

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Забайкальский государственный университет»
(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

КУРСОВОЙ ПРОЕКТ

по дисциплине: Технология разработки программного обеспечения

на тему «Web-приложение ведения складского учета»

Выполнил ст. гр. ИВТ-18
Забаровский Р.В

Проверил доцент кафедры ИВТ и ПМ,
к.т.н., доцент Валова О.В

Чита

2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Забайкальский государственный университет»
(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

ЗАДАНИЕ

на курсовой проект

По дисциплине: Технология разработки программного обеспечения
Студенту Забаровскому Роману Вадимовичу

направление подготовки 09.03.01 Информатика и вычислительная техника

1 Тема курсового проекта: Web-приложение ведения складского учета

2 Срок подачи студентом законченной работы: 06.04.2022 г.

3 Исходные данные к проекту: описание предметной области

4 Перечень подлежащих разработке в курсовом проекте вопросов:

1. Постановка и анализ задачи;
2. Анализ задачи;
3. Программная реализация;
4. Тестирование;
5. Техническое задание;
6. Руководство пользователя.

5 Перечень графического материала: -

Дата выдачи задания 09.02.2022 г.

Руководитель курсового проекта _____ / Валова О.В./
(подпись, расшифровка подписи)

Задание принял к исполнению

«09» февраля 2022 г.

Подпись студента _____ / Забаровский Р.В./

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Забайкальский государственный университет»
(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту

по направлению подготовки 09.03.01 Информатика и вычислительная техника
по дисциплине: Технология разработки программного обеспечения
на тему: Web-приложение ведения складского учета

Выполнил студент группы ИВТ-18 Забаровский Роман Вадимович

Руководитель работы: доцент кафедры ИВТ и ПМ, к.т.н., доцент Валова Ольга Валерьевна

РЕФЕРАТ

Пояснительная записка - 31 с, 11 рис., 7 таб., 3 источников.

ПОЛЬЗОВАТЕЛЬ, АДМИНИСТРАТОР, ТОВАР, БАЗА ДАННЫХ, PHP, MYSQL, СУБД, WEB-ПРИЛОЖЕНИЕ, WEB-ПРИЛОЖЕНИЕ.

Целью проекта является разработка веб-приложения для работников складских помещений.

Поставленная цель достигается путем создания программной части системы. Кроме этого была разработана структура хранения данных проекта.

Данная программа предназначена для автоматизации процессов на предприятии по проведению приема и отгрузки товаров, ведения учета товаров, оформление заказов на товары и услуги.

СОДЕРЖАНИЕ

РЕФЕРАТ	4
ВВЕДЕНИЕ	6
1 Постановка и анализ задачи.....	7
1.1 Описание предметной области.....	7
1.2 Диаграмма Вариантов использования.....	9
1.3 Обоснование выбора средств реализации.....	11
2 Анализ данных	13
2.1 Входные данные	13
2.2 Промежуточные данные	14
2.3 Выходные данные.....	14
3 Программная реализация	16
4 Тестирование	21
5 Документирование.....	23
5.1 Техническое задание	23
5.2 Руководство пользователя	25
ЗАКЛЮЧЕНИЕ	28
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	29
ПРИЛОЖЕНИЯ	30

ВВЕДЕНИЕ

В настоящее время многие организации стараются перенести организационные и производственные аспекты на электронные системы автоматизации, так как хранение информации на бумагах становится все менее выгодно и удобно. Поэтому возможным выходом из данной ситуации является создание сервисов для структуризации и обработки данных.

Множество программ, которые предоставляют возможность вести учет, существующих на современном рынке пользуются спросом в отдельных сферах бизнеса, но они либо не имеют гибкости подстраиваться под нужды организаций, либо стоимость их использования не рациональна для малых бизнесов. Поэтому некоторым организациям проще самостоятельно разработать свою систему, под свои нужды, нежели заимствовать существующее.

Целью данной работы является создание сервиса веб-приложения, которая позволит вести складской учет, то есть прием товаров, отгрузку, оформление заказов, проводить инвентаризацию и оформлять соответствующие документы.

В качестве языка программирования для реализации веб-приложения будут использоваться: PHP, HTML, CSS, JavaScript.

1 Постановка и анализ задачи

1.1 Описание предметной области

За последнее двадцатилетие наука значительно продвинулась в изучении информационных технологий. Во многом производственная и бумажная работа упрощается при их использовании на этапе создания, но и не ограничивается на последующих этапах распределения, поставок, хранения, продажи и обслуживания.

Данная работа выполнена для «Общества с ограниченной ответственностью “Свежий ветер”». Предприятие занимается монтажом систем вентиляции и кондиционирования, установкой холодильного оборудования, технологическим проектированием, монтажом инженерных сетей, к которым относятся: водоснабжение, отопление, электроснабжение, канализация, системы автоматизации и управления.

