ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ «САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М. А. БОНЧ-БРУЕВИЧА»

Факультет ИКСС

Курсовая работа по дисциплине

«Web-технологии»

Разработка информационной системы для управлеия клиентской базой данных и автоматизации процесса продаж
(тема отчета)

Направление/специальность подготовки
09.03.04 Программная инженерия
(код и наименование направления/специальности)

Студент:

Дунаев В.Е. ИКПИ-11

Преподаватель:

Краева Е.В.

Санкт-Петербург
2023

Содержание

1. Актуальность

В наше время все больше и больше компаний реализуют свою деятельность через интернет и используют онлайн-платформы для привлечения клиентов. Учитывая растущую потребность в удобных и эффективных инструментах для учета заказов, создание специализированного приложения для учета покупок и автоматизации процесса продаж становится все более актуальным

Такое приложение позволяет компаниям автоматизировать и оптимизировать процесс учета заказов, значительно улучшая операционную эффективность. Оно предоставляет удобный интерфейс для ввода и обработки заказов, отслеживания их статуса, управления их исполнением, а также генерации отчетности.

Одним из главных преимуществ такого приложения является высокая точность и надежность учета, поскольку все данные записываются и обрабатываются автоматически, исключая возможность человеческой ошибки. Это также значительно сокращает время и ресурсы, которые ранее требовались для ручного ведения учета заказов.

Приложение для учета автоматизации процесса продаж также способствует улучшению коммуникации между различными отделами компании, поскольку все данные и информация о заказах доступны в реальном времени и могут быть легко обменены между сотрудниками. Это позволяет лучше организовать бизнес-процессы и минимизировать возможность возникновения конфликтов или несогласованностей.

Еще одним значимым аспектом такого приложения является повышение уровня обслуживания клиентов. Благодаря быстрому и эффективному управлению покупательской базой данных компании могут оперативно и точно обрабатывать запросы клиентов, сокращая время ожидания и повышая общую удовлетворенность клиента.

В целом, создание приложения для автоматизации процесса продаж является не только актуальной, но и весьма необходимой мерой для компаний, стремящихся к повышению эффективности своей деятельности. Оно помогает автоматизировать процессы, повышает точность и надежность учета, улучшает коммуникацию и уровень обслуживания клиентов. Такое приложение становится незаменимым инструментом для компаний, заинтересованных в оперативности и успешности своих заказов.

2. Используемое ПО

- Язык разметки HTML
- Язык стилей CSS
- Язык программирования Java
- TomCat Server – это веб-сервер с открытым исходным кодом, предназначенный для обработки и доставки веб-страниц и других ресурсов на основе протокола HTTP. Он является одним из самых популярных и широко используемых серверов веб-приложений и часто используется в сочетании с языком программирования Java.
- Intellij IDEA: Мощный текстовый редактор для разработки на Java и . Имеет большой выбор расширений
- Google Chrome: Веб-браузер, обладающий различными инструментами разработчика, которые упрощают отладку и анализ веб-страниц

3. Цель работы

Разработать клиентскую часть сайта для информационной системы для управления клиентской базой данных и автоматизации процесса продаж. Программа должна позволять добавлять, анализировать и изменять аккаунты пользователей, продавцы через программу имеют возможность оформлять покупки клиетнов, а сами клиенты через личный кабинет смотреть совершенные покупки.

4. Задачи

1. Разработать внешний вид сайта. Адаптировать сайт под десктопные устройства.
2. Реализовать клиентскую часть функционала администратора.
3. Реализовать клиентскую часть функционала продавца.
4. Реализовать клиентскую часть функционала рядового пользователя.
5. Реализовать клиентскую часть функционала получения и фильтрации записей из базы данных и отображения их на веб-странице.

5. Реализация

1) Разработать внешний вид сайта. Адаптировать сайт под десктопные устройства.

При разработке внешнего вида сайта был уделен особый акцент на создание приятного и удобного пользовательского интерфейса, который был бы адаптирован к разным устройствам. Файл styles.css содержит основные стили для сайта, включая определение типов шрифтов, цветовую гамму, размеры элементов, их расположение и отступы. Кроме того, данный файл определяет классы и идентификаторы, что обеспечивает единый и согласованный внешний вид всех элементов страницы.

Использование адаптивного дизайна и тщательная работа над медиа-запросами позволили создать универсальный пользовательский интерфейс, который автоматически подстраивается под различные устройства, обеспечивая пользователям удобство и эффективность в использовании сайта.

