

Оценка _____

Руководитель курсовой
работы: Мирвода Сергей Геннадьевич

Члены комиссии _____

Дата защиты _____

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе
по теме: «Военный сайт»
по дисциплине: «Языки и методы программирования»

Студенты: Артамонов Роман Дмитриевич
Петренко Анастасия Сергеевна
Пономаренко Павел Андреевич

Группа: РИ-311055

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1 Цель проекта.....	3
2 Видение проекта.....	3
1 УПРАВЛЕНИЕ СТАДИЯМИ ПРОЕКТА.....	4
2 ПРОЕКТИРОВАНИЕ.....	5
3 РАЗРАБОТКА.....	9
4 АЛГОРИТМ РАБОТЫ.....	11
ЗАКЛЮЧЕНИЕ	17
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	18
ПРИЛОЖЕНИЕ А	19

ВВЕДЕНИЕ

1 Цель проекта

Цель проекта состоит в создании специализированного сайта, который будет предоставлять доступ к важным и конфиденциальным данным, связанным со служащими и вооружением. Эта платформа будет служить инструментом для хранения, управления и обновления информации о важных аспектах военной деятельности.

Основная цель закрытого военного сайта - обеспечить безопасность и конфиденциальность данных. Это позволит предотвратить несанкционированный доступ и гарантировать, что информация останется доступной только тем лицам, которые имеют соответствующие полномочия.

2 Видение проекта

Проект по разработке закрытого военного сайта предполагает создание веб-сайта, на котором будет храниться информация о служащих, технике, сооружениях и других важных данных военного характера.

Доступ к сайту осуществляется строго по приглашению администратора. Это значит, что обычным пользователям невозможно самостоятельно зарегистрироваться на платформе. Вся регистрация и управление аккаунтами осуществляется только администратором.

В будущем, при добавлении нового служащего, администратор будет выдавать ему логин и пароль для получения доступа к закрытому военному сайту. Эти учетные данные позволят служащему войти в систему и получить доступ к информации, необходимой для его служебной деятельности.

1 УПРАВЛЕНИЕ СТАДИЯМИ ПРОЕКТА

Для дальнейшего понимания проекта было принято решение создать функциональную диаграмму, которая описывает основной функционал сайта. Через диаграмму можно визуальным образом отследить взаимодействие различных модулей сайта: какие данные поступают на каждый из модулей и какие данные он возвращает, а также из нее видна последовательность функционала.

Функциональная диаграмма (нотация IDEF0)



Рисунок 1. Функциональная диаграмма.

Чтобы облегчить работу с проектом, был создан репозиторий на GitHub, в котором публиковались текущие версии приложения и диаграммы. Также в нём публиковались новые задачи и требования по проекту, что крайне ускорило как сам процесс разработки, так и проводимый контроль качества, помогающий выявить ошибки и отклонения от изначальной задумки продукта.

Ссылка на репозиторий: <https://github.com/nastushka283/coursework>

2 ПРОЕКТИРОВАНИЕ

Архитектура данного проекта основана на трех основных компонентах: фронтенд, бэкенд и база данных.

Фронтенд представляет собой пользовательский интерфейс, через который пользователи могут авторизоваться и взаимодействовать с сайтом. Он отвечает за отображение информации пользователю и обработку его действий, таких как отправка запросов на сервер.

Бэкенд обрабатывает все запросы от пользователей, взаимодействует с базой данных и обеспечивает логику работы сайта. Он отвечает за обработку запросов и обеспечивает безопасность данных. Бэкенд также обрабатывает запросы на получение и обновление информации в базе данных.

База данных используется для хранения данных о служащих, вооружении, офицерском составе и другой информации. Для этого проекта была выбрана СУБД PostgreSQL, которая обеспечивает надежность, масштабируемость и поддержку на разных операционных системах. PostgreSQL также обеспечивает различные механизмы безопасности, такие как контроль доступа и шифрование данных, что позволяет защитить информацию от несанкционированного доступа.

В проекте была выбрана СУБД PostgreSQL ввиду ее совместимости с различными операционными системами. Чтобы получить более подробную информацию о системных требованиях, можно обратиться к файлу в репозитории на GitHub. Там можно найти информацию о поддерживаемых ОС и другие технические детали, необходимые для успешного использования PostgreSQL в данном проекте.

Для визуализации архитектуры проекта были созданы несколько диаграмм:

1. Архитектурная диаграмма

Архитектурная диаграмма показывает взаимодействие компонентов проекта от пользователя до базы данных. Она отображает потоки данных и взаимодействие между пользователем, сайтом, сервером и базой данных. На диаграмме можно увидеть, как пользователи отправляют запросы через фронтенд, которые затем

обрабатываются и взаимодействуют с базой данных для получения и обновления информации.



Рисунок 2. Архитектурная диаграмма.

2. Функциональная диаграмма

Функциональная диаграмма была построена с использованием нотации IDEF0.

IDEF0 — это графическая нотация, используемая для создания функциональной модели системы, включающей в себя структуру и функции системы, а также потоки информации и материальных объектов, связывающих эти функции. На функциональной диаграмме можно увидеть, как различные функции проекта взаимодействуют друг с другом и обрабатывают информацию.

Функциональная диаграмма (нотация IDEF0)

