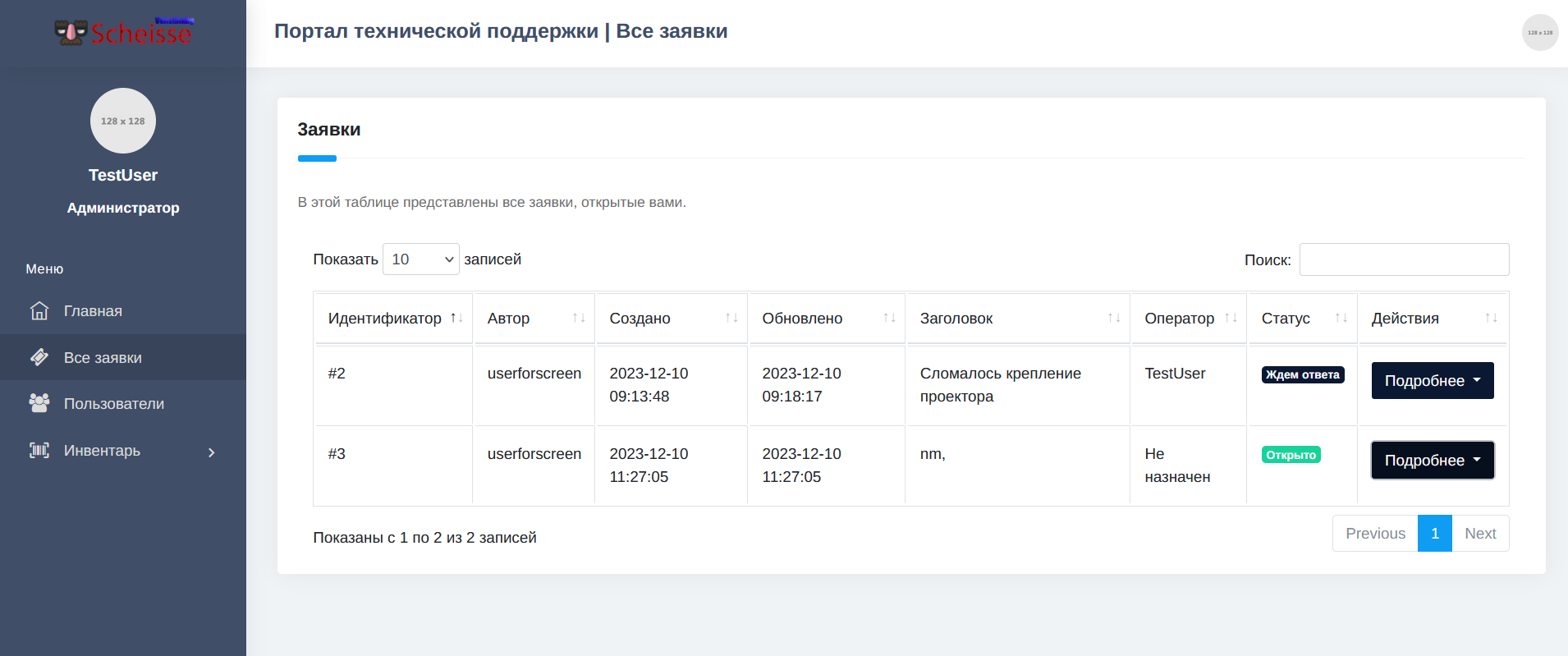


**Рисунок 17. Данные о заявке в БД**

([см. приложение 7](#Прил7))

## **Страница «Все заявки»**

На странице «Все заявки» администратор может просмотреть все заявки, на которые он когда-либо реагировал и новые заявки соответственно (рис.18)

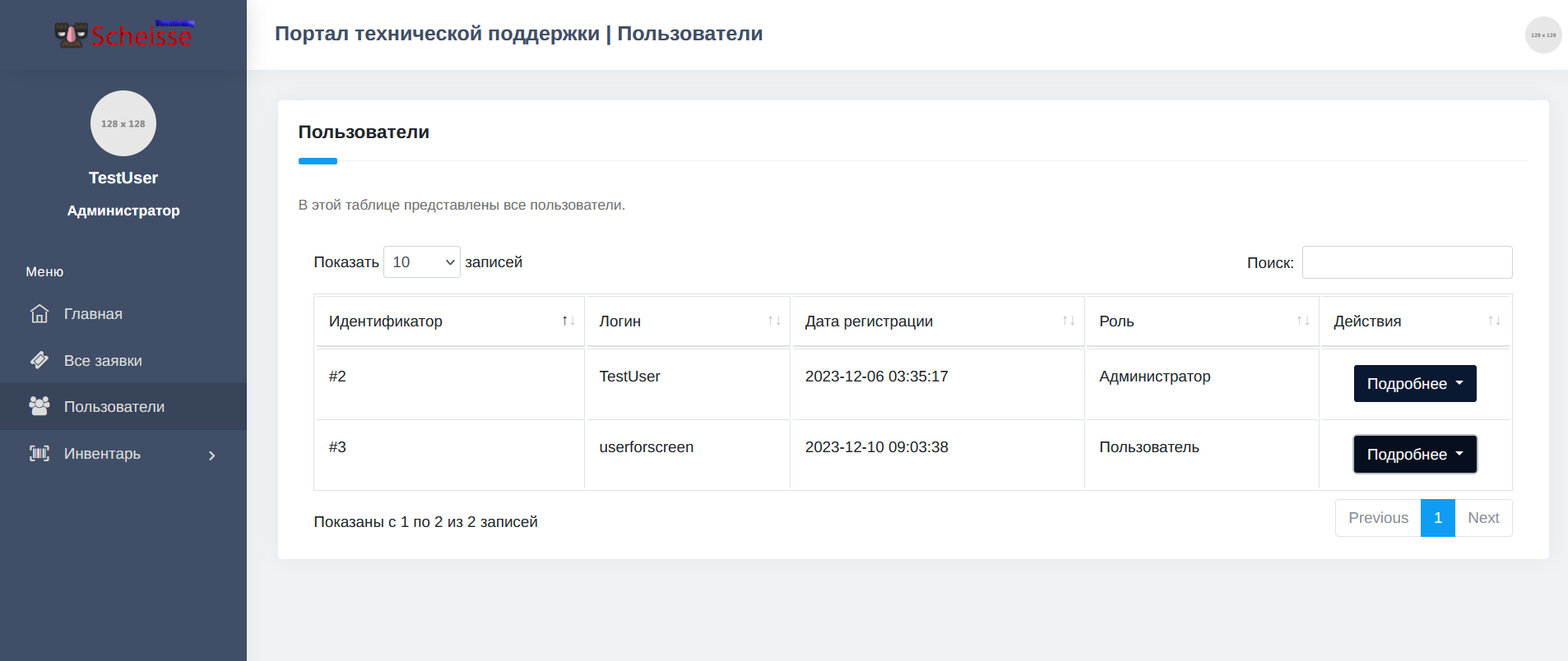


**Рисунок 18. Страница «Все заявки»**

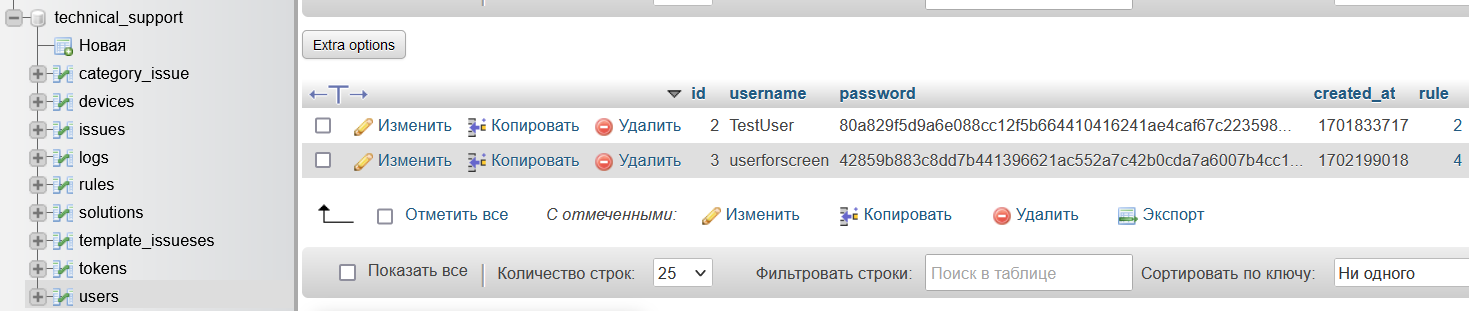
([см. приложение 5](#Прил5))

## **Страница «Пользователи»**

На странице «Пользователи» администратору видны все пользователи, зарегистрированные на портале. Вся информация хранится в БД (рис.19-20)



**Рисунок 19. Страница «Пользователи»**



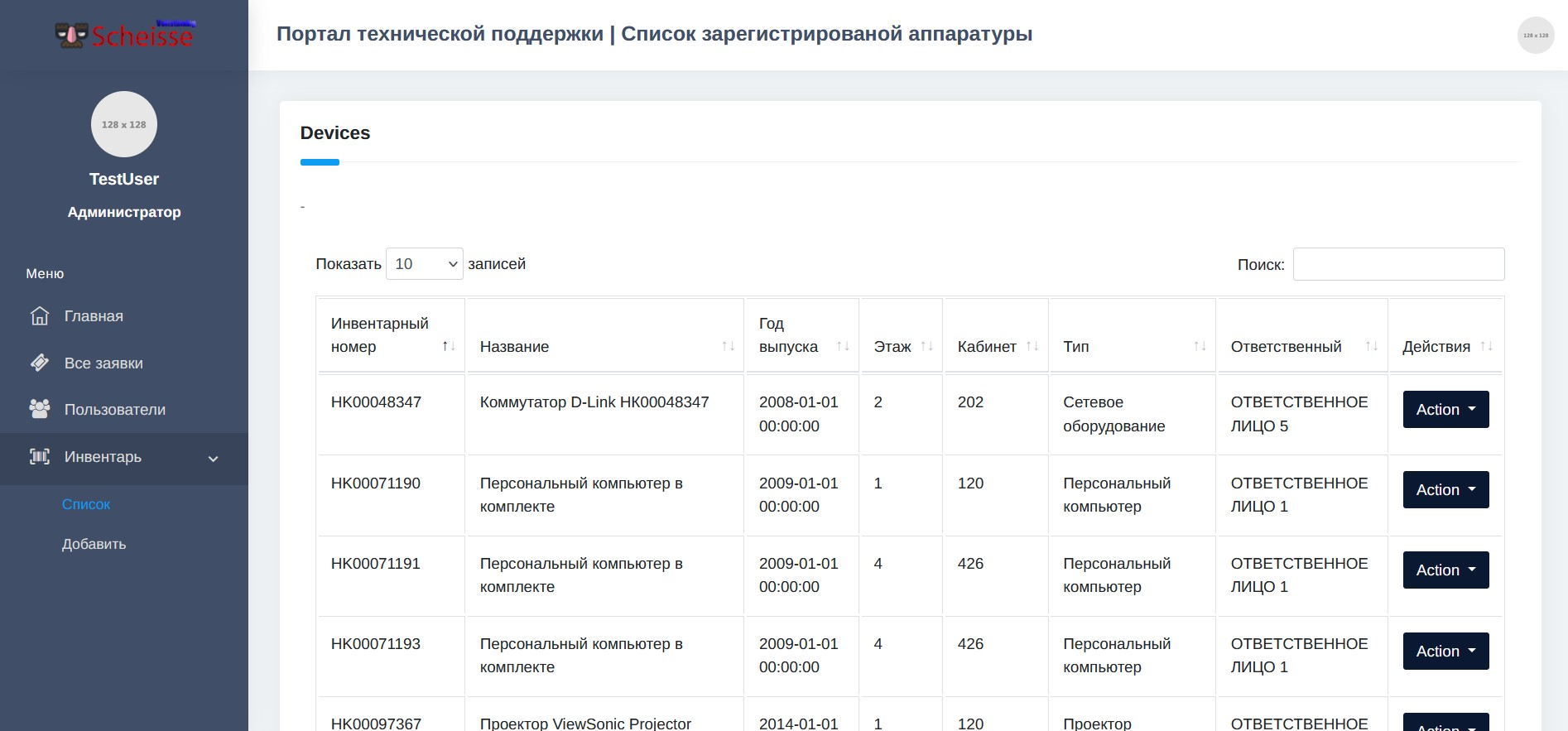
**Рисунок 10. Данные о пользователях в БД**