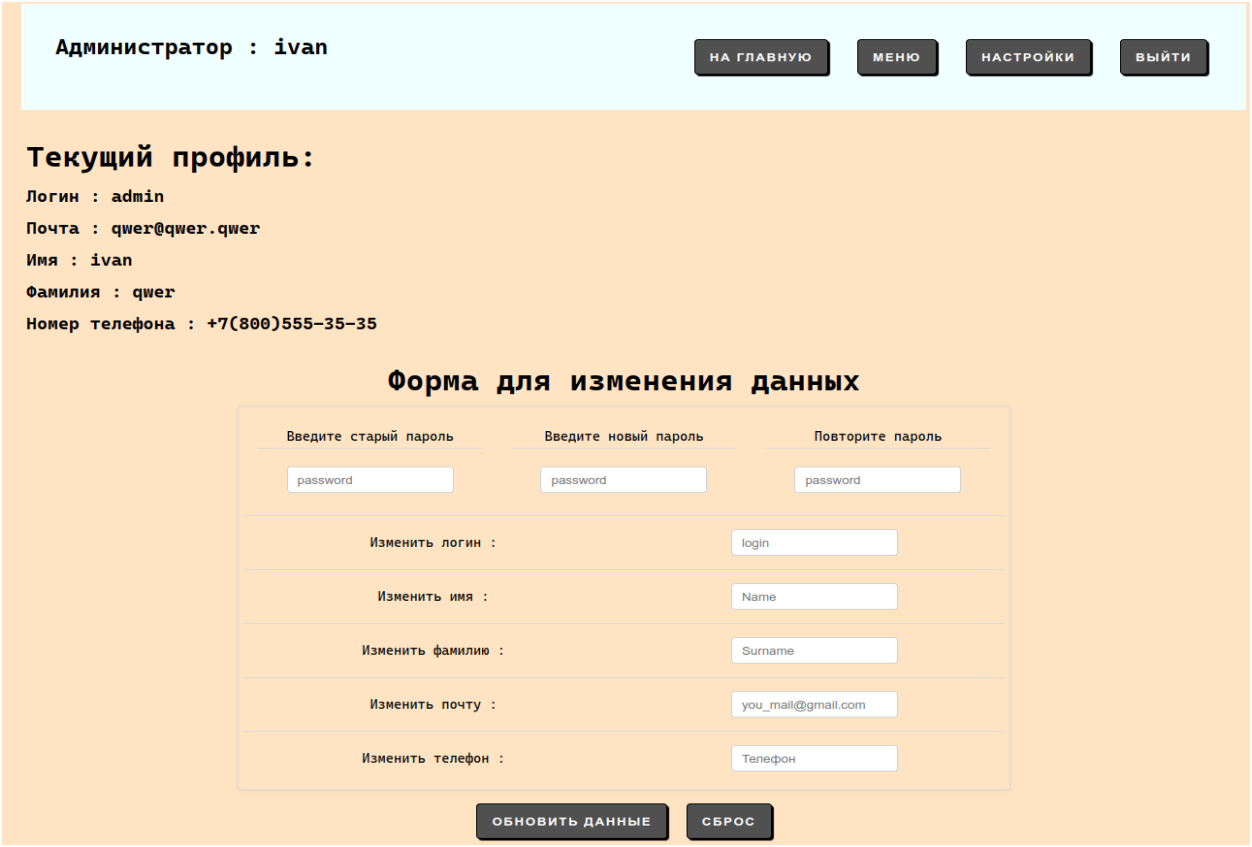

Рисунок 1 – Внешний вид десктоп версии сайта

2) Реализовать клиентскую часть функционала администратора.

Ключевой функционал добавления записей о новых товарах и юзерах был реализован для обеспечения удобства пользователя на веб-сайте. Создание этого функционала началось с разработки интерактивной формы, предоставляющей пользователю простой и интуитивно понятный интерфейс для ввода информации о каждом параметре товара.

В этой форме предусмотрены различные поля для ввода данных, таких как название продукта, доступное количество, указание цены и его штрих-кода. Эти параметры обеспечивают возможность детального описания каждого товара, что позволяет приложению вести точный учет расходов и пополнений товаров.

Когда пользователь заполняет данную форму и нажимает кнопку отправки, происходит инициирование процесса отправки данных на сервер. Для этого используются современные технологии, такие как Java с использованием объектов HttpRequest и HttpResponse. Эти инструменты позволяют передавать заполненные пользователем данные на сервер для последующей обработки.

После отправки данных клиент ожидает ответа от сервера. В случае успешной обработки данных на сервере, пользователю предоставляется информационное сообщение об успешном добавлении записи. Это позволяет пользователям убедиться, что их данные были успешно приняты и сохранены.

В случае возникновения каких-либо проблем или ошибок в процессе отправки данных или обработки на сервере, пользователю сообщается о возникшей ошибке. Это позволяет пользователям быть в курсе любых возможных проблем и, при необходимости, предпринять дополнительные действия для решения проблемы.

После успешного добавления записи, форма автоматически очищается и закрывается для комфортного ввода новых данных. Это помогает пользователям оперативно переходить к следующей финансовой операции, поддерживая удобство и эффективность управления своими финансами на веб-сайте.

Кроме того, важной частью разработки функционала добавления записей о финансах было обеспечение проверки данных на корректность перед отправкой на сервер. Это заключается в валидации полей формы на стороне клиента для обеспечения правильного ввода информации.

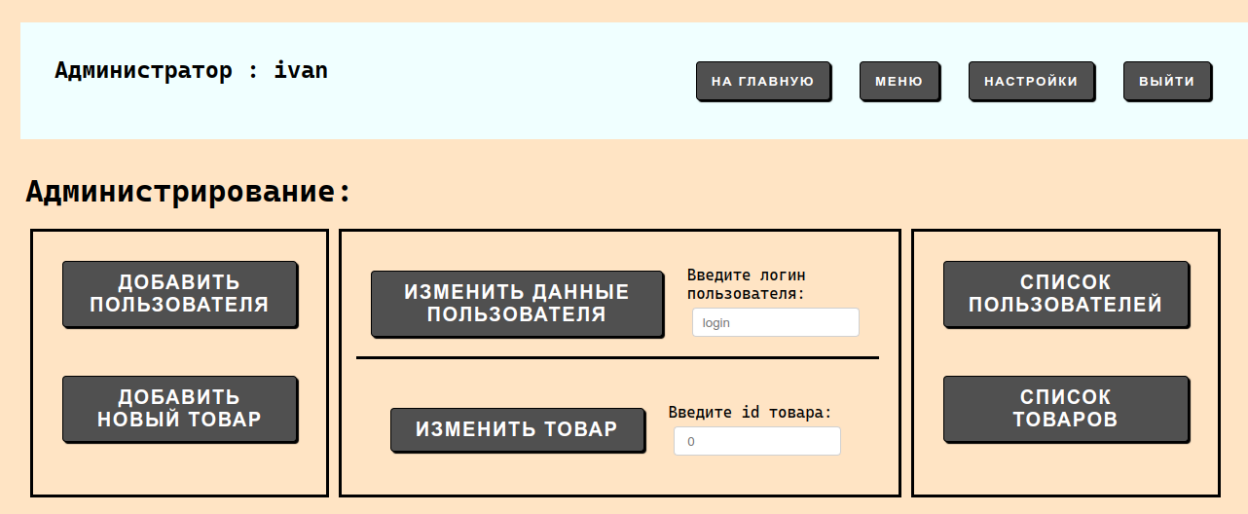

Рисунок 2 – Функции доступные администартору

3) Реализовать клиентскую часть функционала продавца.

На веб-сайте был создан функционал для оформления новых продаж, учета проданных товаров и автоматизации их покупки и оплаты. Этот функционал разработан для удобства сотрудников, позволяя им автоматизировать процесс учета товаров, взамен бумажной волокиты.

Основой данного функционала является сканер штрих-кодов и корзина отсканированных товаров. Сотрудник подносит товар к веб-камере и сканер сам считывает данные штрих-кода и ищет его в базе данных. Штрих-

код правильный, то этот товар добавляется в корзину. После того как все товары отсканированы продавец переходит на страницу оплаты.

Если у покупателя существует аккаунт в приложении, то он может сказать свой номер телефона для привязки текущей покупки.

На этой стадии продавец проверяет правильность заполнения корзины, после чего выбирает способ оплаты. Покупатель может заплатить наличной, банковской картой или оформить договор.