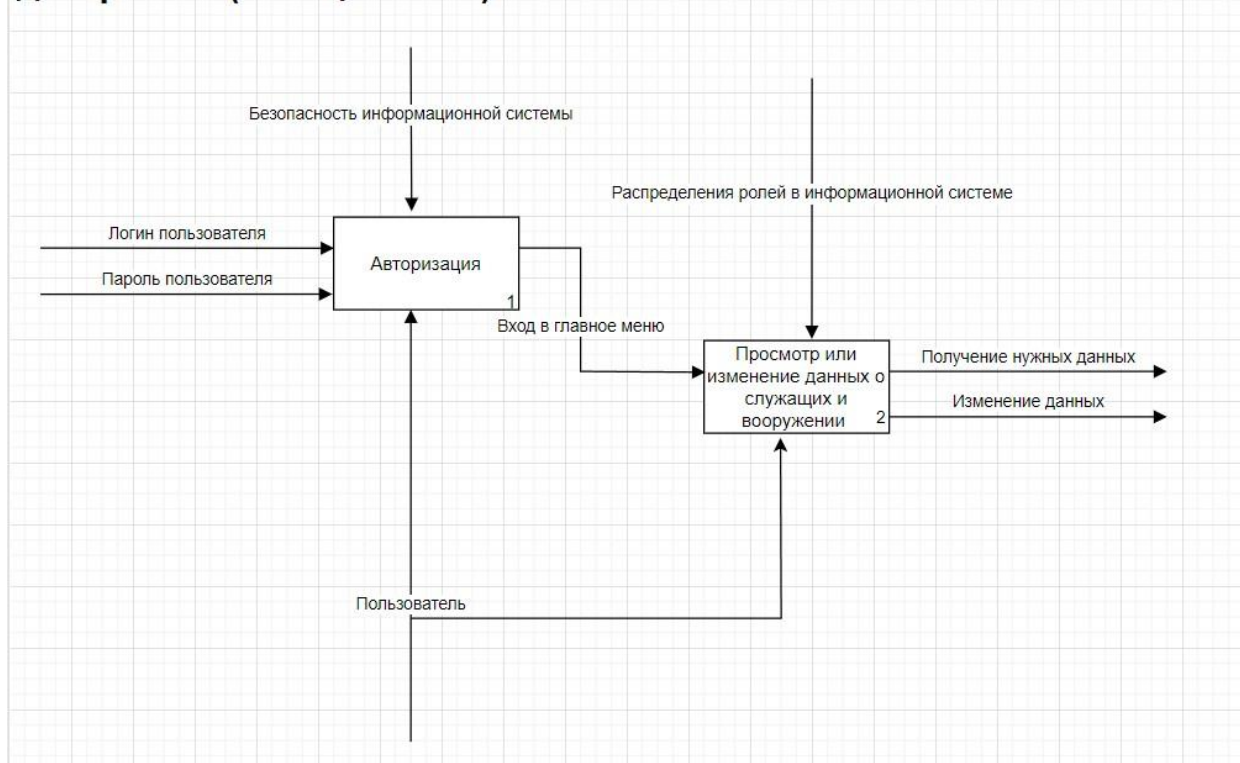


Рисунок 3. Функциональная диаграмма

3. Диаграмма последовательности

Для построения диаграммы последовательности была выбрана функция изменения таблицы вооружения. Эта функция доступна только для администратора и показывает последовательность действий, которые происходят при обновлении данных в таблице вооружения. На диаграмме последовательности можно увидеть, как различные компоненты системы взаимодействуют друг с другом и обрабатывают запросы на изменение данных.

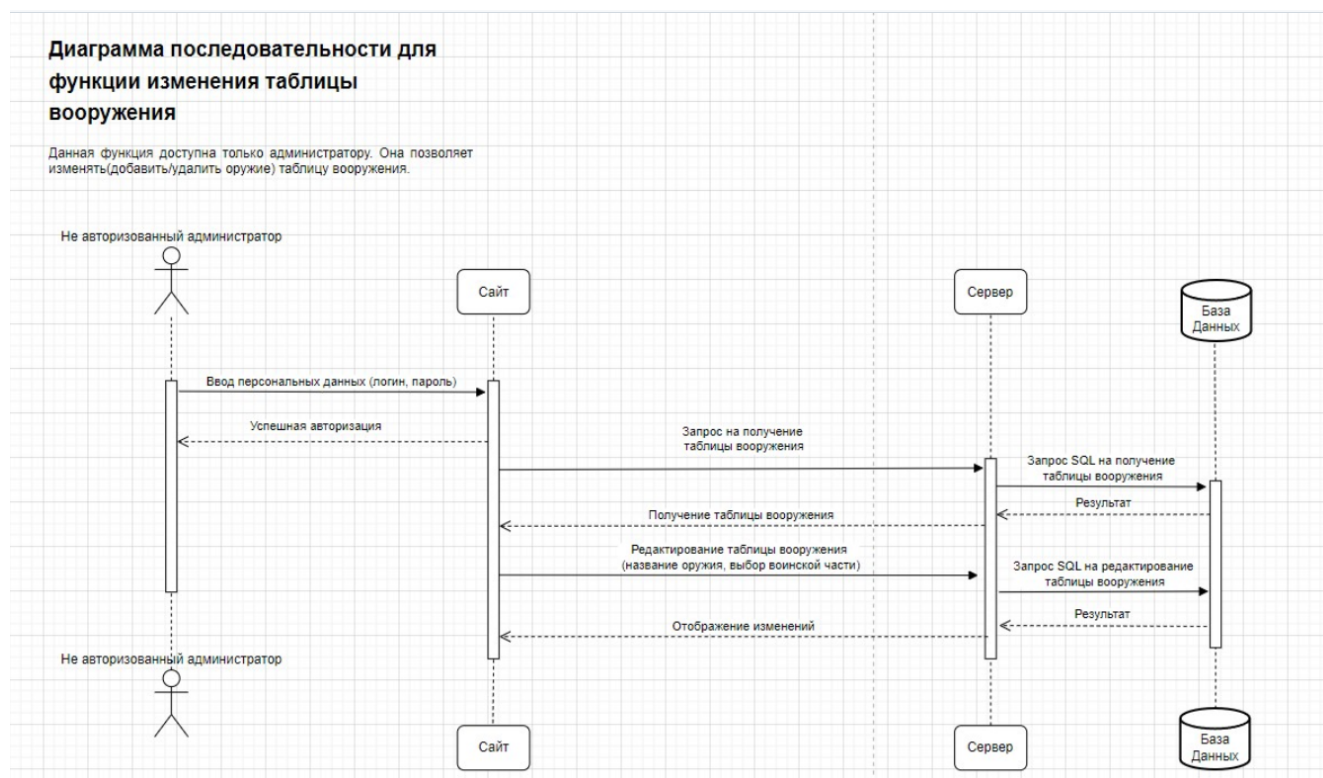


Рисунок 4. Диаграмма последовательности.

4. Концептуальная модель базы данных

Концептуальная модель визуализирует основные сущности и связи между ними в проекте. Она отображает общую структуру данных, позволяя лучше понять организацию данных и их взаимосвязи. Концептуальная модель базы данных помогает определить структуру таблиц и связи между ними, что является важным шагом при проектировании базы данных.

В нашем репозитории сделаны 3 концептуальные модели, в пояснительной записке представлена одна из них.

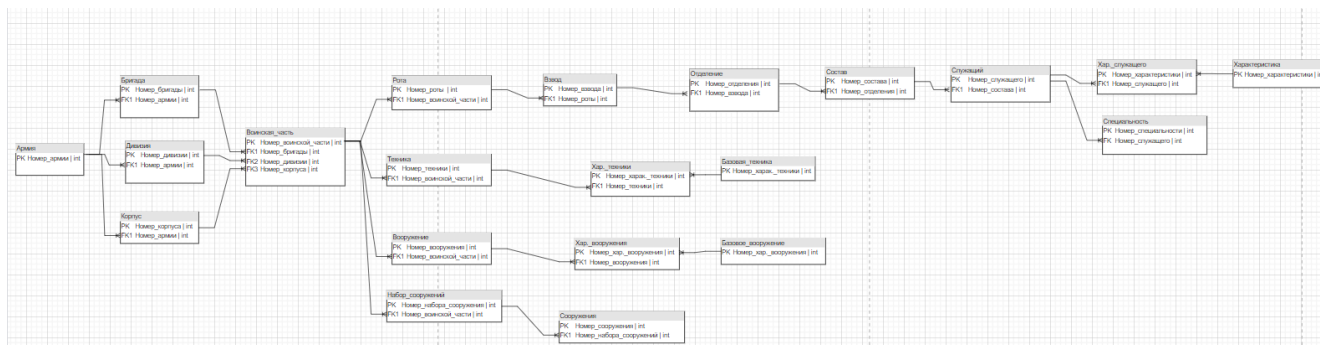


Рисунок 5. Концептуальная модель.

Все задачи, выданные руководителем проекта, были выполнены.

Результаты построения диаграмм были загружены в репозиторий на GitHub.

3 РАЗРАБОТКА

Так как наш проект представлен в виде закрытого военного сайта, то были выбраны следующие языки программирования, обоснованные системным ПО:

1. PHP: Использование PHP обосновано его широким распространением в веб-разработке и легкостью интеграции с web-серверами. Он поддерживает взаимодействие с базами данных, что согласуется с использованием PostgreSQL в качестве СУБД.