Цель разработки web-приложения – оперативный учет складских операций, печать складских документов, анализ количества запасов, их стоимости. В данном приложении необходимо предусмотреть разный функционал для двух видов пользователей: администратор базы данных, складской рабочий. Также в процессе разработки web-приложения должны быть приобретены навыки проектирования баз данных, требующиеся для полной работоспособности системы приложения.

Одним из важных шагов при разработке приложения, - это выбор правильного стека технологий. От него не в последнюю очередь зависит успех всего проекта. Он влияет на количество материальных вложений в проект, сроки разработки, безопасность и модифицируемость. В первую очередь перед тем как говорить о выборе стека технологий, нужно пояснить, что такое веб-приложение – это сайт, в котором клиент взаимодействует с веб-сервером при помощи браузера, то есть может нажимать кнопки,

вводить данные в формы, оформлять документы и т.п. Примером веб-приложения могут выступать интернет магазины, социальные сети, поисковые службы и т.д.

Архитектура веб-приложения состоит из двух частей. Первая клиентская – она реализует пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы. Также называется фронтенд (front-end). Вторая серверная – она реализует логику. Также называется бэкенд (back-end). К технологиям фронтенда относятся HTML, CSS, JavaScript, а бэкенда относят PHP, Java, Python и др.

Подход к разработке также делится на несколько видов:

- Single Page Application (SPA) или веб-приложение с одной страницей.
- Multi Page Application (MPA) – многостраничные приложения.
- Progressive Web Application (PWA) – гибрид сайта и мобильного приложения.

В настоящее время множество предприятий уже используют электронные системы, для эффективной работы. Например, такой гигант как «1С: Торговля и склад» удерживает лидерство в предпочтении крупных предприятий, но малый бизнес предпочитает платить меньшие деньги за обслуживание и функциональное насыщение. В результате чего на рынке существует сильный спрос на аналоги данного программного обеспечения, которые проще в функциональной части. На что разработчики выпустили множество простых и понятных программ складского учета для определенных торговых специфик: «ЕКАМ», «Дебет плюс», «МойСклад» и др. Но основной проблемой данных программ является большой разброс цены и возможности настройки предприятием программы для своего направления, если оно имеет специфичные товары и услуги.

1.2 Диаграмма Вариантов использования

В веб-приложении предусмотрено два уровня доступа к данным, поэтому пользователи системы поделены на две группы. На диаграмме вариантов использования они представлены в виде двух действующих лиц:

- администратор базы данных;
- пользователь.

Для работы веб-приложения необходимо реализовать функции, показанные на диаграмме использования для администратора и пользователя.

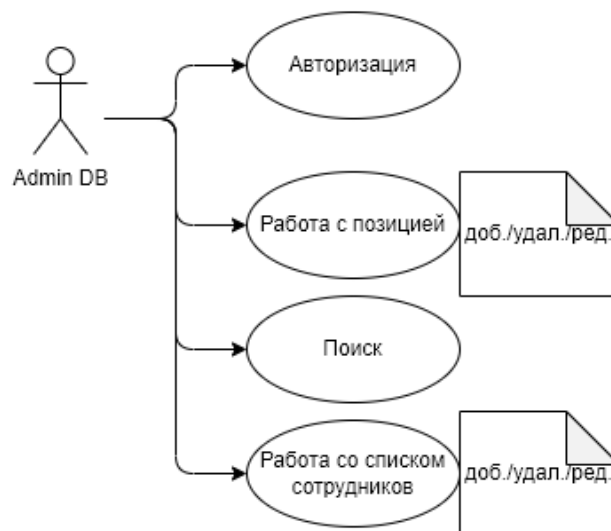


Рисунок 1 - Диаграмма вариантов использования для администратора

Для действующего лица администратора предполагаются следующие варианты использования:

- авторизация, которая предусматривает поддержание сессии текущего пользователя;
- работа с позицией, которая предусматривает добавление, удаление и редактирование типов товаров в базе данных;
- пользование поиском;

– работа со списком сотрудников, которая предусматривает добавление, удаление и редактирования сотрудников в базе данных.

Администратор веб-приложения перед началом работы должен авторизоваться, после этого ему будут доступны функции для работы со списком товаров и списком сотрудников. Диаграмма вариантов использования администратора представлена на рисунке 1.



Рисунок 2 - Диаграмма вариантов использования для пользователя

Для пользователя веб-приложения предполагаются следующие варианты использования:

- авторизация, которая предусматривает поддержание сессии текущего пользователя;
- работа с товаром, которая предусматривает прием, отгрузку;
- оформление накладной, которая предусматривает работу с документом на товар: резервирование под заказ, объединение в один заказ;
- инвентаризация, которая предусматривает инвентаризацию склада, списание излишков и переоценку товаров;

– учет, которая предусматривает просмотр атрибутов товаров, характерных для них.