После оплаты данные о покупке привязываются к пользователю, а также система вычитает из базы данных нужное количество товаров.

Данные отправляются на сервер посредством Java Servlet. Аналогично процессу добавления записей, данные передаются на сервер с помощью HttpRequest для обновления информации о заказах, используя уникальный идентификатор (ID) выбранной записи.

После отправки запроса на сервер, клиент ожидает ответа. В случае успешного обновления данных на сервере, пользователю отображается сообщение об успешной покупке. В случае возникновения ошибки обработки данных на сервере, пользователь получает сообщение об ошибке для уведомления о возникшей проблеме.

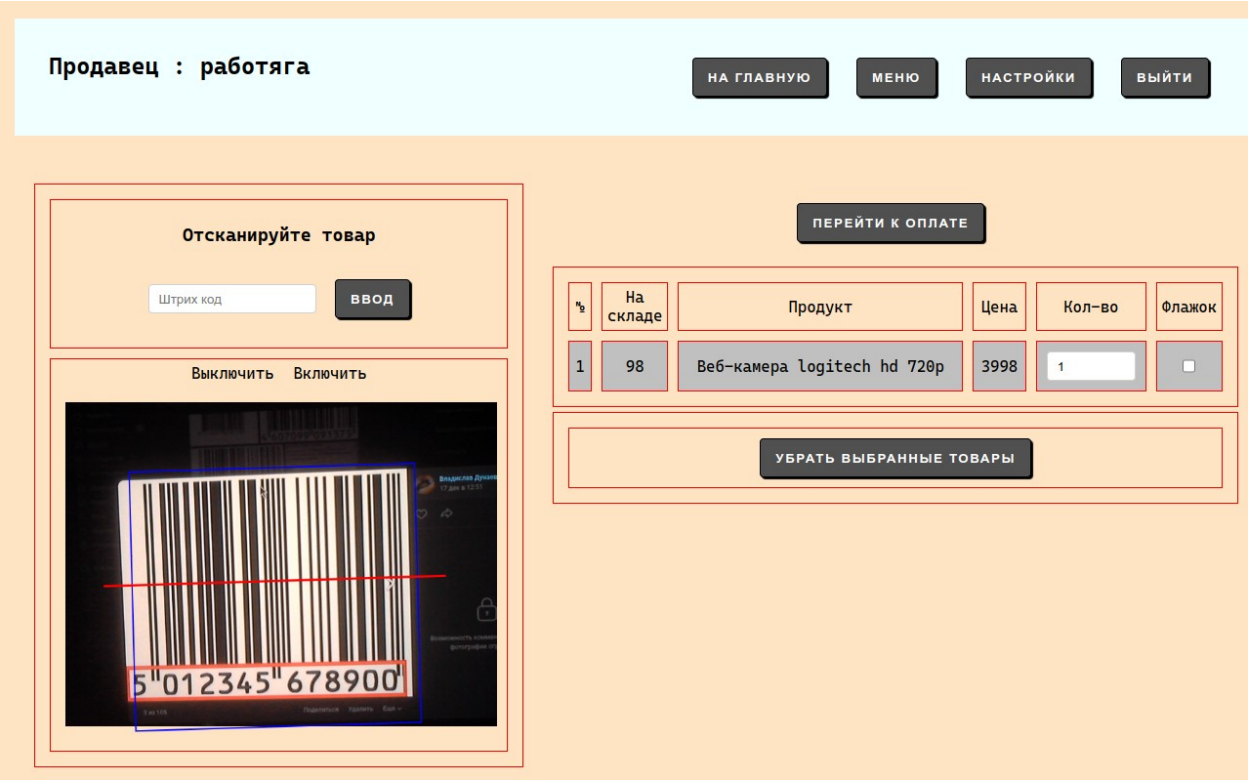

Рисунок 3 – Главное меню продавца

4) Реализовать клиентскую часть функционала рядового пользователя.

На сайте реализован функционал для просмотра как для списка совершенных покупок, так и для одной конкретной покупки с деталями.

Интерфейс предоставляет пользователю возможность нажать на конкретую запись для просмотра ее содежимого. При наведении курсора запись подсвечивается, значит на нее можно нажать. Этой функция создана для интуитивно понятного управления.

На сайте реализован функционал удаления записей о заказе, позволяющий пользователям удалять конкретную запись о заказе. Этот функционал создан для обеспечения гибкости и возможности управления данными, позволяя пользователям удалять ненужные или ошибочные записи из базы данных.

После того как пользователь выбирает запись, данные отправляются на сервер для обработки. Это осуществляется с помощью Java Servlet, который отправляет запрос в базу данных для дальнейшего отображения подробной информации.

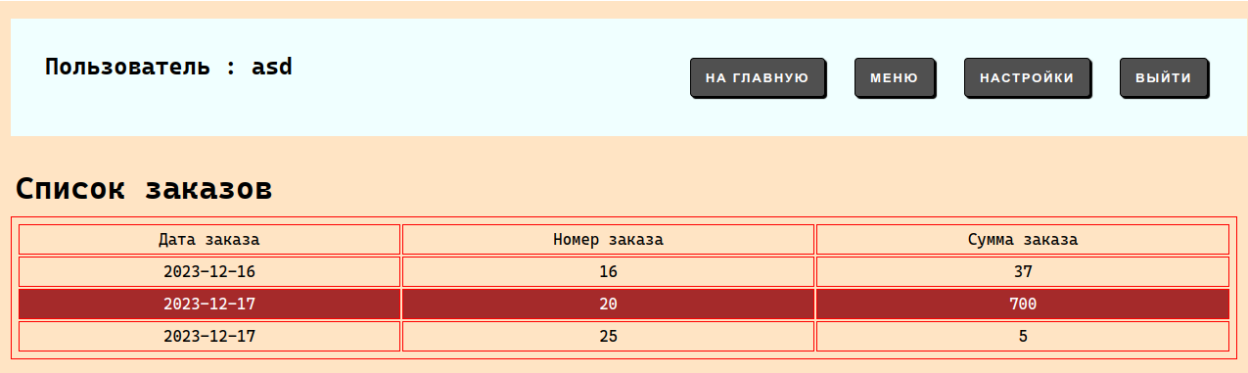

Рисунок 4 – Список совершенных покупок пользователем asd

После успешного просмотра записи, пользователь может продолжить работу с приложением или выполнить другие операции.

Также пользователь имеет возможность посмотреть данные своего аккаунта и в случае необходимости изменить их. В случае некорректно введенных данных сайт сообщит об ошибке.