2. JavaScript (включая jQuery): JavaScript является основным языком для веб-разработки и обеспечивает динамическое взаимодействие на стороне клиента. jQuery облегчает работу с DOM и обеспечивает удобство в написании клиентского кода.

3. HTML и CSS: HTML используется для структурирования содержимого web-страниц, а CSS для стилизации и оформления. Они стандартны и необходимы для веб-разработки.

4. PostgreSQL: Выбор PostgreSQL обоснован его поддержкой различных операционных систем, таких как Windows и Linux, что соответствует требованиям проекта. Эта СУБД предоставляет надежное хранилище данных для проектов среднего и большого масштаба.

Все выбранные технологии хорошо интегрируются между собой и соответствуют требованиям системного ПО. Этот стек позволяет создать эффективный web-проект для управления информацией о служащих, технике и других данных военного характера.

Все необходимые коды представлены в репозитории, а также в приложении А.

Диаграмма пакетов UML

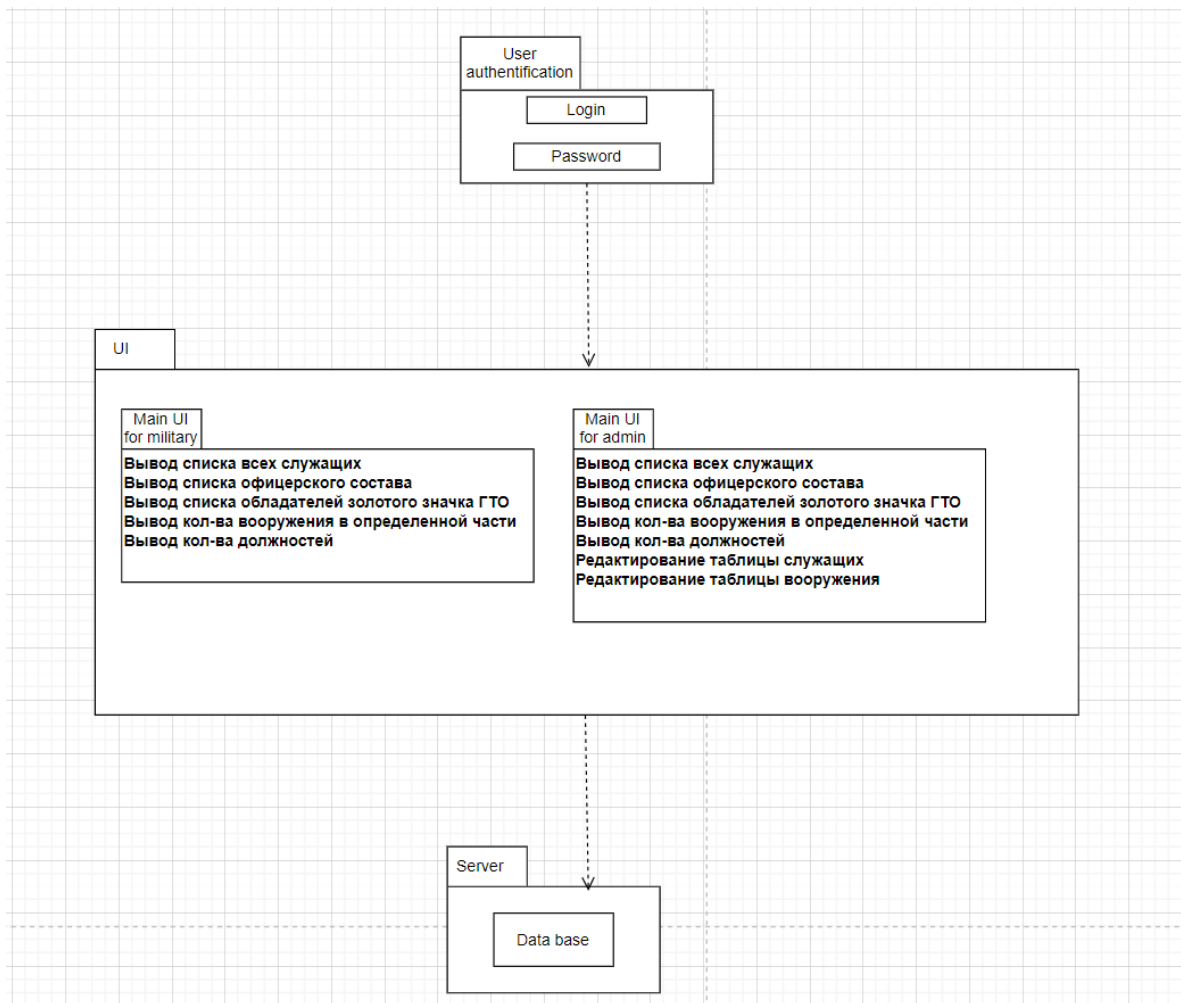


Рисунок 6.

Физическая модель базы данных

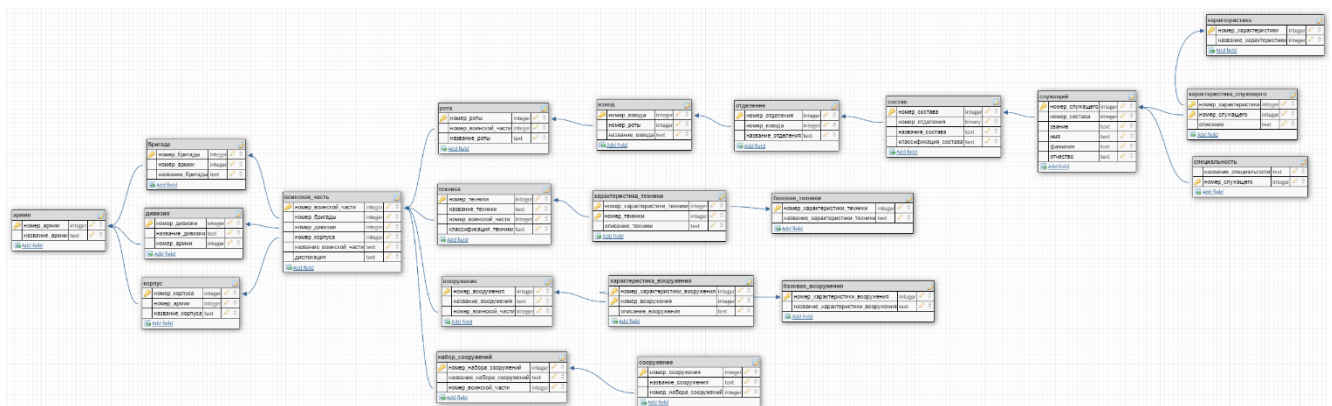


Рисунок 7.

4 АЛГОРИТМ РАБОТЫ

В нашем случае возможно зайти на сайт под ролью администратора, солдата и генерала. Проверим работу администратора.

Для начала авторизуемся под администратором.

**Введите имя и пароль для
входа в систему**

Имя:

Пароль:

Рисунок 8.

Открывается меню.

Вывод списка всех служащих	Вывод списка офицерских составов
Вывод списка всех обладателей золотого значка ГТО	Вывод количества вооружения в определённой части
Вывод количества должностей	
Редактирование таблицы служащих	Редактирование таблицы вооружений
Просмотр аудита	

Рисунок 9.

Выведем список всех военнослужащих.