Также, как и администратор пользователь перед началом должен авторизоваться, после этого ему станут доступны функции работы с товаром, оформления накладных, инвентаризации и учета. Диаграмма использования для пользователя представлена на рисунке 2.

В свою очередь работу с товаром, оформление накладной и инвентаризацию можно представить в виде отдельных диаграмм. Диаграммы вариантов использования представлены на рисунке 3.

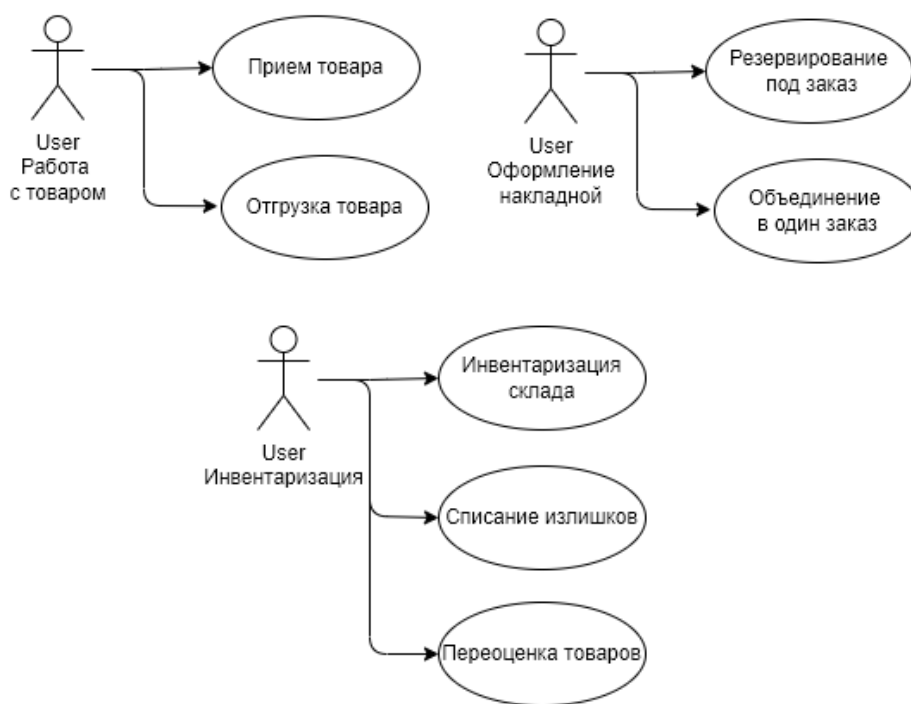


Рисунок 3 - Диаграммы вариантов использования

1.3 Обоснование выбора средств реализации

Реализация веб-приложения произведена на следующем стеке технологий:

- язык гипертекстовой разметки HTML;
- формальный язык внешнего вида документа CSS;

- скриптовый язык общего назначения PHP;
- СУБД PostgreSQL. На данной СУБД разработана база данных для хранения данных о товарах и сотрудниках;
- язык клиентской части веб-приложения JavaScript;
- все страницы в проекте динамические и построены на шаблонах.

Системой управления базами данных (СУБД) была выбрана PostgreSQL. Эта СУБД является бесплатной и её достаточно просто настроить для работы с базами данных. Сервером базы данных была выбрана вычислительная машина под управлением операционной системы семейства Windows, но также сервером может выступать машина под управлением других операционных систем таких как Linux и macOS.

Шаблонизация заключается в следующем: разделение представления данных от исполняемого кода. Для этого создаются отдельные файлы (шаблоны) HTML для каждого элемента, содержимое которого может отличаться. Описав шаблон единожды, его можно использовать множество раз, а изменения в шаблоне будут видны везде, где он используется.

Языком программирования для реализации веб-приложения были выбраны: PHP, JavaScript. PHP – это скриптовый язык общего назначения, с его помощью серверная часть принимает запросы, работает с базой данных, производит необходимую проверку данных и позволяет идентифицировать пользователя с помощью сессий. Его преимущества: большой набор встроенных средств для разработки веб-приложений, бесплатен, большое количество открытой документации, простой синтаксис. JavaScript – динамический язык, используемый для создания интерактивных веб-приложений. Он используется в клиентской части приложений: клиент-серверных программ, где клиентом выступает браузер, а сервером – веб-сервер, имеющих распределенную логику. Выбор представлен эффективностью и возможностями языков программирования.

2 Анализ данных

Данные, с которыми работает система делятся на три группы. Первая группа входных данных – данные, которые поступают от пользователя в систему. Вторая группа промежуточных данных – данных, которые система использует во время своей работы. И третья группа выходных данных – данные, которые система выводит пользователю.

2.1 Входные данные

Входной информацией в разрабатываемом приложении будут выступать такие данные:

- логин;
- пароль;
- информация о пользователе: имя, фамилия, дата рождения, телефонный номер, роль;
- информация о товаре: наименование, материал, длина, размер сечения, количество.