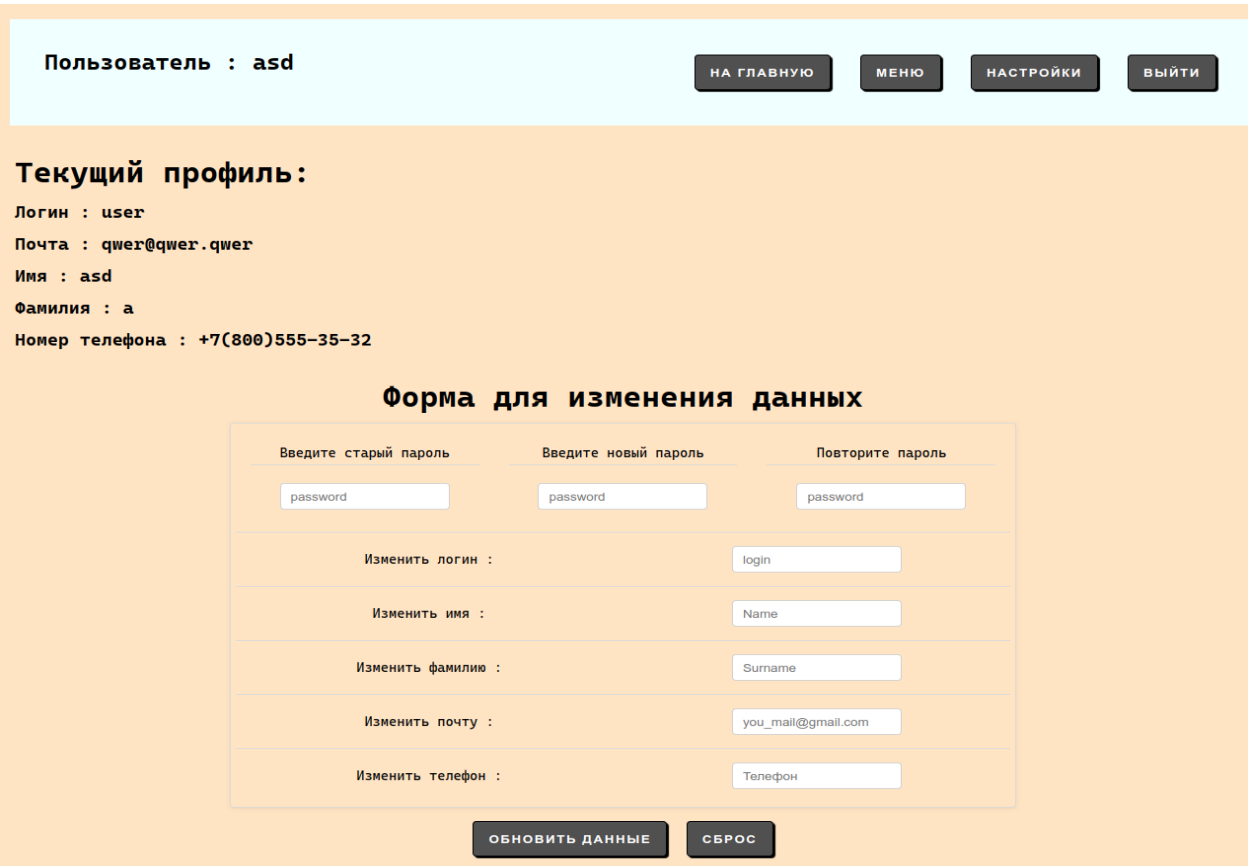

Рисунок 5 – Форма для изменения данных

5) Реализовать клиентскую часть функционала получения и фильтрации записей из базы данных и отображения их на веб-странице.

На веб-сайте реализован функционал получения данных о заказах из базы данных и их дальнейшего отображения на веб-странице. Этот функционал обеспечивает пользователям доступ к информации о своих заказах, предоставляя удобный способ отслеживать и управлять своими расходами

При успешном входе в свой аккаунт пользователь может выбрать необходимую функцию и полученить данных о заказах помощью запроса на сервер.

Полученные данные об операциях обрабатываются на серверной стороне и динамически встраиваются в HTML-структуру веб-страницы. Для отображения данных используются таблицы, где каждая запись о заказе представлена в удобном для пользователя формате.

Для более удобного отображения данных о заказах на странице применяются различные методы стилизации и форматирования с помощью CSS. Это позволяет пользователю легко читать и анализировать информацию о заказах.

В целом, функционал получения и отображения финансовых данных обеспечивает удобство ведения учета расходов и доходов, помогая администраторам принимать осознанные финансовые решения. Этот функционал становится важным инструментом для эффективного управления финансами и контроля над финансовым положением.

6. Заключение

В ходе курсовой работы был реализован сайт для автоматизации процесса продаж с применением следующих технологий: HTML, CSS, Java, MySQL и TomCat Server.

Однако, важно отметить, что проект обладает потенциалом для дальнейшего развития и расширения функционала. В частности, в будущем можно рассмотреть возможность разработки приложений для мобильных устройств, что позволит пользователям и администраторам удобно просматривать и управлять заказами непосредственно через мобильные устройства.

Дополнительно можно рассмотреть внедрение функционала анализа финансов. Разработка инструментов для анализа расходов, доходов и финансовых операций поможет пользователям получать ценную информацию и делать обоснованные финансовые решения.

Эти шаги позволят усовершенствовать сайт, сделать его более универсальным и функциональным, повысить удобство использования для пользователей.

Другим интересным направлением развития проекта может стать реализация функций безопасности и защиты данных пользователей. Интеграция механизмов шифрования данных, двухфакторной аутентификации и постоянное обновление системы безопасности помогут обеспечить конфиденциальность и надежность информации, что важно для пользователей в условиях возрастающих угроз.

