

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Образовательная программа «Программная инженерия»


**СОГЛАСОВАНО**

Научный руководитель, приглашённый  
преподаватель, доцент департамента  
программной инженерии, канд. техн. наук

  
\_\_\_\_\_ А.Д. Брейман  
« 10 » \_\_\_\_\_ мая \_\_\_\_\_ 2023 г.

**УТВЕРЖДЕНО**

Академический руководитель  
образовательной программы  
«Программная инженерия», профессор  
департамента программной инженерии,  
канд. техн. наук

  
\_\_\_\_\_ В.В. Шилов.  
« 10 » \_\_\_\_\_ мая \_\_\_\_\_ 2023 г.

**ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ РЕЛЯЦИОННОЙ БАЗОЙ ДАННЫХ  
В УДОБНОЙ ДЛЯ ПОЛЬЗОВАТЕЛЯ ФОРМЕ НА ПРИМЕРЕ POSTGRESQL**


**Пояснительная записка**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.02.07-01 81 01-1-ЛУ**

Исполнитель

студент группы БПИ213

  
\_\_\_\_\_ А.С. Абрамов  
« 7 » \_\_\_\_\_ мая \_\_\_\_\_ 2023 г.

Подп. и дата	
Инв. № дубл	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

УТВЕРЖДЕН

RU.17701729.02.07-01 81 01-1-ЛУ

**ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ РЕЛЯЦИОННОЙ БАЗОЙ ДАННЫХ  
В УДОБНОЙ ДЛЯ ПОЛЬЗОВАТЕЛЯ ФОРМЕ НА ПРИМЕРЕ POSTGRESQL**

**Пояснительная записка**

**RU.17701729.02.07-01 81 01-1**

**Листов: 103**

Инв. № подл	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата

## АННОТАЦИЯ

Пояснительная записка – документ, содержащий информацию о структуре и алгоритмах работы программного обеспечения, а также обосновывающий принятые при разработке технические и технико-экономические решения.

Настоящая пояснительная записка к техническому заданию на разработку «Веб-приложения для управления реляционной базой данных в удобной для пользователя форме на примере PostgreSQL» содержит следующие разделы: «Введение», «Назначение и область применения», «Технические характеристики», «Ожидаемые технико-экономические показатели», «Источники, использованные при разработке» и приложения.

В разделе «Введение» указаны наименование программы и документы, на основании которых ведется разработка.

В разделе «Назначение и область применения» указаны функциональное и эксплуатационное назначение программы, а также краткая характеристика её области применения.

Раздел «Технические характеристики» содержит постановку задачи на разработку программного продукта, описывает и обосновывает ограничения и допущения решения поставленной задачи, а также структуру, функционирование и алгоритмы работы программы, методы организации входных и выходных данных и состав технических и программных средств.

В разделе «Ожидаемые технико-экономические показатели» указаны технико-экономические показатели, обосновывающие преимущества выбранного технического решения по сравнению с аналогами, а также описаны ориентировочная экономическая эффективность и предполагаемая потребность продукта.

В разделе «Источники, использованные при разработке» указан перечень публикаций, документов и других материалов, использованных при решении поставленной задачи.

Настоящий документ разработан в соответствии с требованиями:

- 1) ГОСТ 19.101-77<sup>[1]</sup>: Виды программ и программных документов.
- 2) ГОСТ 19.102-77<sup>[2]</sup>: Стадии разработки.
- 3) ГОСТ 19.103-77<sup>[3]</sup>: Обозначения программ и программных документов.
- 4) ГОСТ 19.104-78<sup>[4]</sup>: Основные надписи.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- 5) ГОСТ 19.105-78<sup>[5]</sup>: Общие требования к программным документам.
- 6) ГОСТ 19.106-78<sup>[6]</sup>: Требования к программным документам, выполненным печатным способом.
- 7) ГОСТ 19.404-78<sup>[7]</sup>: Пояснительная записка. Требования к содержанию и оформлению.
- 8) ГОСТ 19.602-78<sup>[8]</sup>: Правила дублирования, учета и хранения программных документов, выполненных печатным способом.

Изменения к настоящему документу должны быть оформлены согласно ГОСТ 19.603-78<sup>[9]</sup> и ГОСТ 19.604-78<sup>[10]</sup>.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## СОДЕРЖАНИЕ

<b>1. ВВЕДЕНИЕ.....</b>	<b>7</b>
1.1. Наименование программы.....	7
1.2. Документ(ы), на основании которого(ых) ведётся разработка.....	7
<b>2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ.....</b>	<b>8</b>
2.1. Функциональное назначение.....	8
2.2. Эксплуатационное назначение.....	8
2.3. Краткая характеристика области применения.....	9
<b>3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ.....</b>	<b>10</b>
3.1. Постановка задачи на разработку программы.....	10
3.2. Описание и обоснование ограничений и допущений решения поставленной задачи.....	10
3.3. Описание и обоснование архитектуры программы.....	11
3.3.1. Описание архитектуры программы.....	11
3.3.2. Обоснование архитектуры программы.....	12
3.4. Описание и обоснование функционирования и алгоритмов работы программы.....	13
3.4.1. Описание и обоснование алгоритма аутентификации запросов.....	13
3.4.1.1. Описание алгоритма аутентификации запросов.....	13
3.4.1.2. Обоснование алгоритма аутентификации запросов.....	14
3.4.2. Описание и обоснование алгоритма авторизации пользователей.....	14
3.4.2.1. Описание алгоритма авторизации пользователей.....	14
3.4.2.2. Обоснование алгоритма авторизации пользователей.....	16
3.4.3. Описание и обоснование функционирования основных элементов интерфейса.....	16
3.4.3.1. Описание функционирования основных элементов интерфейса.....	16
3.4.3.2. Обоснование функционирования основных элементов интерфейса.....	19
3.4.4. Описание и обоснование функционирования страницы “База данных”.....	20
3.4.4.1. Описание функционирования страницы “База данных”.....	20

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.4.4.2. Обоснование функционирования страницы “База данных” .....	22
3.4.5. Описание и обоснование функционирования страниц панели управления, соответствующих таблицам целевой базы данных.....	23
3.4.5.1. Описание функционирования страниц панели управления, соответствующих таблицам целевой базы данных.....	23
3.4.5.2. Обоснование функционирования страниц панели управления, соответствующих таблицам целевой базы данных.....	28
3.4.6. Описание и обоснование функционирования страницы “Аккаунты” .....	29
3.4.6.1. Описание функционирования страницы “Аккаунты” .....	29
3.4.6.2. Обоснование функционирования страницы “Аккаунты” .....	32
3.4.7. Описание и обоснование функционирования страницы “История” .....	33
3.4.7.1. Описание функционирования страницы “История” .....	33
3.4.7.2. Обоснование функционирования страницы “История” .....	34
3.5. Описание и обоснование методов организации входных и выходных данных.....	35
3.5.1. Описание и обоснование метода организации входных данных.....	35
3.5.1.1. Описание метода организации входных данных.....	35
3.5.1.2. Обоснование метода организации входных данных.....	39
3.5.2. Описание и обоснование метода организации выходных данных.....	39
3.5.2.1. Описание метода организации выходных данных.....	39
3.5.2.2. Обоснование метода организации выходных данных.....	42
3.6. Описание и обоснование выбора состава технических и программных средств.....	43
3.6.1. Описание и обоснование требований к серверному оборудованию.....	43
3.6.2. Описание и обоснование требований к клиентскому оборудованию.....	43
3.6.3. Описание и обоснование выбора состава программных средств.....	43
3.6.3.1. Описание состава программных средств.....	43
3.6.3.2. Обоснование выбора состава программных средств.....	44
4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ.....	45

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4.1. Ориентировочная экономическая эффективность.....	45
4.2. Предполагаемая потребность.....	45
4.3. Экономические преимущества разработки по сравнению с лучшими отечественными и зарубежными образцами или аналогами.....	45
5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ.....	49
ПРИЛОЖЕНИЕ 1. ОПИСАНИЕ МЕТОДОВ ПОЛУЧЕНИЯ ИНФОРМАЦИИ О ЦЕЛЕВОЙ БАЗЕ ДАННЫХ.....	54
ПРИЛОЖЕНИЕ 2. ОПИСАНИЕ И НАЗНАЧЕНИЕ ФАЙЛОВ ПРОГРАММЫ.....	59
ПРИЛОЖЕНИЕ 3. ОПИСАНИЕ ВНУТРЕННЕЙ БАЗЫ ДАННЫХ ПРИЛОЖЕНИЯ.....	65
ПРИЛОЖЕНИЕ 4. ОПИСАНИЕ И НАЗНАЧЕНИЕ ВНЕШНИХ ЗАВИСИМОСТЕЙ.....	67
ПРИЛОЖЕНИЕ 5. ОПИСАНИЕ И НАЗНАЧЕНИЕ ПАРАМЕТРОВ ВНЕШНИХ ЗАВИСИМОСТЕЙ.....	69
ПРИЛОЖЕНИЕ 6. ОПИСАНИЕ И НАЗНАЧЕНИЕ ФУНКЦИЙ.....	76
ПРИЛОЖЕНИЕ 7. ОПИСАНИЕ И НАЗНАЧЕНИЕ КЛАССОВ.....	82
ПРИЛОЖЕНИЕ 8. ОПИСАНИЕ И НАЗНАЧЕНИЕ ПОЛЕЙ И МЕТОДОВ КЛАССОВ.....	84
ПРИЛОЖЕНИЕ 9. ОПИСАНИЕ И НАЗНАЧЕНИЕ ЭКСПОРТИРУЕМЫХ ПЕРЕМЕННЫХ....	96
ПРИЛОЖЕНИЕ 10. ОПИСАНИЕ И НАЗНАЧЕНИЕ CSS-СТИЛЕЙ.....	97
ТЕРМИНЫ И СОКРАЩЕНИЯ.....	102
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ.....	103

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## 1. ВВЕДЕНИЕ

### 1.1. Наименование программы

Наименование программы: «Веб-приложение для управления реляционной базой данных в удобной для пользователя форме на примере PostgreSQL».

Наименование программы на английском языке: «Web app for relational database management in a user-friendly way using PostgreSQL».

Краткое наименование программы: «DBViewer».

### 1.2. Документ(ы), на основании которого(ых) ведётся разработка

Разработка ведется на основании учебного плана подготовки бакалавров по направлению 09.03.04 “Программная инженерия” Факультета компьютерных наук Национального исследовательского университета “Высшая школа экономики” и утвержденной академическим руководителем программы темы курсового проекта.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



## 2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

### 2.1. Функциональное назначение

«DBViewer» реализует средство управления реляционной базой данных, позволяющее просматривать и редактировать данные с помощью браузера на любом устройстве.

Основная функциональная часть приложения - страницы, содержащие меню навигации по целевой БД и отображающие данные, связанные с ней: строки таблиц, информацию о нагрузке на сервер, занимаемый на диске объём памяти и др. При этом важно, что все отображаемые данные либо понятны неподготовленному пользователю, либо снабжены достаточным количеством подсказок, позволяющих любому человеку без труда разобраться в интерфейсе приложения. Пользователи имеют возможность просматривать данные и изменять некоторые из них, в частности - строки таблиц: «DBViewer» реализует возможность вставки новых записей, а также редактирования и удаления уже существующих. Все действия пользователей передаются серверу базы данных путём генерации соответствующих SQL-запросов. Для этого приложение устанавливает соединение с базой данных, используя реквизиты, указываемые в файле конфигурации при запуске приложения.

Для обеспечения безопасности данных приложение дополнительно реализует простейшую авторизацию пользователей:

- 1) При первичном запуске приложения создаётся один аккаунт, задаваемый в файле конфигурации и обладающий полным доступом к работе с приложением.
- 2) Для управления базой данных пользователь должен авторизоваться с помощью одного из существующих аккаунтов.
- 3) Администраторы имеют возможность создания дополнительных аккаунтов, редактирования уже существующих, а также настройки их уровней доступа с помощью соответствующих меню специальной страницы.
- 4) Администраторы на отдельной странице имеют возможность просмотра списка запросов, сделанных всеми пользователями.

### 2.2. Эксплуатационное назначение

Для эффективного сопровождения программного продукта нередко возникает необходимость просмотра и ручного редактирования содержимого базы данных (например, для быстрой и

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

качественной работы поддержки пользователей). Тем не менее приложений, позволяющих делать это пользователю, не знакомому с принципами проектирования и работы реляционных БД, нет, что вынуждает заказчиков оплачивать услуги специалистов в этой области на протяжении всего жизненного цикла приложения, что может оказаться невозможным для разработок с небольшим бюджетом.

«DBViewer» решает эту проблему, реализуя программу, которая позволяет производить базовые операции с БД любому человеку с минимальной подготовкой после единоразовой настройки специалистом. Таким образом, приложение позволяет безопасно для работоспособности всей системы просматривать и редактировать содержимое базы данных без необходимости оплаты услуг специалистов, что даёт возможность уменьшить затраты во время эксплуатации программного продукта и способствует увеличению количества небольших проектов на рынке ПО.

Таким образом, целевой аудиторией проекта являются команды поддержки программных продуктов, не обладающие достаточным бюджетом для постоянной оплаты услуг квалифицированных в области баз данных специалистов, но готовые приобрести приложение, решающее их задачи, единоразовым платежом.

### 2.3. Краткая характеристика области применения

«DBViewer» - веб-приложение, позволяющее пользователю производить простейшие операции с реляционной базой данных (в частности, PostgreSQL) без особенных знаний её устройства и языка запросов, требующее для работы лишь единоразовую настройку. Основной целью разработки программы является упрощение поддержки программных продуктов - избавление от необходимости постоянного содержания команды специалистов в области баз данных для эффективного решения возникающих проблем и помощи пользователям.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

### 3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

#### 3.1. Постановка задачи на разработку программы

В рамках проекта требуется разработать веб-приложение, позволяющее безопасно производить основные операции с базой данных PostgreSQL в удобной для пользователя форме. Основными требованиями являются:

- 1) Авторизация пользователей с возможностью распределения уровней доступа администраторами: чтение, вставка, редактирование, удаление, администрирование.
- 2) Отображение содержимого таблиц базы данных с помощью браузера с возможностью фильтрации и сортировки записей.
- 3) Возможность редактирования содержимого таблиц базы данных с помощью браузера.
- 4) Возможность удаления записей из таблиц базы данных с помощью браузера.
- 5) Возможность создания новых записей в существующих таблицах базы данных с помощью браузера.
- 6) Отображение статистики нагрузки на отдельные таблицы базы данных и базу данных в целом, а также другой информации, которая может быть полезна для диагностирования и исправления ошибок базы данных.
- 7) Сохранение информации о произведённых всеми пользователями операциях с возможностью её удобного просмотра администраторами.

#### 3.2. Описание и обоснование ограничений и допущений решения поставленной задачи

Решение поставленной задачи в виде веб-приложения в соответствии с описанными в техническом задании требованиями накладывает ряд ограничений на применяемые при реализации алгоритмы, а также на структуру баз данных, с которыми программа может корректно работать.

- 1) Приложение отображает данные только из обычных таблиц и не производит обработку представлений и материализованных представлений, что обусловлено значительными отличиями в устройстве этих объектов: большая часть статистики, доступной для обычных таблиц, не имеет смысла для представлений, а редактирование записей представлений при правильном проектировании почти всегда и вовсе невозможно.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- 2) Приложение может работать некорректно, если изменяемая таблица целевой базы данных содержит триггеры, правила и другие настройки, изменяющие поведение сервера при создании, редактировании и удалении значений. В таких случаях пользователь может неожиданно получать сообщения об ошибках базы данных, связанных с работой упомянутых триггеров, правил и других настроек.
- 3) Работа приложения с таблицами, содержащими столбцы неподдерживаемых типов данных, может быть нарушена: создание, редактирование и удаление значений могут быть недоступны из-за некорректности или невозможности представления данных в текстовом виде, который используется для отображения значений всех неизвестных типов данных. Приложение гарантирует корректную работу лишь со значениями следующих типов: bool, int2 (smallint), int4 (int), int8 (bigint), float4 (real), float8 (double), numeric, char, varchar, text, date, time, timetz, timestamp и timestampz
- 4) Для повышения эффективности своей работы приложение допускает ввод и попытку сохранения некорректных значений (нарушающих требование уникальности значений, внешний ключ или др.). При этом соответствующий запрос завершается полученной от сервера базы данных ошибкой, которая сообщается пользователю.
- 5) Сохранение изменений может работать некорректно, если запись базы данных была изменена “извне” в промежутке времени между ее загрузкой пользователем и сохранением изменением, так как это может нарушить идентификацию записи.

### 3.3. Описание и обоснование архитектуры программы

#### 3.3.1. Описание архитектуры программы

Приложение реализовано в виде трёх взаимодействующих частей: веб-сервера, реализующего обработку запросов пользователей, внутренней базы данных PostgreSQL, организующей хранение необходимой приложению информации, а также хранилища данных Redis, позволяющего удобно поддерживать часто используемые данные в оперативной памяти с целью уменьшения количества обращений к серверу PostgreSQL. Для корректной работы программы также необходимо подключение четвёртой части - внешней, целевой базы данных, информацию о которой должно отображать приложение.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Основная часть (веб-сервер) реализована с использованием присущего языку JavaScript функционального подхода с помощью высоко настраиваемого фреймворка fastify, который позволяет эффективно дополнять объекты сервера, запроса и ответа собственными полями и методами, а также организывает цепочку ответственности при обработке запросов. Более того, фреймворк позволяет декомпозировать работу на несколько независимых модулей, которые могут быть полностью инкапсулированы. Клиентская часть приложения реализована с учётом требований общедоступности приложения WCAG 2.0<sup>[\[11\]](#)</sup>.

Общий процесс взаимодействия с пользователем следующий:

- 1) Первичная обработка, аутентификация и авторизация запроса. При этом происходит взаимодействие с хранилищами данных Redis и PostgreSQL для получения актуальной информации об аккаунте пользователя, если это необходимо.
- 2) Обработка запроса пользователя: получение необходимой информации у целевой и внутренней баз данных, а также изменение их состояния или состояния веб-сервера.
- 3) Генерация исходного кода необходимой веб-страницы - ответа на запрос - и его отправка пользователю.
- 4) В случае успешного ответа клиент инициирует установление двустороннего соединения с сервером для получения дополнительной информации посредством эффективной отправки сообщений в формате JSON.
- 5) При получении сообщения сервер производит обращение к одной или нескольким базам данных, генерирует ответ в формате JSON и отправляет его клиенту в виде ответного сообщения.
- 6) Клиент считывает ответ и вносит соответствующие изменения в содержимое веб-страницы.
- 7) По окончании работы пользователь закрывает веб-страницу, что автоматически прерывает установленное соединение.

Подробные описания всех использованных при разработке внешних зависимостей и их параметров, структуры программы и составляющих её объектов, а также протоколов взаимодействия частей приложения представлены в приложениях 1 - 10.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

### 3.3.2. Обоснование архитектуры программы

Описанная архитектура способствует понятной организации исходного кода и разделению ответственности за различные этапы обработки запросов пользователей между различными частями программы. Использование фреймворка fastify позволяет применять модульную организацию веб-сервера, что обеспечивает возможность легкого изменения этапов обработки запросов пользователей и расширения приложения с добавлением новых методов. Поддержание всех шаблонов исходных кодов веб-страниц и их компонентов “в одном месте” позволяет без труда производить расширение и улучшение клиентской составляющей приложения и уменьшает вероятность нежелательных утечек информации с сервера из-за ошибки разработчика, а следование требованиям WCAG при реализации интерфейса обеспечивает высокое качество веб-страниц и их корректную и эффективную работу на устройствах большинства пользователей.

Помимо прочего, архитектура приложения содействует его масштабируемости: все сервисы могут быть запущены независимо на различных устройствах, а основная часть - веб-сервер - допускает и горизонтальное масштабирование - запуск копий одной программы параллельно на нескольких устройствах для распределения нагрузки.

## 3.4. Описание и обоснование функционирования и алгоритмов работы программы

### 3.4.1. Описание и обоснование алгоритма аутентификации запросов

#### 3.4.1.1. Описание алгоритма аутентификации запросов

Для повышения безопасности приложения и защиты от различных атак, включая CSRF, все запросы проходят аутентификацию в соответствии с RFC 6749<sup>[12]</sup>:

- 1) Для каждого запроса случайным образом генерируется криптографически-сильный ключ размером 16 байт, представляемый в формате hex. В хранилище данных Redis сгенерированному ключу ставится в соответствие URL запроса и IP-адрес пользователя, а ключ считается действительным лишь определённое указываемое в файле конфигурации время.
- 2) При отправке запросов к уязвимым точками доступа пользователь должен указывать выданный ключ в теле запроса. При этом сервер производит проверку переданного значения посредством получения ассоциированной с ним информации из хранилища данных Redis и сравнения IP-адресов пользователя, которому был выдан ключ, и

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

пользователя, который отправил запрос. При их несовпадении сервер не доверяет запросу и отклоняет его. Использование одного ключа для аутентификации нескольких запросов недопустимо.

#### 3.4.1.2. Обоснование алгоритма аутентификации запросов

Описанный алгоритм позволяет эффективно отклонять запросы, сделанные некорректно, а также запросы, с высокой вероятностью не являющиеся легитимными запросами пользователей. Он широко применяется для защиты от атак, основанных на подмене злоумышленниками или злонамеренными программами назначения информации, введённой пользователем. Несмотря на то, что описанный алгоритм не может гарантировать обнаружение всех нелегитимных запросов, при своей относительной простоте он значительно повышает безопасность сервиса.

### 3.4.2. Описание и обоснование алгоритма авторизации пользователей

#### 3.4.2.1. Описание алгоритма авторизации пользователей

При первом взаимодействии с приложением пользователю доступна лишь страница авторизации (рисунок 1), где необходимо ввести выданные администратором идентификационные данные аккаунта.