([см. приложение 8](#Прил8))

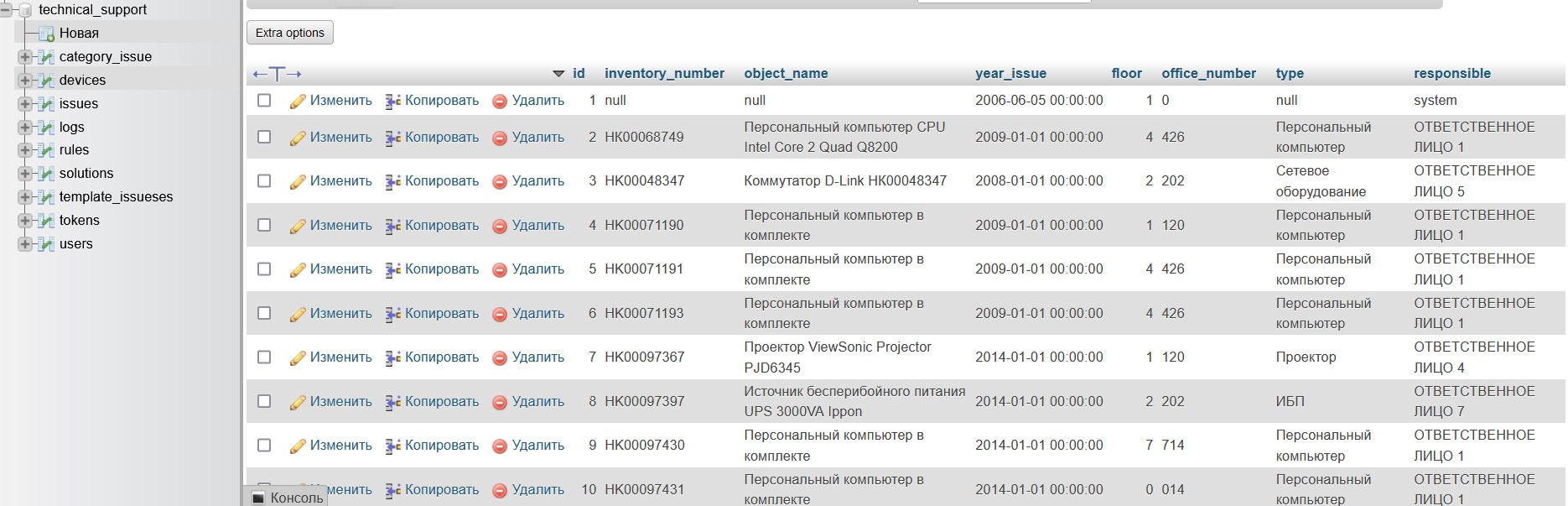
## **Инвентарь. Список аппаратуры**

Администратору доступен раздел «Инвентарь», содержащий 2 страницы: «Список» и «Добавить».

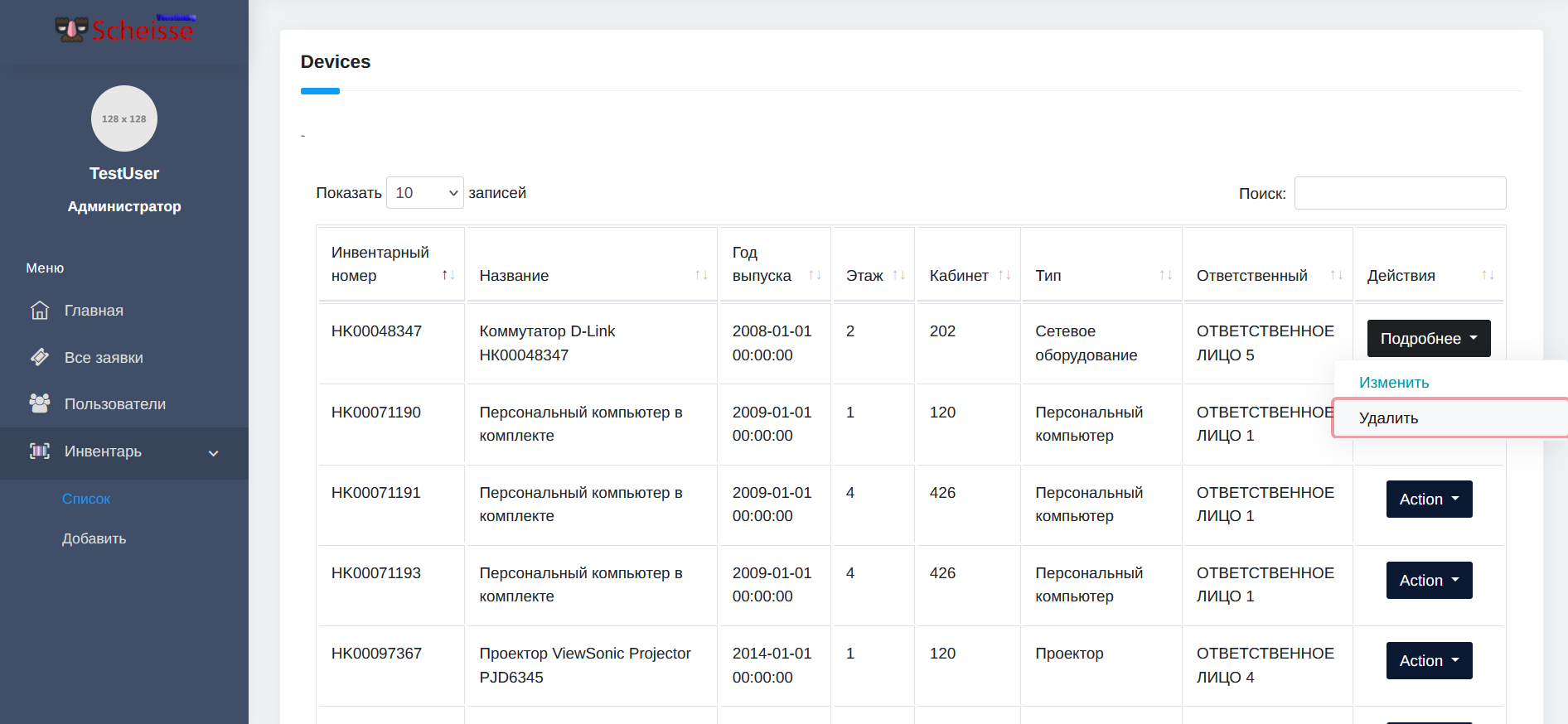
На странице «Список аппаратуры» администратору видны все сведения о технике, прошедшей инвентаризацию на предприятии. Данные хранятся в БД. (рис.21-22) При нажатии на кнопку «Подробнее» в колонке «Действия», администратор может изменить сведения о единице техники, либо удалить ее (рис.23)



**Рисунок 21. Страница «Список аппаратуры»**



**Рисунок 22. Данные о технике в БД**

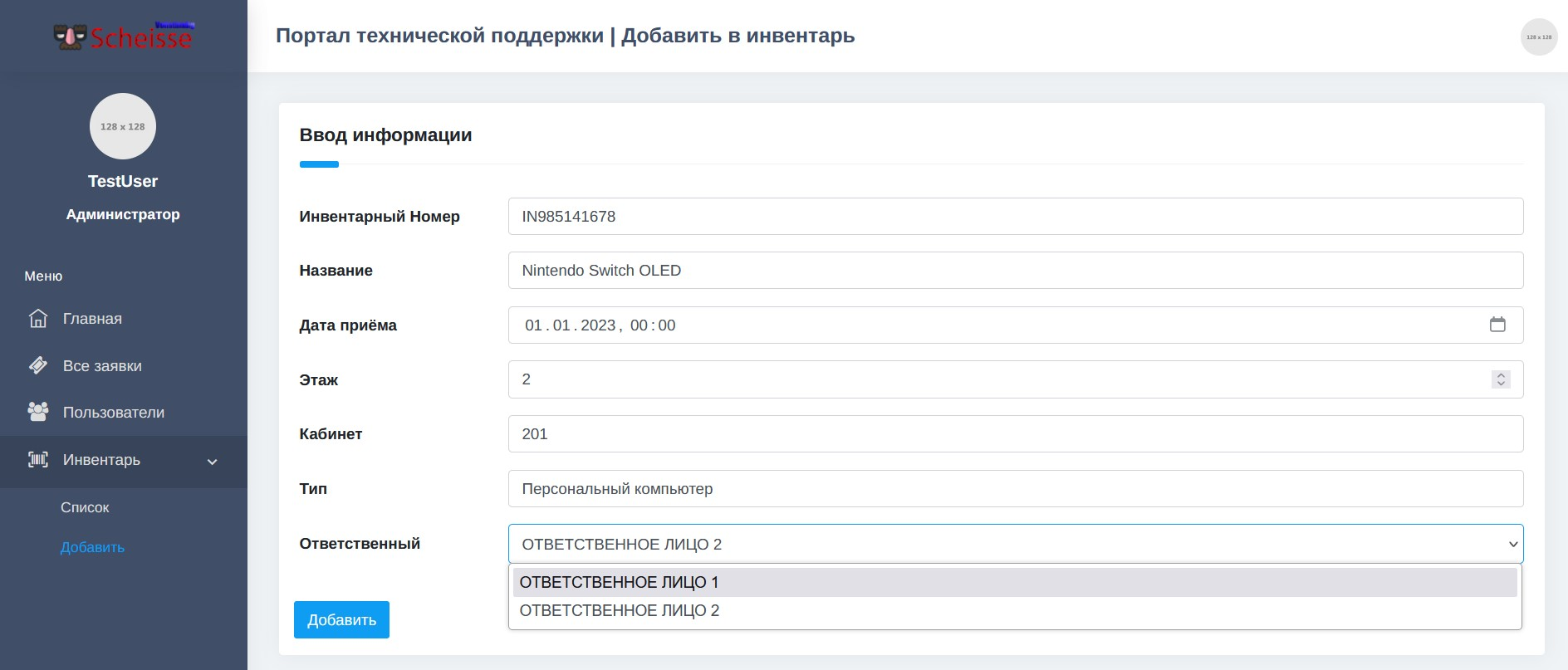


**Рисунок 23. Возможные действия со списком техники**

([см. приложение 9](#Прил9)[-10](#Прил10))

## **Инвентарь. Страница «Добавить»**

На странице «Добавить» в разделе «Инвентарь» администратор может заносить новую технику в БД. (рис.24) ([см. приложение 11](#Прил11))



**Рисунок 24. Страница «Добавить в инвентарь»**

## **Скрипты**

Также, мной были написаны скрипты для взаимодействия с БД

([см. приложение 12](#Прил12))

# **Заключение**

Курсовая работа на тему «Разработка ИС технической поддержки предприятия. Разработка подсистемы клиентов. Разработка подсистемы первой линии техподдержки» была выполнена в соответствии с поставленной задачей. В процессе выполнения данной работы были проанализированы программы и технологии, необходимые для выполнения поставленной цели.