Вывод списка всех служащих

Высоцкий	Семён	Артёмович	Генерал
Козлов	Сергей	Игоревич	Капитан
Морозов	Дмитрий	Петрович	Старшина
Иванов	Иван	Иванович	Сержант
Петров	Иван	Александрович	Сержант
Ковалева	Анна	Игоревна	Солдат
Морозова	Ирина	Дмитриевна	Майор
Иванов	Андрей	Сергеевич	Лейтенант
Кузнецова	Анна	Николаевна	Прапорщик
Козлов	Семён	Иванович	Старший сержант
Захаров	Иван	Петрович	Старший солдат
Николаева	Елена	Сергеевна	Ефрейтор
Смирнов	Антон	Иванович	Ефрейтор
Петров	Максим	Андреевич	Подполковник
Семенова	Ольга	Александровна	Старший лейтенант
Смирнова	Елена	Сергеевна	Младший сержант
Попов	Дмитрий	Андреевич	Младший солдат

Рисунок 10.

Так как редактировать списки может только администратор, проверим данную функцию от его роли.

Добавим нового служащего.

<input type="text" value="Артамонов"/>	<input type="text" value="Роман"/>	<input type="text" value="Дмитриевич"/>	<input type="text" value="Сержант"/>	<input type="text" value="Северный щит"/>	<input type="button" value="Добавить"/>
--	------------------------------------	---	--------------------------------------	---	---

Рисунок 11.

Проверим, появился ли в списках.

Редактирование списка служащих

Высоцкий	Семён	Артёмович	Генерал	Северный щит	Удалить
Козлов	Сергей	Игоревич	Капитан	Северный щит	Удалить
Морозов	Дмитрий	Петрович	Старшина	Сибирская сталь	Удалить
Иванов	Иван	Иванович	Сержант	Северный Грозовой Клин	Удалить
Петров	Иван	Александрович	Сержант	Сибирская сталь	Удалить
Артамонов	Роман	Дмитриевич	Сержант	Северный щит	Удалить
Ковалева	Анна	Игоревна	Солдат	Белые медведи	Удалить
Морозова	Ирина	Дмитриевна	Майор	Северный Грозовой Клин	Удалить
Иванов	Андрей	Сергеевич	Лейтенант	Северный Грозовой Клин	Удалить
Кузнецова	Анна	Николаевна	Прапорщик	Сибирский Орёл	Удалить
Козлов	Семён	Иванович	Старший сержант	Сибирский Орёл	Удалить
Захаров	Иван	Петрович	Старший солдат	Южный Легион Героев	Удалить
Николаева	Елена	Сергеевна	Ефрейтор	Южный Легион Героев	Удалить
Смирнов	Антон	Иванович	Ефрейтор	Белые медведи	Удалить
Петров	Максим	Андреевич	Подполковник	Северный Ветеран	Удалить
Семенова	Ольга	Александровна	Старший лейтенант	Северный Ветеран	Удалить
Смирнова	Елена	Сергеевна	Младший сержант	Южный Ястреб	Удалить
Попов	Дмитрий	Андреевич	Младший солдат	Сибирская Снегурочка	Удалить

Рисунок 12. Отображение в таблице служащих.

Вывод списка всех служащих

Высоцкий	Семён	Артёмович	Генерал
Козлов	Сергей	Игоревич	Капитан
Морозов	Дмитрий	Петрович	Старшина
Артамонов	Роман	Дмитриевич	Сержант
Иванов	Иван	Иванович	Сержант
Петров	Иван	Александрович	Сержант
Ковалева	Анна	Игоревна	Солдат
Морозова	Ирина	Дмитриевна	Майор
Иванов	Андрей	Сергеевич	Лейтенант
Кузнецова	Анна	Николаевна	Прапорщик
Козлов	Семён	Иванович	Старший сержант
Захаров	Иван	Петрович	Старший солдат
Николаева	Елена	Сергеевна	Ефрейтор
Смирнов	Антон	Иванович	Ефрейтор
Петров	Максим	Андреевич	Подполковник
Семенова	Ольга	Александровна	Старший лейтенант
Смирнова	Елена	Сергеевна	Младший сержант
Попов	Дмитрий	Андреевич	Младший солдат

Рисунок 13. Вывод списка служащих.

Вывод количества определённой должности в определённой воинской части

Северный щит	▼	Сержант	▼
Количество количество должностей = 2			

Рисунок 14.

Теперь удалим нового военнотружущего. Проверим, пропал ли он.

Редактирование списка служащих

Высоцкий	Семён	Артёмович	Генерал	Северный щит	Удалить
Козлов	Сергей	Игоревич	Капитан	Северный щит	Удалить
Морозов	Дмитрий	Петрович	Старшина	Сибирская сталь	Удалить
Иванов	Иван	Иванович	Сержант	Северный Грозовой Клин	Удалить
Петров	Иван	Александрович	Сержант	Сибирская сталь	Удалить
Ковалева	Анна	Игоревна	Солдат	Белые медведи	Удалить
Морозова	Ирина	Дмитриевна	Майор	Северный Грозовой Клин	Удалить
Иванов	Андрей	Сергеевич	Лейтенант	Северный Грозовой Клин	Удалить
Кузнецова	Анна	Николаевна	Прапорщик	Сибирский Орёл	Удалить
Козлов	Семён	Иванович	Старший сержант	Сибирский Орёл	Удалить
Захаров	Иван	Петрович	Старший солдат	Южный Легион Героев	Удалить
Николаева	Елена	Сергеевна	Ефрейтор	Южный Легион Героев	Удалить
Смирнов	Антон	Иванович	Ефрейтор	Белые медведи	Удалить
Петров	Максим	Андреевич	Подполковник	Северный Ветеран	Удалить
Семенова	Ольга	Александровна	Старший лейтенант	Северный Ветеран	Удалить
Смирнова	Елена	Сергеевна	Младший сержант	Южный Ястреб	Удалить
Попов	Дмитрий	Андреевич	Младший солдат	Сибирская Снегурочка	Удалить

Рисунок 15.

Вывод списка всех служащих

Высоцкий	Семён	Артёмович	Генерал
Козлов	Сергей	Игоревич	Капитан
Морозов	Дмитрий	Петрович	Старшина
Иванов	Иван	Иванович	Сержант
Петров	Иван	Александрович	Сержант
Ковалева	Анна	Игоревна	Солдат
Морозова	Ирина	Дмитриевна	Майор
Иванов	Андрей	Сергеевич	Лейтенант
Кузнецова	Анна	Николаевна	Прапорщик
Козлов	Семён	Иванович	Старший сержант
Захаров	Иван	Петрович	Старший солдат
Николаева	Елена	Сергеевна	Ефрейтор
Смирнов	Антон	Иванович	Ефрейтор
Петров	Максим	Андреевич	Подполковник
Семенова	Ольга	Александровна	Старший лейтенант
Смирнова	Елена	Сергеевна	Младший сержант
Попов	Дмитрий	Андреевич	Младший солдат

Рисунок 16.

Вывод количества определённой должности в определённой воинской части

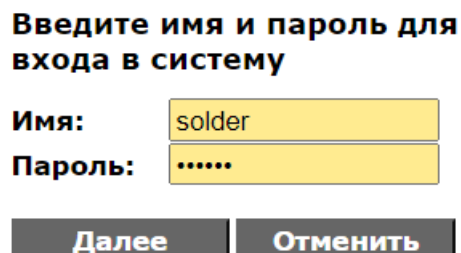
Северный щит
Сержант

Количество количество должностей = 1

Рисунок 17.