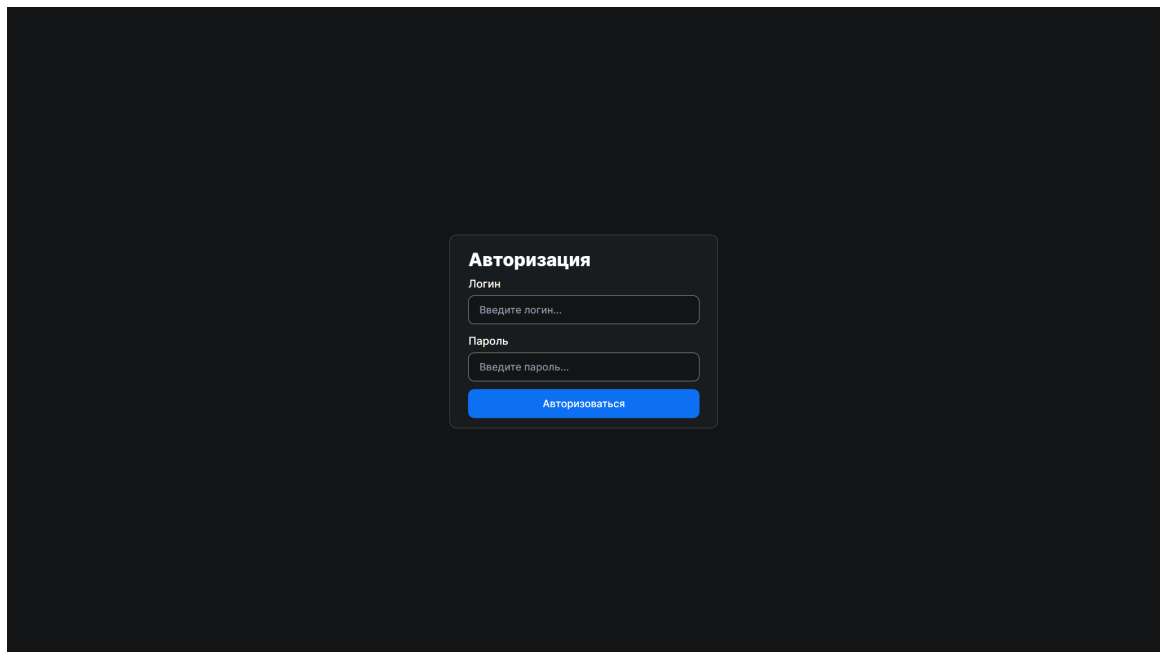


Рисунок 1 - Страница авторизации.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

При вводе необходимых значений и нажатии кнопки “авторизоваться” происходит отправка POST-запроса на сервер, где производится его обработка по следующему алгоритму:

- 1) Аутентификация запроса для повышения безопасности приложения.
- 2) Получение информации об аккаунте пользователя из внутренней базы данных по указанному имени.
- 3) Проверка введенного пользователем пароля на совпадение с паролем полученного аккаунта с помощью криптографического алгоритма bcrypt в соответствии с требованиями RFC 2898<sup>[13]</sup>.
- 4) Если указанный пароль верен, производится генерация и отправка успешного ответа на запрос в соответствии с требованиями RFC 6265<sup>[14]</sup>:
  - a) Генерация криптографически-сильного идентификатора сессии размером 32 байта, представленного в виде строки над алфавитом из 64 символов.
  - b) Сохранение данных пользователя в Redis для повышения эффективности проверки авторизации.
  - c) Подпись cookie-файла, содержащего сгенерированный ID сессии, и его установка в соответствующий заголовок ответа для дальнейшей обработки браузером пользователя.
  - d) Перенаправление пользователя на страницу “База данных” панели управления.
- 5) В случае ошибки на любом шаге ответом на запрос является соответствующий статус и исходный код страницы ошибки.

После входа в аккаунт пользователь получает доступ к страницам панели управления. При этом все дальнейшие запросы при обработке проходят авторизацию, которая работает следующим образом:

- 1) Проверка подписи прикрепленного пользователем к запросу cookie-файла и получение идентификатора сессии из него.
- 2) Получение информации об аккаунте, который ассоциирован с полученным идентификатором сессии, из хранилища данных Redis.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



- 3) Если данные не найдены, авторизация считается неуспешной - идентификатор сессии недействителен: истёк или некорректен. Процесс авторизация завершается, а дальнейшая обработка запроса происходит также, как и для неавторизованного пользователя.
- 4) Если данные найдены, то происходит проверка их актуальности при помощи поля `expiration`, сохраняемого в Redis вместе с информацией об аккаунте пользователя.
- 5) Если данные не актуальны, то происходит их актуализация путём обращения к таблице `users` внутренней базы данных. При этом если информация о нужном аккаунте не найдена, авторизация считается неуспешной. Иначе производится обновление данных в Redis.
- 6) Если авторизация успешна, управление передаётся непосредственному обработчику запроса. В ином случае процесс завершается ошибкой 401, указывающей пользователю на ошибку авторизации.

#### **3.4.2.2. Обоснование алгоритма авторизации пользователей**

Описанный алгоритм позволяет эффективно и достаточно безопасно производить авторизацию пользователей. Идентификатор сессии размером 32 байта обеспечивает его высокую криптографическую стойкость и достаточную безопасность приложения, а использование алгоритма `bcrypt` для шифрования паролей пользователей является достаточной защитой от утечек этой информации даже при компрометации сервера. Более того, подпись `cookie`-файлов и аутентификация POST-запросов на вход в аккаунт добавляют уровень защиты от подмены идентификатора сессии злоумышленниками.

Использование Redis для сохранения актуальной информации об аккаунтах авторизованных пользователей повышает эффективность алгоритма при большом количестве запросов, а регулярная актуализация данных с внутренней БД обеспечивает низкую вероятность работы с устаревшими значениями.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

### 3.4.3. Описание и обоснование функционирования основных элементов интерфейса

#### 3.4.3.1. Описание функционирования основных элементов интерфейса

Все страницы панели управления содержат ряд основных неизменных элементов: верхний колонтитул и меню навигации по приложению, представленное в виде левой боковой панели.

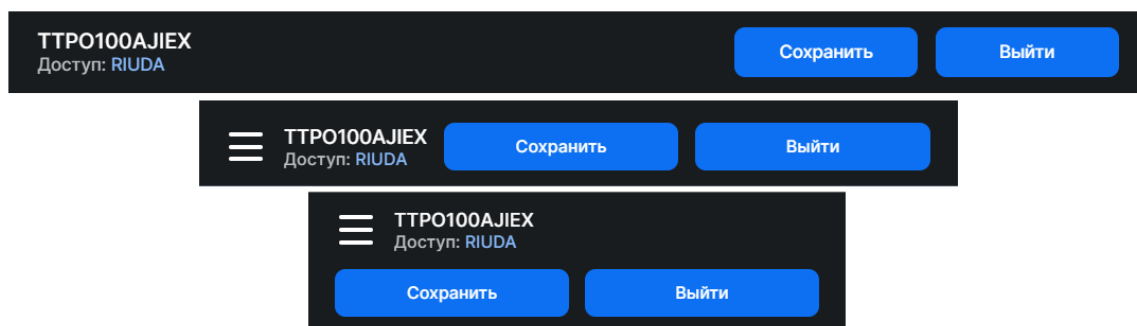


Рисунок 2 - Верхний колонтитул страниц панели управления на различных устройствах.

Верхний колонтитул страниц (рисунок 2) содержит следующие элементы:

- 1) Представленная в виде трёх горизонтальных полос кнопка открытия меню навигации на устройствах с маленькой шириной экрана, где невозможно корректное отображение левой боковой панели и содержимого страницы одновременно.
- 2) Блок с информацией об аккаунте пользователя: имя и код уровня доступа. Для повышения понятности интерфейса при наведении на строку с кодом уровня доступа отображается подсказка, указывающая все права аккаунта явно.
- 3) Блок с кнопками управления приложением, всегда содержащий кнопку выхода из аккаунта (“Выйти”), которая отправляет GET-запрос, в результате которого происходит удаление cookie-файла, содержащего идентификатор сессии. В зависимости от страницы могут быть также отображены кнопки сохранения изменений или создания аккаунта.
- 4) Невидимая пользователями форма для отправки изменений на сервер при нажатии кнопки “сохранить” на определённых страницах.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

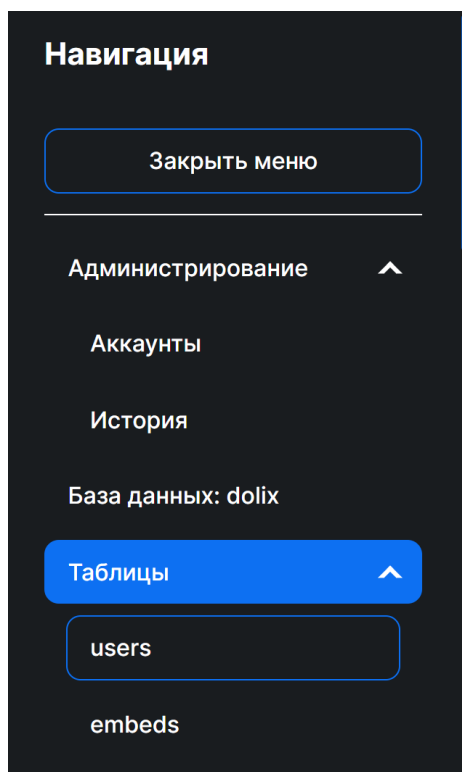


Рисунок 3 - Меню навигации страниц панели управления.

Меню навигации (рисунок 3) содержит следующие элементы:

- 1) Заголовок второго уровня (“Навигация”)
- 2) Кнопка закрытия меню (“Заккрыть меню”), видимая только на устройствах с маленькой шириной экрана, где невозможно корректное отображение левой боковой панели и содержимого страницы одновременно
- 3) Многоуровневый список, обеспечивающий доступ ко всем страницам панели управления. Для удобства пользования части списка можно скрывать нажатием кнопок “Администрирование” и “Таблицы”.
  - а) Кнопки “Аккаунты” и “История”, скрытые для пользователей, не имеющих право “администрирование”, производят загрузку соответствующих страниц панели администратора.
  - б) Кнопка, содержимое которой начинается со слов “База данных”, обеспечивает доступ к странице “База данных” панели управления.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

с) Кнопки группы “Таблицы” производят загрузку страниц, отображающих информацию о соответствующих таблицах целевой базы данных

Адаптация элементов для корректного отображения на различных устройствах происходит в соответствии со следующим алгоритмом:

- 1) Если ширина экрана не менее 1000 пикселей, меню навигации отображается в виде левой боковой панели.
- 2) Если ширина экрана менее 1000 пикселей, меню навигации скрыто, а в верхнем колонтитуле присутствует кнопка его открытия, при нажатии на которую происходит отображение меню, а взаимодействие с остальными частями страницы становится невозможным. Для закрытия меню навигации пользователю необходимо нажать в любое свободное место на экране или воспользоваться соответствующей кнопкой боковой панели.
- 3) Если ширина части страницы, отведённой для основного содержимого, менее 700 пикселей, верхний колонтитул отображается в виде двух строк, первая из которых содержит кнопку открытия меню навигации и блок с информацией об аккаунте, а вторая - кнопки управления приложением.
- 4) Если ширина части страницы, отведённой для основного содержимого, не менее 700 пикселей, но менее 900 пикселей, верхний колонтитул отображается в виде одной строки, но элемент разделения его частей отсутствует: кнопки управления приложением занимают всё свободное место.
- 5) Если ширина основной части не менее 900 пикселей, верхний колонтитул страницы отображается в виде одной строки, а его части выравниваются по разным сторонам элемента.

#### **3.4.3.2. Обоснование функционирования основных элементов интерфейса**

Описанные элементы обеспечивают удобную навигацию между страницами приложения, позволяют пользователям легко управлять сервисом, а также отображают основную информацию об аккаунте пользователя. При этом обеспечено корректное отображение элементов на различных устройствах независимо от их размера, что является одним из ключевых требований WCAG.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.4.4. Описание и обоснование функционирования страницы “База данных”

3.4.4.1. Описание функционирования страницы “База данных”

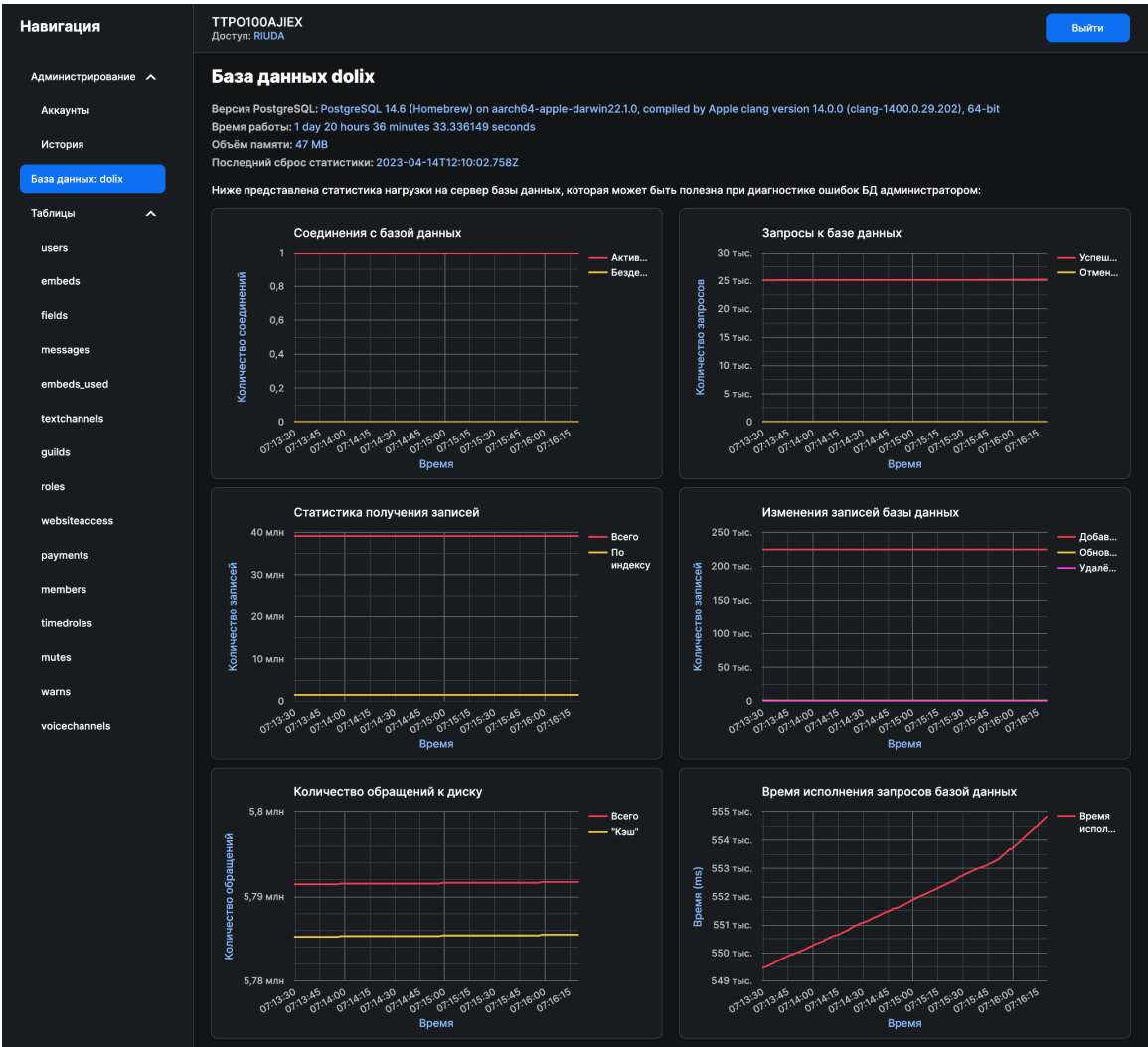


Рисунок 4 - Страница “База данных” панели управления

Страница “База данных” панели управления (рисунок 4) помимо основных содержит следующие элементы:

- 1) Заголовок страницы первого уровня, содержащий название целевой базы данных
- 2) Набор текстовой информации о целевой БД: версия PostgreSQL, время работы, объём занимаемой на диске памяти и дата последнего сброса статистики.
- 3) Графическая информация о базе данных:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- a) График количества активных и бездействующих соединений с базой данных с заголовком “Соединения с базой данных”
- b) График количества успешных и отменённых запросов к базе данных с момента последнего обновления статистики сервером PostgreSQL с заголовком “Запросы к базе данных”
- c) График количества записей, полученных исполнителем запросов с момента последнего обновления статистики сервером PostgreSQL с заголовком “Статистика получения записей”
- d) График количества изменённых (добавленных, обновлённых, удалённых) с момента последнего обновления статистики сервером PostgreSQL записей с заголовком “Изменения записей базы данных”.
- e) График количества прочитанных исполнителем запросов с момента последнего обновления статистики сервером PostgreSQL блоков диска с заголовком “Количество обращений к диску”.
- f) График времени, потраченного базой данных на исполнение запросов, с момента последнего обновления статистики сервером PostgreSQL с заголовком “Время исполнения запросов базой данных”.

Страница предоставляется по точке доступа GET /database, а обработка соответствующего запроса происходит по следующему алгоритму:

- 1) Авторизация пользователя.
- 2) Проверка, есть ли у аккаунта пользователя доступ на уровне “чтение” или выше.
- 3) Запрос к целевой базе данных для генерации основных элементов панели управления.
- 4) Генерация исходного кода веб-страницы по шаблону и его отправка пользователю.

Для реализации обновления данных “в прямом эфире” был реализован алгоритм, работающий следующим образом:

- 1) По окончании загрузки страницы происходит установление двустороннего соединения с сервером с использованием протокола WebSocket. Для этого на сервере зарегистрирована специальная точка доступа, по обращению к которой происходит установление соединения, настройка обработки сообщений и отправка клиенту

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

конфигурационной информации - максимального количества отображаемых на графиках записей, размера страниц при загрузке таблиц и интервала обновления данных.

- 2) Клиентская часть приложения производит настройку страницы: сохраняет полученные конфигурационные данные и инициализирует графики.
- 3) Клиент отправляет сообщение типа `database_data` для получения актуальной на данный момент информации о целевой базе данных.
- 4) Сервер запрашивает у целевой базы данных необходимую информацию и отправляет клиенту в ответ сообщение типа `database_data`.
- 5) Клиент производит обновление всех текстовых данных путём редактирования свойства `innerText` соответствующих HTML-элементов, а также добавляет записи на соответствующие графики.
- 6) По окончании обработки данных клиент планирует переход к шагу 3 настоящего алгоритма через полученный в конфигурационном сообщении период времени для следующей синхронизации значений.

#### 3.4.4.2. Обоснование функционирования страницы “База данных”

Описанная реализация позволяет эффективно предоставлять пользователю первичный ответ на запрос в виде исходного кода, реализующего лишь разметку веб-страницы без наполнения необходимыми значениями, с последующей загрузкой текстовых и графических данных, когда пользователь уже может взаимодействовать с элементами страницы. Это позволяет улучшить пользовательский интерфейс, так как не требует длительного ожидания начального ответа сервера без отображения интерфейса приложения на экране устройства пользователя.

Более того, описанный алгоритм позволяет добавлять на страницу новую информацию о целевой базе данных, не внося значительных изменений в архитектуру приложения и лишь незначительно увеличивая время начального ответа сервера.

Использование протокола `WebSocket` также повышает эффективность взаимодействия, так как позволяет избежать повторных затрат на установление соединения при каждой синхронизации данных, требуя это лишь единожды при загрузке страницы.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.4.5. Описание и обоснование функционирования страниц панели управления, соответствующих таблицам целевой базы данных

3.4.5.1. Описание функционирования страниц панели управления, соответствующих таблицам целевой базы данных

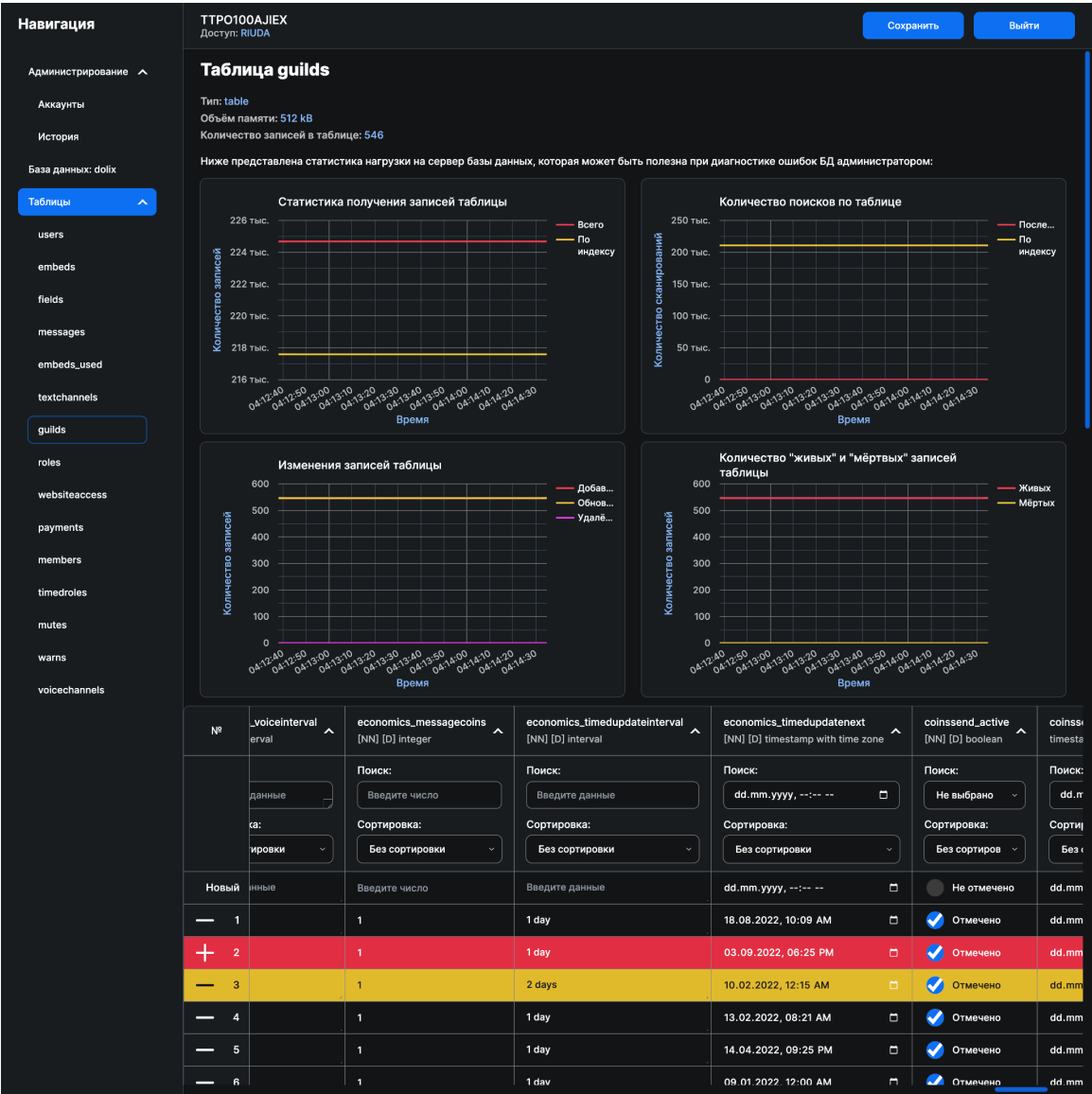


Рисунок 5 - Страница панели управления, соответствующая таблице целевой базы данных.