Для реализации проекта были выбраны следующие программные средства: веб-приложение PhpMyAdmin, текстовый редактор Notepad++, текстовый редактор VSCode, язык разметки HTML, фреймворк Bootstrap, фреймворк для создания веб приложений на языке Python – Flask, распределённая система управления версиями Git.

Была разработана ИС технической поддержки предприятия, а также подсистема клиентов и подсистема первой линии техподдержки. Реализация отвечает всем поставленным требованиям.

# **Список источников**

1. Прохоренок Н. А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера. — 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2010. — 912 с.: ил. + Видеокурс (на CD-ROM) — (Профессиональное программирование).
2. Кириченко А.В., Дубовик Е.В. Динамические сайты на HTML, CSS, JavaScript и Bootstrap. – М.: Наука и техника, 2018. – 251 с.
3. Мартишин С. А. Базы данных. Практическое применение СУБД SQL- и NoSQL-типа для применения проектирования информационных систем. – М.: Форум, 2018. – 368 с.

# **Приложение**

Приложение 1

<html class=" js flexbox flexboxlegacy canvas canvastext webgl no-touch geolocation postmessage no-websqldatabase indexeddb hashchange history draganddrop websockets rgba hsla multiplebgs backgroundsize borderimage borderradius boxshadow textshadow opacity cssanimations csscolumns cssgradients no-cssreflections csstransforms csstransforms3d csstransitions fontface generatedcontent video audio localstorage sessionstorage webworkers no-applicationcache svg inlinesvg smil svgclippaths">

<head>

{% set page\_name = 'Login' %} {% include 'modules/head.html' %}

</head>

<body style="overflow: visible;">

<!-- Begin page -->

<div class="accountbg"></div>

<div class="wrapper-page">

<div class="card">

<div class="card-body">

{% include 'modules/logo.html' %}

<h4 class="text-muted text-center font-18"><b>Вход в учетную запись</b></h4>

<div class="p-3">

{% if error == 1 %}

<div class="alert alert-warning alert-dismissible">

<button type="button" class="close" data-dismiss="alert" aria-hidden="true">×</button>

Сессия истекла: <b>требуется авторизация</b>

</div>

{% endif %}

<form class="form-horizontal m-t-20" method="POST" action="{{ url\_for('routes.login') }}">

{% if status == 400 %}

<div class="alert alert-danger alert-dismissible">

<button type="button" class="close" data-dismiss="alert" aria-hidden="true">×</button>

Неправильный <b>Логин</b> или <b>Пароль</b>!

</div>

{% endif %}

{% if status == 401 %}

<div class="alert alert-danger alert-dismissible">

<button type="button" class="close" data-dismiss="alert" aria-hidden="true">×</button>

Ошибка <b>Пользователь не найден</b>!

</div>

{% endif %}

{% if status == 200 %}

<div class="alert alert-success alert-dismissible">

<button type="button" class="close" data-dismiss="alert" aria-hidden="true">×</button>

Авторизовались, токен: <b>{{ token }}</b>

</div>

{% endif %}

{% if status == 999 %}

<div class="alert alert-danger alert-dismissible">

<button type="button" class="close" data-dismiss="alert" aria-hidden="true">×</button>

Вы не можете войти с ролью: <b>{{rule}}</b>!

</div>

{% endif %}

<div class="form-group row">

<div class="col-12">

<input name="username" class="form-control" type="text" required="" placeholder="Логин">

</div>

</div>

<div class="form-group row">

<div class="col-12">

<input name="password" class="form-control" type="password" required="" placeholder="Пароль">

</div>

</div>

<div class="form-group row">

<label class="col-sm-2 col-form-label">Роль</label>

<div class="col-sm-10">

<select name="rule" class="form-control">

<option>Пользователь</option>

<option>Сотрудник</option>

<option>Администратор</option>

</select>

</div>

</div>

<b>&nbsp</b>

<div class="form-group row">

<div class="col-12">

<div class="custom-control custom-checkbox">

<input type="checkbox" class="custom-control-input" id="customCheck1">

<label class="custom-control-label" for="customCheck1">Запомнить меня</label>

</div>

</div>

</div>

<div class="form-group text-center row m-t-20">

<div class="col-12">

<button class="btn btn-info btn-block waves-effect waves-light" type="submit">Войти</button>

</div>

</div>

<div class="form-group m-t-10 mb-0 row">

<div class="col-12 m-t-20 text-center" >

<a href="register" class="text-muted"><i class="mdi mdi-account-circle"></i> Создать аккаунт</a>

</div>

</div>

</form>

</div>

</div>

</div>

</div>

{% include 'modules/scripts.html' %}

</body></html>

Приложение 2

<!DOCTYPE html>

<html>

<head>

{% set page\_name = 'Register' %} {% include 'modules/head.html' %}

</head>

<body>

<!-- Begin page -->

<div class="accountbg"></div>

<div class="wrapper-page">

<div class="card">

<div class="card-body">

<div id="myModal" class="modal fade" tabindex="-1" role="dialog">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h4 class="modal-title">Пользовательское соглашение</h4>

</div>

<div class="modal-body">

<div class="embed-responsive embed-responsive-16by9">

<iframe class="embed-responsive-item" src="https://www.youtube.com/watch?v=dQw4w9WgXcQ&pp=ygUXbmV2ZXIgZ29ubmEgZ2l2ZSB5b3UgdXA%3D" allowfullscreen></iframe>

</div>

<p>Не воруйте наш код, пожалуйста</p>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">Закрыть</button>

</div>

</div>

</div>

</div>

<h3 class="text-center mt-0 m-b-15">

<a href="index.html" class="logo logo-admin"><img src="assets/images/logo.png" height="30" alt="logo"></a>

</h3>

<h4 class="text-muted text-center font-18"><b>Создать учетную запись</b></h4>

<div class="p-3">

<form class="form-horizontal m-t-20" method="POST" action="{{ url\_for('routes.register') }}">

{% if status == 409 %}

<div class="alert alert-danger alert-dismissible">

<button type="button" class="close" data-dismiss="alert" aria-hidden="true">×</button>

<p>Пользователь уже существует</p>

</div>

{% endif %}

{% if status == 400 %}

<div class="alert alert-danger alert-dismissible">

<button type="button" class="close" data-dismiss="alert" aria-hidden="true">×</button>

<p>{{ \_error|safe }}</p>

</div>

{% endif %}

{% if status == 200 %}

<div class="alert alert-success alert-dismissible">

<button type="button" class="close" data-dismiss="alert" aria-hidden="true">×</button>

Регистрация прошла успешно, статус: <b>{{ status }}</b>

</div>

{% endif %}

<div class="form-group row">

<div class="col-12">

<input class="form-control" name="username" type="text" required="" placeholder="Логин">

</div>

</div>

<div class="form-group row">

<div class="col-12">

<input class="form-control" name="password" type="password" required="" placeholder="Пароль">

</div>

</div>

<div class="form-group row">

<div class="col-12">

<div class="custom-control custom-checkbox">

<input name="user\_agreement" type="checkbox" class="custom-control-input" id="termsId" required="Примите условия пользовательского соглашения">

<label class="custom-control-label font-weight-normal" for="termsId">Я ознакомлен с <a href="#" data-toggle="modal" data-target="#myModal" class="text-muted">пользовательским соглашением</a></label>

</div>

</div>

</div>

<div class="form-group text-center row m-t-20">

<div class="col-12">

<button class="btn btn-info btn-block waves-effect waves-light" type="submit">Зарегестрироваться</button>

</div>

</div>

<div class="form-group m-t-10 mb-0 row">

<div class="col-12 m-t-20 text-center">

<a href="login" class="text-muted">У вас уже есть учетная запись?</a>

</div>

</div>

</form>

</div>

</div>

</div>

</div>

{% include 'modules/scripts.html' %}

<script>

$(document).ready(function() {

$('a[href="#"]').click(function(e) {

e.preventDefault();

var target = $(this).data('target');

$(target).modal('show');

});

});

</script>

</body>

</html>

Приложение 3

<!DOCTYPE html>

<html>

<head>

{% set page\_name = 'User Panel' %} {% include 'modules/head.html' %}

</head>

<body class="fixed-left">

<!-- Loader -->

<div id="preloader"><div id="status"><div class="spinner"></div></div></div>

<!-- Begin page -->

<div id="wrapper">

{% include "./modules/left.html" %}

<!-- Start right Content here -->

<div class="content-page">

<!-- Start content -->

<div class="content">

<!-- Top Bar Start -->

<div class="topbar">

<nav class="navbar-custom">

<ul class="list-inline float-right mb-0">

<li class="list-inline-item dropdown notification-list">

<a class="nav-link dropdown-toggle arrow-none waves-effect nav-user" data-toggle="dropdown" href="#" role="button"

aria-haspopup="false" aria-expanded="false">

<img src="assets/images/users/avatar-5.jpg" alt="user" class="rounded-circle">

</a>

<div class="dropdown-menu dropdown-menu-right profile-dropdown ">

<a class="dropdown-item" href="logout"><i class="mdi mdi-logout m-r-5 text-muted"></i> Выход</a>

</div>

</li>

</ul>

<ul class="list-inline menu-left mb-0">

<li class="list-inline-item">

<button type="button" class="button-menu-mobile open-left waves-effect">

<i class="ion-navicon"></i>

</button>

</li>

<li class="hide-phone list-inline-item app-search">

<h3 class="page-title">Портал технической поддержки | Помощь</h3>

</li>

</ul>

<div class="clearfix"></div>

</nav>

</div>

<!-- Top Bar End -->

<div class="page-content-wrapper ">

<div class="container-fluid">

{% for category in template\_issueses['categories'] %}

<div class="col-lg-6">

<div class="card m-b-30">

<div class="card-body">

<h4 class="mt-0 header-title"> {{ category['category'] }} </h4>

<div class="list-group">

{% for issue in category['list'] %}

<a data-toggle="modal" data-target="#myModal{{ loop.index0 }}" class="list-group-item list-group-item-action">{{ issue['issue'] }}</a>

<div class="modal fade" id="myModal{{ loop.index0 }}" tabindex="-1" role="dialog" aria-labelledby="myModalLabel{{ loop.index0 }}" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="myModalLabel{{ loop.index0 }}">Возможные проблемы по вашему вопросу</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Закрыть">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

<div id="accordion" role="tablist" aria-multiselectable="true">

<div class="card">

<div class="card-header" role="tab" id="headingOne{{ loop.index0 }}">

<h5 class="mb-0 mt-0 font-16">

<a data-toggle="collapse" data-parent="#accordion" href="#collapseOne{{ loop.index0 }}" aria-expanded="false" aria-controls="collapseOne{{ loop.index0 }}" class="text-dark collapsed">

{{ issue['title'] }}

</a>

</h5>

</div>

<div id="collapseOne{{ loop.index0 }}" class="collapse" role="tabpanel" aria-labelledby="headingOne{{ loop.index0 }}" style="">

<div class="card-body">

{{issue['solution']|safe}}

</div>

</div>

</div>

</div>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-dismiss="modal">Закрыть</button>

</div>

</div>

</div>

</div>

{% endfor %}

</div>

</div>

</div>

</div>

{% endfor %}

</div>

</div>

</div> <!-- content -->

<footer class="footer">

© 2023 ParinovYT <i class="mdi mdi-heart text-danger"></i>

</footer>

</div>

<!-- End Right content here -->

</div>

<!-- END wrapper -->

<script>

$(document).ready(function () {

// Добавляем обработчик события клика для всех элементов с классом 'clickable'