Видим, что функция администратора работает корректно.

Теперь зайдем под ролью солдата.



**Введите имя и пароль для
входа в систему**

Имя:

Пароль:

Рисунок 18.

Видим, что при редактировании таблиц выдаёт ошибку.

Редактирование списка служащих

Ошибка: ERROR: permission denied for table воинская_часть

Рисунок 19.

ЗАКЛЮЧЕНИЕ

Проект создания закрытого военного сайта — это важная задача, требующая высокой безопасности и конфиденциальности. Главная цель - обеспечить защищенное взаимодействие между военными подразделениями и передачу информации о военнослужащих и снаряжении. Реализация проекта требует создания удобного интерфейса и возможности масштабирования системы. В итоге, закрытый военный сайт улучшит эффективность и оперативность служебных действий военнослужащих. В ходе выполнения были выполнены все задачи руководителя. Цель работы также была достигнута.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. М. Фаулер. UML. Основы, 3-е издание.
2. What is Product Vision? [Электронный ресурс]. URL - <https://www.tryexponent.com/blog/what-is-product-vision>
3. Что такое построение архитектурных диаграмм? [Электронный ресурс]. URL - <https://aws.amazon.com/ru/what-is/architecture-diagramming/>
4. Функциональные диаграммы - Технология разработки программного обеспечения [Электронный ресурс]. URL - https://studref.com/311808/informatika/funktsionalnye_diagrammy

ПРИЛОЖЕНИЕ А

Поскольку наш код написан на нескольких языках программирования, разделим их на группы.

Подключение к базе данных (файл config.php)

```
<?php
// Объявление констант
$hostName      = 'pg-jail';    //Имя хоста
$Port          = '5432';      //Порт
$DatBaseName   = 'test';      //Имя базы данных
$UserName      = '';          //Имя пользовател
$Password      = '';

ini_set('display_errors', '0');

//Включение работы с глобальными переменными типа $_SESSION['xxx']
session_start();

//Чтение глобальной переменной Логин
if ( !isset($_SESSION['Login']) )
{
    $_SESSION['Login'] = 'admin';
}
$UserName = $_SESSION['Login'];

//Чтение глобальной переменной Пароль
if ( !isset($_SESSION['Password']) )
{
    $_SESSION['Password'] = 'js7gdGrnLhrvR6gkFjgD';
}
$Password = $_SESSION['Password'];

// Подключение к базе данных
function CheckBDBResult($dbconn)
{
    if ( $dbconn == false )
    {
        echo 'Ошибка: ' . pg_last_error(); //Распечатка ошибки
        die;                               //Завершение и
        Выход
    }
}
?>
```

Отрисовка главного меню (файл mainform.php)

```
<?php
echo "<table>\n";
    echo "<tr>\n";
        echo "        <td><button    class='buttonClass'    style='width:    300px;'
onClick='showForm(\"form1.php\")';>Вывод списка всех служащих</button></td>\n";
```

```

                echo "<td><button class='buttonClass' style='width: 300px;'
onClick='showForm(\"form2.php\");'>Вывод списка офицерских составов</button></td>\n";
                echo "</tr>\n";
                echo "<tr>\n";
                echo "<td><button class='buttonClass' style='width: 300px;'
onClick='showForm(\"form3.php\");'>Вывод списка всех обладателей золотого значка
ГТО</button></td>\n";
                echo "<td><button class='buttonClass' style='width: 300px;'
onClick='showForm(\"form4.php?nomer_voinsk_ch=1\");'>Вывод количества вооружения в
определённой части</button></td>\n";
                echo "</tr>\n";
                echo "<tr>\n";
                echo "<td colspan='2'><button class='buttonClass' style='width: 604px;'
onClick='showForm(\"form5.php?nv=1&rt=1\");'>Вывод количества должностей</button></td>
<!-- nv-звание rt-войсковая часть -->\n";
                echo "</tr>\n";
                echo "<tr>\n";
                echo "<td><button class='buttonClass' style='width: 300px;'
onClick='showForm(\"form6.php\");'>Редактирование таблицы служащих</button></td>\n";
                echo "<td><button class='buttonClass' style='width: 300px;'
onClick='showForm(\"form7.php\");'>Редактирование таблицы вооружений</button></td>\n";
                echo "</tr>\n";
                echo "<tr>\n";
                echo "<td colspan='2'><button class='buttonClass' style='width: 604px;'
onClick='showForm(\"form8.php\");'>Просмотр аудита</button></td>\n";
                echo "</tr>\n";
                echo "</table>\n";

                //Новая форма для отрисовки дополнительны форм
                echo "<div id='FormIdNew'></div>\n";
?>

```

Отрисовки и работа кнопок (файлы form1.php – form8.php)

```

<?php
require ('config.php'); //Чтение файла настроек

echo "<h4>Вывод списка всех служащих</h4>";

// Подключение к базе данных
$dbconn = pg_connect("host=" . $hostName . " port=" . $Port . " dbname=" . $DataBaseName . "
user=" . $UserName . " password=" . $Password);
CheckBDResult($dbconn);

// Формирование SQL запроса
$query = " select служащий.фамилия, служащий.имя, служащий.отчество,
звание.название_звания
from служащий, звание
where служащий.номер_звания = звание.номер_звания";

$result = pg_query($query);
CheckBDResult($result);

// Вывод результата в табличном виде
echo "<table>\n";

```

```

while ($line = pg_fetch_array($result, null, PGSQL_ASSOC))
{
    echo "<tr>\n";
    foreach ($line as $col_value)
    {
        echo "<td>$col_value</td>\n";
    }
    echo "</tr>\n";
}
echo "</table>\n";
// Очистка результата
pg_free_result($result);
?>

<?php
require ('config.php'); //Чтение файла настроек

echo "<h4>Вывод списка офицерских составов</h4>";
// Подключение к базе данных
$dbconn = pg_connect("host=" . $hostName . " port=" . $Port . " dbname=" . $DataBaseName . "
user=" . $UserName . " password=" . $Password);
CheckBDResult($dbconn);

// Формирование SQL запроса
$query = "select название_состава, классификация_состава from состав
        where (классификация_состава = 'Офицерский состав')";

$result = pg_query($query);
CheckBDResult($result);

// Вывод результата в табличном виде
echo "<table>\n";
while ($line = pg_fetch_array($result, null, PGSQL_ASSOC))
{
    echo "<tr>\n";
    foreach ($line as $col_value)
    {
        echo "<td>$col_value</td>\n";
    }
    echo "</tr>\n";
}
echo "</table>\n";
// Очистка результата
pg_free_result($result);
?>

<?php
require ('config.php'); //Чтение файла настроек

echo "<h4>Вывод списка всех обладателей золотого значка ГТО</h4>";
// Подключение к базе данных
$dbconn = pg_connect("host=" . $hostName . " port=" . $Port . " dbname=" . $DataBaseName . "
user=" . $UserName . " password=" . $Password);