Страницы, соответствующие таблицам целевой базы данных (рисунок 5), помимо основных содержат следующие элементы:

- 1) Заголовок страницы первого уровня, содержащий имя таблицы

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



- 2) Набор текстовой информации о таблице: объём занимаемой на диске памяти и оценочное количество записей в таблице
- 3) Графическая информация о таблице:
  - a) График количества записей таблицы, полученных исполнителем запросов с момента последнего обновления статистики сервером PostgreSQL с заголовком “Статистика получения записей таблицы”.
  - b) График количества произведённых исполнителем запросов последовательных сканирований таблицы и поисков по её индексам с момента последнего обновления статистики сервером PostgreSQL с заголовком “Количество поисков по таблице”.
  - c) График количества изменённых (добавленных, обновлённых, удалённых) с момента последнего обновления статистики сервером PostgreSQL записей таблицы с заголовком “Изменения записей таблицы”
  - d) График количества “живых” и “мёртвых” записей в таблице с заголовком “Количество "живых" и "мёртвых" записей таблицы”.
- 4) Таблица управления данными, хранящимися в таблице целевой БД, с возможностью сортировки, фильтрации и редактирования, состоящая из следующих частей:
  - a) Заголовок таблицы, содержащий названия столбцов, их типы данных и ограничения на их значения.
  - b) Всплывающее меню настройки параметров фильтрации и сортировки записей по значению выбранного столбца, открывающееся по нажатию на соответствующую часть заголовка таблицы.
  - c) Блоки добавления и отображения данных, содержащие строки, отображающие записи таблицы целевой БД, с полями ввода, соответствующими типам данных значений столбцов таблицы.

Страница предоставляется по точке доступа GET /table с указанием идентификатора таблицы в качестве единственного параметра запроса. Обработка запроса производится аналогично странице “База данных” панели управления, а именно:

- 1) Авторизация пользователя.
- 2) Проверка, есть ли у аккаунта пользователя доступ на уровне “чтение” или выше.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- 3) Запрос к целевой базе данных для генерации основных элементов панели управления и получения информации о столбцах таблицы: их именах, типах данных и ограничениях на значения.
- 4) Генерация исходного кода веб-страницы по шаблону и его отправка пользователю. При этом выполняется также создание HTML-шаблона строки таблицы управления данными с учётом уровня доступа пользователя

Для реализации обновления данных “в прямом эфире” был реализован алгоритм, работающий следующим образом:

- 1) По окончании загрузки страницы происходит установление двустороннего соединения с сервером с использованием протокола WebSocket. Для этого на сервере зарегистрирована специальная точка доступа, по обращению к которой происходит установление соединения, настройка обработки сообщений и отправка клиенту конфигурационной информации - максимального количества отображаемых на графиках записей, размера страниц при загрузке таблиц и интервала обновления данных.
- 2) Клиентская часть приложения производит настройку страницы: сохраняет полученные конфигурационные данные, инициализирует графики и таблицу.
- 3) Клиент отправляет сообщение типа `table_data` для получения актуальной на данный момент информации о просматриваемой таблице целевой базы данных.
- 4) Сервер запрашивает у целевой базы данных необходимую информацию о таблице и отправляет клиенту в ответ сообщение типа `table_data`.
- 5) Клиент производит обновление всех текстовых данных путём редактирования свойства `innerText` соответствующих HTML-элементов, а также добавляет записи на соответствующие графики.
- 6) По окончании обработки данных клиент планирует переход к шагу 3 настоящего алгоритма через полученный в конфигурационном сообщении период времени для следующей синхронизации значений.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Для отображения записей, хранящихся в таблице, и предоставления пользователю возможности их сортировки, фильтрации и редактирования, был также реализован алгоритм управления таблицей, работающие следующим образом:

- 1) При инициализации таблицы производится поиск шаблона строки и динамическое создание двух классов на его основе: строки блока добавления данных и строки блока отображения данных.
- 2) Создание строки в блоке добавления данных, если он существует:
  - a) Если блок добавления данных не пуст, выполняется отображение кнопки удаления последней строки блока, если она скрыта
  - b) Клонирование содержимого шаблона строки блока добавления данных и вставка полученных элементов в конец соответствующего HTML-контейнера.
- 3) Перезагрузка блока отображения данных:
  - a) Удаление всех элементов блока отображения данных
  - b) Загрузка очередной страницы блока отображения данных:
    - i) Получения списка применённых пользователем в заголовке таблицы параметров фильтрации и сортировки записей
    - ii) Отправка сообщения соответствующего типа по указанному WebSocket-соединению и ожидание ответа сервера со списком записей очередной страницы
    - iii) Отображение полученных данных путём клонирования содержимого шаблона строки блока отображения данных и вставки полученных элементов в конец соответствующего HTML-контейнера
  - c) При появлении последней строки блока отображения данных на экране устройства пользователя происходит повторение пункта 3b настоящего алгоритма
  - d) При нажатии кнопки удаления строки блока отображения данных вся строка выделяется красным цветом
  - e) При редактировании любого поля ввода строки блока отображения данных вся строка выделяется жёлтым цветом

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- 4) При редактировании любого поля ввода последней строки блока добавления данных происходит повторение пункта 2 настоящего алгоритма.
- 5) При редактировании задаваемых в заголовке таблицы параметров фильтрации и сортировки записей блока отображения данных происходит повторение пункта 3 настоящего алгоритма

Для обработки принятого по протоколу WebSocket запроса на получение списка записей очередной страницы веб-сервер действует в соответствии со следующим алгоритмом:

- 1) Если у пользователя нет доступа на уровне “чтение”, обработка запроса не производится и алгоритм завершается.
- 2) Генерация SQL-запроса на получение всех строк целевой таблицы, удовлетворяющих заданным фильтрам в заданном порядке
- 3) Если полученный SQL-запрос не совпадает с SQL-запросом, который был сгенерирован при обработке предыдущего сообщения пользователя, производятся следующие операции:
  - a) Завершение выполнения предыдущего SQL-запроса с закрытием курсора и освобождением клиента
  - b) Выполнение нового запроса с созданием нового курсора PostgreSQL
  - c) Добавление соответствующей записи в историю сделанных пользователями запросов
- 4) Получение запрошенного пользователем количества записей с помощью курсора PostgreSQL и их отправка пользователю

При нажатии пользователем кнопки сохранения изменений происходит сбор необходимой информации из таблицы и её отправка веб-серверу по протоколу HTTP в соответствии со следующим алгоритмом:

- 1) Получение объектов значений новых, созданных пользователем строк таблицы.
- 2) Получение объектов-идентификаторов удалённых записей таблицы, то есть наборов исходных значений, позволяющих однозначно определить запись, которую необходимо удалить. Такими наборами значений считаются:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- a) Значения, соответствующие столбцам, которые являются ключом таблицы
  - b) Если в таблице нет ключа, то значения, соответствующие столбцам, имеющим ограничение уникальности значений.
  - c) Если в таблице нет ключа и ограничений уникальности значений, то значения всех ячеек строки
- 3) Получение объектов значений и идентификаторов изменённых строк таблицы
  - 4) Объединение полученных значений в массив и отправка POST-запроса к точке доступа /data веб-сервера.

Обработка такого запроса производится следующим образом:

- 1) Авторизация пользователя.
- 2) Проверка, есть ли у аккаунта пользователя доступ к осуществлению запрошенных изменений.
- 3) Генерация массива insert, update или delete SQL-запросов к целевой базе данных, отражающих сделанные пользователем изменения, а также массива запросов к внутренней базе данных для создания соответствующих записей истории запросов, сделанных пользователями.
- 4) Отправка сгенерированных запросов соответствующим базам данных
- 5) В случае ошибки баз данных, пользователю отображается соответствующее сообщение. Иначе производится перезагрузка страницы приложения.

#### **3.4.5.2. Обоснование функционирования страниц панели управления, соответствующих таблицам целевой базы данных**

Описанная реализация обеспечивает высокую эффективность веб-страницы, так как не требует получения всех отображаемых значений для отправки первичного ответа клиенту. Использование протокола WebSocket для получения данных после загрузки страницы также позволяет повысить скорость работы программы, не требуя установления соединения и авторизации пользователей при каждом обращении к серверу.

Более того, реализованные алгоритмы позволяют модифицировать и расширять работу веб-страницы без внесения значительных изменений в архитектуру приложения. В частности, архитектура страницы позволяет добавлять поддержку новых типов данных

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

PostgreSQL, не прибегая к внесению изменений в протоколы взаимодействия клиента и сервера.

Применение курсоров PostgreSQL для постраничного чтения строк таблицы целевой базы данных также повышает эффективность приложения, так как позволяет избежать загрузки больших массивов данных в память, а также обеспечивает невозможность дублирования или пропуска отдельных записей из-за изменения содержимого таблицы БД другими программами во время работы.

Производимый алгоритмами анализ структуры таблицы также повышает корректность работы приложения и позволяет безошибочно, хотя и не всегда наиболее эффективно, определять набор значений, идентифицирующий строку таблицы.

### 3.4.6. Описание и обоснование функционирования страницы “Аккаунты”

#### 3.4.6.1. Описание функционирования страницы “Аккаунты”

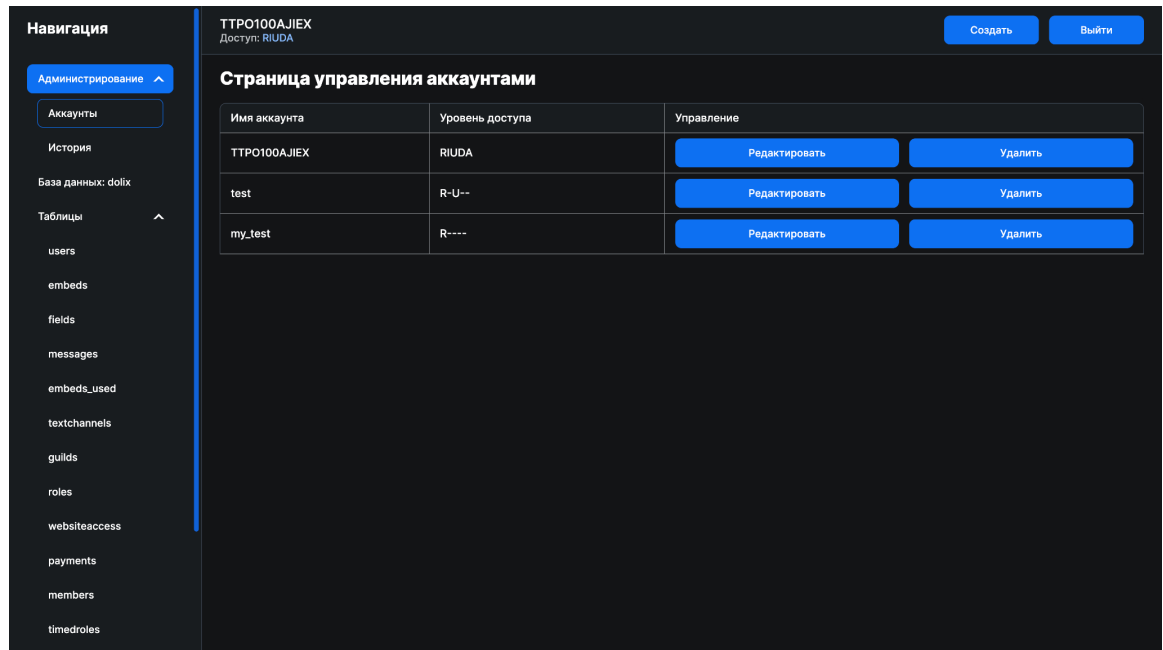


Рисунок 6 - Страница “Аккаунты” панели администратора.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

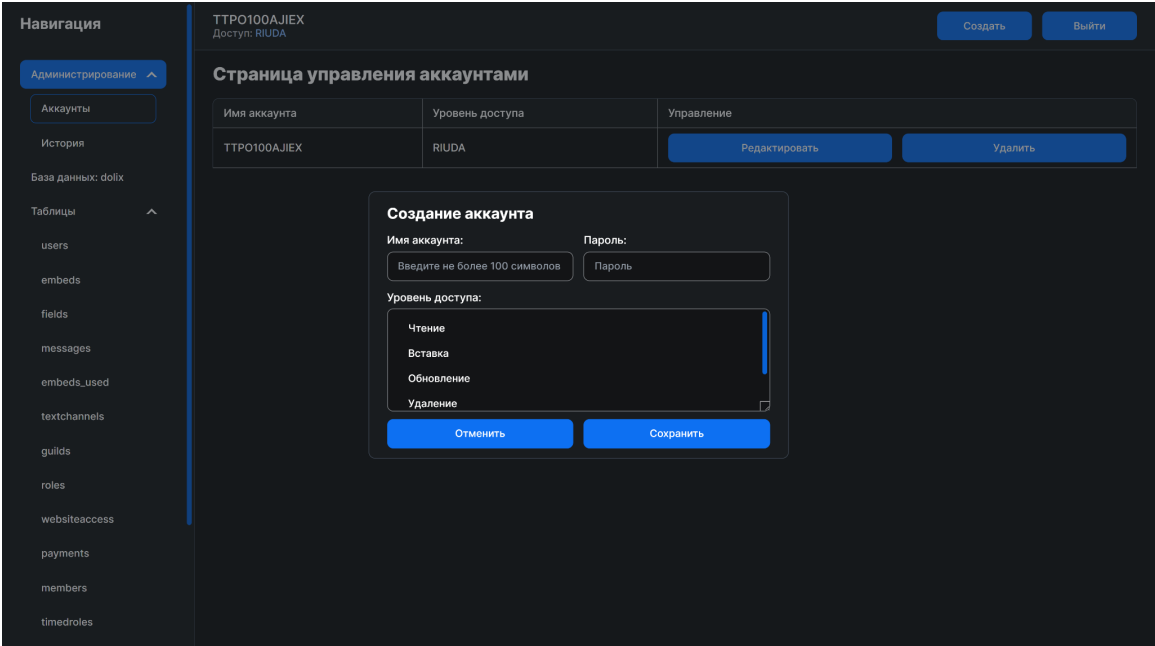


Рисунок 7 - Всплывающее окно создания аккаунта страницы “Аккаунты”

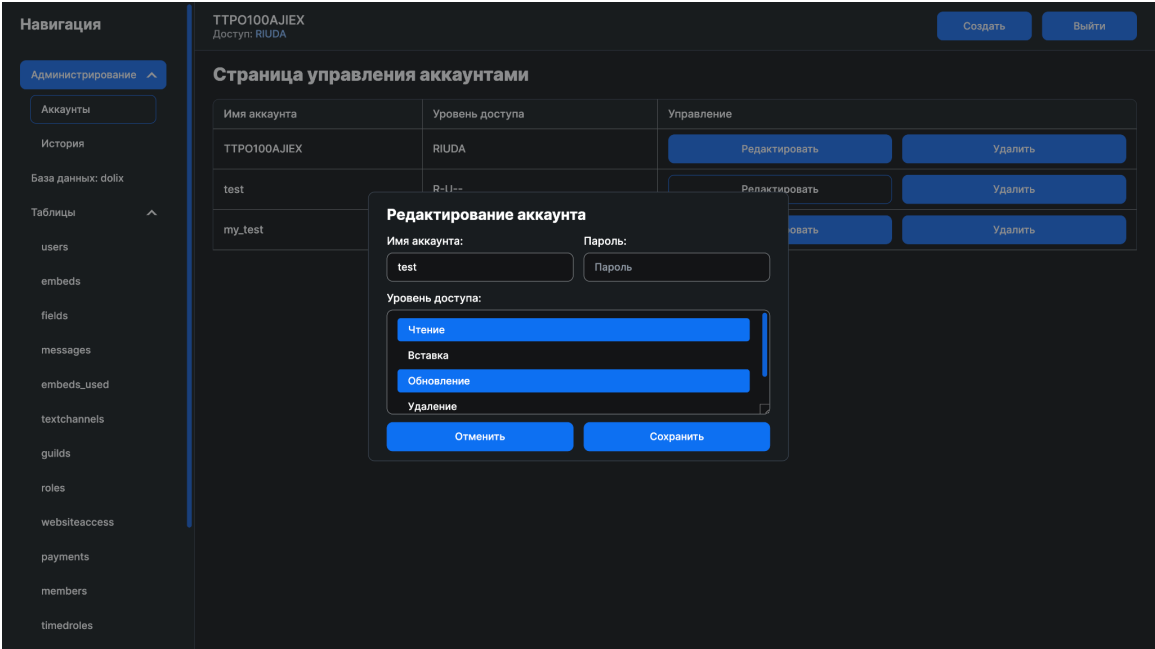


Рисунок 8 - Всплывающее окно редактирования аккаунта страницы “Аккаунты”

Страница “Аккаунты” панели администратора (рисунок 6) помимо основных элементов панели управления содержит следующие части:

- 1) Заголовок страницы первого уровня
- 2) Список зарегистрированных аккаунтов пользователей в виде таблицы:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- a) Первый столбец таблицы указывает имя аккаунта
  - b) Второй столбец таблицы указывает код уровня доступа пользователя.
  - c) Третий столбец таблицы содержит кнопки редактирования и удаления аккаунта
- 3) Всплывающее окно создания аккаунта (рисунок 7) и всплывающие окна редактирования аккаунтов (рисунок 8), содержащие заголовки второго уровня, поля ввода соответствующих данных аккаунтов, а также кнопки закрытия всплывающих окон и применения изменений.

Страница предоставляется по точке доступа GET /admin/accounts, а обработка соответствующего запроса производится по следующему алгоритму:

- 1) Авторизация пользователя.
- 2) Проверка, есть ли у аккаунта пользователя доступ на уровне “администрирование”.
- 3) Запрос к целевой базе данных для генерации основных элементов панели управления.
- 4) Запрос к внутренней базе данных для получения списка зарегистрированных аккаунтов
- 5) Генерация исходного кода веб-страницы по шаблону и его отправка пользователю.

Для обеспечения возможности управления аккаунтами веб-страница реагирует на следующие действия пользователя:

- 1) При нажатии кнопки “Создать” открывается всплывающее окно создания аккаунта. При нажатии кнопки “Отменить” происходит закрытие окна, а при использовании кнопки “Сохранить” происходит отправка POST-запроса на создание аккаунта.
- 2) При нажатии кнопки “Редактировать” открывается всплывающее окно редактирования соответствующего аккаунта, всем полям ввода которого, кроме поля ввода пароля, установлены начальные значения. При нажатии кнопки “Отменить” происходит закрытие окна, а при использовании кнопки “Сохранить” происходит отправка POST-запроса на редактирование соответствующего аккаунта.
- 3) При нажатии кнопки “Удалить” происходит отправка POST-запроса на удаление соответствующего аккаунта.

Обработка запросов на создание, редактирование и удаления аккаунтов происходит по точкам доступа POST /admin/accounts/create, POST /admin/accounts/edit и POST /admin/accounts/delete соответственно следующим образом

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



- 1) Авторизация пользователя.
- 2) Проверка, есть ли у аккаунта пользователя доступ на уровне “администрирование”.
- 3) Шифрование указанного пароля для повышения безопасности его хранения в базе данных, если необходимо
- 4) Запрос к внутренней базе данных на создание, редактирование или удаление записи таблицы аккаунтов пользователей
- 5) В случае успешного выполнения всех операций происходит обновление страницы приложения

#### **3.4.6.2. Обоснование функционирования страницы “Аккаунты”**

Описанные алгоритмы позволяют эффективно реализовать страницу “Аккаунты”, не прибегая к излишним усложнениям архитектуры приложения. Получение информации обо всех аккаунтах сразу при загрузке страницы избавляет от необходимости установления двустороннего соединения с сервером и, следовательно, повышает эффективность приложения, хотя и может привести к значительному увеличению времени первичного ответа сервера при регистрации большого количества аккаунтов.