$('.clickable').on('click', function () {

// Переключаем состояние видимости collapse при клике

$($(this).data('target')).collapse('toggle');

});

});

</script>

{% include 'modules/scripts.html' %}

</body>

</html>

Приложение 4

<!DOCTYPE html>

<html>

<head>

{% set page\_name = 'Add Ticket' %} {% include 'modules/head.html' %}

</head>

<body class="fixed-left">

<!-- Loader -->

<div id="preloader">

<div id="status">

<div class="spinner"></div>

</div>

</div>

<!-- Begin page -->

<div id="wrapper">

{% include "./modules/left.html" %}

<!-- Start right Content here -->

<div class="content-page">

<!-- Start content -->

<div class="content">

<!-- Top Bar Start -->

<div class="topbar">

<nav class="navbar-custom">

<ul class="list-inline float-right mb-0">

<li class="list-inline-item dropdown notification-list">

<a class="nav-link dropdown-toggle arrow-none waves-effect nav-user" data-toggle="dropdown" href="#" role="button" aria-haspopup="false" aria-expanded="false">

<img src="assets/images/users/avatar-5.jpg" alt="user" class="rounded-circle">

</a>

<div class="dropdown-menu dropdown-menu-right profile-dropdown ">

<a class="dropdown-item" href="logout">

<i class="mdi mdi-logout m-r-5 text-muted"></i> Выход </a>

</div>

</li>

</ul>

<ul class="list-inline menu-left mb-0">

<li class="list-inline-item">

<button type="button" class="button-menu-mobile open-left waves-effect">

<i class="ion-navicon"></i>

</button>

</li>

<li class="hide-phone list-inline-item app-search">

<h3 class="page-title">Портал технической поддержки | Создание заявки</h3>

</li>

</ul>

<div class="clearfix"></div>

</nav>

</div>

<!-- Top Bar End -->

<div class="page-content-wrapper ">

<div class="container-fluid">

<div class="row">

<div class="col-12">

<div class="card m-b-30">

<div class="card-body">

<h4 class="mt-0 header-title">Создание заявки</h4>

<p class="text-muted m-b-30 font-14">Пожалуйста, не забудьте заполнить поля, отмеченные красной звездочкой.</p>

<form class="form-horizontal m-t-20" method="POST" action="{{ url\_for('routes.addticket') }}">

<div class="form-group row">

<label for="label\_desc" class="col-sm-2 col-form-label">Заголовок <a style='color:red'>\*</a>

</label>

<div class="col-sm-10">

<input name='desc\_ticket' class="form-control form-control-sm" type="text" required placeholder="Введите краткую информацию" id="label\_desc">

</div>

</div>

<div class="card">

<div class="card-header" role="tab" id="headingOne">

<h5 class="mb-0 mt-0 font-16">

<a data-toggle="collapse" data-parent="#accordion" href="#collapseOne" aria-expanded="true" aria-controls="collapseOne" data-toggle="collapse" class="text-dark"> Дополнительная инфомация (Заполнять по желанию) </a>

</h5>

</div>

<div id="collapseOne" class="collapse show" role="tabpanel" aria-labelledby="headingOne" style="">

<div class="card-body">

<div class="form-group row">

<label for="example-number-input" class="col-sm-2 col-form-label">Этаж</label>

<div class="col-sm-10">

<input name="floor" class="form-control" type="number" value="" id="example-number-input">

</div>

</div>

<div class="form-group row">

<label for="example-number-input" class="col-sm-2 col-form-label">Кабинет</label>

<div class="col-sm-10">

<input name="office" class="form-control" type="text" value="" id="example-number-input">

</div>

</div>

<div class="card">

<div class="card-header" role="tab" id="headingTwo">

<h5 class="mb-0 mt-0 font-16">

<a data-toggle="collapse" data-parent="#accordion" href="#collapseTwo" aria-expanded="true" aria-controls="collapseTwo" data-toggle="collapse" class="text-dark"> Информация о устройстве </a>

</h5>

</div>

<div id="collapseTwo" class="collapse show" role="tabpanel" aria-labelledby="headingTwo" style="">

<div class="card-body">

<p class="text-muted m-b-30 font-14">В данном поле, выберите устройство или введите <a style='color:red'>поисковую фразу</a>! </p>

<div class="form-group row">

<label for="device-name" class="col-sm-2 col-form-label">Поисковая фраза</label>

<div class="col-sm-10">

<input name="device-name" class="form-control" type="text" value="" id="device-name">

</div>

</div>

<div class="form-group row">

<label for="device-name" class="col-sm-2 col-form-label">Варианты</label>

<div class="col-sm-10">

<select name='device\_name' id="device-name-list" class="form-control">

<option value='none'>Выбрать (наименование - инвентаризационный номер | Этаж/Кабинет)</option>

{% for device in devices %}

<option>{{device[2]}} - {{device[1]}} | {{device[4]}}/{{device[5]}}</option>

{% endfor %}

</select>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

<div class="form-group row"></div>

<textarea name="text\_ticket" id="editor1"></textarea>

<div class="form-group row"></div>

<input name="create" hidden value="true">

<button type="submit" class="btn btn-primary waves-effect m-l-5"> Отправить </button>

</div>

</div>

</form>

</div>

<!-- end col -->

</div>

<!-- end row -->

</div>

<!-- container -->

</div>

<!-- Page content Wrapper -->

</div>

<!-- content -->

<footer class="footer"> © 2023 ParinovYT <i class="mdi mdi-heart text-danger"></i>

</footer>

</div>

<!-- End Right content here -->

</div>

<!-- END wrapper --> {% include 'modules/scripts.html' %} <script>

$(document).ready(function() {

var editor1 = Jodit.make('#editor1', {

height: 300

});

});

</script>

<script>

$(document).ready(function() {

var devices = [];

$('#device-name-list option').each(function() {

devices.push($(this).text().trim());

});

var deviceNameInput = $("#device-name");

var deviceNameList = $("#device-name-list");

deviceNameInput.on('input', function() {

deviceNameList.empty();

var inputText = deviceNameInput.val().toLowerCase();

if (inputText.length > 0) {

for (var i = 0; i < devices.length; i++) {

if (devices[i].toLowerCase().indexOf(inputText) !== -1) {

var option = $("<option>", {

"class": "dropdown-item", href: "#", text: devices[i]

});

option.click(function() {

deviceNameInput.val(devices[i]);

// deviceNameList.empty();

});

deviceNameList.append(option);

}

}

}

});

});

</script>

</body>

</html>

Приложение 5

<!DOCTYPE html>

<html>

<head>

{% set page\_name = 'Tickets' %} {% include 'modules/head.html' %}

</head

<body class="fixed-left">

<!-- Loader -->

<div id="preloader"><div id="status"><div class="spinner"></div></div></div>

<!-- Begin page -->

<div id="wrapper">

{% include "./modules/left.html" %}

<!-- Start right Content here -->

<div class="content-page">

<!-- Start content -->

<div class="content">

<!-- Top Bar Start -->

<div class="topbar">

<nav class="navbar-custom">

<ul class="list-inline float-right mb-0">

<li class="list-inline-item dropdown notification-list">

<a class="nav-link dropdown-toggle arrow-none waves-effect nav-user" data-toggle="dropdown" href="#" role="button"

aria-haspopup="false" aria-expanded="false">

<img src="assets/images/users/avatar-5.jpg" alt="user" class="rounded-circle">

</a>

<div class="dropdown-menu dropdown-menu-right profile-dropdown ">

<a class="dropdown-item" href="logout"><i class="mdi mdi-logout m-r-5 text-muted"></i> Выход</a>

</div>

</li>

</ul>

<ul class="list-inline menu-left mb-0">

<li class="list-inline-item">

<button type="button" class="button-menu-mobile open-left waves-effect">

<i class="ion-navicon"></i>

</button>

</li>

<li class="hide-phone list-inline-item app-search">

<h3 class="page-title">Портал технической поддержки | Мои заявки</h3>

</li>

</ul>

<div class="clearfix"></div>

</nav>

</div>

<!-- Top Bar End -->

<div class="page-content-wrapper ">

<div class="container-fluid">

<div class="row">

<div class="col-12">

<div class="card m-b-30">

<div class="card-body">

<h4 class="mt-0 header-title">3аявки</h4>

<p class="text-muted m-b-30 font-14">

В этой таблице представлены все заявки, открытые вами.

</p>

<table id="datatable" class="table table-bordered">

<thead>

<tr>

<th>Идентификатор</th>

{% if drawing\_creator\_ticket == 1 %}

<th>Автор</th>

{% endif %}

<th>Создано</th>

<th>Обновлено</th>

<th>Заголовок</th>

{% if drawing\_creator\_ticket == 1 %}

<th>Оператор</th>

{% endif %}

<th>Статус</th>

<th>Действия</th>

</tr>

</thead>

<tbody>

{% for ticket in tickets %}

<tr>

<td>#{{ ticket[0] }}</td>

{% if drawing\_creator\_ticket == 1 %}

<td>{{ ticket[6] }}</td>

{% endif %}

<td>{{ ticket[1] }}</td>

<td>{{ ticket[2] }}</td>

<td>{{ ticket[3] }}</td>

{% if drawing\_creator\_ticket == 1 %}

{% if ticket[7] == '' %}

<td>Не назначен</td>

{% else %}

<td>{{ ticket[7] }}</td>

{% endif %}

{% endif %}

<td> <span class="badge badge-{{ ticket[5] }}">{{ ticket[4] }}</span></td>

<td class="text-center">

<button type="button" class="btn btn-dark waves-effect waves-light dropdown-toggle" data-toggle="dropdown" aria-expanded="false">

Action

</button>

<div class="dropdown-menu" style="">

<a class="btn btn-outline-info waves-effect waves-light dropdown-item view\_ticket" href="view\_ticket?ticket\_id={{ ticket[0] }}">View</a>

{% if ticket[4] == 'Закрыт' %}

{% else %}

<div class="dropdown-divider"></div>

<a class="btn btn-outline-danger waves-effect waves-light dropdown-item delete\_ticket" onclick="close\_ticket('{{ ticket[0] }}')">Close</a>

{% endif %}

</div>

</td>

</tr>

{% endfor %}

</tbody>

</table>

</div>

</div>

</div> <!-- end col -->

</div> <!-- end row -->

</div> <!-- Page content Wrapper -->

</div> <!-- content -->

<footer class="footer">

© 2023 ParinovYT <i class="mdi mdi-heart text-danger"></i>

</footer>

</div>

<!-- End Right content here -->

</div>

<!-- END wrapper -->

{% include 'modules/scripts.html' %}

<script>

function close\_ticket(id) {

$.post('/add\_comment', { ticket\_id: id, sender: 'system', text: 'Пользователь закрыл обращение' }, function(response) {

location.reload(); // Перезагрузка страницы после добавления комментария

});

}

</script>

</body>

</html>

Приложение 6

<!DOCTYPE html>

<html>

<head>

{% set page\_name = 'Admin Panel' %} {% include './modules/head.html' %}

</head>

<body class="fixed-left">

<!-- Loader -->

<div id="preloader">

<div id="status"><div class="spinner"></div></div>

</div>

<!-- Begin page -->

<div id="wrapper">

{% include "./modules/left.html" %}

<!-- Start right Content here -->

<div class="content-page">

<!-- Start content -->

<div class="content">

<!-- Top Bar Start -->

<div class="topbar">

<nav class="navbar-custom">

<ul class="list-inline float-right mb-0">

<li class="list-inline-item dropdown notification-list">

<a class="nav-link dropdown-toggle arrow-none waves-effect nav-user" data-toggle="dropdown" href="#" role="button" aria-haspopup="false" aria-expanded="false">