```

```

CheckBDBResult($dbconn);

// Формирование SQL запроса
$query = "select фамилия, имя, отчество, описание from служащий as sl
        inner join характеристика_служащего as hs on hs.номер_служащего =
sl.номер_служащего
        inner join характеристика as hr on hr.номер_характеристики =
hs.номер_характеристики
        where (hr.название_характеристики = 'Значёк ГТО') and (hs.описание
= 'Золотой')";

$result = pg_query($query);
CheckBDBResult($result);

// Вывод результата в табличном виде
echo "<table>\n";
while ($line = pg_fetch_array($result, null, PGSQL_ASSOC))
{
    echo "<tr>\n";
    foreach ($line as $col_value)
    {
        echo "<td>$col_value</td>\n";
    }
    echo "</tr>\n";
}
echo "</table>\n";
// Очистка результата
pg_free_result($result);
?>

<?php
require ('config.php'); //Чтение файла настроек

echo "<h4>Вывод количества вооружения в определённой части</h4>";

$Num_voinsk_ch = $_GET['nomer_voinsk_ch']; //Чтение номера воинской части из строки
запроса

// Подключение к базе данных
$dbconn = pg_connect("host=" . $hostName . " port=" . $Port . " dbname=" . $DataBaseName . "
user=" . $UserName . " password=" . $Password);
CheckBDBResult($dbconn);

// Формирование запроса о наименования воинских частей
$query = "select номер_воинской_части, название_воинской_части from воинская_часть";
$result = pg_query($query);
CheckBDBResult($result);

// Вывод результата в выпадающий список
echo
        id='someSelect'
onChange='showFormWithParam(\"form4.php?nomer_voinsk_ch=\", \"someSelect\");'>\n";
while ($line = pg_fetch_array($result, null, PGSQL_ASSOC))
{

```

```

        if ( $line['номер_воинской_части'] == $Num_voinsk_ch )
        {
            echo "<option value=\"" . $line['номер_воинской_части']. " selected>" .
$line['название_воинской_части']. "</option>\n";
        } else
        {
            echo "<option value=\"" . $line['номер_воинской_части']. ">" .
$line['название_воинской_части']. "</option>\n";
        }
    }
    echo "</select>\n";

```

// Формирование SQL запроса с расчётом количества вооружений в выбранной воинской части

```
$query = "select kolicestvo_voorujenie (" . $Num_voinsk_ch . ")";
```

```
$result = pg_query($query);
```

```
CheckBDResult($result);
```

// Вывод результата в табличном виде

```
echo "<table>\n";
```

```
while ($line = pg_fetch_array($result, null, PGSQL_ASSOC))
```

```

{
    echo "<tr>\n";
    foreach ($line as $col_value)
    {
        echo "<td>Количество вооружений = $col_value</td>\n";
    }
    echo "</tr>\n";
}

```

```
echo "</table>\n";
```

// Очистка результата

```
pg_free_result($result);
```

```
?>
```

```
<?php
```

```
require ('config.php'); //Чтение файла настроек
```

echo "<h4>Вывод количества определённой должности в определённой воинской части</h4>";

```
$Num_zvanie = $_GET['nv']; //Чтение номера звания из строки запроса
```

```
$Num_voinsk_ch = $_GET['rt']; //Чтение номера воинской части из строки запроса
```

```
$Name_zvanie = 'Рядовой';
```

// Подключение к базе данных

```
$dbconn = pg_connect("host=" . $hostName . " port=" . $Port . " dbname=" . $DataBaseName . "
user=" . $UserName . " password=" . $Password);
```

```
CheckBDResult($dbconn);
```

// Формирование запроса о наименованиях воинских частей

```
$query = "select номер_воинской_части, название_воинской_части from воинская_часть";
```

```

$result = pg_query($query);
CheckBDResult($result);

// Вывод результата в выпадающий список
echo "<select id='someSelectIDRT' onChange='showFormWith2Param(\"form5.php\",
\"someSelectIDNV\", \"someSelectIDRT\")';>\n";
while ($line = pg_fetch_array($result, null, PGSQL_ASSOC))
{
    if ( $line['номер_воинской_части'] == $Num_voinsk_ch )
    {
        echo "<option value=\"" . $line['номер_воинской_части']. " " selected>" .
$line['название_воинской_части']. "</option>\n";
    } else
    {
        echo "<option value=\"" . $line['номер_воинской_части']. " ">" .
$line['название_воинской_части']. "</option>\n";
    }
}
echo"</select>\n";

// Фомирование запроса о наименованиях воинских званий
$query = " select номер_звания, название_звания from звание";

$result = pg_query($query);
CheckBDResult($result);

// Вывод результата в выпадающий список
echo "<select id='someSelectIDNV' onChange='showFormWith2Param(\"form5.php\",
\"someSelectIDNV\", \"someSelectIDRT\")';>\n";
while ($line = pg_fetch_array($result, null, PGSQL_ASSOC))
{
    if ( $line['номер_звания'] == $Num_zvanie )
    {
        echo "<option value=\"" . $line['номер_звания']. " " selected>" .
$line['название_звания']. "</option>\n";
        $Name_zvanie = $line['название_звания'];
    } else
    {
        echo "<option value=\"" . $line['номер_звания']. " ">" . $line['название_звания'].
"</option>\n";
    }
}
echo"</select>\n";
echo "<table>\n";
while ($line = pg_fetch_array($result, null, PGSQL_ASSOC))
{
    echo "<tr>\n";
    foreach ($line as $col_value)
    {
        echo "<td>$col_value</td>\n";
    }
    echo "</tr>\n";
}

```



```

}
echo "</table>\n";

// Формирование SQL запроса с обращением к функции БД "nazvanie_chasti"
$query = "select nazvanie_chasti (" . $Name_zvanie . ", " . $Num_voinsk_ch . ")";
$result = pg_query($query);
CheckBDRresult($result);

// Вывод результата в табличном виде
echo "<table>\n";
while ($line = pg_fetch_array($result, null, PGSQL_ASSOC))
{
    echo "<tr>\n";
    foreach ($line as $col_value)
    {
        echo "<td>Количество количество должностей = $col_value</td>\n";
    }
    echo "</tr>\n";
}
echo "</table>\n";

// Очистка результата
pg_free_result($result);
?>

<?php
require ('config.php'); //Чтение файла настроек

//Отображение страницы
echo "<h4>Редактирование списка служащих</h4>";
// Подключение к базе данных
$dbconn = pg_connect("host=" . $HostName . " port=" . $Port . " dbname=" . $DataBaseName . "
user=" . $UserName . " password=" . $Password);
CheckBDRresult($dbconn);

// Формирование SQL запроса
$query = "
select служащий.номер_служащего, служащий.фамилия, служащий.имя,
служащий.отчество, звание.название_звания, воинская_часть.название_воинской_части
from служащий, воинская_часть, звание
where (служащий.номер_состава =
воинская_часть.номер_воинской_части) and (служащий.номер_звания = звание.номер_звания)";
$result = pg_query($query);
CheckBDRresult($result);

// Вывод результата в табличном виде
echo "<table>\n";
while ($line = pg_fetch_array($result, null, PGSQL_ASSOC))
{
    echo "<tr>\n";
    foreach ($line as $Key => $col_value)
    {
        if ( $Key != 'номер_служащего' ) //Столбец 'номер_служащего' не выводим
на экран