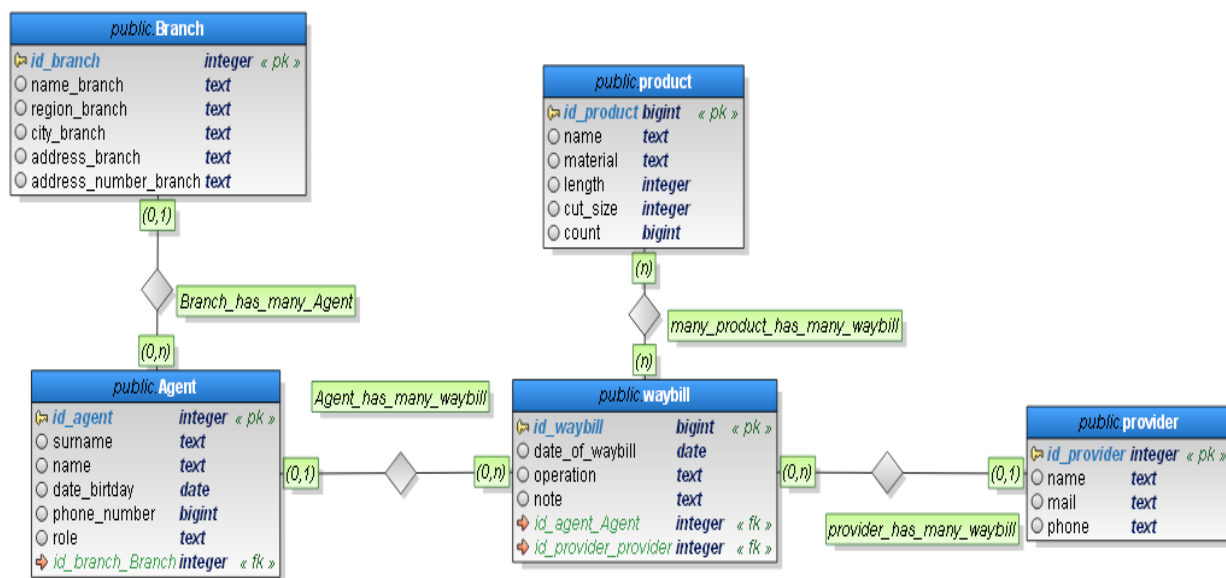


Рисунок 4 – Модель базы данных

Каждая таблица базы данных, которая была разработана состоит из атрибутов для хранения информации о том или ином объекте предметной области. Связи между таблицами показаны на рисунке 4.

2.2 Промежуточные данные

Во время работы приложения промежуточными данными являются таблицы базы данных. Вся информация о данных, содержащихся в таблицах, находится в приложении А.

Таблица Admin содержит данные о логине и пароле администратора базы данных для входа в приложение и личную информацию о пользователе, такую как имя, фамилия, дата рождения, телефонный номер.

Таблица Agent содержит информацию о паролях и логинах пользователей, а также личную информацию: имя, фамилию, дату рождения, телефонный номер.

Таблица Product хранит информацию о всех товарах, такую как наименование, материал, размер сечения, длина, количество, цена.

Таблица Waybill хранит данные о всех операциях, совершенных пользователем.

Таблица Provider содержит информацию о поставщиках, их имя, электронную почту и телефонный номер.

Таблица Branch хранит информацию о всех складских помещениях организации.

2.3 Выходные данные

К выходным данным можно отнести:

- информационные веб-страницы веб-приложения;
- списки товаров;
- списки операций;
- списки поставщиков;

- списки пользователей;
- отчеты.

Пример отчета приведен на рисунке 11, в разделе «Руководство пользователя».

3 Программная реализация

В ходе программной реализации разработаны алгоритмы для авторизации, внесения данных в базу данных, изменения данных в базе данных, шаблонизации страниц. Приложение имеет модульную структуру и содержит несколько модулей, которые представлены в виде диаграммы компонентов в соответствии с рисунком 5.