<img src="assets/images/users/avatar-5.jpg" alt="user" class="rounded-circle" />

</a>

<div class="dropdown-menu dropdown-menu-right profile-dropdown">

<a class="dropdown-item" href="logout"><i class="mdi mdi-logout m-r-5 text-muted"></i> Выход</a>

</div>

</li>

</ul>

<ul class="list-inline menu-left mb-0">

<li class="list-inline-item">

<button type="button" class="button-menu-mobile open-left waves-effect">

<i class="ion-navicon"></i>

</button>

</li>

<li class="hide-phone list-inline-item app-search">

<h3 class="page-title">Портал технической поддержки | Главная</h3>

</li>

</ul>

<div class="clearfix"></div>

</nav>

</div>

<!-- Top Bar End -->

<div class="page-content-wrapper">

<div class="row col-12">

<div class="col-md-6 col-xl-3">

<div class="mini-stat clearfix bg-white">

<span class="mini-stat-icon"><i class="fa fa-ticket"></i></span>

<div class="mini-stat-info text-right text-light">

<span class="counter text-white">{{tickets\_done}}</span>

Решеные/закрытые заявки

</div>

</div>

</div>

<div class="col-md-6 col-xl-3">

<div class="mini-stat clearfix bg-success">

<span class="mini-stat-icon">

<i class="fa fa-ticket"></i>

</span>

<div class="mini-stat-info text-right text-light">

<span class="counter text-white">{{tickets\_work}}</span>

Заявки в работе

</div>

</div>

</div>

<div class="col-md-6 col-xl-3">

<div class="mini-stat clearfix bg-info">

<span class="mini-stat-icon">

<i class="ti-user"></i>

</span>

<div class="mini-stat-info text-right text-light">

<span class="counter text-white">{{all\_useers}}</span>

Всего пользователей

</div>

</div>

</div>

</div>

</div>

<div class="row">

<div class="col-12">

<div class="card m-b-30">

<div class="card-body">

<h4 class="mt-0 header-title">открытые заявки</h4>

<table id="datatable" class="table table-bordered">

<thead>

<tr>

<th>Идентификатор</th>

<th>Автор</th>

<th>Создано</th>

<th>Обновлено</th>

<th>Заголовок</th>

<th>Статус</th>

<th>Действия</th>

</tr>

</thead>

<tbody>

{% for ticket in tickets %}

<tr>

{% if ticket[4] == 'Открыто' %}

<td>#{{ ticket[0] }}</td>

<td>{{ ticket[6] }}</td>

<td>{{ ticket[1] }}</td>

<td>{{ ticket[2] }}</td>

<td>{{ ticket[3] }}</td>

<td>

<span class="badge badge-{{ ticket[5] }}">{{ ticket[4] }}</span>

</td>

<td class="text-center">

<button type="button" class="btn btn-dark waves-effect waves-light dropdown-toggle" data-toggle="dropdown" aria-expanded="false">Action</button>

<div class="dropdown-menu" style="">

<a class="btn btn-outline-info waves-effect waves-light dropdown-item view\_ticket" href="view\_ticket?ticket\_id={{ ticket[0] }}">View</a>

<div class="dropdown-divider"></div> <a class="btn btn-outline-danger waves-effect waves-light dropdown-item delete\_ticket" onclick="close\_ticket('{{ ticket[0] }}')">Close</a>

</div>

</td>

{% endif %}

</tr>

{% endfor %}

</tbody>

</table>

</div>

</div>

</div>

<!-- end col -->

</div>

<!-- end row -->

</div>

<!-- content -->

<footer class="footer">

© 2023 ParinovYT

<i class="mdi mdi-heart text-danger"></i>

</footer>

</div>

<!-- End Right content here -->

</div>

<!-- END wrapper -->

{% include './modules/scripts.html' %}

<script>

function close\_ticket(id) {

$.post(

"/add\_comment",

{

ticket\_id: id,

sender: "system",

text: "Ваше обращение закрыто. Всего доброго!",

},

function (response) {

location.reload(); // Перезагрузка страницы после добавления комментария

}

);

}

</script>

</body>

</html>

Приложение 7

<!DOCTYPE html>

<html>

<head>

{% set page\_name = 'ViewTicket' %} {% include 'modules/head.html' %}

{% include 'modules/custom\_style\_viewticket.html' %}

</head>

<body class="fixed-left">

<!-- Loader -->

<div id="preloader">

<div id="status">

<div class="spinner"></div>

</div>

</div>

<!-- Begin page -->

<div id="wrapper">

{% include "./modules/left.html" %}

<!-- Start right Content here -->

<div class="content-page">

<!-- Start content -->

<div class="content">

<!-- Top Bar Start -->

<div class="topbar">

<nav class="navbar-custom">

<ul class="list-inline float-right mb-0">

<li class="list-inline-item dropdown notification-list">

<a class="nav-link dropdown-toggle arrow-none waves-effect nav-user" data-toggle="dropdown" href="#" role="button" aria-haspopup="false" aria-expanded="false">

<img src="assets/images/users/avatar-5.jpg" alt="user" class="rounded-circle">

</a>

<div class="dropdown-menu dropdown-menu-right profile-dropdown ">

<a class="dropdown-item" href="logout">

<i class="mdi mdi-logout m-r-5 text-muted"></i> Выход </a>

</div>

</li>

</ul>

<ul class="list-inline menu-left mb-0">

<li class="list-inline-item">

<button type="button" class="button-menu-mobile open-left waves-effect">

<i class="ion-navicon"></i>

</button>

</li>

<li class="hide-phone list-inline-item app-search">

<h3 class="page-title">Портал технической поддержки | Просмотр заявки</h3>

</li>

</ul>

<div class="clearfix"></div>

</nav>

</div>

<!-- Top Bar End -->

<div class="page-content-wrapper ">

<div class="container-fluid">

<div class="row">

<div class="col-12">

<div class="card m-b-30">

<div class="card-body">

{% if rule == 'Пользователь' %}

<h4 class="mt-0 header-title">Заявка #{{id}}</h4>

<p class="text-muted m-b-30 font-14">В этом поле вы можете взаимодействовать с вашей заявкой.</p>

{% else %}

<h4 class="mt-0 header-title">Заявка #{{id}}</h4>

<p class="text-muted m-b-30 font-14">В этом поле вы можете взаимодействовать с вашей заявкой.</p>

{% endif %}

{% for message in history\_chat %}

{% for comment in message['comments'] %}

{% if comment['sender'] == 'user' %}

<div class="user-message"><strong>Пользователь:</strong> {{ comment['text']|safe }}</div>

{% elif comment['sender'] == 'support' %}

<div class="support-message"><strong>Поддержка:</strong> {{ comment['text']|safe }}</div>

{% elif comment['sender'] == 'system' %}

<div class="system-message"><strong>Система:</strong> {{ comment['text']|safe }}</div>

{% endif %}

{% endfor %}

{% endfor %}

<div class="c1">

</div>

{% if status\_ticket == 3 %}

{% else %}

<textarea id="new-comment-text" ></textarea>

<div class="form-group row">

</div>

{% if rule == 'Пользователь' %}

<button class="btn btn-primary" onclick="addComment('user')">Добавить комментарий</button>

{% else %}

<div class="alert alert-danger alert-dismissible fade show" role="alert">

<strong style="text-align: right;">При ответе в чате заявки, вы становитесь оператором данной заявки</strong>

</div>

<button class="btn btn-primary" onclick="addComment('support')">Добавить комментарий</button>

<!-- <button class="btn btn-primary" onclick="close\_the\_request()">Закрыть обращение (решено)</button> -->

{% endif %}

{% endif %}

</div>

<!-- end col -->

</div>

<!-- end row -->

</div>

<!-- container -->

</div>

<!-- Page content Wrapper -->

</div>

<!-- content -->

<footer class="footer"> © 2023 ParinovYT <i class="mdi mdi-heart text-danger"></i>

</footer>

</div>

<!-- End Right content here -->

</div>

<!-- END wrapper -->

{% include 'modules/scripts.html' %}

<script>

$(document).ready(function() {

var editor1 = Jodit.make('#new-comment-text', {

height: 300

});

});

function addComment(sender) {

var editor = Jodit.instances['new-comment-text']; // получаем объект-редактор Jodit

var text = editor.value; // получаем текст из редактора Jodit

var text1 = '{{operator}}'

console.log(text1)

$.post('/add\_comment', { ticket\_id: {{id}}, sender: sender, text: text, operator: text1 }, function(response) {

location.reload(); // Перезагрузка страницы после добавления комментария

});

}

</script>

</body>

</html>

Приложение 8

<!DOCTYPE html>

<html>

<head>

{% set page\_name = 'Users' %} {% include './modules/head.html' %}

</head>

<body class="fixed-left">

<!-- Loader -->

<div id="preloader"><div id="status"><div class="spinner"></div></div></div>

<!-- Begin page -->

<div id="wrapper">

{% include "./modules/left.html" %}

<!-- Start right Content here -->

<div class="content-page">

<!-- Start content -->

<div class="content">

<!-- Top Bar Start -->

<div class="topbar">

<nav class="navbar-custom">

<ul class="list-inline float-right mb-0">

<li class="list-inline-item dropdown notification-list">

<a class="nav-link dropdown-toggle arrow-none waves-effect nav-user" data-toggle="dropdown" href="#" role="button"

aria-haspopup="false" aria-expanded="false">

<img src="assets/images/users/avatar-5.jpg" alt="user" class="rounded-circle">

</a>

<div class="dropdown-menu dropdown-menu-right profile-dropdown ">

<a class="dropdown-item" href="logout"><i class="mdi mdi-logout m-r-5 text-muted"></i> Выход</a>

</div>

</li>

</ul>

<ul class="list-inline menu-left mb-0">

<li class="list-inline-item">

<button type="button" class="button-menu-mobile open-left waves-effect">

<i class="ion-navicon"></i>

</button>

</li>

<li class="hide-phone list-inline-item app-search">

<h3 class="page-title">Портал технической поддержки | Пользователи</h3>

</li>

</ul>

<div class="clearfix"></div>

</nav>

</div>

<!-- Top Bar End -->

<div class="page-content-wrapper ">

<div class="container-fluid">

<div class="row">

<div class="col-12">

<div class="card m-b-30">

<div class="card-body">

<h4 class="mt-0 header-title">Пользователи</h4>

<p class="text-muted m-b-30 font-14">

В этой таблице представлены все пользователи.