Отсутствие необходимости отображения паролей пользователей при редактировании аккаунтов позволяет производить их шифрование при сохранении в базе данных, что повышает безопасность данных пользователей приложения.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

### 3.4.7. Описание и обоснование функционирования страницы “История”

#### 3.4.7.1. Описание функционирования страницы “История”

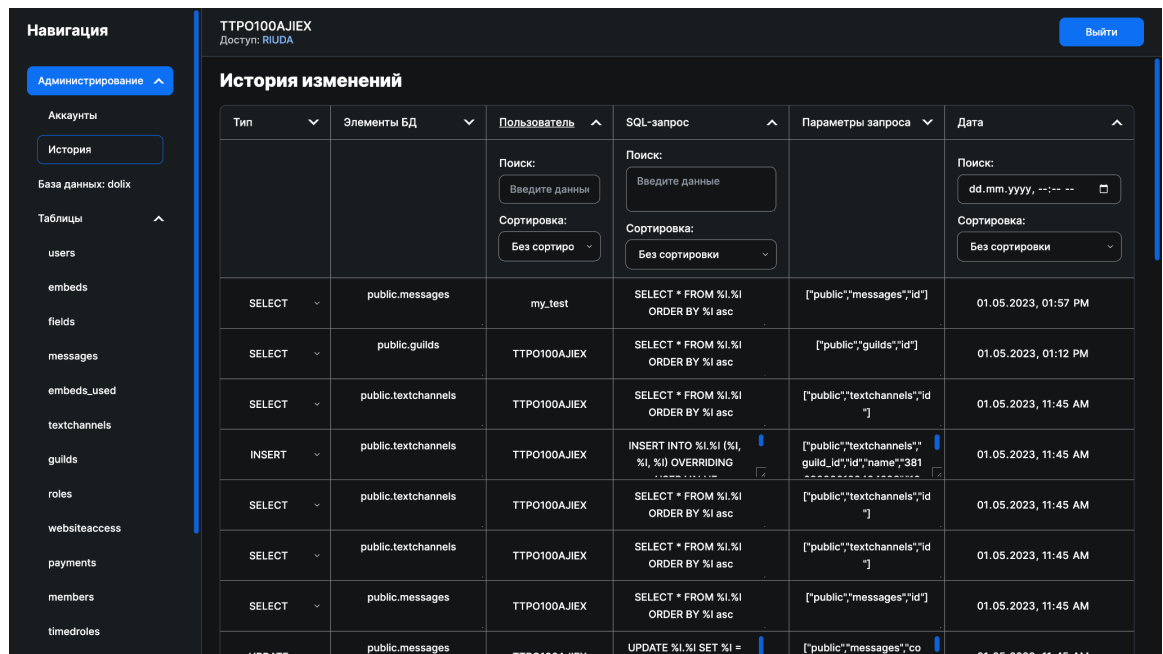


Рисунок 9 - Страница “История” панели администратора.

Страница “История” панели администратора (рисунок 9) помимо основных элементов панели управления содержит следующие части:

- 1) Заголовок страницы первого уровня
- 2) Таблица отображения списка запросов, сделанных пользователями приложения к целевой базе данных. Причём её реализация аналогична применённой на страницах, соответствующих таблицам целевой базы данных, за исключением отсутствия блока добавления данных и недопустимости редактирования и удаления записей блока отображения данных.

Страница предоставляется по точке доступа GET /admin/logs, а обработка запроса производится следующим образом:

- 1) Авторизация пользователя.
- 2) Проверка, есть ли у аккаунта пользователя доступ на уровне “администрирование”.
- 3) Запрос к целевой базе данных для генерации основных элементов панели управления.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- 4) Обращение к внутренней базе данных для получения информации о типах данных столбцов таблицы, содержащей историю изменений.
- 5) Генерация по шаблону и отправка пользователю исходного кода веб-страницы аналогично описанному ранее за исключением отсутствия элементов, отображающих информацию о таблице.

Для организации корректной загрузки данных “в прямом эфире” был реализован алгоритм, работающий следующим образом:

- 1) По окончании загрузки страницы происходит установление двустороннего соединения с сервером с использованием протокола WebSocket. Для этого на сервере зарегистрирована специальная точка доступа, по обращении к которой происходит установление соединения, настройка обработки сообщений и отправка клиенту конфигурационной информации - максимального количества отображаемых на графиках записей, размера страниц при загрузке таблиц и интервала обновления данных.
- 2) Клиентская часть приложения производит настройку страницы: сохраняет полученные конфигурационные данные и инициализирует таблицу, которая работает по алгоритму, описанному ранее, используя для получения данных установленное соединение.

#### **3.4.7.2. Обоснование функционирования страницы “История”**

Описанный алгоритм позволяет приложению эффективно предоставлять функции для удобного просмотра и анализа отображаемых данных администраторами. Загрузка разметки страницы без данных обеспечивает высокую скорость генерации первичного ответа сервера, а использование протокола WebSocket для дальнейшего взаимодействия позволяет эффективно заполнять таблицу данными и производить фильтрацию и сортировку записей.

Использование тех же алгоритмов работы таблицы, что и в других частях панели управления, добавляет странице “История” панели администратора все те же преимущества, а также позволяет переиспользовать исходный код, что улучшает архитектуру приложения.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

### 3.5. Описание и обоснование методов организации входных и выходных данных

#### 3.5.1. Описание и обоснование метода организации входных данных

##### 3.5.1.1. Описание метода организации входных данных

При запуске сервера входные данные должны быть размещены в файле конфигурации config.json директории common/configs:

- 1) Поле stage должно указывать режима запуска приложения: testing в режиме тестирования или release в обычном режиме
- 2) Поле application должно указывать название приложения
- 3) Поля PostgreSQL.target и PostgreSQL.internal должны содержать реквизиты подключения к целевой и внутренней базам данных соответственно:
  - a) Поля host и port должны указывать адрес соответствующего сервера PostgreSQL в сети
  - b) Поле database должно указывать название соответствующей базы данных
  - c) Поля user и password должны указывать имя и пароль для подключения к соответствующей базе данных
- 4) Поле redis должно содержать реквизиты подключения к хранилищу данных Redis
  - a) Поля host и port должны указывать адрес сервера Redis в сети
  - b) Поле database должно указывать номер базы данных Redis, с которой должно работать приложение
  - c) Поле password должно указывать пароль для подключения к серверу Redis
- 5) Поле website должно содержать параметры запускаемого сервера
  - a) Поле host должно указывать имя сервера в сети “Интернет”
  - b) Поля ssl\_key и ssl\_cert должны указывать пути к файлам с полной цепью обмена ключей и с закрытым ключом SSL-сертификата защищённого соединения
  - c) Поле https\_port должно указывать порт, по которому должны приниматься запросы по протоколу HTTPS и поддерживаться соединения по протоколу WebSocket

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- d) Поле `secret` должно указывать секретный ключ, используемый для организации работы веб-сервера. В частности, указываемый ключ используется для подписи создаваемых приложением cookie-файлов.
- e) Поле `authentication_expiration` должно указывать время в миллисекундах, через которое ключ аутентификации, сгенерированный приложением, становится недействительным.
- f) Поле `authorization_expiration` должно указывать время в миллисекундах, через которое идентификатор сессии, выданный пользователю, становится недействительным.
- g) Поле `user_data_expiration` должно указывать периодичность синхронизации данных авторизованных пользователей с внутренней базой данных в миллисекундах.
- 6) Поле `bcrypt.saltRounds` должно указывать количество повторений алгоритма `bcrypt` при шифровании паролей пользователей.
- 7) Поле `defaultAccount` должно указывать начальные данные стандартного аккаунта:
  - a) Поле `login` должно указывать имя аккаунта
  - b) Поле `password` должно указывать пароль для доступа к аккаунту
- 8) Поле `graphs_records` должно указывать максимальное количество отображаемых на графиках записей
- 9) Поле `page_size` должно указывать размер страниц при загрузке таблиц панели управления
- 10) Поле `update_interval` должно указывать интервал обновления данных в панели управления в миллисекундах

Во время работы приложения входными данными его клиентской составляющей являются взаимодействия пользователя с веб-страницей, которые в полной мере обрабатываются браузером. Входными данными баз данных PostgreSQL и Redis являются соответствующие запросы, выполняемые серверами PostgreSQL и Redis, а входными данными веб-сервера являются HTTP-запросы и сообщения по протоколу WebSocket в соответствии с таблицей 1. Причем все запросы к веб-серверу, требующие авторизации пользователей, должны содержать подписанный приложением cookie-файл с именем

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

\_\_Secure-authorization, содержащий идентификатор сессии пользователя. Отправка лишних данных в теле или параметрах запроса не допускается, а нарушающие запросы отклоняются веб-сервером.

Таблица 1 - описание входных данных точек доступа веб-сервера

Точка доступа	Описание входных данных
GET /, GET /main	Отсутствуют
POST /oauth	Тело запроса: <ul style="list-style-type: none"> <li>– authentication - ключ аутентификации запроса</li> <li>– login - имя аккаунта</li> <li>– password - пароль для доступа к аккаунту</li> </ul>
GET /logout	Отсутствуют
GET /database	Отсутствуют
GET /table	Параметры запроса: <ul style="list-style-type: none"> <li>– id - идентификатор таблицы целевой базы данных</li> </ul>
GET /data, websocket	Отсутствуют
POST /data	Тело запроса: <ul style="list-style-type: none"> <li>– authentication - ключ аутентификации запроса</li> <li>– table - идентификатор таблицы целевой базы данных</li> <li>– actions - JSON-строка, описывающая массив действий, которые необходимо выполнить. Каждое действие представляется объектом со следующими полями: <ul style="list-style-type: none"> <li>– type - тип действия (INSERT, UPDATE, DELETE)</li> <li>– data - объект, содержащий данные новой записи или изменённые данные редактируемой записи. Поле должно отсутствовать, если type равно DELETE.</li> <li>– id - объект, идентифицирующий удаляемую или редактируемую запись. Поле должно отсутствовать, если type равно INSERT.</li> </ul> </li> </ul>
GET /admin/accounts	Отсутствуют
POST /admin/accounts/create	Тело запроса: <ul style="list-style-type: none"> <li>– authentication - ключ аутентификации запроса</li> <li>– login - имя создаваемого аккаунта</li> <li>– password - пароль создаваемого аккаунта</li> <li>– permissions - массив символов кода уровня доступа аккаунта</li> </ul>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 1

POST /admin/accounts/delete	Тело запроса: – authentication - ключ аутентификации запроса – id - идентификатор удаляемого аккаунта
POST /admin/accounts/edit	Тело запроса: – authentication - ключ аутентификации запроса – id - идентификатор редактируемого аккаунта – login - новое имя аккаунта – password - новый пароль аккаунта – permissions - массив символов кода нового уровня доступа аккаунта
GET /admin/logs	Отсутствуют
Статические файлы (GET /static/*)	Отсутствуют
Сообщения, передаваемые по протоколу WebSocket	– type - тип запроса – data - тело запроса
Сообщения типа database_data	Тело запроса отсутствует
Сообщения типа table_data	Тело запроса: – id - идентификатор таблицы целевой базы данных
Сообщения типа table_rows	Тело запроса: – tableid - идентификатор таблицы базы данных (для сообщений типа table_rows) – limit - максимальное количество возвращаемых записей – filters - массив объектов - параметров фильтрации: – name - название столбца, по которому должна производиться фильтрация – value - искомое значение столбца – comparison - способ сравнения значений: полное совпадение (exact) или подстрока (substring) – sorts - массив объектов - параметров сортировки: – name - название столбца, по которому должна производиться сортировка – order - порядок сортировки: возрастающий (asc) или убывающий (desc)
Сообщения типа logs	

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

### 3.5.1.2. Обоснование метода организации входных данных

Описанный метод организации входных данных обеспечивает эффективность и безопасность приложения, а также способствует понятному структурированию его исходного кода.

Размещение всех конфигурационных файлов веб-сервера в одной директории упрощает настройку приложения, а задание строгого формата запросов обеспечивает низкую вероятность нарушения его работы из-за случайной или намеренной ошибки пользователя.

Использование стандартных методов организации входных данных серверов PostgreSQL и Redis, а также клиентской составляющей приложения дополнительно обеспечивает высокую отказоустойчивость приложения, удобство пользования и простоту сопровождения.

### 3.5.2. Описание и обоснование метода организации выходных данных

#### 3.5.2.1. Описание метода организации выходных данных

Выходными данными клиентской составляющей приложения являются отображение веб-страниц в браузере пользователя и отправка запросов и сообщений на сервер в ответ на действия пользователя.

Выходными данными серверов PostgreSQL и Redis являются ответы на полученные от веб-сервера запросы.

Выходными данными веб-сервера являются ответы на полученные HTTP-запросы и сообщения по протоколу WebSocket в соответствии с таблицей 2. Помимо этого, для всех ответов по протоколу HTTP устанавливаются заголовки в соответствии с RFC 9110<sup>[15]</sup>, позволяющие клиенту однозначно интерпретировать данные и обеспечивающие безопасность легитимных пользователей. Если обработка запроса завершилась ошибкой, дополнительно устанавливается соответствующий статус ответа и отправляется исходный код страницы с описанием ошибки.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



Таблица 2 - описание выходных данных точек доступа веб-сервера

Точка доступа	Описание выходных данных
GET /, GET /main	Исходный код начальной страницы или, если пользователь авторизован, перенаправление на страницу “База данных”
POST /oauth	Заголовок создания cookie-файла с идентификатором сессии и перенаправление на страницу “База данных”
GET /logout	Заголовок удаления cookie-файла с идентификатором сессии и перенаправление на страницу авторизации
GET /database	Исходный код страницы “База данных”
GET /table	Исходный код страницы, отображающей информацию о таблице целевой базы данных
GET /data, websocket	Сообщение с конфигурационной информацией: <ul style="list-style-type: none"> <li>– graphs_records - максимальное количество отображаемых на графике записей</li> <li>– page_size - размер страниц при загрузке таблиц</li> <li>– update_interval - интервал обновления данных в миллисекундах</li> </ul>
POST /data	Обновление страницы
GET /admin/accounts	Исходный код страницы “Аккаунты”
POST /admin/accounts/create	Перенаправление на страницу “Аккаунты”
POST /admin/accounts/delete	Перенаправление на страницу “Аккаунты”
POST /admin/accounts/edit	Перенаправление на страницу “Аккаунты”
GET /admin/logs	Исходный код страницы “История”
Статические файлы (GET /static/*)	Содержимое запрошенного файла
Ответы, передаваемые по протоколу WebSocket	<ul style="list-style-type: none"> <li>– eventName - тип ответа</li> <li>– data - тело ответа</li> </ul>
Ответы типа table_rows	Тело ответа представляет массив объектов, описывающих запрошенные записи таблицы целевой базы данных
Ответы типа logs	Тело ответа представляет массив объектов, описывающих запрошенные записи истории запросов

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## Продолжение таблицы 2

Ответы типа database_data	<p>Тело ответа:</p> <ul style="list-style-type: none"> <li>– database_name - имя целевой базы данных</li> <li>– postgres_version - версия PostgreSQL целевой базы данных</li> <li>– uptime - время работы целевой базы данных</li> <li>– database_size - объём памяти, занимаемый целевой базой данных на диске</li> <li>– connections - данные графика “Соединения с базой данных” <ul style="list-style-type: none"> <li>– active - количество активных соединений</li> <li>– idle - количество бездействующих соединений</li> </ul> </li> <li>– queries - данные графика “Запросы к базе данных” <ul style="list-style-type: none"> <li>– success - количество успешных запросов</li> <li>– cancelled - количество отменённых запросов</li> </ul> </li> <li>– reads - данные графика “статистика получения записей” <ul style="list-style-type: none"> <li>– total - суммарное количество полученных записей</li> <li>– index - количество записей, для получения которых использовался ускоренный поиск по индексу</li> </ul> </li> <li>– updates - данные графика “Изменения записей базы данных” <ul style="list-style-type: none"> <li>– inserted - количество добавленных записей</li> <li>– updated - количество обновлённых записей</li> <li>– deleted - количество удалённых записей</li> </ul> </li> <li>– blocks - данные графика “Количество обращений к диску” <ul style="list-style-type: none"> <li>– total - количество обращений к диску</li> <li>– cache - количество обращений к диску, которые удалось выполнить проверкой “кэша”</li> </ul> </li> <li>– worktime.worktime - время исполнения запросов базой данных</li> <li>– reset - дата последнего сброса статистики сервером PostgreSQL</li> </ul>
---------------------------	---

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 2

Ответы типа table_data	<p>Тело ответа:</p> <ul style="list-style-type: none"> <li>– table - информация о таблице целевой базы данных</li> <li>– name - имя таблицы</li> <li>– type - тип таблицы</li> <li>– size - объём памяти, занимаемый таблицей на диске</li> <li>– rows - оценочное количество записей в таблице</li> <li>– reads - данные графика “Статистика получения записей таблицы”</li> <li>– total - суммарное количество полученных записей</li> <li>– index - количество записей, для получения которых использовался ускоренный поиск по индексу</li> <li>– scans - данные графика “Количество поисков по таблице”</li> <li>– sequential - количество произведённых последовательных сканирований таблицы</li> <li>– index - количество произведённых поисков по индексу таблицы</li> <li>– updates - данные графика “Изменения записей базы данных”</li> <li>– inserted - количество добавленных записей</li> <li>– updated - количество обновлённых записей</li> <li>– deleted - количество удалённых записей</li> <li>– live - данные графика “Количество "живых" и "мёртвых" записей таблицы”</li> <li>– live - количество “живых” записей таблицы</li> <li>– dead - количество “мёртвых” записей таблицы</li> </ul>
------------------------	--

### 3.5.2.2. Обоснование метода организации выходных данных

Описанный метод организации выходных данных обеспечивает корректность ответов всех сервисов на запросы, что позволяет настроить между ними эффективное взаимодействие и легко производить интерпретацию и обработку получаемых данных. Более того, установка заголовков ответа веб-сервером не только позволяет однозначно интерпретировать результаты запросов, но и обеспечивает безопасность пользователей приложения и их защиту от типичных атак при использовании браузера, удовлетворяющего RFC 9110<sup>[15]</sup>.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

### 3.6. Описание и обоснование выбора состава технических и программных средств

#### 3.6.1. Описание и обоснование требований к серверному оборудованию

Для корректной работы приложения серверное оборудование должно соответствовать следующим минимальным требованиям:

- 1) 2 виртуальных ядра процессора Intel Xeon E5 2620 или AMD EPYC 7402P
- 2) 4 ГБ оперативной памяти
- 3) 8 ГБ SSD хранилища
- 4) Скорость интернет-соединения 500 Мбит/с.
- 5) Операционная система Linux Debian 11 с возможностью запуска Node.JS приложений на портах 80 и 443.
- 6) Наличие безопасного доступа к серверу и возможности загрузки на него необходимых файлов.

Серверного оборудования, удовлетворяющего указанным требованиям, достаточно для обработки запросов не менее, чем от 100 активных пользователей, и сохранения истории запросов не менее, чем за месяц.

#### 3.6.2. Описание и обоснование требований к клиентскому оборудованию

Для корректной работы приложения клиентское оборудование должно соответствовать следующим требованиям:

- 1) Поддержка последней версии установленной операционной системы.
- 2) Свободный доступ в интернет со скоростью соединения не менее 5 Мбит/с.

Приложение выполняется в браузере, что обеспечивает его низкую требовательность к ресурсам и другим параметрам системы. Тем не менее приложению необходим стабильный доступ в интернет для поддержания двустороннего соединения по протоколу WebSocket.

#### 3.6.3. Описание и обоснование выбора состава программных средств

##### 3.6.3.1. Описание состава программных средств

Серверная часть реализована на языке программирования JavaScript в соответствии со спецификацией ECMAScript 13<sup>[\[16\]](#)</sup> в среде Node.JS версии 19.0.0<sup>[\[17\]](#)</sup> с использованием

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

веб-фреймворка fastify версии 4.15.0<sup>[18]</sup>. Для соединения с базами данных PostgreSQL версии 15.1<sup>[19]</sup> использована библиотека pg версии 8.10.0<sup>[20]</sup>, а для взаимодействия с хранилищем данных Redis версии 7.0.8<sup>[21]</sup> использована библиотека ioredis версии 5.3.2<sup>[22]</sup>. Другие зависимости, используемые при реализации веб-сервера, указаны в приложении 4.

Клиентская часть реализована с помощью языка разметки HTML версии 5<sup>[23]</sup>, языка стилей CSS версии 3<sup>[24]</sup> и языка JavaScript в соответствии со спецификацией ECMAScript 13<sup>[16]</sup>. Для динамической генерации разметки использован шаблонизатор ejs версии 3.1.9<sup>[25]</sup>, а линейные графики панели управления реализованы при помощи библиотеки google charts<sup>[26]</sup>. Корректное отображение веб-страниц гарантируется при их использовании в следующих браузерах:

- 1) Google Chrome или Microsoft Edge версии 105 или выше;
- 2) Mozilla Firefox версии 110 или выше;
- 3) Opera версии 91 или выше;
- 4) Safari версии 16.0 или выше;

Взаимодействие между клиентом и сервером происходит по протоколам HTTPS в соответствии с RFC 9110<sup>[15]</sup> и WebSocket в соответствии с RFC 6455<sup>[27]</sup>.

### 3.6.3.2. Обоснование выбора состава программных средств

Использование фреймворка fastify позволяет применять модульную организацию веб-сервера, что обеспечивает понятную, легко поддерживаемую структуру исходного кода, а большое количество доступных библиотек позволяет эффективно обеспечить защиту приложения от большинства типичных атак. Использование наиболее современных версий Redis и PostgreSQL также способствует повышению эффективности и безопасности программы.

Программные средства, необходимые для корректной работы клиентской составляющей приложения, также позволяют хорошо структурировать исходный код и обеспечивают высокую эффективность и безопасность приложения. Несмотря на то, что при реализации использован ряд методов, описанных лишь современными стандартами HTML и CSS, более 97% людей уже используют браузеры поддерживаемых версий, что позволяет абсолютному большинству пользователей успешно взаимодействовать с приложением.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

#### 4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

##### 4.1. Ориентировочная экономическая эффективность

В рамках проекта расчёт экономической эффективности программного продукта не производился.

##### 4.2. Предполагаемая потребность

Ожидается, что приложение будет пользоваться спросом для снижения затрат на этапе поддержки программного продукта, особенно среди работающих над небольшими проектами команд, которые нуждаются в возможности взаимодействия с базой данных вручную для эффективного выполнения задач сопровождения продукта. Команды, не обладающие достаточным для постоянного содержания группы специалистов бюджетом, смогут единоразово оплатить настройку приложения вместо полного отказа от разработки в связи с невозможностью её сопровождения.