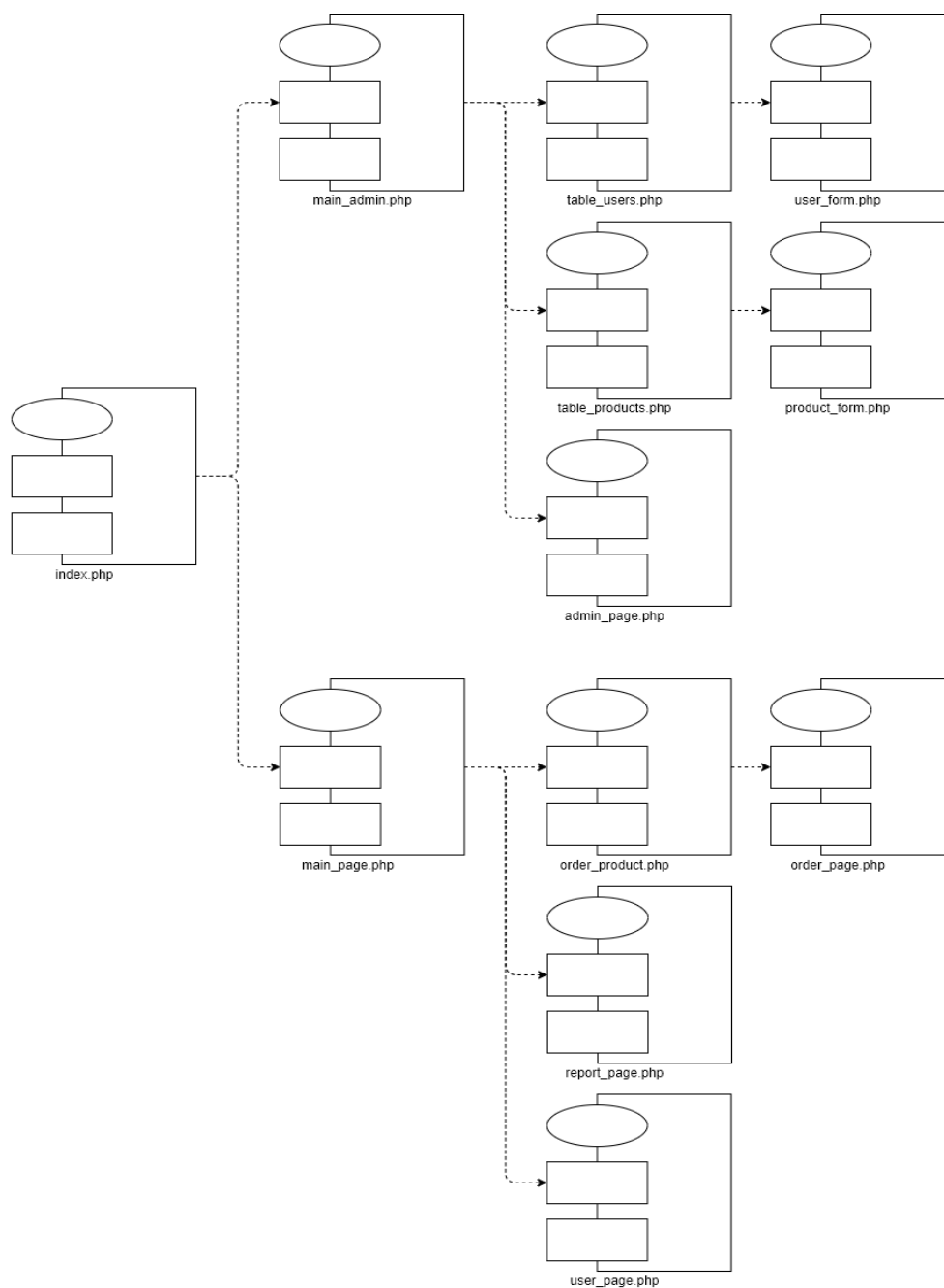


Рисунок 5 – Диаграмма компонентов

Таблица 1 – Описание компонентов

Компонент	Назначение
index.php	Точка входа в приложение
main_admin.php	Модуль, главной страницы администратора
main_page.php	Модуль, главной страницы пользователя
table_users.php	Модуль, со списком пользователей
table_product.php	Модуль, содержащий список товаров и услуг
admin_page.php	Модуль, личной страницы администратора
user_form.php	Модуль, содержащий данные о пользователе
product_form.php	Модуль, содержащий данные о товаре
order_product.php	Модуль, оформления товара или услугу
report_page.php	Модуль, отвечающий за создание отчетного документа
user_page.php	Модуль, личной страницы пользователя
order_page.php	Модуль, страницы заказа

При авторизации для каждого пользователя в зависимости от уровня доступа загружается свой интерфейс. На странице авторизации загружается модуль layout.php, в котором для не авторизованного пользователя также подгружается свой интерфейс из модуля navigation_template.php. Пример алгоритма:

```
require_once 'database.php';
require_once 'query.php';

if (isset($_POST['name']) && !empty($_POST['name'])) {
    $login = $_POST["name"];
    $pass = $_POST["password"];

    $db = new DB();

    $query = "select \"surname\" from \"Agent\"
    where '$login' = \"surname\" and '$pass' = \"pass\" ";
    $query_result = pg_query($db->get_link(), $query);
```

```

$result = pg_fetch_all($query_result, PGSQL_ASSOC);

$role = get_role($login);
var_dump($result);
if (!empty($result)) {
    session_start();
    $_SESSION['name'] = $login;
    if ($role[0]["role_personal"] == "admin") {
        header('Location:http://localhost/admins/main_admin_page.php')
;    } else {
        header('Location:http://localhost/users/main_page.php');
    }
} else{
    $error = "reg error";
    header("Location:http://localhost/index.php?error=$error");
}
}

```

Если авторизовался пользователь с правами администратора, то модуль layout.php подгрузит модуль navigation_admin_template.php, который отвечает за отображение навигационного меню администратора, а модуль log-in.php подгрузит модуль главной страницы администратора main_admin_page.php, где содержатся таблицы пользователей, товаров, поставщиков и операций.