7. Источники

1. MySQL Documentation [Электронный ресурс]. - Режим доступа: https://dev.mysql.com/doc
2. Mozilla Developer Network (MDN) Web Docs - HTML [Электронный ресурс]. - Режим доступа: https://developer.mozilla.org/en-US/docs/Web/HTML
3. CSS-Tricks [Электронный ресурс]. - Режим доступа: https://css-tricks.com
4. MySQL Tutorial [Электронный ресурс]. - Режим доступа: https://www.mysqltutorial.org
5. Web Platform [Электронный ресурс]. - Режим доступа: https://webplatform.org
6. StackExchange [Электронный ресурс]. - Режим доступа: https://stackexchange.com
7. Google Developers [Электронный ресурс]. - Режим доступа: https://developers.google.com

8. Приложение

Разработанный сайт можно скачать из моего репозитория на github https://github.com/Krasin1/labs-from-potato/tree/main/5/web

Файлы .java

AddProduct.java

```
package lab.course.controller.admin;


import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import lab.course.model.Product;


import java.io.IOException;


import static lab.course.model.Product.newProduct;



@WebServlet(name = "AddProduct", urlPatterns = "/addProduct")
public class AddProduct extends HttpServlet {


    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws IOException {
        // получаем параметры из формы
        String name = request.getParameter("productName");
        String barcode = request.getParameter("barcode");
        String count = request.getParameter("count");
        String price = request.getParameter("price");


        if (name == null || name.isEmpty()) {
            request.getSession().setAttribute("errorAddProduct", "Наименоавние не
может быть пустым");
            response.sendRedirect("addProduct");
            return;
        } else if (barcode == null || barcode.isEmpty()) {
            request.getSession().setAttribute("errorAddProduct", "Штрих-код не может
быть пустым");
            response.sendRedirect("addProduct");
            return;
        } else if (!Product.checkUniqueField("barcode", barcode)) {
            request.getSession().setAttribute("errorAddProduct", "Штрих-код уже
занят");
            response.sendRedirect("addProduct");
            return;
        } else if (count == null || count.isEmpty() || Integer.parseInt(count) < 0) {
            request.getSession().setAttribute("errorAddProduct", "Неправильно задано
кол-во");
            response.sendRedirect("addProduct");
            return;
        } else if (price == null || price.isEmpty() || Integer.parseInt(price) < 0) {
            request.getSession().setAttribute("errorAddProduct", "Неправильно задана
цена");
            response.sendRedirect("addProduct");
            return;
        }



        // если insert прошел успешно, то выводим уведомление
```

```
                if (newProduct(name, barcode, Integer.valueOf(count), Integer.valueOf(price)))
{
                    request.getSession().removeAttribute("errorAddProduct");
                    request.getSession().setAttribute("successAddProduct", "Товар
зарегестрированн");
                } else {
                    request.getSession().setAttribute("errorAddProduct", "Что то пошло не
так");
                }
                response.sendRedirect("addProduct");


            }



            protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                request.getRequestDispatcher("admin/addProduct.jsp").forward(request,
response);
            }
        }
```

AddUser.java

```
        package lab.course.controller.admin;


        import jakarta.servlet.ServletException;
        import jakarta.servlet.annotation.WebServlet;
        import jakarta.servlet.http.HttpServlet;
        import jakarta.servlet.http.HttpServletRequest;
        import jakarta.servlet.http.HttpServletResponse;
        import lab.course.model.User;


        import java.io.IOException;
        import java.sql.Connection;
        import java.sql.DriverManager;
        import java.sql.PreparedStatement;



        @WebServlet(name = "AddUser", urlPatterns = "/addUser")
        public class AddUser extends HttpServlet {
            private static final String URL = "jdbc:mariadb://localhost:3306/javaCourse";
            private static final String USERNAME = "user";
            private static final String PASSWORD = "qwer";


            protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws IOException {
                // получаем параметры из формы
                String login = request.getParameter("login");
                // хэшируем пароль
                String password =
User.getHashedPassword(request.getParameter("password").trim());
                String passwordSecond =
User.getHashedPassword(request.getParameter("passwordSecond").trim());
                String name = request.getParameter("name");
                String surname = request.getParameter("surname");
                String email = request.getParameter("email");
                String phone = request.getParameter("phone");
                String role = request.getParameter("roleInput");


                // подготавливаем запрос
                Connection connection = null;
                PreparedStatement preparedStatement = null;
                String SQL = "INSERT INTO `Users`(`login`, `password`, `email`, `name`,
`surname`, `phone`, `role`) VALUES (?,?,?,?,?,?,?)";
                int resultExecute = 0;


                if (!passwordSecond.equals(password)) {
```

```
                    request.getSession().setAttribute("errorAddUser", "Пароли не совпадают");
                    response.sendRedirect("addUser");
                    return;
                } else if (!User.checkEmail(email)) {
                    request.getSession().setAttribute("errorAddUser", "Неправильно введена
почта");
                    response.sendRedirect("addUser");
                    return;
                } else if (!User.checkUniqueField("phone",phone)) {
                    request.getSession().setAttribute("errorAddUser", "Аккаунт с этим номером
уже существует");
                    response.sendRedirect("addUser");
                    return;
                } else if (!User.checkUniqueField("login", login)) {
                    request.getSession().setAttribute("errorAddUser", "Логин занят");
                    response.sendRedirect("addUser");
                    return;
                }


                // подключаем базу данных
                // если прошли проверки, то заносим пользователя в бд
                try {
                    Class.forName("org.mariadb.jdbc.Driver");
                    connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
                    preparedStatement = connection.prepareStatement(SQL);
                    preparedStatement.setString(1, login);
                    preparedStatement.setString(2, password);
                    preparedStatement.setString(3, email);
                    preparedStatement.setString(4, name);
                    preparedStatement.setString(5, surname);
                    preparedStatement.setString(6, phone);
                    preparedStatement.setString(7, role);
                    resultExecute = preparedStatement.executeUpdate();
                    // preparedStatement.execute();


                    connection.close();
                } catch (Exception e) {
                    e.printStackTrace();
                }


                // если insert прошел успешно, то выводим уведомление
                if (resultExecute != 0) {
                    request.getSession().removeAttribute("errorAddUser");
                    request.getSession().setAttribute("successAddUser", "Пользователь
зарегестрированн");
                } else {
                    request.getSession().setAttribute("errorAddUser", "Что то пошло не так");
                }
                response.sendRedirect("addUser");


            }



            protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                request.getRequestDispatcher("admin/addUser.jsp").forward(request, response);
            }
        }
```