</p>

<table id="datatable" class="table table-bordered">

<thead>

<tr>

<th>Идентификатор</th>

<th>Логин</th>

<th>Дата регистрации</th>

<th>Роль</th>

<th>Действия</th>

</tr>

</thead>

<tbody>

{% for user in users %}

<tr>

<td>#{{ user[0] }}</td>

<td>{{ user[1] }}</td>

<td>{{ user[2] }}</td>

<td>{{ user[3] }}</td>

<td class="text-center">

<button type="button" class="btn btn-dark waves-effect waves-light dropdown-toggle" data-toggle="dropdown" aria-expanded="false">

Action

</button>

<div class="dropdown-menu" style="">

<a class="btn btn-outline-info waves-effect waves-light dropdown-item" href="view\_profile?user\_id={{ user[0] }}">View</a>

</div>

</td>

</tr>

{% endfor %}

</tbody>

</table>

</div>

</div>

</div> <!-- end col -->

</div> <!-- end row -->

</div> <!-- Page content Wrapper -->

</div> <!-- content -->

<footer class="footer">

© 2023 ParinovYT <i class="mdi mdi-heart text-danger"></i>

</footer>

</div>

<!-- End Right content here -->

</div>

<!-- END wrapper -->

{% include './modules/scripts.html' %}

<script>

function close\_ticket(id) {

$.post('/add\_comment', { ticket\_id: id, sender: 'system', text: 'Пользователь закрыл обращение' }, function(response) {

location.reload(); // Перезагрузка страницы после добавления комментария

});

}

</script>

</body>

</html>

Приложение 9

<!DOCTYPE html>

<html>

<head>

{% set page\_name = 'inventory\_list' %} {% include './modules/head.html' %}

</head>

<body class="fixed-left">

<!-- Loader -->

<div id="preloader"><div id="status"><div class="spinner"></div></div></div>

<!-- Begin page -->

<div id="wrapper">

{% include "./modules/left.html" %}

<!-- Start right Content here -->

<div class="content-page">

<!-- Start content -->

<div class="content">

<!-- Top Bar Start -->

<div class="topbar">

<nav class="navbar-custom">

<ul class="list-inline float-right mb-0">

<li class="list-inline-item dropdown notification-list">

<a class="nav-link dropdown-toggle arrow-none waves-effect nav-user" data-toggle="dropdown" href="#" role="button"

aria-haspopup="false" aria-expanded="false">

<img src="assets/images/users/avatar-5.jpg" alt="user" class="rounded-circle">

</a>

<div class="dropdown-menu dropdown-menu-right profile-dropdown ">

<a class="dropdown-item" href="logout"><i class="mdi mdi-logout m-r-5 text-muted"></i> Выход</a>

</div>

</li>

</ul>

<ul class="list-inline menu-left mb-0">

<li class="list-inline-item">

<button type="button" class="button-menu-mobile open-left waves-effect">

<i class="ion-navicon"></i>

</button>

</li>

<li class="hide-phone list-inline-item app-search">

<h3 class="page-title">Портал технической поддержки | Список зарегистрированой аппаратуры</h3>

</li>

</ul>

<div class="clearfix"></div>

</nav>

</div>

<!-- Top Bar End -->

<div class="page-content-wrapper ">

<div class="container-fluid">

<div class="row">

<div class="col-12">

<div class="card m-b-30">

<div class="card-body">

<h4 class="mt-0 header-title">Devices</h4>

<p class="text-muted m-b-30 font-14">

-

</p>

<table id="datatable" class="table table-bordered">

<thead>

<tr>

<th>Инвентарный номер</th>

<th>Название</th>

<th>Год выпуска</th>

<th>Этаж</th>

<th>Кабинет</th>

<th>Тип</th>

<th>Ответственный</th>

<th>Действия</th>

</tr>

</thead>

<tbody>

{% for device in devices %}

<tr>

<td>{{ device[1] }}</td>

<td>{{ device[2] }}</td>

<td>{{ device[3] }}</td>

<td>{{ device[4] }}</td>

<td>{{ device[5] }}</td>

<td>{{ device[6] }}</td>

<td>{{ device[7] }}</td>

<td class="text-center">

<button type="button" class="btn btn-dark waves-effect waves-light dropdown-toggle" data-toggle="dropdown" aria-expanded="false">

Action

</button>

<div class="dropdown-menu" style="">

<a class="btn btn-outline-info waves-effect waves-light dropdown-item" href="inventory\_edit?inv\_number={{ device[1] }}">Edit</a>

<a class="btn btn-outline-danger waves-effect waves-light dropdown-item" href="inventory\_edit?delete=true&inv\_number={{ device[1] }}">Delete</a>

</div>

</td>

</tr>

{% endfor %}

</tbody>

</table>

</div>

</div>

</div> <!-- end col -->

</div> <!-- end row -->

</div> <!-- Page content Wrapper -->

</div> <!-- content -->

<footer class="footer">

© 2023 ParinovYT <i class="mdi mdi-heart text-danger"></i>

</footer>

</div>

<!-- End Right content here -->

</div>

<!-- END wrapper -->

{% include './modules/scripts.html' %}

</body>

</html>

Приложение 10

<!DOCTYPE html>

<html>

<head>

{% set page\_name = 'Add Ticket' %} {% include 'modules/head.html' %}

</head>

<body class="fixed-left">

<!-- Loader -->

<div id="preloader"><div id="status"><div class="spinner"></div></div></div>

<!-- Begin page -->

<div id="wrapper">

{% include "./modules/left.html" %}

<!-- Start right Content here -->

<div class="content-page">

<!-- Start content -->

<div class="content">

<!-- Top Bar Start -->

<div class="topbar">

<nav class="navbar-custom">

<ul class="list-inline float-right mb-0">

<li class="list-inline-item dropdown notification-list">

<a class="nav-link dropdown-toggle arrow-none waves-effect nav-user" data-toggle="dropdown" href="#" role="button"

aria-haspopup="false" aria-expanded="false">

<img src="assets/images/users/avatar-5.jpg" alt="user" class="rounded-circle">

</a>

<div class="dropdown-menu dropdown-menu-right profile-dropdown ">

<a class="dropdown-item" href="logout"><i class="mdi mdi-logout m-r-5 text-muted"></i> Выход</a>

</div>

</li>

</ul>

<ul class="list-inline menu-left mb-0">

<li class="list-inline-item">

<button type="button" class="button-menu-mobile open-left waves-effect">

<i class="ion-navicon"></i>

</button>

</li>

<li class="hide-phone list-inline-item app-search">

<h3 class="page-title">Портал технической поддержки | Изменить --</h3>

</li>

</ul>

<div class="clearfix"></div>

</nav>

</div>

<!-- Top Bar End -->

<div class="page-content-wrapper ">

<div class="container-fluid">

<div class="row">

<div class="col-12">

<div class="card m-b-30">

<div class="card-body">

<h4 class="mt-0 header-title">Ввод информации</h4>

<form class="form-horizontal m-t-20" method="POST" action="{{ url\_for('routes.inventoryedit') }}">

<div class="form-group row">

<label for="example-text-input" class="col-sm-2 col-form-label">Инвентарный Номер</label>

<div class="col-sm-10">

<input name="inventory\_number" class="form-control" type="text" value="{{inventory\_number}}" id="example-text-input">

</div>

</div>

<div class="form-group row">

<label for="example-text-input" class="col-sm-2 col-form-label">Название</label>

<div class="col-sm-10">

<input name="inventory\_name" class="form-control" type="text" value="{{inventory\_name}}" id="example-text-input">

</div>

</div>

<div class="form-group row">

<label for="example-datetime-local-input" class="col-sm-2 col-form-label">Дата приёма</label>

<div class="col-sm-10">

<input name="inventory\_date" class="form-control" type="datetime-local" value="{{inventory\_date}}" id="example-datetime-local-input">

</div>

</div>

<div class="form-group row">

<label for="example-number-input" class="col-sm-2 col-form-label">Этаж</label>

<div class="col-sm-10">

<input name="inventory\_floor" class="form-control" type="number" value="{{inventory\_floor}}" id="example-number-input">

</div>

</div>

<div class="form-group row">

<label for="example-number-input" class="col-sm-2 col-form-label">Кабинет</label>

<div class="col-sm-10">

<input name="inventory\_office" class="form-control" type="text" value="{{inventory\_office}}" id="example-number-input">

</div>

</div>

<div class="form-group row">

<label for="example-number-input" class="col-sm-2 col-form-label">Тип</label>

<div class="col-sm-10">

<input name="inventory\_type" class="form-control" type="text" value="{{inventory\_type}}" id="example-number-input">

</div>

</div>

<div class="form-group row">

<label for="example-number-input" class="col-sm-2 col-form-label">Ответственный</label>

<div class="col-sm-10">

<select name="inventory\_responsible" class="form-control">

<option>{{inventory\_responsible}}</option>

{% for user in users %}

{% if user != inventory\_responsible %}

<option>{{user}}</option>

{% endif %}

{% endfor %}

</select>

</div>

</div>

<!-- button -->

</div>

<div class="col-lg-3">

<button type="submit" class="btn btn-primary mb-3 me-auto">ОК</button>

</div>

</form>

</div>

</div> <!-- end col -->

</div> <!-- end row -->

</div><!-- container -->

</div>

</div> <!-- content -->

<footer class="footer">

© 2023 ParinovYT - ft by >.< Навалил. <i class="mdi mdi-heart text-danger"></i>

</footer>

</div>

<!-- End Right content here -->

</div>

<!-- END wrapper -->

{% include 'modules/scripts.html' %}

</body>

</html>

Приложение 11

<!DOCTYPE html>

<html>

<head>

{% set page\_name = 'Add Ticket' %} {% include 'modules/head.html' %}

</head>

<body class="fixed-left">

<!-- Loader -->

<div id="preloader"><div id="status"><div class="spinner"></div></div></div>

<!-- Begin page -->

<div id="wrapper">

{% include "./modules/left.html" %}

<!-- Start right Content here -->

<div class="content-page">

<!-- Start content -->

<div class="content">

<!-- Top Bar Start -->

<div class="topbar">

<nav class="navbar-custom">

<ul class="list-inline float-right mb-0">

<li class="list-inline-item dropdown notification-list">

<a class="nav-link dropdown-toggle arrow-none waves-effect nav-user" data-toggle="dropdown" href="#" role="button"

aria-haspopup="false" aria-expanded="false">

<img src="assets/images/users/avatar-5.jpg" alt="user" class="rounded-circle">

</a>

<div class="dropdown-menu dropdown-menu-right profile-dropdown ">

<a class="dropdown-item" href="logout"><i class="mdi mdi-logout m-r-5 text-muted"></i> Выход</a>

</div>

</li>

</ul>

<ul class="list-inline menu-left mb-0">

<li class="list-inline-item">

<button type="button" class="button-menu-mobile open-left waves-effect">

<i class="ion-navicon"></i>

</button>

</li>

<li class="hide-phone list-inline-item app-search">

<h3 class="page-title">Портал технической поддержки | Добавить в инвентарь</h3>

</li>

</ul>

<div class="clearfix"></div>

</nav>

</div>

<!-- Top Bar End -->

<div class="page-content-wrapper ">

<div class="container-fluid">

<div class="row">

<div class="col-12">

<div class="card m-b-30">

<div class="card-body">

<h4 class="mt-0 header-title">Ввод информации</h4>

<form class="form-horizontal m-t-20" method="POST" action="{{ url\_for('routes.inventoryadd') }}">

<div class="form-group row">

<label for="example-text-input" class="col-sm-2 col-form-label">Инвентарный Номер</label>

<div class="col-sm-10">

<input name="inventory\_number" class="form-control" type="text" value="" id="example-text-input">

</div>

</div>

<div class="form-group row">

<label for="example-text-input" class="col-sm-2 col-form-label">Название</label>

<div class="col-sm-10">

<input name="inventory\_name" class="form-control" type="text" value="" id="example-text-input">

</div>

</div>

<div class="form-group row">

<label for="example-datetime-local-input" class="col-sm-2 col-form-label">Дата приёма</label>

<div class="col-sm-10">

<input name="inventory\_date" class="form-control" type="datetime-local" value="" id="example-datetime-local-input">

</div>

</div>

<div class="form-group row">

<label for="example-number-input" class="col-sm-2 col-form-label">Этаж</label>

<div class="col-sm-10">

<input name="inventory\_floor" class="form-control" type="number" value="" id="example-number-input">

</div>

</div>

<div class="form-group row">

<label for="example-number-input" class="col-sm-2 col-form-label">Кабинет</label>

<div class="col-sm-10">

<input name="inventory\_office" class="form-control" type="text" value="" id="example-number-input">

</div>

</div>

<div class="form-group row">

<label for="example-number-input" class="col-sm-2 col-form-label">Тип</label>

<div class="col-sm-10">

<input name="inventory\_type" class="form-control" type="text" value="" id="example-number-input">