```

```

        {
            echo "<td>$col_value</td>\n";
        }
    }
    echo " <td><button onClick=\"DeleteWarior(" . $line['номер_служащего'] .
");\">Удалить</button></td>";
    echo "</tr>\n";
}

//Вывод полей для редактирования
echo "<tr>\n";
echo "<td><input id=FamiliaId value='Иванов'></input></td>\n";
echo "<td><input id=NameId value='Иван'></input></td>\n";
echo "<td><input id=OtchestvoId value='Иванович'></input></td>\n";
echo "<td>";
    // Фомирование запроса о наименованиях воинских званий
    $query = "select номер_звания, название_звания from звание";
    $result = pg_query($query);
    CheckBDRResult($result);
    // Вывод результата в выпадающий список
    echo "<select id='ZvanieId'>\n";
    while ($line = pg_fetch_array($result, null, PGSQL_ASSOC))
    {
        echo " <option value=\" . $line['номер_звания']. ">\" .
$line['название_звания']. "</option>\n";
    }
    echo "</select>\n";
echo "</td>";

echo "<td>";
    // Фомирование запроса о наименованиях воинских частей
    $query = "select номер_воинской_части, название_воинской_части from
воинская_часть";
    $result = pg_query($query);
    CheckBDRResult($result);
    // Вывод результата в выпадающий список
    echo "<select id='VoinChastId'>\n";
    while ($line = pg_fetch_array($result, null, PGSQL_ASSOC))
    {
        echo " <option value=\" . $line['номер_воинской_части']. ">\" .
$line['название_воинской_части']. "</option>\n";
    }
    echo "</select>\n";
echo "</td>";
echo "<td><button onClick=\"WriteWarior();\">Добавить</button></td>";
echo "</tr>\n";
echo "</table>\n";
// Очистка результата
pg_free_result($result);
?>

<?php
//Запись в таблицу Служащий

```

```

require ('config.php'); //Чтение файла настроек

$Familia      = $_GET['f']; //Чтение фамилии
$Name          = $_GET['n']; //Чтение имя
$Otchestvo    = $_GET['o']; //Чтение отчество
$Zvanie        = $_GET['z']; //Чтение звание (номер)
$VoinChast     = $_GET['VCh']; //Воинская часть

// Подключение к базе данных
$dbconn = pg_connect("host=" . $HostName . " port=" . $Port . " dbname=" . $DataBaseName . "
user=" . $UserName . " password=" . $Password);
CheckBDResult($dbconn);

//Формирование запроса на запись воина
$query = "insert into служащий
        (номер_состава, номер_звания, имя, фамилия, отчество)
        values
        (" . $VoinChast . " , " . $Zvanie . " , " . $Name . " , " . $Familia . " , " .
$Otchestvo . ")";

$result = pg_query($query);
CheckBDResult($result);
pg_free_result($result);
?>

<?php
//Удаление из таблицы Служащий
require ('config.php'); //Чтение файла настроек

$WariorNumber   = $_GET['WariorNum']; //Номер служащего

// Подключение к базе данных
$dbconn = pg_connect("host=" . $HostName . " port=" . $Port . " dbname=" . $DataBaseName . "
user=" . $UserName . " password=" . $Password);
CheckBDResult($dbconn);

//Формирование запроса на запись воина
$query = "delete from служащий where номер_служащего = " . $WariorNumber;

$result = pg_query($query);
CheckBDResult($result);
pg_free_result($result);
?>

<?php
require ('config.php'); //Чтение файла настроек

//Отображение страницы
echo "<h4>Редактирование списка вооружений</h4>";
// Подключение к базе данных
$dbconn = pg_connect("host=" . $HostName . " port=" . $Port . " dbname=" . $DataBaseName . "
user=" . $UserName . " password=" . $Password);
CheckBDResult($dbconn);

```

```

// Формирование SQL запроса
$query = " select вооружение.номер_вооружения, вооружение.название_вооружения,
воинская_часть.название_воинской_части
          from вооружение, воинская_часть
          where вооружение.номер_воинской_части =
воинская_часть.номер_воинской_части";

$result = pg_query($query);
CheckBDResult($result);

// Вывод результата в табличном виде
echo "<table>\n";
while ($line = pg_fetch_array($result, null, PGSQL_ASSOC))
{
    echo "<tr>\n";
    foreach ($line as $Key => $col_value)
    {
        if ( $Key != 'номер_вооружения' ) //Столбец 'номер_вооружения' не
выводим на экран
        {
            echo "<td>$col_value</td>\n";
        }
    }
    echo " <td><button onClick=\"DeleteWeapon(" . $line['номер_вооружения'] .
");\">Удалить</button></td>";
    echo "</tr>\n";
}

//Вывод полей для редактирования
echo "<tr>\n";
echo "<td><input id=WeaponNameId></input></td>\n";
echo "<td>";
    // Формирование запроса о наименованиях воинских частей
    $query = "select номер_воинской_части, название_воинской_части from
воинская_часть";
    $result = pg_query($query);
    CheckBDResult($result);
    // Вывод результата в выпадающий список
    echo "<select id='VoinChastId'>\n";
    while ($line = pg_fetch_array($result, null, PGSQL_ASSOC))
    {
        echo "<option value=\"" . $line['номер_воинской_части']. "\"> " .
$line['название_воинской_части']. "</option>\n";
    }
    echo"</select>\n";
    echo "</td>";
    echo "<td><button onClick=\"WriteWeapon();\">Добавить</button></td>";
echo "</tr>\n";
echo "</table>\n";
// Очистка результата
pg_free_result($result);
?>

```

```

<?php
//Запись в таблицу Оружие
require ('config.php'); //Чтение файла настроек

$WeaponName      = $_GET['WeapNam'];    //Название оружия
$VoinChast       = $_GET['VCh'];        //Воинская часть

// Подключение к базе данных
$dbconn = pg_connect("host=" . $hostName . " port=" . $Port . " dbname=" . $DataBaseName . "
user=" . $UserName . " password=" . $Password);
CheckBDResult($dbconn);

//Формирование запроса на запись воина
$query = "insert into вооружение
        (название_вооружения, номер_воинской_части)
        values
        (" . $WeaponName . " , " . $VoinChast . ")";

$result = pg_query($query);
CheckBDResult($result);
pg_free_result($result);
?>

```

```

<?php
//Удаление из таблицы Служащий
require ('config.php'); //Чтение файла настроек

$WeaponNumber     = $_GET['WeaponNum']; //Номер Оружия

// Подключение к базе данных
$dbconn = pg_connect("host=" . $hostName . " port=" . $Port . " dbname=" . $DataBaseName . "
user=" . $UserName . " password=" . $Password);
CheckBDResult($dbconn);

//Формирование запроса на запись воина
$query = "delete from вооружение where номер_вооружения = " . $WeaponNumber;

$result = pg_query($query);
CheckBDResult($result);
pg_free_result($result);
?>

```

```

<?php
require ('config.php'); //Чтение файла настроек

echo "<h4>Вывод таблицы аудита</h4>";

// Подключение к базе данных
$dbconn = pg_connect("host=" . $hostName . " port=" . $Port . " dbname=" . $DataBaseName . "
user=" . $UserName . " password=" . $Password);
CheckBDResult($dbconn);

```

```

// Формирование SQL запроса
$query = "    select * from audit_of_changes";

$result = pg_query($query);
CheckBDRresult($result);

// Вывод результата в табличном виде
echo "<table>\n";
while ($line = pg_fetch_array($result, null, PGSQL_ASSOC))
{
    echo "<tr>\n";
    foreach ($line as $col_value)
    {
        echo "<td>$col_value</td>\n";
    }
    echo "</tr>\n";
}
echo "</table>\n";
// Очистка результата
pg_free_result($result);
?>