##### 4.3. Экономические преимущества разработки по сравнению с лучшими отечественными и зарубежными образцами или аналогами

Приложение имеет немалое количество аналогов. Для оценки преимуществ проекта было проведено сравнение его функциональных характеристик с некоторыми из них. Результат представлен в таблице 3. Здесь “+” и “-” означают наличие и отсутствие указанной функции соответственно, а “?” означает, что функция присутствует только на не поддерживаемых более версиях, только в “PRO” версии или работает некорректно.

Таблица 3 - сравнение приложения с аналогами

	psql <sup>[28]</sup>	pgAdmin <sup>[29]</sup>	DBeaver <sup>[30]</sup>	DataGrip <sup>[31]</sup>	phppgad min <sup>[32]</sup>	pgWeb <sup>[33]</sup>	DBViewer
Наличие собственной авторизации	-	-	-	+	-	-	+
Возможность настройки доступа к приложению	-	-	-	-	-	-	+
Возможность настройки доступа к разным элементам БД	-	-	-	-	-	-	-
Возможность просмотра истории действий	-	-	-	-	-	-	+

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 3

Возможность работы с несколькими БД одновременно		+	+	+	+	+	+	-
Отображаемая информация (без SQL)	Версия PostgreSQL	+	-	-	-	+	+	+
	Размер БД	+	-	-	-	+	-	+
	Время работы БД	-	-	-	-	-	-	+
	Статистика нагрузки на БД	-	+	-	-	-	-	+
	Другая информация о БД	+	+	+	-	+	-	-
	Размер таблицы	+	-	+	+	-	+	+
	Количество записей в таблице	-	-	+	+	+	+	+
	Статистика нагрузки на таблицу	-	-	-	-	-	-	+
	Другая информация о таблице	+	+	+	-	+	+	-
	Tables	+	+	+	+	+	+	+
	Sequences	+	+	+	+	+	+	-
	Views	+	+	+	+	+	+	-
	Types, Domains	+	+	+	+	+	-	-
	Functions, Triggers, Casts	+	+	+	+	+	+	-

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 3

Возможность просмотра записей (без SQL)	-	-	+	+	+	+	+
Возможность фильтрации данных (без SQL)	-	-	+	-	-	+	+
Возможность сортировки данных (без SQL)	-	-	+	+	+	+	+
Визуализация данных	-	+	?	-	-	?	-
Возможность редактирования записей (без SQL)	-	+	+	+	+	?	+
Возможность вставки записей (без SQL)	-	+	+	+	+	?	+
Возможность удаления записей (без SQL)	-	+	+	+	+	?	+
Построение диаграмм отношений (например, ER)	-	+	+	+	-	-	-
Наличие интерфейса	-	+	+	+	+	+	+
Возможность использования через браузер	-	?	?	-	+	+	+
Возможность скачивания приложения	+	+	+	+	-	-	-
Доступность на устройствах	Windows	+	+	+	+	+	+
	MacOS	+	+	+	+	+	+
	Linux	+	+	+	+	+	+
	IOS	-	-	-	-	+	+
	Android	-	-	-	-	+	+
Поддержка новых версий PostgreSQL	+	+	+	+	-	+	+

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



Таким образом, приложение имеет ряд значимых деталей, отличающих его от аналогов. Несмотря на то, что некоторые возможности, доступные в других приложениях, не будут реализованы, DBViewer предлагает ряд особенностей, специально направленных на упрощение работы с приложением для обычных пользователей, что является его главной задачей.

Большинство приложений аналогов направлены на администрирование баз данных специалистами, на содействие разработчикам при создании БД и настройке взаимодействия с ней, в то время как DBViewer ставит основной целью именно просмотр и редактирование данных на этапе эксплуатации программного продукта пользователями, которые не являются специалистами в области баз данных. Именно с этой целью приложение не реализует продвинутое взаимодействие с целевой БД, ограничиваясь лишь тем, что с наибольшей вероятностью может потребоваться при сопровождении ПО. Именно это является главной конкурентной особенностью DBViewer, которая способствует его продвижению на рынке и позволит обеспечить превосходство над аналогами в первую очередь среди команд, занимающихся поддержкой программных продуктов с небольшим бюджетом.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## 5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

- 1) ГОСТ 19.101-77: Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 2) ГОСТ 19.102-77: Стадии разработки. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 3) ГОСТ 19.103-77: Обозначения программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 4) ГОСТ 19.104-78: Основные надписи. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 5) ГОСТ 19.105-78: Общие требования к программным документам. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 6) ГОСТ 19.106-78: Требования к программным документам, выполненным печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 7) ГОСТ 19.404-79: Пояснительная записка. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 8) ГОСТ 19.602-78: Правила дублирования, учета и хранения программных документов, выполненных печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 9) ГОСТ 19.603-78: Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 10) ГОСТ 19.604-78: Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 11) Web Content Accessibility Guidelines (WCAG) 2.0. [Электронный ресурс] / World Wide Web Consortium (W3C). 11 December 2008. Режим доступа: <https://www.w3.org/TR/WCAG20>, свободный. (дата обращения: 02.05.2023)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- 12) RFC 6749: The OAuth 2.0 Authorization Framework [Электронный ресурс] / Internet Engineering Task Force (IETF). D. Hardt, Ed. Microsoft. October 2012. Режим доступа: <https://www.ietf.org/rfc/rfc6749.txt>, свободный. (дата обращения: 02.05.2023)
- 13) RFC 2898: PKCS #5: Password-Based Cryptography Specification. Version 2.0. [Электронный ресурс] / Network Working Group. B. Kaliski. RSA Laboratories. September 2000. Режим доступа: <https://www.ietf.org/rfc/rfc2898.txt>, свободный. (дата обращения: 25.01.2023)
- 14) RFC 6265: HTTP State Management Mechanism. [Электронный ресурс] / Internet Engineering Task Force (IETF). A. Barth. U.C. Berkeley. April 2011. Режим доступа: <https://www.ietf.org/rfc/rfc6265.txt>, свободный. (дата обращения: 25.01.2023)
- 15) RFC 9110: HTTP Semantics. [Электронный ресурс] / Internet Engineering Task Force (IETF). R. Fielding, Ed. Adobe. M. Nottingham, Ed. Fastly. J. Reschke, Ed. greenbytes. June 2022. Режим доступа: <https://www.ietf.org/rfc/rfc9110.txt>, свободный. (дата обращения: 25.01.2023)
- 16) ECMAScript 2022 Language Specification. [Электронный ресурс] / Ecma International. ECMA-262, 13th edition, June 2022. Режим доступа: <https://262.ecma-international.org/13.0>, свободный. (дата обращения: 26.01.2023)
- 17) Node.js v19.0.0 documentation. [Электронный ресурс] / OpenJS Foundation and Node.js contributors. 17 октября 2022. Режим доступа: <https://nodejs.org/docs/v19.0.0/api>, свободный. (дата обращения: 26.01.2023)
- 18) Fastify documentation (v4.15.0). [Электронный ресурс] / OpenJS Foundation and The Fastify team. 2023. Режим доступа: <https://www.fastify.io/docs/v4.15.x>, свободный. (дата обращения: 02.05.2023).
- 19) PostgreSQL 15.1: Documentation [Электронный ресурс] / The PostgreSQL Global Development Group. 10th November 2022. Режим доступа: <https://www.postgresql.org/docs/current/index.html>, свободный. (дата обращения: 26.01.2023)
- 20) node-postgres documentation. [Электронный ресурс] / Brian Carlson. 2023. Режим доступа: <https://node-postgres.com>, свободный. (дата обращения: 02.05.2023)
- 21) Redis documentation. [Электронный ресурс] / Redis Ltd. 2023. Режим доступа: <https://redis.io/docs>, свободный. (дата обращения: 02.05.2023)
- 22) ioredis. [Электронный ресурс] / Zihua Li. 2023. Режим доступа: <https://www.npmjs.com/package/ioredis>, свободный. (дата обращения: 26.01.2023)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- 23) HTML: Living Standard. [Электронный ресурс] / The Web Hypertext Application Technology Working Group. 26 January 2023. Режим доступа: <https://html.spec.whatwg.org>, свободный. (дата обращения: 26.01.2023)
- 24) CSS Snapshot 2021. [Электронный ресурс] / World Wide Web Consortium (W3C). 31 December 2021. Режим доступа: <https://www.w3.org/TR/css-2021>, свободный. (дата обращения: 26.01.2023)
- 25) EJS: Embedded JavaScript templating. [Электронный ресурс] / Matthew Eernisse. 2022. Режим доступа: <https://ejs.co>, свободный. (дата обращения: 02.05.2023)
- 26) Using Google Charts. [Электронный ресурс] / Google Developers. 2022. Режим доступа: <https://developers.google.com/chart/interactive/docs>, свободный. (дата обращения: 26.01.2023)
- 27) RFC 6455: The WebSocket Protocol. [Электронный ресурс] / Internet Engineering Task Force (IETF). I. Fette, Google, Inc. A. Melnikov, Isode Ltd. December 2011. Режим доступа: <https://www.ietf.org/rfc/rfc6455.txt>, свободный. (дата обращения: 25.01.2023)
- 28) Продукт psql [Электронный ресурс] / The PostgreSQL Global Development Group. Режим доступа: <https://www.postgresql.org/docs/current/app-psql.html>, свободный. (дата обращения: 29.04.2023)
- 29) Продукт pgAdmin [Электронный ресурс] / The pgAdmin Development Team. Режим доступа: <https://www.pgadmin.org/>, свободный. (дата обращения: 29.04.2023)
- 30) Продукт DBeaver [Электронный ресурс] / Режим доступа: <https://dbeaver.io/>, свободный. (дата обращения: 29.04.2023)
- 31) Продукт DataGrip [Электронный ресурс] / JetBrains. Режим доступа: <https://www.jetbrains.com/datagrip/>, свободный. (дата обращения: 29.04.2023)
- 32) Продукт phppgadmin [Электронный ресурс] / Режим доступа: <https://github.com/phppgadmin/phppgadmin>, свободный. (дата обращения: 29.04.2023)
- 33) Продукт pgWeb [Электронный ресурс] / Режим доступа: <https://github.com/sosedoff/pgweb>, свободный. (дата обращения: 29.04.2023)
- 34) JSON Schema: Specification. [Электронный ресурс] / OpenJS Foundation and JSON Schema contributors. Режим доступа: <https://json-schema.org/specification.html>, свободный. (дата обращения: 02.05.2023)
- 35) fastify-plugin. [Электронный ресурс] / OpenJS Foundation and The Fastify team. 2023. Режим доступа: <https://www.npmjs.com/package/fastify-plugin>, свободный. (дата обращения: 02.05.2023).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- 36) @fastify/helmet. [Электронный ресурс] / OpenJS Foundation and The Fastify team. 2023. Режим доступа: <https://www.npmjs.com/package/@fastify/helmet>, свободный. (дата обращения: 02.05.2023).
- 37) @fastify/cors. [Электронный ресурс] / OpenJS Foundation and The Fastify team. 2023. Режим доступа: <https://www.npmjs.com/package/@fastify/cors>, свободный. (дата обращения: 02.05.2023).
- 38) @fastify/rate-limit. [Электронный ресурс] / OpenJS Foundation and The Fastify team. 2023. Режим доступа: <https://www.npmjs.com/package/@fastify/rate-limit>, свободный. (дата обращения: 02.05.2023).
- 39) @fastify/formbody. [Электронный ресурс] / OpenJS Foundation and The Fastify team. 2023. Режим доступа: <https://www.npmjs.com/package/@fastify/formbody>, свободный. (дата обращения: 02.05.2023).
- 40) @fastify/websocket. [Электронный ресурс] / OpenJS Foundation and The Fastify team. 2023. Режим доступа: <https://www.npmjs.com/package/@fastify/websocket>, свободный. (дата обращения: 02.05.2023).
- 41) @fastify/compress. [Электронный ресурс] / OpenJS Foundation and The Fastify team. 2023. Режим доступа: <https://www.npmjs.com/package/@fastify/compress>, свободный. (дата обращения: 02.05.2023).
- 42) @fastify/static. [Электронный ресурс] / OpenJS Foundation and The Fastify team. 2023. Режим доступа: <https://www.npmjs.com/package/@fastify/static>, свободный. (дата обращения: 02.05.2023).
- 43) @fastify/view. [Электронный ресурс] / OpenJS Foundation and The Fastify team. 2023. Режим доступа: <https://www.npmjs.com/package/@fastify/view>, свободный. (дата обращения: 02.05.2023).
- 44) html-minifier. [Электронный ресурс] / Juriy "kangax" Zaytsev. 2019. Режим доступа: <https://www.npmjs.com/package/html-minifier>, свободный. (дата обращения: 02.05.2023).
- 45) @fastify/circuit-breaker. [Электронный ресурс] / OpenJS Foundation and The Fastify team. 2023. Режим доступа: <https://www.npmjs.com/package/@fastify/circuit-breaker>, свободный. (дата обращения: 02.05.2023).
- 46) @fastify/under-pressure. [Электронный ресурс] / OpenJS Foundation and The Fastify team. 2023. Режим доступа: <https://www.npmjs.com/package/@fastify/under-pressure>, свободный. (дата обращения: 02.05.2023).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- 47) @fastify/cookie. [Электронный ресурс] / OpenJS Foundation and The Fastify team. 2023. Режим доступа: <https://www.npmjs.com/package/@fastify/cookie>, свободный. (дата обращения: 02.05.2023).
- 48) Ajv JSON schema validator. [Электронный ресурс] / Evgeny Poberezkin. 2023. Режим доступа: <https://ajv.js.org/>, свободный. (дата обращения: 02.05.2023).
- 49) RFC 8927: JSON Type Definition. [Электронный ресурс] / Independent Submission. U. Carion, Segment. November 2020. Режим доступа: <https://www.ietf.org/rfc/rfc8927.txt>, свободный. (дата обращения: 02.05.2023)
- 50) Ajv JSON schema validator. ajv-formats. [Электронный ресурс] / Evgeny Poberezkin. 2022. Режим доступа: <https://ajv.js.org/packages/ajv-formats.html>, свободный. (дата обращения: 02.05.2023).
- 51) bcrypt [Электронный ресурс] / Nicholas Campbell. 2022. Режим доступа: <https://www.npmjs.com/package/bcrypt>, свободный. (дата обращения: 02.05.2023).
- 52) pg-format [Электронный ресурс] / Datalanche, Inc. 2014. Режим доступа: <https://www.npmjs.com/package/pg-format>, свободный. (дата обращения: 02.05.2023).
- 53) node-postgres documentation. Cursor. [Электронный ресурс] / Brian Carlson. 2023. Режим доступа: <https://node-postgres.com/apis/cursor>, свободный. (дата обращения: 02.05.2023)
- 54) postgres-interval. [Электронный ресурс] / Ben Drucker. 2021. Режим доступа: <https://www.npmjs.com/package/postgres-interval>, свободный. (дата обращения: 02.05.2023)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## ПРИЛОЖЕНИЕ 1

## ОПИСАНИЕ МЕТОДОВ ПОЛУЧЕНИЯ ИНФОРМАЦИИ О ЦЕЛЕВОЙ БАЗЕ ДАННЫХ

Для получения необходимой информации о целевой базе данных и её элементах были использованы следующие системные функции и таблицы:

## 1) Информация о базе данных

## a) Системные функции

- i) `current_database` возвращает имя текущей базы данных.
- ii) `version` возвращает версию PostgreSQL текущей базы данных
- iii) `pg_postmaster_start_time` возвращает время запуска сервера текущей базы данных
- iv) `pg_database_size` возвращает объём памяти, занимаемый текущей базой данных на диске

## b) Системные таблицы статистики

- i) `pg_stat_activity`, содержащая информацию об активности процессов сервера базы данных:
  - (1) Столбец `datname` указывает имя базы данных, к которой подключен процесс
  - (2) Столбец `state` указывает статус процесса: активный, бездействующий и др.
- ii) `pg_stat_database`, содержащая информацию о всех базах данных сервера и статистику нагрузки на них
  - (1) Столбец `datname` указывает имя базы данных
  - (2) Столбец `xact_commit` указывает количество успешных запросов к базе данных с момента последнего обновления статистики сервером PostgreSQL
  - (3) Столбец `xact_rollback` указывает количество отменённых запросов к базе данных с момента последнего обновления статистики сервером PostgreSQL
  - (4) Столбец `tup_returned` указывает количество записей базы данных, полученных исполнителем запросов с момента последнего обновления статистики сервером PostgreSQL.
  - (5) Столбец `tup_fetched` указывает количество записей базы данных, полученных исполнителем запросов с момента последнего обновления статистики сервером PostgreSQL с помощью ускоренного поиска по индексу.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- (6) Столбец `tup_inserted` указывает количество обновлённых с момента последнего обновления статистики сервером PostgreSQL записей.
- (7) Столбец `tup_updated` указывает количество добавленных с момента последнего обновления статистики сервером PostgreSQL записей.
- (8) Столбец `tup_deleted` указывает количество удалённых с момента последнего обновления статистики сервером PostgreSQL записей
- (9) Столбец `blks_read` указывает количество запросов блоков диска, сделанных исполнителем с момента последнего обновления статистики сервером PostgreSQL.
- (10) Столбец `blks_hit` указывает количество запросов блоков диска, сделанных исполнителем с момента последнего обновления статистики сервером PostgreSQL, для ответа на которые не потребовалось непосредственное обращение к диску (данные содержались в “кэше” PostgreSQL).
- (11) Столбец `active_time` указывает время, потраченное сервером базы данных на исполнение запросов с момента последнего обновления статистики сервером PostgreSQL.
- (12) Столбец `stats_reset` указывает момент последнего обновления статистики сервером PostgreSQL.

## 2) Информация об элементах базы данных

### а) Системные функции

- i) `pg_total_relation_size` возвращает объём памяти, занимаемый элементом данных на диске
- ii) `format_type`, преобразующая идентификатор типа данных PostgreSQL и его модификатор в человеко-читаемое представление

### б) Системные таблицы информации

- i) `pg_namespace`, содержащая информацию о пространствах имён базы данных
  - (1) Столбец `oid` содержит идентификатор пространства имён
  - (2) Столбец `nsrname` содержит название пространства имён
- ii) `pg_class`, содержащая информацию об элементах базы данных
  - (1) Столбец `oid` указывает идентификатор элемента

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



- (2) Столбец relname указывает имя элемента
  - (3) Столбец relnamespace указывает идентификатор пространства имён, которому принадлежит элемент
  - (4) Столбец relkind указывает тип элемента:
    - (a) r - обычная таблица
  - (5) Столбец reltuples указывает приблизительное количество записей в элементе
- iii) pg\_attribute, содержащая информацию о столбцах таблиц
- (1) Столбец attrelid указывает идентификатор таблицы
  - (2) Столбец attname указывает имя столбца
  - (3) Столбец attnum указывает номер столбца
  - (4) Столбец atttypid указывает идентификатор типа данных столбца:
    - (a) 16 - bool
    - (b) 18 - char
    - (c) 20 - int8
    - (d) 21 - int2
    - (e) 23 - int4
    - (f) 25 - text
    - (g) 700 - float4
    - (h) 701 - float8
    - (i) 1043 - varchar
    - (j) 1082 - date
    - (k) 1083 - time
    - (l) 1114 - timestamp
    - (m) 1184 - timestamptz
    - (n) 1266 - timetz
    - (o) 1700 - numeric

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- (5) Столбец atttypmod указывает модификатор типа данных столбца (для типов CHAR и VARCHAR - длина значения)
  - (6) Столбец attisdropped указывает, был ли столбец удален, но еще не очищен на диске
  - (7) Столбец attidentity указывает, является ли столбец генерируемым идентификатором
  - (8) Столбец attnotnull указывает, разрешены ли в столбце пустые значения
  - (9) Столбец atthasdef указывает, имеет ли столбец стандартное значение
  - (10) Столбец attgenerated указывает, является ли столбец генерируемым
- iv) pg\_constraint, содержащая информацию об ограничениях на значения строк таблицы
- (1) Столбец contype указывает тип ограничения:
    - (a) p - Primary key
    - (b) u - Unique
  - (2) Столбец conrelid указывает идентификатор таблицы, к записям которой относится ограничение
  - (3) Столбец conkey содержит список номеров столбцов таблицы, на которые действует ограничение
- v) pg\_enum, содержащая информацию о всех перечислениях, доступных в базе данных
- (1) Столбец enumtypid указывает идентификатор типа данных
  - (2) Столбец enumlabel указывает название допустимого значения перечисления
- с) Системные таблицы статистики
- i) pg\_stat\_user\_tables, содержащая статистику обращений к пользовательским таблицам базы данных
- (1) Столбец relname указывает имя таблицы
  - (2) Столбец seq\_scan указывает количество произведённых последовательных сканирований таблицы с момента последнего обновления статистики сервером PostgreSQL
  - (3) Столбец idx\_scan указывает количество произведённых поисков по индексу в таблице с момента последнего обновления статистики сервером PostgreSQL

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- (4) Столбец seq\_tup\_read указывает количество записей таблицы, полученных исполнителем запросов с момента последнего обновления статистики сервером PostgreSQL с помощью последовательного сканирования.
- (5) Столбец idx\_tup\_fetch указывает количество записей таблицы, полученных исполнителем запросов с момента последнего обновления статистики сервером PostgreSQL с помощью ускоренного поиска по индексу.
- (6) Столбец n\_tup\_ins указывает количество обновлённых с момента последнего обновления статистики сервером PostgreSQL записей таблицы.
- (7) Столбец n\_tup\_upd указывает количество добавленных с момента последнего обновления статистики сервером PostgreSQL записей таблицы.
- (8) Столбец n\_tup\_del указывает количество удалённых с момента последнего обновления статистики сервером PostgreSQL записей таблицы.
- (9) Столбец n\_live\_tup указывает количество “живых” записей в таблице
- (10) Столбец n\_dead\_tup указывает количество “мёртвых” записей в таблице

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## ПРИЛОЖЕНИЕ 2

## ОПИСАНИЕ И НАЗНАЧЕНИЕ ФАЙЛОВ ПРОГРАММЫ

Исходный код приложения разделён на следующие файлы и директории:

- 1) Файл `package.json` описывает общую информацию о приложении, содержит список зависимостей и набор скриптов для удобства запуска и настройки веб-сервера и внутренней базы данных
- 2) Директория `common` содержит универсальные реализации, которые могут быть использованы различными частями приложения во избежание дублирования исходного кода
  - a) Директория `configs` содержит файлы конфигурации приложения, а также подпрограмму, производящую их считывание
    - i) Директория `ssl` содержит данные тестового SSL-сертификата защищённого соединения веб-приложения
      - (1) Файл `fullchain.pem` содержит полную цепь обмена ключей при установлении защищённого соединения по протоколу HTTPS.
      - (2) Файл `privkey.pem` содержит закрытый ключ шифрования
    - ii) Файл `config.json` является основным файлом конфигурации приложения и содержит все влияющие на поведение программы доступные к изменению настройки
    - iii) Файл `config.js` реализует подпрограмму, которая производит считывание файла конфигурации и его реэкспорт в виде JSON-объекта для удобного использования другими частями приложения.
  - b) Директория `postgreSQL` содержит подпрограммы-утилиты для удобного взаимодействия с базами данных PostgreSQL
    - i) Файл `postgreSQL.js` реализует класс, используемый для организации взаимодействия с базами данных PostgreSQL, а также создаёт соединения с целевой и с внутренней базами данных.
  - c) Директория `redis` содержит подпрограммы-утилиты для удобного взаимодействия с хранилищем данных Redis

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- i) Файл `redis.js` реализует класс, используемый для реализации взаимодействия с хранилищем данных Redis, а также создаёт соединение для использования другими частями приложения.
- d) Директория `utils` реализует ряд методов, которые могут быть удобны при разработке основной части приложения
  - i) Файл `Utils.js` описывает реализованное в виде класса пространство имён `Utils`, содержащее ряд универсальных утилит для удобства разработки других подпрограмм приложения
  - ii) Директория `implementations` содержит реализации утилит, определённых в файле `Utils.js`
    - (1) Директория `Interval` содержит реализацию утилиты `Interval` для удобства представления промежутков времени в программе
      - (a) Файл `Interval.js` реализует класс `Interval`, представляющий интервал времени
    - (2) Директория `object_manipulation_by_path` содержит ряд методов для удобства управления полями объекта JavaScript на любом уровне вложенности по массиву строк, который описывает путь до необходимого значения
      - (a) Файлы `get_field_at_path.js` и `set_field_at_path.js` реализуют одноимённые функции, позволяющие получать и изменять значения полей объекта JavaScript описанным образом.
- 3) В директории PostgreSQL представлены подпрограммы для управления внутренней базой данных приложения.
  - a) Файл `create.js` реализует подпрограмму для создания внутренней базы данных приложения
  - b) Файл `setup.sql` содержит SQL-запросы настройки (в т.ч. создания таблиц) внутренней базы данных приложения
  - c) Файл `setup.js` реализует подпрограмму для настройки внутренней базы данных приложения при помощи запросов, описанных в файле `setup.sql`
  - d) Файл `fill.js` реализует подпрограмму для заполнения внутренней базы данных приложения начальными значениями
  - e) Файл `delete.js` реализует подпрограмму для удаления внутренней базы данных приложения
- 4) В директории `website` представлена реализация основного сервиса - веб-сервера.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- a) Директория `plugins` содержит исходные коды алгоритмов первичной обработки, аутентификации и авторизации запросов пользователей, реализованных как модули фреймворка `fastify`.
- i) Файл `security.js` реализует модуль, добавляющий дополнительные стадии обработки запроса, которые позволяют защитить веб-сервер и пользователей от различных типичных атак, что повышает безопасность приложения.
  - ii) Файл `request.js` реализует модуль, позволяющий веб-серверу корректно обрабатывать запросы различных типов с различными форматами содержимого.
  - iii) Файл `response.js` реализует модуль, позволяющий веб-серверу корректно и эффективно отвечать на запросы пользователей: генерировать исходные коды веб-страниц по шаблонам, выбирать понятные сообщения об обнаруженных ошибках, а также сжимать и кэшировать ответы.
  - iv) Файл `utility.js` реализует модуль, позволяющий отслеживать состояние веб-сервера во время его работы и отлавливать критические ошибки при обработке запросов.
  - v) Файл `oauth.js` реализует модуль, производящий аутентификацию и авторизацию запросов пользователей.
  - vi) Файл `routes.js` реализует модуль, производящий сбор и регистрацию точек доступа к веб-серверу.
  - vii) Файл `other.js` реализует модуль, добавляющий к веб-серверу необходимые утилиты, не реализованные другими модулями.
- b) Директория `routes` содержит исходные коды методов непосредственной обработки запросов пользователей и генерации ответов на них.
- i) Директория `utils` содержит ряд функций, которые могут использоваться веб-сервером при обработке нескольких различных запросов
  - (1) Директория `database` содержит функции, производящие получение определённой информации из указанной базы данных
    - (a) Файлы `get_database_info.js`, `get_table_layout.js` и `get_table_name.js` реализуют одноимённые функции получения соответствующей информации из указанной базы данных.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- (2) Директория `schemas` содержит значения, используемые для удобной валидации входных данных веб-сервером.
- (a) Файл `types.json` содержит описания часто используемых типов вводимых значений.
  - (b) Файл `schemas.js` содержит описания типичных форматов входных данных в виде схем JSON Schema<sup>[34]</sup>, которые могут быть переданы фреймворку `fastify` при регистрации точки доступа для произведения эффективной валидации входных данных.
- (3) Файл `compile_websocket_handler.js` реализует одноимённую функцию для удобной обработки сообщений, принятых веб-сервером по протоколу `WebSocket`.
- ii) Директория `websocket_data_routes` реализует методы непосредственной обработки сообщений, полученных сервером по протоколу `WebSocket`.
- (1) Директория `utils` содержит методы, которые могут использоваться веб-сервером при обработке нескольких различных сообщений
- (a) Файл `table_rows_query.js` реализует функцию получения SQL-запроса записей указанной таблицы базы данных.
- (2) Файлы `database_data.js`, `table_data.js`, `table_rows.js` и `logs.js` реализуют одноимённые функции обработки сообщений соответствующих типов.
- iii) Файл `main.js` содержит функции непосредственной обработки запросов к публичным точкам доступа веб-сервера.
- iv) Файл `database.js` содержит функции непосредственной обработки запросов панели управления: запросов страниц панели управления, а также запросов установления соединения по протоколу `WebSocket` и сохранения внесённых пользователем изменений.
- v) Файл `admin.js` содержит функции непосредственной обработки запросов панели администратора: запросов страниц “Аккаунты” и “История”, а также запросов создания, редактирования и удаления аккаунтов.
- с) Директория `static` содержит статические файлы клиентской части приложения
- i) Директория `css` содержит CSS-описания стилей элементов, используемых на страницах приложения
- (1) Файл `general.css` описывает основные используемые стили элементов веб-страниц

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ii) Директория images содержит картинки, отображаемые в приложении пользователю

(1) Директория favicon содержит значок приложения в нескольких форматах

- (a) Файл favicon.svg представляет значок приложения в векторном формате
- (b) Файл favicon.ico представляет значок приложения в формате ico
- (c) Файл favicon\_48x48.png представляет значок приложения размером 48x48 пикселей в формате png
- (d) Файл favicon\_96x96.png представляет значок приложения размером 96x96 пикселей в формате png
- (e) Файл favicon\_144x144.png представляет значок приложения размером 144x144 пикселя в формате png
- (f) Файл favicon\_192x192.png представляет значок приложения размером 192x192 пикселя в формате png
- (g) Файл apple-touch-icon.png представляет значок приложения в формате png, в первую очередь предназначенный для отображения на устройствах от компании Apple

iii) Директория js содержит скрипты клиентской части приложения, описывающие алгоритмы работы веб-страниц и их элементов

(1) Директория utils содержит универсальные скрипты, используемые несколькими страницами

- (a) Файл general.js реализует алгоритмы работы выпадающих окон и меню, применимые на всех страницах приложения.
- (b) Файл chart.js реализует класс Chart, используемый для отображения графиков на страницах приложения
- (c) Файл table.js реализует класс Table, используемый для отображения таблиц данных на страницах приложения

(2) Директория admin содержит скрипты, используемые на страницах панели администратора

- (a) Файл logs.js описывает алгоритмы работы страницы “История”

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



- (3) Файл database.js описывает алгоритмы работы страницы “База данных”
- (4) Файл table.js описывает алгоритмы работы страниц, соответствующих таблицам целевой базы данных
- d) Директория views содержит EJS-шаблоны исходных кодов веб-страниц приложения
- i) Директория general содержит шаблоны элементов, которые повторяются на нескольких страницах приложения
- (1) Файл head.ejs описывает “голову” страниц приложения, где представлены метаданные страницы, подключаются основные шрифты, стили и скрипты, а также указывается значок приложения.
- (2) Файл header.ejs описывает шаблон верхнего колонтитула страниц панели управления.
- (3) Файл input.ejs реализует шаблон полей ввода, отображаемых в таблицах страниц панели управления
- (4) Файл nav.ejs представляет шаблон меню навигации панели управления
- ii) Файл error.ejs описывает шаблон исходного кода страницы ошибки, отображаемой пользователю при невозможности успешного выполнения запроса
- iii) Директория admin содержит шаблоны исходных кодов страниц панели администратора
- (1) Файл accounts.ejs описывает шаблон исходного кода страницы “Аккаунты”
- (2) Файл logs.ejs описывает шаблон исходного кода страницы “История”
- iv) Файл database.ejs описывает шаблон исходного кода страницы “База данных”
- v) Файл oauth.ejs описывает шаблон исходного кода начальной страницы приложения
- vi) Файл table.ejs описывает шаблон исходного кода страниц, соответствующих таблицам целевой базы данных
- e) Файл index.js реализует точку входа серверной части приложения: инициализирует и запускает веб-сервер, подключая все необходимые модули.
- f) Файлы robots.txt и sitemap.xml предназначены для будущего размещения соответственно параметров индексирования страниц приложения роботами поисковых систем, а также “карты сайта” для улучшения навигации между страницами приложения.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## ПРИЛОЖЕНИЕ 3

## ОПИСАНИЕ ВНУТРЕННЕЙ БАЗЫ ДАННЫХ ПРИЛОЖЕНИЯ

Внутренняя база данных приложения состоит из двух таблиц: users (см. табл. 1), которая содержит информацию о зарегистрированных аккаунтах пользователей, и logs (см. табл. 2), предназначенная для сохранения истории запросов, сделанных всеми пользователями приложения к целевой базе данных. Для удобства организации хранения данных также создан тип REQUEST\_TYPE, использующийся для указания типа запроса, соответствующего записи таблицы logs.

Таблица 1 - описание и назначение столбцов таблицы users внутренней базы данных

Наименование	Тип	Ограничения	Назначение
id	SMALLINT	– GENERATED ALWAYS AS IDENTITY – PRIMARY KEY	Идентификатор, генерируемый автоматически при создании аккаунта
login	VARCHAR(100)	– NOT NULL – UNIQUE	Имя аккаунта
password	CHAR(60)	– NOT NULL	Зашифрованный пароль для доступа к приложению
read	BOOL	– NOT NULL – DEFAULT FALSE	Флаг, указывающий на наличие уровня доступа “чтение”
insert	BOOL	– NOT NULL – DEFAULT FALSE	Флаг, указывающий на наличие уровня доступа “вставка”
update	BOOL	– NOT NULL – DEFAULT FALSE	Флаг, указывающий на наличие уровня доступа “редактирование”
delete	BOOL	– NOT NULL – DEFAULT FALSE	Флаг, указывающий на наличие уровня доступа “удаление”
admin	BOOL	– NOT NULL – DEFAULT FALSE	Флаг, указывающий на наличие уровня доступа “администрирование”
permissions	CHAR(5)	– GENERATED ALWAYS AS ( ... ) STORED	Код уровня доступа аккаунта, генерируемый при каждом обращении

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 2 - описание и назначение столбцов таблицы logs внутренней базы данных

Наименование	Тип	Ограничения	Назначение
id	SMALLINT	– GENERATED ALWAYS AS IDENTITY – PRIMARY KEY	Идентификатор записи таблицы, генерируемый автоматически при её добавлении
date	TIMESTAMPZ	– NOT NULL – DEFAULT NOW()	Дата выполнения описываемого запроса
type	REQUEST_TYPE	– NOT NULL	Тип описываемого запроса
data	TEXT	– NOT NULL	Данные, с которыми работал запрос
userid	SMALLINT	– REFERENCES users(id) MATCH FULL ON DELETE SET NULL ON UPDATE CASCADE	Идентификатор пользователя, который инициировал запрос
query	TEXT	– NOT NULL	Текст сгенерированного SQL-запроса
query_params	TEXT	– NOT NULL – DEFAULT '[' ]'	Параметры сгенерированного SQL-запроса

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## ПРИЛОЖЕНИЕ 4

## ОПИСАНИЕ И НАЗНАЧЕНИЕ ВНЕШНИХ ЗАВИСИМОСТЕЙ

Для повышения эффективности и безопасности приложения реализация полагается на ряд npm-зависимостей, упрощающих разработку.

Таблица 1 - описание и назначение внешних зависимостей

Наименование	Версия	Назначение
fastify <sup>[18]</sup>	4.15.0	Фреймворк, использованный для реализации веб-сервера
fastify-plugin <sup>[35]</sup>	4.5.0	Расширение фреймворка fastify, позволяющее реализовывать и поддерживать независимые модули.
@fastify/helmet <sup>[36]</sup>	10.1.0	Расширение фреймворка fastify, реализующее установку основных заголовков ответа, повышающих безопасность приложения
@fastify/cors <sup>[37]</sup>	8.2.1	Расширение фреймворка fastify, реализующее установку веб-сервером заголовков механизма CORS для повышения безопасности.
@fastify/rate-limit <sup>[38]</sup>	7.6.0	Расширение фреймворка fastify, реализующее ограничение количества запросов, которые может отправить пользователь к серверу за указанный промежуток времени, для защиты от определённых типов атак.
@fastify/formbody <sup>[39]</sup>	7.4.0	Расширение фреймворка fastify, реализующее обработку запросов, указывающих тело в формате application/x-www-form-urlencoded
@fastify/websocket <sup>[40]</sup>	7.2.0	Расширение фреймворка fastify, реализующее установление соединения по протоколу WebSocket
@fastify/compress <sup>[41]</sup>	6.2.1	Расширение фреймворка fastify, реализующее сжатие отправляемых веб-сервером ответов, а также распаковку получаемых запросов
@fastify/static <sup>[42]</sup>	6.10.1	Расширение фреймворка fastify, реализующее эффективную отправку статических файлов пользователям.
@fastify/view <sup>[43]</sup>	7.4.1	Расширение фреймворка fastify, реализующее генерацию исходных кодов веб-страниц по их шаблонам при отправке пользователям
ejs <sup>[25]</sup>	3.1.9	Реализует обработку EJS-шаблонов и генерацию исходных кодов веб-страниц по ним. Используется для корректной настройки библиотеки @fastify/view

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 1

html-minifier <sup>[44]</sup>	4.0.0	Производит минификацию исходных кодов веб-страниц перед отправкой пользователю для уменьшения их размера, что повышает эффективность библиотеки @fastify/view
@fastify/circuit-breaker <sup>[45]</sup>	3.2.0	Расширение фреймворка fastify, обнаруживающее ошибки, связанные с бесконечной обработкой запросов из-за циклического вызова обработчиков.
@fastify/under-pressure <sup>[46]</sup>	8.2.0	Расширение фреймворка fastify, позволяющее отслеживать количество ресурсов, используемых веб-сервером, и автоматически отклонять запросы, если сервер перегружен, во избежание нарушения его работы.
@fastify/cookie <sup>[47]</sup>	8.3.0	Расширение фреймворка fastify, реализующее чтение, создание, удаление и подпись cookie-файлов приложением.
ajv <sup>[48]</sup>	8.12.0	Позволяет производить валидацию, сериализацию и парсинг JSON-объектов по указанной в формате JSON Schema <sup>[34]</sup> или JSON Type Definition <sup>[49]</sup> схеме. Используется фреймворком fastify для валидации входных данных точек доступа, а также приложением для, помимо прочего, проверки корректности сообщений, получаемых по протоколу WebSocket.
ajv-formats <sup>[50]</sup>	2.1.1	Расширяет возможности ajv путём добавления дополнительных типов данных.
bcrypt <sup>[51]</sup>	5.1.0	Реализует алгоритмы шифрования данных, используемые приложением для безопасного хранения и проверки паролей пользователей.
ioredis <sup>[22]</sup>	5.3.2	Реализует взаимодействие с хранилищем данных Redis
pg <sup>[20]</sup>	8.10.0	Реализует взаимодействие с базами данных PostgreSQL
pg-format <sup>[52]</sup>	1.0.4	Реализует функцию, аналогичную функции format базы данных PostgreSQL, которая повышает безопасность приложения, предотвращая атаки типа SQL injection.
pg-cursor <sup>[53]</sup>	2.9.0	Предоставляет возможность получения результатов выполнения SQL-запросов постранично с помощью курсоров PostgreSQL
postgres-interval <sup>[54]</sup>	4.0.0	Используется приложением для реализации класс Interval “пространства имён” Utils, а также библиотекой pg для представления и работы со значениями типа INTERVAL при их получении в результате выполнения запросов.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## ПРИЛОЖЕНИЕ 5

## ОПИСАНИЕ И НАЗНАЧЕНИЕ ПАРАМЕТРОВ ВНЕШНИХ ЗАВИСИМОСТЕЙ

Таблица 1 - описание и назначение параметров фреймворка fastify

Наименование	Значение	Назначение
http2	true	Позволяет веб-серверу использовать вторую версию протокола HTTP для установления более эффективных соединений с клиентами.
https.allowHTTP1	true	Позволяет веб-серверу использовать первую версию протокола HTTP, если клиент не поддерживает использование более новых версий.
https.key	Содержимое файла, указываемого значением website.ssl_key файла конфигурации	Указание закрытого ключа шифрования SSL-сертификата защищённого соединения
https.cert	Содержимое файла, указываемого значением website.ssl_cert файла конфигурации	Указание полной цепи обмена ключей SSL-сертификата защищённого соединения
forceCloseConnections	true	Требуется обязательного закрытия всех соединений с клиентами при завершении работы веб-сервера
ignoreTrailingSlash	true	Требуется игнорирования веб-сервером лишних /-символов в конце адреса точки доступа
ignoreDuplicateSlashes	true	Требуется игнорирования веб-сервером повторяющихся /-символов в адресах точек доступа
logger	true, если приложение запущено в режиме тестирования. Иначе false	Включает вывод информации о входящих запросах в стандартный поток вывода в режиме тестирования приложения
disableRequestLogging	true	Отключает вывод излишней информации о входящих запросах
ajv.plugins	ajv_formats в полном режиме	Подключает к используемому валидатору входных данных библиотеку ajv_formats
ajv.customOptions.removeAdditional	true	Требуется удаления всех лишних значений из входных данных при валидации

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 1

ajv.customOptions.useDefaults	true	Позволяет клиенту не передавать поля входных данных, если требуется их стандартное значение
ajv.customOptions.coerceTypes	true	Позволяет клиенту передавать значения неверного типа данных, если он тривиально преобразуем в необходимый

Таблица 2 - описание и назначение параметров библиотеки @fastify/helmet

Наименование	Значение	Назначение
global	true	Подключение модуля ко всем входящим запросам
enableCSPNonces	true	Генерация nonce-значений Content-Security-Policy
contentSecurityPolicy	-	Установка заголовка ответа Content-Security-Policy для защиты от XSS-атак
crossOriginEmbedderPolicy	true	Установка заголовка ответа Cross-Origin-Embedder-Policy для защиты от XSS-атак
crossOriginOpenerPolicy.policy	same-origin	Установка заголовка ответа Cross-Origin-Opener-Policy для защиты от XSS-атак
crossOriginResourcePolicy.policy	same-origin	Установка заголовка ответа Cross-Origin-Resource-Policy для защиты от XSS-атак
expectCt	{ maxAge: 24 * 60 * 60, enforce: true, reportUri: "/ct-violation-report" }	Установка заголовка ответа Expect-CT для защиты от использования некорректного SSL-сертификата
referrerPolicy.policy	strict-origin-when-cross-origin	Установка заголовка ответа Referrer-Policy во избежание утечки конфиденциальной информации посредством заголовка запроса Referer
hsts	{ maxAge: 7 * 24 * 60 * 60, includeSubDomains: false, preload: true }	Установка заголовка ответа Strict-Transport-Security для запрета взаимодействия по незащищённому протоколу
noSniff	true	Установка заголовка ответа X-Content-Type-Options, запрещающего браузеру "угадывать" тип содержимого ответа
originAgentCluster	true	Установка заголовка ответа Origin-Agent-Cluster во избежание утечки конфиденциальной информации

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 2

dnsPrefetchControl.allow	true	Установка заголовка ответа X-DNS-Prefetch-Control для повышения скорости соединения путём предобработки DNS-зоны
frameguard.action	SAMEORIGIN	Установка заголовка ответа X-Frame-Options для запрета встраивания страниц приложения на другие сайты во избежание утечек информации.
hidePoweredBy	true	Удаление заголовка ответа X-Powered-By во избежание атак, направленных специально на нарушение работы веб-сервера, реализованного с использованием фреймворка fatify
xssFilter	true	Удаление заголовка ответа X-XSS-Protection, который работает некорректно в некоторых браузерах

Таблица 3 - описание и назначение параметров библиотеки @fastify/cors

Наименование	Значение	Назначение
origin	Адрес веб-сервера	Установка заголовка ответа Access-Control-Allow-Origin для повышения безопасности приложения
methods	'GET,POST'	Установка заголовка ответа Access-Control-Allow-Methods для разрешения только GET и POST запросов к серверу
maxAge	300	Установка заголовка ответа Access-Control-Max-Age во избежание многократного повторения “preflight” запросов