</div>

</div>

<div class="form-group row">

<label for="example-number-input" class="col-sm-2 col-form-label">Ответственный</label>

<div class="col-sm-10">

<select name="inventory\_responsible" class="form-control">

{% for responsible in devices %}

<option>{{responsible}}</option>

{% endfor %}

</select>

</div>

</div>

<!-- button -->

</div>

<div class="col-lg-3">

<button type="submit" class="btn btn-primary mb-3 me-auto">Добавить</button>

</div>

</form>

</div>

</div> <!-- end col -->

</div> <!-- end row -->

</div><!-- container -->

</div>

</div> <!-- content -->

<footer class="footer">

© 2023 ParinovYT <i class="mdi mdi-heart text-danger"></i>

</footer>

</div>

<!-- End Right content here -->

</div>

<!-- END wrapper -->

{% include 'modules/scripts.html' %}

</body>

</html>

Приложение 12

import calendar

from datetime import datetime

import json

import time

from imports import Blueprint, redirect, render\_template, request, User, Report, MySqlBase, TemplateIssue, session, getConfigurate, url\_for, is\_valid\_password, is\_valid\_username

routes = Blueprint('routes', \_\_name\_\_)

cfg = getConfigurate()

def get\_users():

db\_connection = MySqlBase().connection()

db\_connection.open()

try:

cursor = db\_connection.connection.cursor()

cursor.execute("SELECT username FROM users")

row = cursor.fetchall()

return row

finally:

db\_connection.close()

def get\_id\_by\_login():

db\_connection = MySqlBase().connection()

db\_connection.open()

try:

cursor = db\_connection.connection.cursor()

cursor.execute("SELECT id FROM users WHERE username=%s", (session.get('login'),))

row = cursor.fetchone()

if row:

rule\_id = int(row[0])

return rule\_id

else:

return -1

finally:

db\_connection.close()

def get\_login\_by\_id(id):

db\_connection = MySqlBase().connection()

db\_connection.open()

try:

cursor = db\_connection.connection.cursor()

cursor.execute("SELECT username FROM users WHERE id=%s", (id,))

row = cursor.fetchone()

if row:

rule\_id = str(row[0])

return rule\_id

else:

return ''

finally:

db\_connection.close()

def checkLogin():

result = False

if 'login' in session and 'time' in session and 'token' in session and 'type' in session:

print(f"login: {session.get('login')} | time: {session.get('time')} | token: {session.get('token')} | type: {session.get('type')}")

result = True

timestamp = calendar.timegm(time.gmtime())

if timestamp > session.get('time'):

print('pop time/login/token')

session.pop('time', None)

session.pop('login', None)

session.pop('token', None)

session.pop('type', None)

print('redirect to login page | EXPIRED sessions')

result = False

else:

result = False

print('redirect to login page | none sessions')

return result

def get\_rule\_by\_login(value: str):

from src.core.classes.mysql import MySqlBase

db\_connection = MySqlBase().connection()

db\_connection.open()

try:

cursor = db\_connection.connection.cursor()

cursor.execute("SELECT rule FROM users WHERE username=%s", (value,))

row = cursor.fetchone()

if row:

rule\_id = int(row[0])

return rule\_id

else:

return 0

finally:

db\_connection.close()

def get\_stats():

stats = []

db\_connection = MySqlBase().connection()

db\_connection.open()

try:

cursor = db\_connection.connection.cursor()

cursor.execute("SELECT COUNT(\*) FROM issues")

tickets\_all = cursor.fetchone()[0]

cursor.execute("SELECT COUNT(\*) FROM users")

users\_all = cursor.fetchone()[0]

cursor.execute("SELECT COUNT(\*) FROM issues WHERE status = 0")

tickets\_work = cursor.fetchone()[0]

tickets\_closed = tickets\_all - tickets\_work

stats = [tickets\_closed, tickets\_work, users\_all]

return stats

finally:

db\_connection.close()

def get\_all\_tickets():

issues = []

db\_connection = MySqlBase().connection()

db\_connection.open()

try:

cursor = db\_connection.connection.cursor()

cursor.execute('SELECT \* FROM issues')

row = cursor.fetchall()

finally:

db\_connection.close()

for rows in row:

issue\_status = 'В работе'

issue\_color = 'danger'

if rows[8] == 0:

issue\_color = 'info'

issue\_status = 'Открыто'

elif rows[8] == 1:

issue\_color = 'success'

issue\_status = 'Решено'

elif rows[8] == 2:

issue\_color = 'danger'

issue\_status = 'В работе'

elif rows[8] == 3:

issue\_color = 'warning'

issue\_status = 'Закрыт'

issue\_date = datetime.fromtimestamp(int(rows[5])).strftime('%Y-%m-%d %H:%M:%S')

issue\_date\_update = datetime.fromtimestamp(int(rows[6])).strftime('%Y-%m-%d %H:%M:%S')

issue = [str(rows[0]), issue\_date, issue\_date\_update, str(rows[2]), issue\_status, issue\_color, get\_login\_by\_id(rows[1])]

issues.append(issue)

return issues

@routes.route('/login', methods=['GET', 'POST'])

def login():

if checkLogin() == True:

return redirect(url\_for('.home\_general'))

\_status = 0

\_token = 0

\_rule\_selected = 0

\_rule\_selected\_index = 0

block\_login = 0

if request.method == 'POST':

block\_login = 0

user\_obj = User().sign\_in()

user\_obj.execute(request.form['username'], request.form['password'], int(cfg['login\_timeout']))

\_status = user\_obj.status\_code

\_token = user\_obj.get\_token

\_rule\_selected = request.form['rule']

if \_rule\_selected == 'Пользователь':

\_rule\_selected\_index = 4

if \_rule\_selected == 'Сотрудник':

\_rule\_selected\_index = 3

if \_rule\_selected == 'Администратор':

\_rule\_selected\_index = 2

if get\_rule\_by\_login(request.form['username']) != \_rule\_selected\_index:

if get\_rule\_by\_login(request.form['username']) != 2 or get\_rule\_by\_login(request.form['username']) != 3 and \_rule\_selected\_index != 4:

block\_login = 1

if \_status == 200:

if block\_login == 1:

return render\_template('login.html', status=999, token='', rule=\_rule\_selected)

session['login'] = request.form['username']

session['time'] = calendar.timegm(time.gmtime()) + int(cfg['login\_timeout'])

session['token'] = \_token

session['type'] = \_rule\_selected

return redirect(url\_for('.home\_general'))

#return render\_template('login.html', status=\_status, token=\_token)

return render\_template('login.html', status=\_status, token=\_token)

@routes.route('/register', methods=['GET', 'POST'])

def register():

username = None

password = None

user\_agreement\_checked = None

if request.method == 'POST':

username = request.form['username']

password = request.form['password']

user\_agreement\_checked = request.form['user\_agreement']

validate\_username = is\_valid\_username(username)

if validate\_username != '':

return render\_template('register.html', status=400, \_error=validate\_username)

validate\_pwd = is\_valid\_password(password)

if validate\_pwd != '':

return render\_template('register.html', status=400, \_error=validate\_pwd)

if username != None and password != None and user\_agreement\_checked == 'on':

user\_obj = User().sign\_up()

user\_obj.execute(username, password)

\_status = user\_obj.status\_code

if \_status == 200:

return render\_template('register.html', status=\_status)

elif \_status == 409:

return render\_template('register.html', status=\_status)

return render\_template('register.html')

@routes.route('/')

def home():

return redirect('/home')

@routes.route('/home', methods=['GET', 'POST'])

def home\_general():

if checkLogin() == False:

return redirect(url\_for('.login'))

rule = session.get('type')

if rule == 'Пользователь':

\_\_obj = TemplateIssue()

obj = \_\_obj.get()

obj.execute()

obj.status\_code

respone=obj.response

print(respone)

text = json.loads(respone)

print(text)

return render\_template('userarea.html', login=session.get('login'), rule=rule, template\_issueses=text)

elif rule == 'Администратор':

stats = get\_stats()

issues = get\_all\_tickets()

return render\_template('admin/adminarea.html', login=session.get('login'), rule=rule, tickets\_work=stats[1], tickets\_done=stats[0], all\_useers=stats[2], tickets=issues)

@routes.route('/logout', methods=['GET', 'POST'])

def logout():

if checkLogin() == False:

return redirect(url\_for('.login'))

user\_obj = User().sign\_out()

user\_obj.execute(session.get('token'))

session.pop('time', None)

session.pop('login', None)

session.pop('token', None)

session.pop('type', None)

return redirect(url\_for('.login'))

def get\_rule\_by\_value(value: str):

db\_connection = MySqlBase().connection()

db\_connection.open()

try:

cursor = db\_connection.connection.cursor()

# Используем параметризованный запрос

cursor.execute("SELECT id FROM rules WHERE name=%s", (value,))

row = cursor.fetchone()

# Проверяем, есть ли результат

if row:

rule\_id = int(row[0])

return rule\_id

else:

# Если не найдено, можно вернуть, например, None

return 1

finally:

db\_connection.close()

@routes.route('/users')

def users():

array = []

if checkLogin() == False:

return redirect(url\_for('.login'))

rule = session.get('type')

if rule != 'Пользователь':

db\_connection = MySqlBase().connection()

db\_connection.open()

try:

cursor = db\_connection.connection.cursor()

cursor.execute('SELECT \* FROM users')

row = cursor.fetchall()

finally:

db\_connection.close()

for rows in row:

registration\_date = datetime.fromtimestamp(int(rows[3])).strftime('%Y-%m-%d %H:%M:%S')

\_rule = None

\_\_rule = rows[4]

if \_\_rule == 4:

\_rule = 'Пользователь'

elif \_\_rule == 3:

\_rule = 'Сотрудник'

elif \_\_rule == 2:

\_rule = 'Администратор'

elif \_\_rule == 1:

\_rule = 'Заблокирован'

\_str = [str(rows[0]), str(rows[1]), registration\_date, \_rule]

array.append(\_str)

return render\_template('admin/users.html', login=session.get('login'), rule=rule, users=array)

@routes.route('/my\_tickets')

def my\_tickets():

issues = []

if checkLogin() == False:

return redirect(url\_for('.login'))

rule = session.get('type')

if rule != 'Пользователь':

db\_connection = MySqlBase().connection()

db\_connection.open()

try:

cursor = db\_connection.connection.cursor()

cursor.execute('SELECT \* FROM issues')

row = cursor.fetchall()

finally:

db\_connection.close()

for rows in row:

issue\_status = 'В работе'

issue\_color = 'danger'

if rows[8] == 0:

issue\_color = 'info'

issue\_status = 'Открыто'

elif rows[8] == 1:

issue\_color = 'success'

issue\_status = 'Решено'

elif rows[8] == 2:

issue\_color = 'danger'

issue\_status = 'Есть ответ'

elif rows[8] == 3:

issue\_color = 'warning'

issue\_status = 'Закрыт'

elif rows[8] == 4:

issue\_color = 'dark'

issue\_status = 'Ждем ответа'

issue\_date = datetime.fromtimestamp(int(rows[5])).strftime('%Y-%m-%d %H:%M:%S')

issue\_date\_update = datetime.fromtimestamp(int(rows[6])).strftime('%Y-%m-%d %H:%M:%S')

issue = [str(rows[0]), issue\_date, issue\_date\_update, str(rows[2]), issue\_status, issue\_color, get\_login\_by\_id(rows[1]), str(rows[9])]

issues.append(issue)

return render\_template('mytickets.html', login=session.get('login'), drawing\_creator\_ticket=True,rule=rule, tickets=issues)

if rule == 'Пользователь':

user\_id = get\_id\_by\_login()

db\_connection = MySqlBase().connection()

db\_connection.open()

try:

cursor = db\_connection.connection.cursor()

cursor.execute('SELECT \* FROM issues WHERE user\_id = %s', (user\_id,))

row = cursor.fetchall()

finally:

db\_connection.close()

for rows in row:

issue\_status = 'В работе'

issue\_color = 'danger'

if rows[8] == 0:

issue\_color = 'info'

issue\_status = 'Открыто'

elif rows[8] == 1:

issue\_color = 'success'

issue\_status = 'Решено'

elif rows[8] == 2:

issue\_color = 'danger'

issue\_status = 'В работе'

elif rows[8] == 3:

issue\_color = 'warning'

issue\_status = 'Закрыт'

elif rows[8] == 4:

issue\_color = 'dark'

issue\_status = 'Есть ответ'

issue\_date = datetime.fromtimestamp(int(rows[5])).strftime('%Y-%m-%d %H:%M:%S')

issue\_date\_update = datetime.fromtimestamp(int(rows[6])).strftime('%Y-%m-%d %H:%M:%S')

issue = [str(rows[0]), issue\_date, issue\_date\_update, str(rows[2]), issue\_status, issue\_color]

issues.append(issue)

return render\_template('mytickets.html', drawing\_creator\_ticket=False, login=session.get('login'), rule=rule, tickets=issues)