EditProduct.java

```
package lab.course.controller.admin;


import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
```

```
import jakarta.servlet.http.HttpServletResponse;
import lab.course.model.Product;


import java.io.IOException;
import java.util.HashMap;


import static lab.course.model.Product.*;




@WebServlet(name = "EditProduct", urlPatterns = "/editProduct")
public class EditProduct extends HttpServlet {


    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {



        // получаем параметры
        int id= Integer.parseInt((String)
request.getSession().getAttribute("selected_product"));
        String productName = request.getParameter("productName");
        String barcode = request.getParameter("barcode");
        String count_string = request.getParameter("count");
        String price_string = request.getParameter("price");



        // инициализируем ассоциативный массив для последующей отправки его в бд
        HashMap<String,String> map = new HashMap<>();


        // проверка на штрих-код
        if (barcode != null && !barcode.isEmpty()) {
            if (!Product.checkUniqueField("barcode",barcode)) {
                request.getSession().setAttribute("errorEditProduct", "Ошибка, штрих-
код уже занят");
                response.sendRedirect("/editProduct");
                return;
            } else {
                map.put("barcode", barcode);
                request.getSession().setAttribute("product_barcode",barcode);
            }
        }


        // проверка на имя
        if (productName!= null && !productName.isEmpty()) {
            map.put("name", productName);
            request.getSession().setAttribute("product_name",productName);
        }


        // проверка на кол-во
        if (count_string!= null && !count_string.isEmpty()) {
            Integer count = Integer.parseInt(count_string);
            if (count < 0) {
                request.getSession().setAttribute("errorEditProduct", "Ошибка, не
может быть отрицательного кол-ва");
                response.sendRedirect("/editProduct");
                return;
            } else {
                map.put("count_in_stok", count_string);
                request.getSession().setAttribute("product_count", count);
            }
        }


        // проверка на цену
        if (price_string!= null && !price_string.isEmpty()) {
            Integer price = Integer.parseInt(price_string);
```

```
                    if (price< 0) {
                        request.getSession().setAttribute("errorEditProduct", "Ошибка, не
может быть отрицательной цены");
                        response.sendRedirect("/editProduct");
                        return;
                    } else {
                        map.put("price", price_string);
                        request.getSession().setAttribute("product_price",price);
                    }
                }



                // загружаем изменения в бд + обработка ошибки
                boolean check = updateProductParam(id, map);
                if (!check) {
                    request.getSession().setAttribute("errorEditProduct", "Произошла ошибка во
время обновления данных");
                    response.sendRedirect("/editProduct");
                    return;
                }


                // если все хорошо, то говорим что все хорошо
                request.getSession().setAttribute("successEditProduct", "Данные обновлены");
                response.sendRedirect("/editProduct");
            }


            protected void doGet(HttpServletRequest request, HttpServletResponse response)
                    throws ServletException, IOException {
                request.getRequestDispatcher("admin/editProduct.jsp").forward(request,
response);
            }
        }
```

EditUser.java

```
package lab.course.controller.admin;


import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import lab.course.model.User;


import java.io.IOException;
import java.util.HashMap;
import java.util.Objects;


import static lab.course.model.User.*;


@WebServlet(name = "EditUser", urlPatterns = "/editUser")
public class EditUser extends HttpServlet {


    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {


        request.getSession().setAttribute("errorEdit", null);
        request.getSession().setAttribute("sucEdit", null);


        // получаем параметры
        String user_login = (String) request.getSession().getAttribute("user_login");
        String login = request.getParameter("login");
        String passwordNew = getHashedPassword(request.getParameter("passwordNew"));
        String passwordSecond =
```

```
getHashedPassword(request.getParameter("passwordNewSecond"));
                String name = request.getParameter("name");
                String surname = request.getParameter("surname");
                String email = request.getParameter("email");
                String phone = request.getParameter("phone");
                String role = request.getParameter("roleInput");



                // инициализируем ассоциативный массив для последующей отправки его в бд
                HashMap<String,String> map = new HashMap<>();


                // правильность пароля
                if ( passwordNew != null && !passwordNew.isEmpty() || passwordSecond != null
&& !passwordSecond.isEmpty()) {
                    if (!passwordNew.equals(passwordSecond)) {
                        request.getSession().setAttribute("errorEdit", "Ошибка: Проверьте
пароль");
                        response.sendRedirect("/editUser");
                        return;
                    } else {
                        map.put("password", passwordNew);
                    }
                }
                // проверка на правилный email
                if (email != null && !email.isEmpty()) {
                    if (!checkEmail(email)) {
                        request.getSession().setAttribute("errorEdit", "Некорректный адрес
почты");
                        response.sendRedirect("/editUser");
                        return;
                    } else {
                        map.put("email", email);
                        request.getSession().setAttribute("user_email", email);
                    }
                }


                // проверка на логин
                if (login != null && !login.isEmpty()) {
                    if (!User.checkUniqueField("login", login)) {
                        request.getSession().setAttribute("errorEdit", "Ошибка, логин уже
занят");
                        response.sendRedirect("/editUser");
                        return;
                    } else {
                        map.put("login", login);
                        request.getSession().setAttribute("user_login", login);
                    }
                }


                // проверка на имя
                if (name != null && !name.isEmpty()) {
                    map.put("name", name);
                    request.getSession().setAttribute("user_name",name);
                }


                // проверка на фамилию
                if (surname != null && !surname.isEmpty()) {
                    map.put("surname", surname);
                    request.getSession().setAttribute("user_surname", surname);
                }


                // проверка на телефон
                if (phone != null && !phone.isEmpty()) {
                    if (!User.checkUniqueField("phone",phone)) {
                        request.getSession().setAttribute("errorEdit", "Ошибка, телефон уже
используется");
                        response.sendRedirect("/editUser");
                        return;
                    } else {
                        map.put("phone", phone);
```

```
                    request.getSession().setAttribute("user_phone", phone);
                }
            }

            // проверка на роль
            if (role != null && !role.isEmpty()) {
                map.put("role", role);
                request.getSession().setAttribute("user_role", role);
            }

            // получаем id пользователя по логину для загрузки в бд
            int id = Objects.requireNonNull(getUserByLogin(user_login)).getId();

            // загружаем изменения в бд + обработка ошибки
            boolean check = updateUserParam(id, map);
            if (!check) {
                request.getSession().setAttribute("errorEdit", "Произошла ошибка во время
обновления данных");
                response.sendRedirect("/editUser");
                return;
            }

            // если все хорошо, то говорим что все хорошо
            request.getSession().setAttribute("sucEdit", "Данные обновлены");
            response.sendRedirect("/editUser");
        }

        protected void doGet(HttpServletRequest request, HttpServletResponse response)
                throws ServletException, IOException {
            request.getRequestDispatcher("admin/editUser.jsp").forward(request, response);
        }
    }
```