```

Окно аутентификации (файлы ChangeUsr.php и index.html)

```

<head>
<title> База данных </title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script type="text/javascript" src="libraries/js/jquery.min.js"></script>
<script type="text/javascript" src="libraries/js/scripts.js"></script>
<link href="/css/main.css" rel="stylesheet" type="text/css"></link>
</head>

<!-- Поле, на котором в дальнейшем будут отображаться таблицы -->
<div id="FormId">
    <table width="100%" height="100%">
        <th>
            <table width="228" border="0" cellpadding="0" align="center">
                <tr><td colspan="3" class="login" style="font-size:11pt;">Введите имя
и пароль для входа в систему</td></tr>
                <tr><td colspan="3" height="10"></td></tr>
                <tr>
                    <td width="110" class="login">Имя:</td>
                    <td width="160"><input id='LoginId' name="name" class="login"
type="text" style="width:135px;"></td>
                    <td width="10"></td>
                </tr>
                <tr>
                    <td align="left" class="login">Пароль:</td>
                    <td><input id='PasswordId' name="pass" class="login"
type="password" style="width:135px;"></td>
                </tr>
                <tr><td colspan="3" height="15"></td></tr>
                <tr><td colspan="3" align="left">
                    <input name="ok" onClick="ChangeUser();" type="submit"
value="Далее" class="but_log">

```

```



```

```

<?php
//Сохранение в глобальных переменных
session_start();
$_SESSION['Login'] = $_GET['NewUser'];
$_SESSION['Password'] = $_GET['NewPassword'];

require ('config.php'); //Чтение файла настроек

//Подключение к БД
$dbconn = pg_connect("host=" . $hostName . " port=" . $Port . " dbname=" . $DataBaseName . "
user=" . $UserName . " password=" . $Password);
CheckBDResult($dbconn);

require ('mainform.php'); //Отрисовка главного меню

?>

```

Скрипты (файл scripts.js)

```

// Смена пользователя
function ChangeUser()
{
    var Login = document.getElementById('LoginId');
    var Password = document.getElementById('PasswordId');
    var urlChangeUser = "ChangeUsr.php?NewUser=" + Login.value + "&NewPassword=" +
Password.value;
    $.get(urlChangeUser, function(data, status) {
        var result = document.getElementById('FormId');
        if ( result != null ) {
            if ( status == "success" ) {
                result.innerHTML = data;
            } else {
                result.innerHTML = "ошибка выполнения запроса";
            }
        }
    }, "html");
};

// Отрисовка дополнительной формы на новой форме
function showForm(url) {
    $.get(url, function(data, status) {
        var result = document.getElementById('FormIdNew');
        if ( result != null ) {

```

```

    if ( status == "success" ) {
        result.innerHTML = data;
    } else {
        result.innerHTML = "ошибка выполнения запроса";
    }
}
}, "html");
};

```

// Отрисовка дополнительной формы на новой форме с дополнительным параметром

```

function showFormWithParam(url, someID) {
    var resultID = document.getElementById(someID);
    if ( resultID != null ) {
        url = url + resultID.value;
        $.get(url, function(data, status) {
            var result = document.getElementById('FormIdNew');
            if ( result != null ) {
                if ( status == "success" ) {
                    result.innerHTML = data;
                } else {
                    result.innerHTML = "ошибка выполнения запроса";
                }
            }
        }
    }, "html");
};

```

// Отрисовка дополнительной формы на новой форме с дополнительными двумя параметрами

```

function showFormWith2Param(url, someID1, someID2) {
    var resultID1 = document.getElementById(someID1);
    if ( resultID1 != null ) {
        var resultID2 = document.getElementById(someID2);
        if ( resultID2 != null ) {
            url = url + "?nv=" + resultID1.value + "&rt=" + resultID2.value;
            $.get(url, function(data, status) {
                var result = document.getElementById('FormIdNew');
                if ( result != null ) {
                    if ( status == "success" ) {
                        result.innerHTML = data;
                    } else {
                        result.innerHTML = "ошибка выполнения запроса";
                    }
                }
            }
        }, "html");
    }
};

```

// Добавление служащего

```

function WriteWarior()
{

```



```

var Familia = document.getElementById('FamiliaId');
var Name = document.getElementById('NameId');
var Otchestvo = document.getElementById('OtchestvoId');
var Zvanie = document.getElementById('ZvanieId');
var VoinChast = document.getElementById('VoinChastId');

var urlWriter = "form61.php?f=" + Familia.value + "&n=" + Name.value + "&o=" +
Otchestvo.value + "&z=" + Zvanie.value + "&VCh=" + VoinChast.value;
$.get(urlWriter, function(data, status)
{
    var result = document.getElementById('FormIdNew');
    if ( result != null )
    {
        if ( status == "success" )
        {
            $.get("form6.php", function(data, status)
            {
                if ( status == "success" )
                {
                    result.innerHTML = data;
                } else
                {
                    result.innerHTML = "ошибка выполнения запроса";
                }
            }, "html");
        } else
        {
            result.innerHTML = "ошибка запроса для записи в БД";
        }
    }
}, "html");

};

// Удаление служащего
function DeleteWarior(WariorNumber)
{
    var urlDeleter = "form62.php?WariorNum=" + WariorNumber;
    $.get(urlDeleter, function(data, status)
    {
        var result = document.getElementById('FormIdNew');
        if ( result != null )
        {
            if ( status == "success" )
            {
                $.get("form6.php", function(data, status)
                {
                    if ( status == "success" )
                    {
                        result.innerHTML = data;
                    } else
                    {
                        result.innerHTML = "ошибка выполнения запроса";

```

```

        }
        }, "html");
    } else
    {
        result.innerHTML = "ошибка запроса для записи в БД";
    }
}
}, "html");
};

// Добавление оружия
function WriteWeapon()
{
    var WeaponName = document.getElementById('WeaponNameId');
    var VoinChast = document.getElementById('VoinChastId');
    var urlWriter = "form71.php?WeapNam=" + WeaponName.value + "&VCh=" +
VoinChast.value;
    $.get(urlWriter, function(data, status)
    {
        var result = document.getElementById('FormIdNew');
        if ( result != null )
        {
            if ( status == "success" )
            {
                $.get("form7.php", function(data, status)
                {
                    if ( status == "success" )
                    {
                        result.innerHTML = data;
                    } else
                    {
                        result.innerHTML = "ошибка выполнения запроса";
                    }
                }, "html");
            } else
            {
                result.innerHTML = "ошибка запроса для записи в БД";
            }
        }
    }, "html");
};

// Удаление Оружия
function DeleteWeapon(WeaponNumber)
{
    var urlDeleter = "form72.php?WeaponNum=" + WeaponNumber;
    $.get(urlDeleter, function(data, status)
    {
        var result = document.getElementById('FormIdNew');
        if ( result != null )
        {

```

```

if ( status == "success" )
{
    $.get("form7.php", function(data, status)
    {
        if ( status == "success" )
        {
            result.innerHTML = data;
        } else
        {
            result.innerHTML = "ошибка выполнения запроса";
        }
    }, "html");
} else
{
    result.innerHTML = "ошибка запроса для записи в БД";
}
}, "html");
};

```