@routes.route('/add\_ticket', methods=['GET', 'POST'])

def addticket():

if checkLogin() == False:

return redirect(url\_for('.login'))

rule = session.get('type')

devices\_ = []

db\_connection = MySqlBase().connection()

db\_connection.open()

if request.method == 'GET':

try:

cursor = db\_connection.connection.cursor()

cursor.execute('SELECT \* FROM devices')

devices\_\_ = cursor.fetchall()

for devices\_\_\_ in devices\_\_:

if devices\_\_\_[1] == 'null':

continue

devices\_.append(devices\_\_\_)

finally:

db\_connection.close()

if rule == 'Пользователь':

if request.method == 'POST':

\_create = request.form['create']

login = session.get('login')

time\_created = calendar.timegm(time.gmtime())

desc = request.form['desc\_ticket']

text = request.form['text\_ticket']

floor = request.form['floor']

office = request.form['office']

info\_device = request.form['device\_name']

info\_device\_id = 'null'

print(f'once add info {\_create}')

if \_create == "true":

if floor:

floor\_text = str(floor)

else:

floor\_text = 'Не указан'

if office:

office\_text = str(office)

else:

office\_text = 'Не указан'

if info\_device == 'none':

info\_device\_text = 'нет'

else:

info\_device\_text = f'<br>{info\_device}'

devID = info\_device.split("-")[-1].strip()

devID = devID.split("|")[0].replace(" ", "")

info\_device\_id = devID

comment\_info = {'sender': 'system', 'text': f'<br>Дополнительная информация о заявке:<br>Этаж: {floor\_text}<br>Кабинет: {office\_text}<br>Проблемное устройство: {info\_device\_text}'}

if \_create == "true":

once\_comment = [comment\_info, {"sender": "user", "text": text}]

else:

once\_comment = [{"sender": "user", "text": text}]

text = json.dumps(once\_comment)

user\_obj = Report().create()

user\_obj.execute(session.get('token'), desc, text, info\_device\_id)

print(f'Попытка создания заявки:\nLogin:{login}\nTime:{time\_created}\nDescription:{desc}\nText:{text}\nStatus: {user\_obj.status\_code}')

return render\_template('addticket.html', login=session.get('login'), rule=rule, devices=devices\_)

@routes.route('/add\_comment', methods=['POST'])

def add\_comment():

comments\_data = []

ticket\_id = request.form['ticket\_id'] # Получаем ticket\_id

sender = request.form['sender'] # Получаем отправителя

text = request.form['text'] # Получаем текст комментария

worker = request.form['operator'] # Получаем оператора

print(f"!!!worker!!! -- {worker}")

if worker == 'Не назначен':

operator = f"Вам назначен оператор: {session.get('login')}"

new\_comment\_worker = {'sender': 'system', 'text': operator}

db\_connection = MySqlBase().connection()

db\_connection.open()

try:

cursor = db\_connection.connection.cursor()

cursor.execute('SELECT comments FROM issues WHERE id = %s', (ticket\_id,))

row = cursor.fetchone()

comments\_data = json.loads(row[0]) if row else []

if worker == 'Не назначен':

comments\_data.append(new\_comment\_worker)

cursor.execute("UPDATE issues SET worker = %s WHERE id = %s", (session.get('login'), ticket\_id))

db\_connection.connection.commit()

new\_comment = {'sender': sender, 'text': text}

comments\_data.append(new\_comment)

cursor.execute("UPDATE issues SET comments = %s, updated\_at = %s WHERE id = %s", (json.dumps(comments\_data), calendar.timegm(time.gmtime()), ticket\_id))

if sender == 'system':

cursor.execute("UPDATE issues SET status = %s WHERE id = %s", ('3', ticket\_id))

db\_connection.connection.commit()

if sender == 'user':

cursor.execute("UPDATE issues SET status = %s WHERE id = %s", ('2', ticket\_id))

db\_connection.connection.commit()

if sender == 'support':

cursor.execute("UPDATE issues SET status = %s WHERE id = %s", ('4', ticket\_id))

db\_connection.connection.commit()

finally:

cursor.close()

db\_connection.close()

return f'200'

@routes.route('/view\_ticket', methods=['GET', 'POST'])

def viewticket():

history\_chat\_ = []

if checkLogin() == False:

return redirect(url\_for('.login'))

rule = session.get('type')

if rule == 'Администратор':

if request.method == 'GET':

ticket\_id = request.args.get('ticket\_id')

db\_connection = MySqlBase().connection()

db\_connection.open()

try:

cursor = db\_connection.connection.cursor()

cursor.execute('SELECT \* FROM issues WHERE id = %s', (ticket\_id,))

row = cursor.fetchall()

finally:

db\_connection.close()

for rows in row:

ticket\_id = rows[0]

comments = json.loads(rows[3])

history\_chat\_.append({'comments': comments})

\_status\_ticket = rows[8]

\_operator = rows[9]

if \_operator == '':

\_operator = 'Не назначен'

return render\_template('viewticket.html', login=session.get('login'), rule=rule, id=ticket\_id, history\_chat=history\_chat\_, status\_ticket=\_status\_ticket, operator=\_operator)

if rule == 'Пользователь':

if request.method == 'GET':

ticket\_id = request.args.get('ticket\_id')

user\_id = get\_id\_by\_login()

db\_connection = MySqlBase().connection()

db\_connection.open()

try:

cursor = db\_connection.connection.cursor()

cursor.execute('SELECT \* FROM issues WHERE user\_id = %s AND id = %s', (user\_id, ticket\_id,))

row = cursor.fetchall()

finally:

db\_connection.close()

for rows in row:

ticket\_id = rows[0]

comments = json.loads(rows[3])

history\_chat\_.append({'comments': comments})

\_status\_ticket = rows[8]

\_operator = 'no\_operator'

return render\_template('viewticket.html', login=session.get('login'), rule=rule, id=ticket\_id, history\_chat=history\_chat\_, status\_ticket=\_status\_ticket, operator=\_operator)

@routes.route('/view\_profile', methods=['GET', 'POST'])

def viewprofile():

return render\_template('admin/view\_profile.html', login=session.get('login'), rule=session.get('type'))

@routes.route('/inventory\_list', methods=['GET', 'POST'])

def inventorylist():

\_devices = []

db\_connection = MySqlBase().connection()

db\_connection.open()

try:

cursor = db\_connection.connection.cursor()

cursor.execute('SELECT \* FROM devices')

row = cursor.fetchall()

finally:

db\_connection.close()

for rows in row:

if rows[1] == 'null':

continue

\_devices.append(rows)

return render\_template('admin/materials.html', login=session.get('login'), rule=session.get('type'), devices=\_devices)

@routes.route('/inventory\_add', methods=['GET', 'POST'])

def inventoryadd():

if request.method == 'POST':

inventory\_number = request.form['inventory\_number']

inventory\_name = request.form['inventory\_name']

inventory\_date = request.form['inventory\_date']

inventory\_floor = request.form['inventory\_floor']

inventory\_office = request.form['inventory\_office']

inventory\_type = request.form['inventory\_type']

inventory\_responsible = request.form['inventory\_responsible']

db\_connection = MySqlBase().connection()

db\_connection.open()

try:

cursor = db\_connection.connection.cursor()

insert\_query = "INSERT INTO devices VALUES (NULL, %s, %s, %s, %s, %s, %s, %s);"

params = (

inventory\_number,

inventory\_name,

inventory\_date,

inventory\_floor,

inventory\_office,

inventory\_type,

inventory\_responsible

)

cursor.execute(insert\_query, params)

db\_connection.connection.commit()

finally:

db\_connection.close()

data = []

row=get\_users()

for rows in row:

data.append(rows[0])

return render\_template('admin/material\_add.html', users=data, login=session.get('login'), rule=session.get('type'))

@routes.route('/inventory\_edit', methods=['GET', 'POST'])

def inventoryedit():

if request.method == 'GET':

has\_only\_del = request.args.get('delete')

inv\_number = request.args.get('inv\_number')

if has\_only\_del == 'true':

session['inv\_number'] = inv\_number

db\_connection = MySqlBase().connection()

db\_connection.open()

try:

cursor = db\_connection.connection.cursor()

cursor.execute('DELETE FROM devices WHERE inventory\_number = %s', (session.get('inv\_number'),))

db\_connection.connection.commit()

finally:

db\_connection.close()

return redirect(url\_for('.inventorylist'))

db\_connection = MySqlBase().connection()

db\_connection.open()

try:

cursor = db\_connection.connection.cursor()

cursor.execute('SELECT \* FROM devices WHERE inventory\_number = %s', (inv\_number,))

row\_data = cursor.fetchall()

finally:

db\_connection.close()

for data in row\_data:

inventory\_number = data[1]

inventory\_name = data[2]

inventory\_date = data[3]

inventory\_floor = data[4]

inventory\_office = data[5]

inventory\_type = data[6]

inventory\_responsible = data[7]

if request.method == 'POST':

inventory\_number = request.form['inventory\_number']

inventory\_name = request.form['inventory\_name']

inventory\_date = request.form['inventory\_date']

inventory\_floor = request.form['inventory\_floor']

inventory\_office = request.form['inventory\_office']

inventory\_type = request.form['inventory\_type']

inventory\_responsible = request.form['inventory\_responsible']

db\_connection = MySqlBase().connection()

db\_connection.open()

try:

cursor = db\_connection.connection.cursor()

cursor.execute('DELETE FROM devices WHERE inventory\_number = %s', (session.get('inv\_number'),))

db\_connection.connection.commit()

insert\_query = "INSERT INTO devices VALUES (NULL, %s, %s, %s, %s, %s, %s, %s);"

params = (

inventory\_number,

inventory\_name,

inventory\_date,

inventory\_floor,

inventory\_office,

inventory\_type,

inventory\_responsible

)

cursor.execute(insert\_query, params)

db\_connection.connection.commit()

finally:

db\_connection.close()

data = []

row=get\_users()

for rows in row:

data.append(rows[0])

return render\_template('admin/material\_edit.html', users=data, login=session.get('login'), rule=session.get('type'), inventory\_number=inventory\_number,

inventory\_name=inventory\_name, inventory\_date=inventory\_date,inventory\_floor=inventory\_floor,inventory\_office=inventory\_office,

inventory\_type=inventory\_type, inventory\_responsible=inventory\_responsible)

Приложение 13

<https://github.com/ParinovYT/TechnicalSupportCenterProject> - GitHub-репозиторий с проектом