Если авторизовался пользователь с правами пользователя, то модуль layout.php подгрузит модуль navigation_user_template.php, который отобразит навигационное меню для роли пользователь. Модуль log-in.php же подгрузит модуль main_user_page.php, на которой содержатся элементы пользовательской страницы.

Для внесения данных в базу данных и изменения их в ней были разработаны алгоритмы для связи с базой данных. Модуль database.php открывает подключение к базе данных по заранее известным данным, таким как: логин, пароль, название базы данных, адрес хоста, порт подключения. Открытие соединения для работы с базой данных происходит на этапе авторизации, а закрытие при выходе из системы, либо автоматически через определенный интервал времени. Внесение и изменения данных происходит через модуль query.php, который используется в интерфейсе пользователя, вызовом методов insert и update, либо delete. Ниже приведен алгоритм запросов из базы данных.

```

function query($query){
    $db = new DB();
    return pg_query($db->get_link(), $query);
}

function query_fetch_all($query){
    return pg_fetch_all(query($query), PGSQL_ASSOC);
}

function get_role($surname){
    return pg_fetch_all(query("select \"role_personal\" from
    \"Agent\" where '$surname' = \"surname\""));
}

function get_user($surname){
    return pg_fetch_all(query("select * from \"Agent\" where
    '$surname' = \"surname\""));
}

```

Для построения отображаемых страниц был разработан алгоритм шаблонизации, в который передаются модули шаблонов и данных на них отображаемых. Пример алгоритма:

```

function template($file, $args){

    // проверяем, существует ли файл
    if ( !file_exists( $file ) ) {
        //echo $file;
        return 'Файла не существует';
    }

    // распаковываем массив
    if ( is_array( $args ) ){
        extract( $args );
    }

    // буфферезируем вывод
    ob_start();
    include $file;
    return ob_get_clean();
}

// Рендер страницы с контентом
function render_page($title, $content){
    // Формируем страницу
    $file = dirname(__DIR__, 1).'/templates/layout.php';
    //echo $file;
    $content = ['title' => $title, 'content' => $content];
    $page = template($file, $content);
    print $page;
}

```

```
// Рендер титульной страницы
function render_template_index_form($args) {
    return
    template(dirname(__DIR__, 1).'/templates/index_form_template.php',
    $args);
}
```

Пример работы данного алгоритма можно посмотреть на рисунке 8, в разделе «Руководство пользователя»

4 Тестирование

По мере разработки программного продукта проводились многочисленные тесты для выявления ошибок и уязвимостей системы.

Первый тест был направлен на то, чтобы при неверной комбинации логина и пароля система должна оповестить об этом пользователя.

Если пользователь ввел неверную комбинацию логина и пароля, то он увидит следующее сообщение, как на рисунке 6.

Неверный логин или пароль

Логин

Пароль

Вход

Рисунок 6 – Ошибка введенного логина или пароля

Второй тест направлен на уведомление пользователя о незаполненных полях товаров и услуг. Если пользователь не заполнит ключевые поля, то получит ошибку, как показано на рисунке 7.

Заполните все поля

Наименование	Цена	кол-во
Сток круг. 100ø	350 руб.	2
Утка квадр. 150 ø		1
Оформление		

Рисунок 7 – Ошибка о пустых полях

Результаты тестирования подтверждают, что данные методы защиты работают и повышают работоспособность системы.

На текущий момент система находится в стадии разработки, следовательно, при каждом добавление новых функций происходит тестирование и отладка.

5 Документирование

5.1 Техническое задание

5.1.1. Введение

Проект «Web-приложение для ведения складского учета» предоставляет возможность оформления товаров и услуг складским работником. Программа представлена в виде веб-сервиса.

5.1.2. Назначение разработки

Данный сервис разрабатывается с целью оптимизации работы компании и повышению продаж.

5.1.3. Требования к функциональным характеристикам

В данном сервисе, в соответствии с диаграммой использования предусмотрены два действующих лица:

Администратору приложения программный продукт должен предоставить следующие возможности, связанные с полноценным администрированием веб-сервиса:

- авторизация;
- редактирование списка товаров;
- редактирование списка сотрудников.

Пользователю, как сотруднику организации, требуется другой функционал. Веб-сервис должен предоставить следующие функциональные возможности:

- авторизация;
- возможность приема, отгрузки товара;
- возможность оформления накладной;
- возможность списания излишков;

- возможность переоценки товаров;
- возможность учета товаров.