Таблица 4 - описание и назначение параметров библиотеки @fastify/rate-limit

Наименование	Значение	Назначение
global	true	Подключение модуля ко всем входящим запросам
max	1000000 в режиме тестирования. Иначе 100	Максимальное количество запросов, которое может отправить пользователь за указанный интервал
timeWindow	60000	Интервал сброса информации о количестве запросов, отправленных пользователем
ban	1500000 в режиме тестирования. Иначе 150	Максимальное количество запросов, которое может отправить пользователь до получения 403 Forbidden

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



Продолжение таблицы 4

continueExceeding	true	Обновление таймера, если пользователь продолжает отправлять запросы после превышения лимита
addHeadersOnExceeding	-	Установка заголовков ответа для сообщения клиенту об ошибке после превышения лимита запросов
addHeaders	-	Установка заголовков ответа для сообщения клиенту об ошибке до превышения лимита запросов
cache	10000	Размер хранилища данных
redis	-	Соединение с хранилищем данных Redis, которое используется библиотекой для подсчета запросов
nameSpace	'website-rate-limit-'	Префикс имени добавляемых в Redis значений
skipOnError	true	Игнорирование ошибок Redis, если такие возникают
onExceeding	-	Функция, которая выводит в стандартный поток вывода информацию о превышении лимита запросов пользователем

Таблица 5 - описание и назначение параметров библиотеки @fastify/static

Наименование	Значение	Назначение
root	-	Путь к директории статических файлов
prefix	"/static"	Префикс имён точек доступа, предназначенных для отправки статических файлов
acceptRanges	true	Установка заголовка ответа Accept-Ranges для реализации возможности отправки ответа “по частям” для повышения эффективности приложения
cacheControl	true	Установка заголовка ответа Cache-Control для сохранения ответа браузером на стороне клиента во избежание повторения запросов без необходимости
dotfiles	'ignore'	Запрет отправки клиентам скрытых файлов, имя которых начинается с точки
etag	true	Установка заголовка ответа ETag для уменьшения вероятности использования браузером устаревших ответов
extensions	false	Запрет изменения расширения файлов, если запрошенное не найдено непосредственно

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 5

immutable	false	Указание клиенту, что сохраняемое значение не является неизменяемым, для уменьшения вероятности использования браузером устаревших ответов
lastModified	true	Установка заголовка ответа Last-Modified для уменьшения вероятности использования браузером устаревших ответов
maxAge	0 в режиме тестирования, 604800000 иначе	Указание максимального времени, на которое браузер может запомнить ответ, для уменьшения вероятности использования устаревших ответов
preCompressed	true	Использование алгоритмов сжатия данных перед отправкой, если это возможно
wildcard	false	Явное добавление точек доступа для всех файлов в директории статических файлов во избежание утечек информации с сервера
index	false	Отключение отправки стандартного файла вместо ошибки 404
allowedPath	-	Указание функции, которая дополнительно проверяет, разрешена ли отправка выбранного файла, во избежание утечек информации с сервера

Таблица 6 - описание и назначение параметров библиотеки @fastify/view

Наименование	Значение	Назначение
engine.ejs	-	Используемая библиотека шаблонов - ejs
production	false в режиме тестирования, иначе true	Сохранение сгенерированных исходных кодов веб-страниц для повышения эффективности приложения
propertyName	"render"	Наименование метода для генерации и отправки исходных кодов веб-страниц по шаблонам, который будет добавлен к объекту ответа
root	"./website/views/"	Путь к директории с шаблонами
defaultContext	-	Набор переменных, которые доступны шаблонам
options	-	Параметры, с которыми следует использовать библиотеки ejs и html-minifier при обработке шаблонов для обеспечения генерации эффективно-сжатого, но корректного исходного кода

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 7 - описание и назначение параметров библиотеки @fastify/cookie

Наименование	Значение	Назначение
secret	Значение website.secret файла конфигурации	Секретный ключ шифрования, используемый при подписи cookie-файлов

Таблица 8 - описание и назначение параметров библиотеки bcrypt

Наименование	Значение	Назначение
saltRounds	Значение bcrypt.saltRound файла конфигурации	Количество повторений шифрования для обеспечения надёжности алгоритма

Таблица 9 - описание и назначение параметров библиотеки @fastify/circuit-breaker

Наименование	Значение	Назначение
threshold	5	Максимальное количество ошибок, которые могут произойти при обработке запроса, до его завершения с ошибкой
timeout	10000	Максимальная длительность обработки запроса до его завершения с ошибкой
onCircuitOpen	-	Функция, выводящая в стандартный поток вывода сообщение об ошибке и завершающая обработку запроса со статусом 508
onTimeout	-	Функция, выводящая в стандартный поток вывода сообщение об ошибке и завершающая обработку запроса со статусом 504

Таблица 10 - описание и назначение параметров библиотеки @fastify/under-pressure

Наименование	Значение	Назначение
maxEventLoopDelay	250	Максимальная допустимая задержка цикла событий Node.JS
maxHeapUsedBytes	1048576000	Максимальное количество используемой приложением памяти в пространстве heap
maxRssBytes	1048576000	Максимальное количество используемой приложением памяти в пространстве rss
maxEventLoopUtilization	0.8	Максимальная допустимая утилизация цикла событий Node.JS
message	-	Сообщение, отправляемое в ответ на запросы, которые не могут быть успешно обработаны из-за перегрузки сервера
retryAfter	60	Установка значения заголовка ответа Retry-After, если сервер не может обработать запрос из-за высокой нагрузки.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 10

pressureHandler	-	Функция, печатающая в стандартный поток вывода информацию об ошибке и отправляющая пользователю ответ со статусом 503.
exposeStatusRoute	-	Конфигурация точки доступа просмотра статистики нагрузки на веб-сервер
healthCheck	-	Функция получения информации о нагрузке на веб-сервер, которая отправляется в ответ на запрос к соответствующей точке доступа

Таблица 11 - описание и назначение параметров библиотеки ioredis

Наименование	Значение	Назначение
connectionName	Значение поля application файла конфигурации	Имя соединения с хранилищем данных
enableAutoPipelining	true	Возможность отправки несколько команд одновременно
port, host, database, password	Значения соответствующих полей объекта redis файла конфигурации	Реквизиты подключения к Redis

Таблица 12 - описание и назначение параметров библиотеки pg

Наименование	Значение	Назначение
parseInputDatesAsUTC	true	Указание, что часовой пояс всех дат, получаемые из баз данных postgresSQL, не должен быть изменен
application_name	Значение поля application файла конфигурации	Имя соединения с базой данных
host, port, database, user, password	Значения соответствующих полей файла конфигурации	Реквизиты подключения к PostgreSQL

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## ПРИЛОЖЕНИЕ 6

## ОПИСАНИЕ И НАЗНАЧЕНИЕ ФУНКЦИЙ

Таблица 1 - описание и назначение функций подпрограммы common

Наименование	Исходный код в директории common/utills/implementations	Аргументы	Возвращаемое значение	Назначение
get_field_at_path	object_manipulation_by_path/get_field_at_path.js	<ul style="list-style-type: none"> <li>obj - объект, значение поля которого нужно получить</li> <li>path - массив, описывающий путь к полю, значение которого нужно получить</li> </ul>	Значение запрошенного поля	Получение значения поля, расположенного в объекте по указанному пути
set_field_at_path	object_manipulation_by_path/set_field_at_path.js	<ul style="list-style-type: none"> <li>obj - изменяемый объект</li> <li>path - массив, описывающий путь к полю, значение которого нужно изменить</li> <li>value - новое значение поля</li> </ul>	Изменённый объект	Изменение значения поля, расположенного в объекте по указанному пути

Таблица 2 - описание и назначение функций, реализующих модули веб-сервера fastify

Наименование	Исходный код в директории website/plugins	Аргументы	Назначение
register_security_plugins	security.js	<ul style="list-style-type: none"> <li>app - веб-сервер fastify</li> <li>options - параметры модуля</li> <li>done - функция, которую необходимо вызвать по окончании работы</li> </ul>	Подключение расширений helmet, cors и rate-limit
register_request_plugins	request.js		Подключение расширений formbody и websocket
register_response_plugins	response.js		Подключение расширений compress, static, view и обработка ошибок
register_utility_plugins	utility.js		Подключение расширений circuit-breaker и under-pressure
register_oauth	oauth.js		Аутентификация и авторизация запросов
register_routes	routes.js		Регистрация точек доступа к веб-серверу из файлов директории routes
register_other_plugins	other.js		Подключение расширения cookie

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 3 - описание и назначение “хуков”, добавленных в цепочку обработки входящих запросов

Наименование	Исходный код в директории website/plugins	Аргументы	Назначение
onRequest	response.js	<ul style="list-style-type: none"> <li>– req - объект запроса</li> <li>– res - объект ответа</li> <li>– next - функция, которую необходимо вызвать по окончании работы</li> </ul>	Установка ссылок между объектами запроса и ответа для удобства обработки и генерации исходных кодов веб-страниц по шаблонам
onRequest	security.js		Установка заголовков ответа Report-To и Reporting-Endpoints для получения уведомлений от браузера пользователя о нарушениях политики безопасности, заданной заголовком Content-Security-Policy
preValidation	oauth.js	– req - объект запроса	Аутентификация запросов пользователей
preHandler		– res - объект ответа	Авторизация запросов пользователей

Таблица 4 - описание и назначение функций, обрабатывающих входящие запросы и сообщения

Наименование	Исходный код в директории website/routes	Аргументы	Возвращаемое значение	Назначение
get_main	main.js	<ul style="list-style-type: none"><li>– req - объект запроса</li><li>– res - объект ответа</li></ul>	Объект ответа, для которого произведена отправка результата обработки	Обработка запроса начальной страницы
post_oauth				Обработка запроса на вход в аккаунт
get_logout				Обработка запроса на выход из аккаунта
get_database	database.js			Обработка запроса страницы “База данных”
get_table				Обработка запроса страницы, соответствующей таблице целевой БД
post_data				Обработка запроса на изменение таблицы целевой БД

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 4

get_admin_accounts	admin.js	<ul style="list-style-type: none"><li>– req - объект запроса</li><li>– res - объект ответа</li></ul>	Объект ответа, для которого произведена отправка результата обработки	Обработка запроса страницы “Аккаунты”	
delete_admin_accounts				Обработка запроса на удаление аккаунта	
create_admin_accounts				Обработка запроса на создание аккаунта	
edit_admin_accounts				Обработка запроса на изменение аккаунта	
get_admin_logs				Обработка запроса страницы “История”	
websocket_data	database.js	<ul style="list-style-type: none"><li>– connection - объект соединения</li><li>– req - объект запроса</li></ul>	Отсутствует	Обработка запроса на установление двустороннего соединения	
database_data	websocket_data_routes/database_data.js	<ul style="list-style-type: none"><li>– msg - содержимое полученного сообщения</li><li>– socket - объект соединения</li><li>– req - объект запроса</li></ul>		Обработка сообщения на получение информации о целевой БД	
table_data	websocket_data_routes/table_data.js			Обработка сообщения на получение информации о таблице целевой БД	
logs	websocket_data_routes/logs.js			Обработка сообщения на получение страницы истории запросов	
table_rows	websocket_data_routes/table_rows.js			Обработка сообщения на получение страницы записей таблицы целевой БД	
transform_data		<ul style="list-style-type: none"><li>– data - объект</li></ul>		Обработанный объект	Обработка массива записей таблицы целевой БД для отправки клиенту

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 4

table_rows_query_conditions	websocket_data_routes/utils/table_rows_query.js	<ul style="list-style-type: none"> <li>– filters - массив объектов-параметров фильтрации:</li> <li>– name - название столбца, по которому должен работать фильтр</li> <li>– value - искомое значение столбца</li> <li>– comparison - способ сравнения значений: полное совпадение (exact) или подстрока (substring)</li> </ul>	Объект, содержащий два поля: <ul style="list-style-type: none"> <li>– query - текст части запроса</li> <li>– params - параметры части запроса</li> </ul>	Строит часть SQL-запроса на получение страницы записей таблицы базы данных, отвечающую за фильтрацию записей
table_rows_query_ordering		<ul style="list-style-type: none"> <li>– sorts - массив объектов-параметров сортировки:</li> <li>– name - название столбца, по которому нужно сортировать</li> <li>– order - порядок: возрастающий (asc) или убывающий (desc)</li> </ul>	Объект, содержащий два поля: <ul style="list-style-type: none"> <li>– query - текст части запроса</li> <li>– params - параметры части запроса</li> </ul>	Строит часть SQL-запроса на получение страницы записей таблицы базы данных, отвечающую за сортировку записей
table_rows_query		<ul style="list-style-type: none"> <li>– columns - строка, описывающая необходимые столбцы</li> <li>– table - имя таблицы БД</li> <li>– filters - массив параметров фильтрации</li> <li>– sorts - массив параметров сортировки</li> </ul>	Объект, содержащий два поля: <ul style="list-style-type: none"> <li>– query - текст запроса</li> <li>– params - параметры запроса</li> </ul>	Строит SQL-запрос на получение указанных столбцов записей указанной таблицы базы данных с применением заданных фильтров и порядка сортировки

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



Продолжение таблицы 4

compile_websocket_handler	utils/compile_websocket_handler.js	– folder - имя папки директории website/routes, в которой размещены функции обработки сообщений	Функция обработки сообщений, полученных по протоколу WebSocket	Сбор информации о типах сообщений и создание функции валидации и обработки входных данных.
get_database_info	utils/database/get_database_info.js	– DatabaseConnection - соединение с базой данных	Объект, содержащий два поля: – database_name - имя базы данных – tables - массив объектов, описывающих таблицы (id и name)	Получение имени базы данных и списка её таблиц для отображения на страницах панели управления
get_table_layout	utils/database/get_table_layout.js	– DatabaseConnection - соединение с базой данных – type - тип задаваемого параметром input значения: идентификатор (id) или имя (name) – input - идентификатор или имя таблицы базы данных	Значение типа Promise, разрешаемое в массив объектов, описывающих столбцы таблицы: типы данных и ограничения значений	Получение информации о столбцах таблицы базы данных для корректной генерации элементов страниц панели управления
get_table_name	utils/database/get_table_name.js	– DatabaseConnection - соединение с базой данных – tableid - идентификатор таблицы базы данных	Значение типа Promise, разрешаемое в объект, содержащий два поля: – schema - имя схемы, которой принадлежит таблица – table - имя таблицы	Получение имени таблицы, которое можно использовать в SQL-запросах, по её идентификатору

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 5 - описание и назначение функций клиентской части приложения

Наименование	Исходный код в директории website/static/js	Аргументы	Назначение
requestData	table.js	Отсутствуют	Отправка сообщения типа table_data по протоколу WebSocket
requestData	database.js		Отправка сообщения типа database_data по протоколу WebSocket
dropdown_open	utils/general.js	– dropdown - элемент, являющийся всплывающим меню	Открытие всплывающего меню
dropdown_close			Закрытие всплывающего меню
toggle_dropdown		– ev - событие, объект типа Event	Переключение видимости всплывающего меню
toggle_dialog			Переключение видимости всплывающего окна
open_mobile_navigation		Отсутствуют	Открытие меню навигации при использовании устройств с маленькой шириной экрана
close_mobile_navigation			Закрытие меню навигации при использовании устройств с маленькой шириной экрана

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## ОПИСАНИЕ И НАЗНАЧЕНИЕ КЛАССОВ

Таблица 1 - описание и назначение классов серверной части приложения

Наименование	Исходный код	Аргументы конструктора	Назначение
PostgreSQL	common/postgreSQL/postgreSQL.js	– options - параметры соединения в формате, описываемом библиотекой pg	Соединение и взаимодействие с базой данных PostgreSQL
PostgreSQLCursor	common/postgreSQL/postgreSQL.js	– client - клиент PostgreSQL, используемый для выполнения запроса – query - текст выполняемого SQL-запроса – params - параметры выполняемого SQL-запроса – parser - функция, применяемая к возвращаемым данным при их получении	Выполнение SQL-запроса с использованием переданного клиента PostgreSQL и предоставление его результата постранично
Redis	common/redis/redis.js	– options - параметры соединения в формате, описываемом библиотекой ioredis	Соединение и взаимодействие с хранилищем данных Redis
Utils	common/utils/Utils.js	Отсутствуют	Набор вспомогательных функций
Interval	common/utils/implementations/Interval/Interval.js	– input - число секунд, объект типа PostgresInterval или строка, описывающая интервал времени	Представление интервалов времени в памяти и удобная работа с ними

Таблица 2 - описание и назначение расширенных приложением объектов фреймворка fastify

Наименование	Назначение
Request	Объект запроса к веб-серверу, содержащий всю информацию о входящем запросе и предоставляющий методы его обработки.
Reply	Объект ответа веб-сервера пользователю, содержащий и накапливающий информацию об ответе, а также содержащий методы его редактирования и отправки.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 3 - описание и назначение классов клиентской части приложения

Наименование	Исходный код	Аргументы конструктора	Назначение
Chart	website/static/js/ utils/chart.js	<ul style="list-style-type: none"> <li>– element - HTML-контейнер для отображения графика</li> <li>– records - максимальное количество отображаемых значений</li> <li>– update_interval - интервал обновления графика</li> <li>– title - заголовок графика</li> <li>– vAxisTitle - заголовок вертикальной оси графика</li> <li>– labels - названия линий на графике</li> </ul>	Удобное отображение и редактирование графиков на веб-страницах с помощью библиотеки google charts
TableRow	website/static/js/ utils/table.js	– template - шаблон строки таблицы	Представление записи таблицы и удобное взаимодействие с ней
TableInsertRow		– template - шаблон строки таблицы	Представление записи блока добавления данных таблицы и удобное взаимодействие с ней
TableDisplayRow		<ul style="list-style-type: none"> <li>– template - шаблон строки таблицы</li> <li>– data - данные, отображаемые строкой таблицы</li> </ul>	Представление записи блока отображения данных таблицы и удобное взаимодействие с ней
Table		<ul style="list-style-type: none"> <li>– table - HTML-элемент таблицы</li> <li>– page_size - размер страниц при загрузке таблиц</li> <li>– socket - соединение по протоколу WebSocket, которое должно использоваться для получения данных</li> <li>– socketEventName - тип сообщения, который должен использоваться для получения данных</li> </ul>	Отображение и реализация алгоритмов работы таблиц, отображающих записи базы данных на страницах панели управления

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## ПРИЛОЖЕНИЕ 8

## ОПИСАНИЕ И НАЗНАЧЕНИЕ ПОЛЕЙ И МЕТОДОВ КЛАССОВ

Таблица 1 - описание и назначение полей и методов класса PostgreSQL

1) Описание и назначение полей класса PostgreSQL				
Наименование	Модификаторы доступа	Тип значения		Назначение
#Connection	Приватный	pg.Pool		Соединение с базой данных
#DefaultPostgresOptions	Приватный, статический	Объект		Стандартные настройки соединения с базой данных
2) Описание и назначение методов класса PostgreSQL				
Наименование	Модификаторы доступа	Аргументы	Возвращаемое значение	Назначение
format	Публичный	– fmt - форматная строка запроса – ...args - параметры запроса	Строка запроса с безопасно подставленными значениями	Предотвращение атак типа SQL injection при подстановке параметров в запросы
format	Публичный, статический			
end	Публичный	Отсутствуют	Отсутствует	Остановка соединения
get_raw	Публичный	Отсутствуют	Объект типа pg.Pool	Получение соединения с базой данных
#parse_response	Приватный	– rows - массив объектов, полученных в результате выполнения SQL-запроса	Массив преобразованных объектов	Представление результата выполнения запроса в удобном виде

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 1

query	Публичный	<ul style="list-style-type: none"> <li>– query - текст SQL-запроса</li> <li>– params - массив параметров SQL-запроса</li> <li>– Объект настроек запроса: <ul style="list-style-type: none"> <li>– parse указывает необходимость обработки ответа</li> <li>– one_response указывает необходимость получения только первого результата</li> </ul> </li> </ul>	Результат выполнения SQL-запроса к базе данных	Выполнение SQL-запроса и представление результата в удобной форме
query_multiple	Публичный	<ul style="list-style-type: none"> <li>– queries - объект, описывающий SQL-запросы. Имя поля - имя запроса, значение поля - текст запроса или объект, содержащий текст запроса (query) и настройки запроса (parse и one_response)</li> </ul>	Введённый объект, в котором значения полей заменены результатами выполнения соответствующих запросов	Выполнение нескольких SQL-запросов за одно обращение к базе данных и представление результата в удобной форме
cursor	Публичный	<ul style="list-style-type: none"> <li>– query - текст SQL-запроса</li> <li>– params - параметры SQL-запроса</li> <li>– Объект настроек: <ul style="list-style-type: none"> <li>– parse указывает необходимость обработки ответов</li> </ul> </li> </ul>	Объект типа PostgreSQLCursor	Создание объекта для выполнения SQL-запроса с страничным возвратом результата

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 2 - описание и назначение полей и методов класса PostgreSQLCursor

1) Описание и назначение полей класса PostgreSQLCursor				
Наименование	Модификаторы доступа	Тип значения	Назначение	
#client	Приватный	pg.Client	Клиент PostgreSQL	
#cursor	Приватный	Cursor	“Курсор”, выполняющий запрос и постранично возвращающий результат	
query	Публичный	String	Текст выполняемого запроса	
params	Публичный	Массив любых значений	Параметры выполняемого запроса	
parser	Публичный	Функция	Функция, применяемая к возвращаемым данным для их представления в удобном виде	
2) Описание и назначение методов класса PostgreSQLCursor				
Наименование	Модификаторы доступа	Аргументы	Возвращаемое значение	Назначение
get	Публичный	– maxRows - количество записей, которые необходимо получить	Страница результата выполнения SQL-запроса	Выполнение SQL-запроса и представление страницы его результата в удобной форме
end	Публичный	Отсутствуют	Отсутствуют	Завершение выполнения запроса и освобождение клиента

Таблица 3 - описание и назначение полей и методов класса Redis

1) Описание и назначение полей класса Redis			
Наименование	Модификаторы доступа	Тип значения	Назначение
#Connection	Приватный	ioredis	Соединение с базой данных
#DefaultRedis Options	Приватный, статический	Объект	Стандартные настройки соединения с базой данных