RedirectToFunctions.java

```
package lab.course.controller.admin;

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;

import java.io.IOException;



@WebServlet(name = "RedirectToFunctions", urlPatterns = "/func")
public class RedirectToFunctions extends HttpServlet {

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        String choose =  request.getParameter("choose");
        request.getSession().setAttribute("choose", choose);
        if (choose.equals("editUser")) {
```

```
                    request.getSession().setAttribute("user_login",
request.getParameter("login"));
                }
                if (choose.equals("editProduct")) {
                    request.getSession().setAttribute("selected_product",
request.getParameter("product"));
                }


                response.sendRedirect("/func");
            }



            protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                HttpSession session = request.getSession();
                boolean check = (session.getAttribute("role") != null);
                String choose = (String) session.getAttribute("choose");


                if (check) {
                    if (choose.equals("addUser")) {
                        request.getRequestDispatcher("admin/addUser.jsp").forward(request,
response);
                    } else if (choose.equals("editUser")) {
                        request.getRequestDispatcher("/redirectToEditUser").forward(request,
response);
                    } else if (choose.equals("listUser")) {
                        request.getRequestDispatcher("/listUser").forward(request, response);
                    } else if (choose.equals("deleteUser")){
                        request.getRequestDispatcher("admin/deleteUser.jsp").forward(request,
response);
                    } else if (choose.equals("addProduct")) {
                        request.getRequestDispatcher("admin/addProduct.jsp").forward(request,
response);
                    } else if (choose.equals("editProduct")) {

request.getRequestDispatcher("/redirectToEditProduct").forward(request, response);
                    } else if (choose.equals("listProduct")) {
                        request.getRequestDispatcher("/listProduct").forward(request,
response);
                    } else if (choose.equals("deleteProduct")){

request.getRequestDispatcher("admin/deleteProduct.jsp").forward(request, response);
                    }
                } else {
                    request.getRequestDispatcher("login.jsp").forward(request, response);
                }
            }
        }
```

AccountChange.java

```
package lab.course.controller.auth;


import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import lab.course.model.User;


import java.io.IOException;
import java.util.HashMap;


import static lab.course.model.User.*;


@WebServlet(name = "AccountChange", urlPatterns = "/account")
```

```
public class AccountChange extends HttpServlet {

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        // получаем параметры
        String login = request.getParameter("login");
        String passwordNew = getHashedPassword(request.getParameter("passwordNew"));
        String passwordOld = getHashedPassword(request.getParameter("passwordOld"));
        String passwordSecond =
getHashedPassword(request.getParameter("passwordNewSecond").trim());
        String name = request.getParameter("name");
        String surname = request.getParameter("surname");
        String email = request.getParameter("email");
        String phone = request.getParameter("phone");
        int id = (int) request.getSession().getAttribute("user_id");

        // инициализируем ассоциативный массив для последующей отправки его в бд
        HashMap<String,String> map = new HashMap<>();



        // правильность пароля
        if (passwordOld != null && !passwordOld.isEmpty() ||
                passwordNew != null && !passwordNew.isEmpty() ||
                passwordSecond != null && !passwordSecond.isEmpty()) {
            if (!request.getSession().getAttribute("password").equals(passwordOld)
                    || !passwordNew.equals(passwordSecond)) {
                request.getSession().setAttribute("error", "Ошибка: Проверьте
пароль");
                response.sendRedirect("/account");
                return;
            } else {
                map.put("password", passwordNew);
                request.getSession().setAttribute("password", passwordNew);
            }
        }
        // проверка на правилный email
        if (email != null && !email.isEmpty()) {
            if (!checkEmail(email)) {
                request.getSession().setAttribute("error", "Некорректный адрес
почты");
                response.sendRedirect("/account");
                return;
            } else {
                map.put("email", email);
                request.getSession().setAttribute("email", email);
            }
        }


        // проверка на логин
        if (login != null && !login.isEmpty()) {
            if (!User.checkUniqueField("login", login)) {
                request.getSession().setAttribute("error", "Ошибка, логин уже занят");
                response.sendRedirect("/account");
                return;
            } else {
                map.put("login", login);
                request.getSession().setAttribute("login", login);
            }
        }


        // проверка на имя
        if (name != null && !name.isEmpty()) {
            map.put("name", name);
            request.getSession().setAttribute("name",name);
        }


        // проверка на фамилию
        if (surname != null && !surname.isEmpty()) {
```

```
                    map.put("surname", surname);
                    request.getSession().setAttribute("surname", surname);
                }


                // проверка на телефон
                if (phone != null && !phone.isEmpty()) {
                    if (!User.checkUniqueField("phone",phone)) {
                        request.getSession().setAttribute("error", "Ошибка, телефон уже
используется");
                        response.sendRedirect("/account");
                        return;
                    } else {
                        map.put("phone", phone);
                        request.getSession().setAttribute("phone", phone);
                    }
                }


                // загружаем изменения в бд + обработка ошибки
                boolean check = updateUserParam(id, map);
                if (!check) {
                    request.getSession().setAttribute("error", "Произошла ошибка во время
обновления данных");
                    response.sendRedirect("/account");
                    return;
                }


                // если все хорошо, то говорим что все хорошо
                request.getSession().setAttribute("error", null);
                request.getSession().setAttribute("suc", "Данные обновлены");
                response.sendRedirect("/account");
            }


            protected void doGet(HttpServletRequest request, HttpServletResponse response)
                    throws ServletException, IOException {
                request.getRequestDispatcher("user_profile.jsp").forward(request, response);
            }
        }
```