5.1.4. Требования к надежности

Веб-сервер должен уметь обрабатывать неправильные запросы и отправлять пользователю уведомления об ошибках. Для изучения ошибок и последующего устранения, необходимо заносить сведения обо всех ошибках в журнал ошибок.

При возникновении несерьезных ошибок, которые пользователь может исправить самостоятельно, такие как некорректный ввод данных, сервис должен оповестить об этом пользователя.

5.1.5. Системные требования

Наличие в системе одного из интернет-обозревателей с версиями не ниже указанных:

- Edge 12;
- Firefox 21;
- Chrome 23.

5.1.6. Требования к программной документации

Программная документация должна содержать исчерпывающее руководство пользователя с полным описанием действий пользователя.

5.1.7. Требования к составу и параметрам технических средств

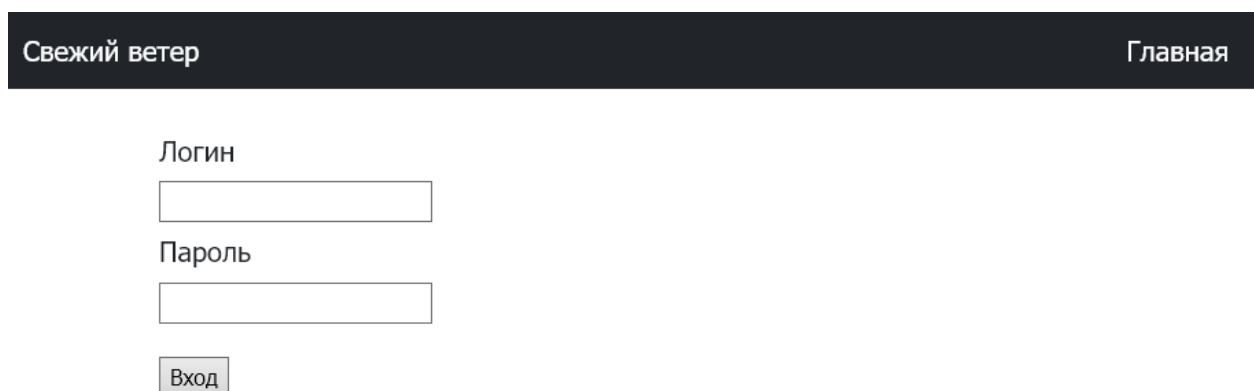
Минимальные требования для корректной работы сервиса необходимо: персональный компьютер с любой операционной системой, встроенной оперативной памятью не менее 512 Мб, также свободного места на диске должно быть не менее 550 Мб, процессор с частотой 600 МГц.

5.1.8. Требования к информационной и программной совместимости

Сервер не требует определенной операционной системы. На рабочей станции должен быть установлен любой браузер, подходящий по системным требованиям, описанным выше.

5.2. Руководство пользователя

При запуске веб-сервиса пользователь попадает на страницу авторизации, которая содержит поля для ввода логина и пароля. Страница авторизации представлена на рисунке 8.



Свежий ветер

Главная

Логин

Пароль

Вход

Рисунок 8 – Страница авторизации

После страницы авторизации пользователю в зависимости от роли будут доступны функции работы с товаром, со списком сотрудников либо работа с товаром, прием товара, отгрузка товара, резервирование под заказ и объединение в один заказ. Пример того как будет выглядеть главная страница представлен на рисунке 9. В верхнем правом углу система оповещает пользователя, что он зашел в свой аккаунт и может его покинуть по своему желанию.