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 3

2) Описание и назначение методов класса Redis				
Наименование	Модификаторы доступа	Аргументы	Возвращаемое значение	Назначение
end	Публичный	Отсутствуют	Отсутствует	Остановка соединения
get_raw	Публичный	Отсутствуют	Объект типа ioredis	Получение соединения с базой данных
set	Публичный	– key - ключ – value - значение	Результат выполнения операции	Добавление или изменение значения, ассоциированного с заданным ключом
set_expire	Публичный	– key - ключ – value - значение – expiration - время жизни в секундах		Добавление или изменение значения, ассоциированного с заданным ключом, с установкой времени его автоматического удаления
set_keepttl	Публичный	– key - ключ – value - значение		Изменение значения, ассоциированного с заданным ключом, с сохранением времени до его автоматического удаления
get	Публичный	– key - ключ	Значение, ассоциированное с заданным ключом	Получение значения по ключу
get_delete	Публичный			Получение и удаление значения по ключу
get_expire	Публичный	– key - ключ – expiration - время жизни в секундах		Получение значения по ключу с обновлением времени до его автоматического удаления
delete	Публичный	– ...keys - ключи	Результат выполнения операции	Удаление значений, ассоциированных с заданными ключами

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



Таблица 4 - описание и назначение полей и методов класса Utils

1) Описание и назначение полей класса Utils			
Наименование	Модификаторы доступа	Тип значения	Назначение
get_field_at_path	Публичный, статический	Функция	Ссылка на функцию get_field_at_path
set_field_at_path		Функция	Ссылка на функцию set_field_at_path
Interval		Класс	Ссылка на класс Interval

Таблица 5 - описание и назначение полей и методов класса Interval

1) Описание и назначение полей класса Interval (помимо наследуемых от PostgresInterval)				
Наименование	Модификаторы доступа	Тип значения		Назначение
#format	Приватный, статический	Regex		Человекочитаемый формат представления интервала
years	Публичный	Number		Количество лет в интервале
months	Публичный	Number		Количество месяцев в интервале
days	Публичный	Number		Количество дней в интервале
hours	Публичный	Number		Количество часов в интервале
minutes	Публичный	Number		Количество минут в интервале
seconds	Публичный	Number		Количество секунд в интервале
milliseconds	Публичный	Number		Количество миллисекунд в интервале
2) Описание и назначение методов класса Interval (помимо наследуемых от PostgresInterval)				
Наименование	Модификаторы доступа	Аргументы	Возвращаемое значение	Назначение
normalize	Публичный	Отсутствуют	Отсутствует	Перераспределение значений полей интервала так, чтобы более узкое поле не содержало значений, которые можно описать более широкими полями
toSeconds	Публичный	Отсутствуют	Number	Получение размера интервала в секундах
toFormat	Публичный	Отсутствуют	String	Получение представления интервала в человекочитаемом формате

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 6 - описание и назначение полей и методов класса Chart

1) Описание и назначение полей класса Chart				
Наименование	Модификаторы доступа	Тип значения		Назначение
#load	Приватный, статический	Promise		Ожидание полной загрузки библиотеки google charts до начала отображения графиков
onLoad	Публичный, статический	Function		Функция разрешения значения поля #load
#options	Приватный	Объект		Хранение настроек графика
#data	Приватный	google.visualization.DataTable		Хранение отображаемых данных
#chart	Приватный	google.visualization.LineChart		Объект графика
records	Публичный	Number		Максимальное количество отображаемых записей
update_interval	Публичный	Number		Интервал обновления графика
labels	Публичный	Объект		Названия линий на графике
#firstDraw	Приватный	Promise		Ожидание первичной настройки и отображения графика
#firstDrawResolve	Приватный	Function		Функция разрешения значения поля #firstDraw
2) Описание и назначение методов класса Chart				
Наименование	Модификаторы доступа	Аргументы	Возвращаемое значение	Назначение
#adjustFontSize	Приватный	Отсутствуют	Отсутствует	Вычисление подходящих размеров шрифтов и их установка в объект настроек #options
#firstDrawChart	Приватный	Отсутствуют	Отсутствует	Первичная настройка и отображение графика
#redraw	Приватный	Отсутствуют	Отсутствует	Перерисовка графика
addRecord	Публичный	– record - объект новой записи	Отсутствует	Добавление записи на график и его перерисовка

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 7 - описание и назначение полей и методов класса TableRow

1) Описание и назначение полей класса TableRow				
Наименование	Модификаторы доступа	Тип значения		Назначение
elements	Публичный	Массив HTML-элементов		Список элементов, из которых составлена запись таблицы
deleteButton	Публичный	HTML-элемент		Кнопка удаления записи
changeHandler	Публичный	Функция		Ссылка на метод changed
#changeCallbacks	Приватный	Массив функций		Массив функций, вызываемых при редактировании записи пользователем
2) Описание и назначение методов класса TableRow				
Наименование	Модификаторы доступа	Аргументы	Возвращаемое значение	Назначение
inputs	Публичный	Отсутствуют	Массив элементов	Получение списка полей ввода записи
addChangeCallback	Публичный	– callback - функция	Отсутствует	Добавление функции, вызываемой при редактировании записи пользователем
removeChangeCallback	Публичный	– callback - функция	Отсутствует	Удаление функции, вызываемой при редактировании записи пользователем
changed	Публичный	– ev - событие, объект типа Event	Отсутствует	Обработка события изменения записи пользователем
getData	Публичный	Отсутствуют	Объект	Получение данных текущей записи в виде ассоциативного контейнера
getInitialData	Публичный	Отсутствуют	Объект	Получение начальных данных текущей записи в виде ассоциативного контейнера
getIdentifier	Публичный	Отсутствуют	Объект	Получение данных, однозначно идентифицирующих текущую запись, в виде ассоциативного контейнера

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 8 - описание и назначение полей и методов класса TableInsertRow

1) Описание и назначение полей класса TableRow (помимо наследуемых от TableRow)				
Наименование	Модификаторы доступа	Тип значения		Назначение
#removeCallbacks	Приватный	Массив функций		Массив функций, вызываемых при удалении записи пользователем
2) Описание и назначение методов класса TableRow (помимо наследуемых от TableRow)				
Наименование	Модификаторы доступа	Аргументы	Возвращаемое значение	Назначение
addRemoveCallback	Публичный	– callback - функция	Отсутствует	Добавление функции, вызываемой при удалении записи пользователем
removeRemoveCallback	Публичный	– callback - функция	Отсутствует	Удаление функции, вызываемой при удалении записи пользователем
removed	Публичный	Отсутствуют	Отсутствует	Обработка события удаления записи
showDeleteButton	Публичный	Отсутствуют	Отсутствует	Отображение кнопки удаления записи

Таблица 9 - описание и назначение полей и методов класса TableDisplayRow

1) Описание и назначение полей класса TableDisplayRow (помимо наследуемых от TableRow)				
Наименование	Модификаторы доступа	Тип значения	Назначение	
deleted	Публичный	Bool	Указание, была ли запись помечена на удаление	
edited	Публичный	Bool	Указание, была ли запись изменена пользователем	
2) Описание и назначение методов класса TableDisplayRow (помимо наследуемых от TableRow)				
Наименование	Модификаторы доступа	Аргументы	Возвращаемое значение	Назначение
#fillData	Приватный	– data - объект данных о записи	Отсутствует	Заполнение полей ввода записи значениями переданного объекта
changed	Публичный	Отсутствуют	Отсутствует	Обработки изменения записи
remove	Публичный	Отсутствуют	Отсутствует	Обработка удаления записи

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 10 - описание и назначение полей и методов класса Table

1) Описание и назначение полей класса Table			
Наименование	Модификаторы доступа	Тип значения	Назначение
#tables	Приватный, статический	Массив ссылок на объекты типа Table	Поддержание списка всех активных таблиц
#saveButton	Приватный, статический	HTML-элемент	Кнопка, по нажатию на которую происходит сохранение изменений
#page_size	Приватный	Number	Размер страниц, которыми происходит загрузка данных
#table	Приватный	HTML-элемент	Элемент таблицы
#head	Приватный	HTML-элемент	Элемент заголовка таблицы
#insertBody	Приватный	HTML-элемент	Элемент блока добавления данных таблицы
#displayBody	Приватный	HTML-элемент	Элемент блока отображения данных таблицы
#insertRow	Приватный	Класс	Класс-обёртка над TableRow для создания записей блока добавления данных таблицы
#displayRow	Приватный	Класс	Класс-обёртка над TableRow для создания записей блока отображения данных таблицы
socket	Публичный	WebSocket	Соединение, которое должно использоваться для получения данных
socketEventName	Публичный	String	Тип сообщения, который должен использоваться для получения данных
#insertRows	Приватный	Массив объектов типа, описываемого полем #insertRow	Список записей блока добавления данных таблицы
#addInsertRow Callback	Приватный	Function	Ссылка на функцию #addInsertRow

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 10

#displayRows	Приватный	Массив объектов типа, описываемого полем #displayRow	Список записей блока отображения данных таблицы	
#displayObserver	Приватный	IntersectionObserver	Наблюдатель, вызывающий функцию #loadNextPage при появлении последней строки блока отображения данных таблицы на экране	
2) Описание и назначение методов класса Table				
Наименование	Модификаторы доступа	Аргументы	Возвращаемое значение	Назначение
registerSaveButton	Публичный, статический	– button - элемент кнопки	Отсутствует	Настройка сохранения изменений по нажатии на указанную кнопку
showSaveButton	Публичный, статический	Отсутствуют	Отсутствует	Отображение кнопки сохранения изменений
#setupHead	Приватный	Отсутствуют	Отсутствует	Настройка заголовка таблицы
#getFilters	Приватный	Отсутствуют	Массив объектов	Получение массива применённых пользователем параметров фильтрации
#getSorts	Приватный	Отсутствуют	Массив объектов	Получение массива применённых пользователем параметров сортировки
#addInsertRow	Приватный	Отсутствуют	Отсутствует	Создание новой записи в блоке добавления данных таблицы
#removeInsertRow	Приватный	– row - объект	Отсутствует	Удаление записи блока добавления данных таблицы
#setupInsert	Приватный	Отсутствуют	Отсутствует	Настройка блока добавления данных таблицы
#getNextPageData	Приватный	Отсутствуют	Значение типа Promise, разрешаемое в массив записей очередной страницы	Получение очередной страницы записей таблицы базы данных с сервера

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 10

#renderData	Приватный	– data - массив записей очередной страницы	Отсутствует	Отображение переданной страницы записей в блоке отображения данных
#loadNextPage	Приватный	Отсутствуют	Отсутствует	Загрузка очередной страницы записей таблицы базы данных
#setupDisplay	Приватный	Отсутствуют	Отсутствует	Настройка блока отображения данных таблицы
getUpdateActions	Публичный	Отсутствуют	Массив объектов	Получение списка изменений, сделанных пользователем в записях таблицы

Таблица 11 - описание и назначение полей и методов, добавленных к классу Request

1) Описание и назначение полей, добавленных к классу Request				
Наименование	Модификаторы доступа	Тип значения	Назначение	
session_id	Публичный	String	Идентификатор сессии пользователя, отправившего запрос	
authentication	Публичный	String	Ключ аутентификации запроса	
authentication_data	Публичный	Объект	Аутентификационные данные запроса, если они были получены	
authorization	Публичный	Объект	Авторизационные данные запроса, если они были получены	
2) Описание и назначение методов, добавленных к классу Request				
Наименование	Модификаторы доступа	Аргументы	Возвращаемое значение	Назначение
get_sessionid	Публичный	Отсутствуют	Строка - идентификатор сессии	Получение идентификатора сессии пользователя из файлов cookie, содержащихся в объекте запроса
authenticate	Публичный	Отсутствуют	Строка - ключ аутентификации запроса	Генерация ключа аутентификации запроса и его сохранение в Redis

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 11

check_authentication	Публичный	– code - ключ аутентификации запроса	Аутентификационные данные запроса, если они были получены	Получение аутентификационных данных, соответствующих указанному ключу, и их проверка для данного запроса
authorize	Публичный	Отсутствуют	Авторизационные данные запроса, если они были получены	Получение авторизационных данных запроса и их актуализация, если это необходимо

Таблица 12 - описание и назначение полей и методов, добавленных к классу Response

1) Описание и назначение методов, добавленных к классу Response				
Наименование	Модификаторы доступа	Аргументы	Возвращаемое значение	Назначение
login	Публичный	– login - имя аккаунта – password - пароль аккаунта	Объект ответа, для которого произведена отправка результата обработки	Обработка запроса на вход в аккаунт с использованием указанных идентификационных данных
logout	Публичный	Отсутствуют		Обработка запроса на выход из аккаунта
error	Публичный	– code - код ошибки – description - описание ошибки – opts - дополнительные параметры генерации исходного кода страницы по шаблону		Генерация исходного кода подходящей страницы ошибки и его отправка пользователю

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



## ПРИЛОЖЕНИЕ 9

## ОПИСАНИЕ И НАЗНАЧЕНИЕ ЭКСПОРТИРУЕМЫХ ПЕРЕМЕННЫХ

Таблица 1 - описание и назначение экспортируемых переменных

Наименование	Определяющий файл	Тип значения	Назначение
config	common/configs/config.js	Повторяет файл конфигурации приложения	Представление файла конфигурации приложения в памяти программы
TargetDatabase	common/postgreSQL/postgreSQL.js	Объект класса PostgreSQL	Взаимодействие с целевой базой данных
InternalDatabase			Взаимодействие с внутренней базой данных
Cache	common/redis/redis.js	Объект класса Redis	Взаимодействие с хранилищем данных Redis
types	website/routes/utills/schemas/schemas.js	Объект, содержащий описания типов данных в формате JSON Schema	Описание часто используемых типов вводимых значений.
EMPTY_GET_SCHEMA		Объект JSON Schema	Схема входных данных для запросов, не имеющих входных данных

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## ОПИСАНИЕ И НАЗНАЧЕНИЕ CSS-СТИЛЕЙ

Таблица 1 - описание и назначение основных CSS-слоёв

Наименование	Назначение
normalization	Основные стили и переменные, позволяющие уменьшить зависимость внешнего вида приложения от используемого браузера
surface	Классы surface и surface_rounded
headings	Заголовки первого, второго и третьего уровней, а также элемент figcaption
lists	Списки ul, ol, dl, menu и их элементы
controls	Элементы управления приложением: кнопки и ссылки (“якоря”)
forms	Формы и их элементы, в том числе поля ввода
dropdowns	Выпадающие меню и элементы управления ими
tables	Таблицы
dialogs	Выпадающие окна
body	Тело веб-страницы, основной элемент
header	Заголовок веб-страницы
nav	Меню навигации и элементы управления им на устройствах с маленькой шириной экрана
graphs	Блок, содержащий графики на веб-страницах приложения
data_table	Таблица отображения данных и её элементы
text	Классы управления стилями текста
sizing	Классы управления размерами элементов
blocks	Классы управления расположением элементов и его детей
padding	Классы управления внутренними отступами элементов
margin	Классы управления отступами между элементами
border_radius	Классы управления закруглениям углов элементов

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 2 - описание и назначение основных CSS-переменных

Наименование	Назначение
--brand-color	Основной цвет приложения
--background-color	Цвет фона приложения
--surface-color	Цвет фона основных элементов приложения
--border-color	Цвет обводок элементов
--shadow-color	Цвет теней, отбрасываемых элементами
--text-brand-color	Цвет важного текста приложения
--text-major-color	Цвет основного текста приложения
--text-minor-color	Цвет дополнительного текста приложения
--text-placeholder-color	Цвет вспомогательного текста приложения
--text-inverse-color	Цвет текста, отображаемого на светлом фоне
--scrollbar-color	Цвет полосы прокрутки веб-страницы
--color-red	Красный цвет
--color-yellow	Желтый цвет
--color-purple	Фиолетовый цвет
Переменные вида --chart-*	HEX-коды некоторых цветов для использования с библиотекой google charts
--line-height	Высота строки текста
Переменные виде --font-size-*	Адаптивные размеры шрифтов, используемых в приложении: small, regular, medium, big, large, huge, enormous
Переменные вида --vertical-sizing-*	Стандартные адаптивные размеры вертикальных отступов
Переменные вида --horizontal-sizing-*	Стандартные адаптивные размеры горизонтальных отступов
Переменные вида --padding-vertical-*	Стандартные адаптивные размеры вертикальных внутренних отступов элементов
Переменные вида --padding-horizontal-*	Стандартные адаптивные размеры горизонтальных внутренних отступов элементов
Переменные вида --padding-horizontal-*-wide	Увеличенные адаптивные размеры горизонтальных внутренних отступов элементов

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 2

Переменные вида --padding-*	Стандартные адаптивные размеры внутренних отступов элементов
Переменные вида --padding-*-wide	Увеличенные адаптивные размеры внутренних отступов элементов
Переменные вида --margin-vertical-*	Стандартные адаптивные размеры вертикальных отступов между элементами
Переменные вида --margin-horizontal-*	Стандартные адаптивные размеры горизонтальных отступов между элементами
Переменные вида --margin-*	Стандартные адаптивные размеры отступов между элементами
--interactive-min-height	Минимальная допустимая высота интерактивных элементов
Переменные вида --border-radius-*	Стандартные размеры закруглений углов элементов

Таблица 3 - описание и назначение основных CSS-классов

Наименование	Назначение
desktop_hidden	Элементы, которые должны быть невидимы на экранах с достаточной шириной экрана
surface	Элементы макро-разметки веб-страницы
surface_rounded	Элементы макро-разметки веб-страницы с закруглёнными углами
no_bullets	Списки без маркеров
control_inline	“Внутристрочные” элементы управления приложением - ссылки
control_block	Элементы управления приложением блочного уровня - кнопки
control_primary	Основные блочные элементы управления приложением
control_secondary	Дополнительные блочные элемент управления приложением
checkboxWrapper	Более красивое отображений полей ввода типа checkbox
dropdown_arrow	Добавление стрелки, указывающей на наличие выпадающего меню, открываемого нажатием на этот элемент
dropdown_arrow_horizontal	Добавление повернутой на 90 градусов стрелки, указывающей на наличие выпадающего меню, открываемого нажатием на этот элемент
dropdown_opened	Открытые всплывающие меню и кнопки, управляющие ими
dropdown_inplace	Всплывающие меню, отображаемые “на месте”

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 3

dropdown_nobg	Всплывающие меню, отображаемые в виде отдельного блока без фона
dropdown_bg	Всплывающие меню, отображаемые в виде отдельного блока с фоном
striped_button	Кнопки управления всплывающими меню, отображаемые в виде трёх полос
cross_button	Кнопки управления всплывающими меню, отображаемые в виде креста
main_body	Элемент веб-страницы с основным содержимым
graphs	Элемент панели управления, содержащий графики
data_table	Таблица данных панели управления
Классы вида text_*	Изменение цвета текста в соответствии с установленными CSS-переменными
Классы вида text_size_*	Изменение размера шрифта в соответствии с установленными CSS-переменными
Классы вида text_align_*	Изменение алгоритма выравнивания текста
text_nowrap	Отключение переноса текста с добавлением многоточия, если текст не помещается в элемент
text_nohover	Отключение изменения стиля курсора и свойства text-decoration при наведении курсора на элемент
text_nodecoration	Отключение свойства text-decoration
minwidth	Установка свойству width значения min-content
maxwidth	Установка свойству width значения 100%
maxheight	Установка свойству height значения 100%
block	Установка свойству display значения block
centered	Горизонтальное выравнивание элементов по центру родителей при помощи значения auto свойств margin
flex_horizontal	Преобразование элементов во flex-контейнеры, располагающие элементы горизонтально
flex_vertical	Преобразование элементов во flex-контейнеры, располагающие элементы вертикально
flex_fill	Установка свойству flex-grow значения 2
grid_1x2	Отображение элементов в виде двух частей одинакового размера, расположенных горизонтально

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 3

grid	Преобразование элементов в grid-контейнеры с двумя столбцами
grid_columns2	Указание grid-контейнеру на отображение двух столбцов
grid_columns3	Указание grid-контейнеру на отображение трёх столбцов
grid_fill_column	Установка свойству grid-row значения 1/-1
grid_fill_row	Установка свойству grid-column значения 1/-1
Классы вида padding_*	Изменение внутренних отступов элементов
Классы вида margin_*	Изменение отступов между элементами
Классы вида border_radius_*	Изменение закруглений углов элементов

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**ТЕРМИНЫ И СОКРАЩЕНИЯ**

- 1) Веб-сервер - основная часть приложения, отвечающая за обработку запросов пользователей и организацию взаимодействия с базами данных.
- 2) Внутренняя база данных - база данных PostgreSQL, используемая приложением для организации своей работы: сохранения данных аккаунтов пользователей и истории сделанных ими запросов.
- 3) Код уровня доступа - строка, описывающая уровень доступа аккаунта и состоящая из пяти символов: R указывает наличие доступа к чтению, I - к вставке, U - к редактированию, D - к удалению и A - к администрированию. В случае отсутствия какого-либо доступа у аккаунта на его месте указывается символ прочерка "-".
- 4) Начальная страница, страница авторизации, страница входа в аккаунт - страница приложения, предназначенная для ввода пользователем идентификационных данных аккаунта.
- 5) Панель администратора - часть панели управления; набор страниц приложения, доступных только пользователям с уровнем доступа "администрирование".
- 6) Панель управления - набор страниц приложения, доступных только авторизованным пользователям.
- 7) Страница "Аккаунты" - страница панели администратора, предоставляющая возможность управления зарегистрированными аккаунтами.
- 8) Страница "База данных" - страница панели управления, отображающая информацию о целевой базе данных.
- 9) Страница "История" - страница панели администратора, предоставляющая возможность просмотра истории сделанных пользователями запросов к целевой базе данных.
- 10) Целевая база данных - база данных PostgreSQL, информацию о которой отображает приложение.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

<b>Лист регистрации изменений</b>
-----------------------------------

[illegible]