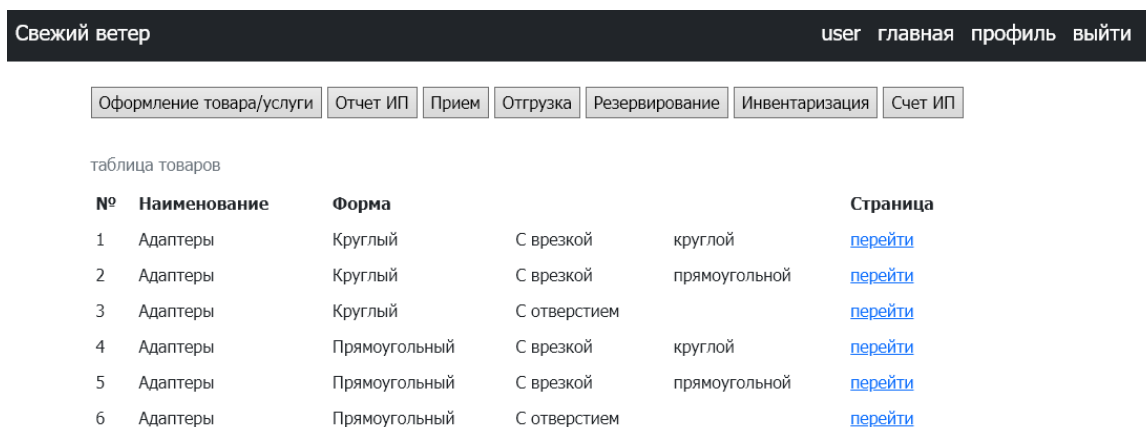


Рисунок 9 – Главная страница пользователя

Для того чтобы оформить товар или услугу нужно нажать на соответствующую кнопку, представленную на главной странице и выбрать товар или услугу, нажав кнопку оформить на странице товара или услуги. Для того чтобы добавить несколько позиций необходимо на странице со списком добавленных товаров нажать кнопку добавить, пример можно посмотреть на рисунке 10.



Рисунок 10 – Список добавленных товаров

После выбора всех нужных товаров либо услуг можно перейти к оформлению, нажав соответствующую кнопку на интерфейсе. В результате пользователь получит документ готовый к печати. Пример документа представлен на рисунке 11.

ПРЕДЛОЖЕНИЕ

Счет на оплату № 24 от 1 апреля 2022 г.

Поставщик ООО "Свежий ветер"

Покупатель

(Заказчик): 0. Без скидки

Основание:

№	Товары (работы, услуги)	Кол-во	Ед.	Цена	Сумма
1	Зонт- Ф100 Оцин 0,5	2	шт.	350	624
2	Хомутподвесной Ф50 Нерж. 0,5	4	шт.	85	340
3	Ниппель Ф100 Оцин 0,5 L=100	3	шт.	85	255

Итого: 1219,00

В том числе НДС: 203,17

Всего к оплате: 1219,00

Всего наименований 3, на сумму 1219,00 руб.

Одна тысяча двести девятнадцать рублей 00 копеек

Рисунок 11 – Пример документа

ЗАКЛЮЧЕНИЕ

Целью работы было создание веб-приложения для ведения складского учета организации.

В ходе работы было создано веб-приложение. В приложении реализованы такие возможности как возможность работы с товаром, оформление товаров в заказ, оформление услуг, создание документов отчетов организации.

Цель была достигнута путем выполнения следующих задач:

- реализованы модули веб-приложения;
- создан интерфейс взаимодействия пользователя и программы при помощи языков программирования PHP и JavaScript;
- создано руководство пользователя.

В дальнейшем возможно развитие программы для более удобного использования и добавления функциональных возможностей, таких как экспорт для 1С, применения скидок и наценок, отчетность по материалам, конвертер масс, калькулятор сечений и др.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- 1 PostgreSQL [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/PostgreSQL> (дата обращения 11.03.2022).
- 2 Документация PHP [Электронный ресурс]. – Режим доступа: <https://www.php.net/manual/ru/> (дата обращения 29.03.2022).
- 3 Веб–приложение [Электронный ресурс]. – Режим доступа: <https://semantica.in/blog/veb–prilozhenie.html> (дата обращения 01.04.2022).

ПРИЛОЖЕНИЕ А

(обязательное)

Описание таблиц базы данных

Таблица А.1 – Таблицы Базы данных

Название таблицы	Описание
Branch	Информация об отделениях
Agent	Информация о пользователях
product	Информация о товарах
waybill	Накладные
provider	Информация о поставщиках

Таблица А.2 – Описание таблицы Branch

Название	Тип	Описание
id_branch	integer	Идентификатор
name_branch	varchar	Название отделения
region_branch	varchar	Регион отделения
city_branch	varchar	Город отделения
address_branch	varchar	Название улицы отделения
address_number_branch	varchar	Номер улицы отделения

Таблица А.3 – Описание таблицы Agent

Название	Тип	Описание
id_agent	integer	Идентификатор
surname	varchar	Фамилия
name	varchar	Имя
date_birthday	date	Дата рождения
phone_number	varchar	Телефонный номер
role_personal	varchar	Роль пользователя
pass	varchar	Пароль пользователя
id_branch_Brarnch	integer	Идентификатор отделения

Таблица А.4 – Описание таблицы product

Название	Тип	Описание
id_product	integer	Идентификатор
name_product	varchar	Наименование
material	varchar	Материал
length	integer	Длина
cut_size	integer	Размер сечения
count	bigint	Количество
price	bigint	Цена за единицу

Таблица А.5 – Описание таблицы waybill

Название	Тип	Описание
id_waybill	integer	Идентификатор
operation	varchar	Название операции
date_of_waybill	varchar	Дата накладной
note	integer	Пояснение
id_agent_FK	integer	Идентификатор пользователя
id_provider_FK	bigint	Идентификатор поставщика

Таблица А.6 – Описание таблицы provider

Название	Тип	Описание
id_provider	integer	Идентификатор
name_provider	varchar	Имя поставщика
mail	varchar	Почта
phone	varchar	Телефонный номер
address	varchar	Адрес