Login.java

```
package lab.course.controller.auth;

import java.nio.charset.StandardCharsets;
import java.security.*;


import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;


import jakarta.servlet.ServletContext;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;


@WebServlet(name = "Login", urlPatterns = "/login")
public class Login extends HttpServlet {
    private static final String URL = "jdbc:mariadb://localhost:3306/javaCourse";
    private static final String USERNAME = "user";
    private static final String PASSWORD = "qwer";
```

```
            String getHashedPassword(final String text) {
                if (text == null) return null;


                MessageDigest md = null;
                      try {
                              md = MessageDigest.getInstance("SHA-256");
                    md.update(text.getBytes("UTF-8"));
                      } catch (Exception e) {
                              e.printStackTrace();
                      }


                if (md != null)
                    return new String(md.digest(), StandardCharsets.UTF_8);
                else return null;
            }


            protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                // считываем логин и хэш пароля
                String login = request.getParameter("login");
                String password = getHashedPassword(request.getParameter("password").trim());


                HttpSession session = request.getSession(true);
                Connection connection = null;
                PreparedStatement preparedStatement = null;


                String SQL = "select * from Users where login = ?";


                try {
                    Class.forName("org.mariadb.jdbc.Driver");
                    connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
                    preparedStatement = connection.prepareStatement(SQL);
                    preparedStatement.setString(1, login);
                    ResultSet resultSet = preparedStatement.executeQuery();


                    if (resultSet.next()) {
                        session.setAttribute("user_id", resultSet.getInt("id"));
                        session.setAttribute("login", resultSet.getString("login"));
                        session.setAttribute("password", resultSet.getString("password"));
                        session.setAttribute("role", resultSet.getString("role"));
                        session.setAttribute("name", resultSet.getString("name"));
                        session.setAttribute("surname", resultSet.getString("surname"));
                        session.setAttribute("email", resultSet.getString("email"));
                        session.setAttribute("phone", resultSet.getString("phone"));
                    }
                    connection.close();
                } catch (Exception e) {
                    e.printStackTrace();
                }


                // проверяем логин и пароль, после перенаправляем на нужную страницу
                if (login.equals(session.getAttribute("login")) &&
                        password.equals(session.getAttribute("password")) &&
                        "admin".equals(session.getAttribute("role")) ) {
                    response.sendRedirect("login");
                } else if (login.equals(session.getAttribute("login")) &&
                        password.equals(session.getAttribute("password")) &&
                        "salesman".equals(session.getAttribute("role")) ) {
                    response.sendRedirect("login");
                } else if (login.equals(session.getAttribute("login")) &&
                        password.equals(session.getAttribute("password")) &&
                        "user".equals(session.getAttribute("role")) ) {
                    response.sendRedirect("login");
                } else {
                    request.getSession().invalidate();
                    request.getSession().setAttribute("hash", getHashedPassword(password));
                    request.getSession().setAttribute("errorMessage", "Логин или пароль
введены неправильно");
```

```
                response.sendRedirect("login");
            }
        }


        protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
            HttpSession session = request.getSession();
            boolean check = (session.getAttribute("role") != null);
            ServletContext servletContext = getServletContext();
            // ServletRequest servletRequest = request.

            if (check && session.getAttribute("role").equals("admin")) {
                request.getRequestDispatcher("admin/").forward(request, response);
                // servletContext.getRequestDispatcher("/admin/").forward(request,
response);
            } else if (check && session.getAttribute("role").equals("salesman")) {
                request.getRequestDispatcher("salesman/").forward(request, response);
                // servletContext.getRequestDispatcher("/salesman/").forward(request,
response);
            } else if (check && session.getAttribute("role").equals("user")) {
                request.getRequestDispatcher("user/").forward(request, response);
                // servletContext.getRequestDispatcher("/user/").forward(request,
response);
            } else {
                request.getRequestDispatcher("login.jsp").forward(request, response);
                // servletContext.getRequestDispatcher("/login.jsp").forward(request,
response);
            }
        }
    }
```

Logout.java

```
package lab.course.controller.auth;


import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;


import java.io.IOException;


@WebServlet(urlPatterns = "/logout")
public class Logout extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws IOException {
        HttpSession session = request.getSession(false);
        if (session != null) session.invalidate();
        response.sendRedirect("/login");
    }
}
```

Register.java

```
package lab.course.controller.auth;


import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
```

```
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import lab.course.model.User;


@WebServlet(name = "Register", urlPatterns = "/register")
public class Register extends HttpServlet {
    private static final String URL = "jdbc:mariadb://localhost:3306/javaCourse";
    private static final String USERNAME = "user";
    private static final String PASSWORD = "qwer";


    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws IOException {
        // получаем параметры из формы
        String login = request.getParameter("login");
        // хэшируем пароль
        String password =
User.getHashedPassword(request.getParameter("password").trim());
        String passwordSecond =
User.getHashedPassword(request.getParameter("passwordSecond").trim());
        String name = request.getParameter("name");
        String surname = request.getParameter("surname");
        String email = request.getParameter("email");
        String phone = request.getParameter("phone");
        // новые пользователи получают роль user
        String role = "user";


        // подготавливаем запрос
        Connection connection = null;
        PreparedStatement preparedStatement = null;
        String SQL = "INSERT INTO `Users`(`login`, `password`, `email`, `name`,
`surname`, `phone`, `role`) VALUES (?,?,?,?,?,?,?)";
        int resultExecute = 0;


        if (!passwordSecond.equals(password)) {
            request.getSession().setAttribute("errorInput", "Пароли не совпадают");
            response.sendRedirect("register");
            return;
        } else if (!User.checkEmail(email)) {
            request.getSession().setAttribute("errorInput", "Неправильно введена
почта");
            response.sendRedirect("register");
            return;
        } else if (!User.checkUniqueField("phone",phone)) {
            request.getSession().setAttribute("errorInput", "Аккаунт с этим номером
уже существует");
            response.sendRedirect("register");
            return;
        } else if (!User.checkUniqueField("login", login)) {
            request.getSession().setAttribute("errorInput", "Логин занят");
            response.sendRedirect("register");
            return;
        }


        // подключаем базу данных
        // если прошли проверки, то заносим пользователя в бд
        try {
            Class.forName("org.mariadb.jdbc.Driver");
            connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
            preparedStatement = connection.prepareStatement(SQL);
            preparedStatement.setString(1, login);
            preparedStatement.setString(2, password);
            preparedStatement.setString(3, email);
            preparedStatement.setString(4, name);
            preparedStatement.setString(5, surname);
            preparedStatement.setString(6, phone);
            preparedStatement.setString(7, role);
            resultExecute = preparedStatement.executeUpdate();
            // preparedStatement.execute();
```

```
                    connection.close();
                } catch (Exception e) {
                    e.printStackTrace();
                }


                // если insert прошел успешно, то выводим уведомление
                request.getSession().invalidate();
                if (resultExecute != 0) {
                    request.getSession().setAttribute("success", "Пользователь
зарегестрированн");
                } else {
                    request.getSession().setAttribute("errorInput", "Что то пошло не так");
                }
                response.sendRedirect("register");


            }



            protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                    request.getRequestDispatcher("register.jsp").forward(request, response);
            }
        }
```

CheckBarcode.java

```
        package lab.course.controller.sales;


        import java.io.IOException;


        import jakarta.servlet.ServletException;
        import jakarta.servlet.annotation.WebServlet;
        import jakarta.servlet.http.HttpServlet;
        import jakarta.servlet.http.HttpServletRequest;
        import jakarta.servlet.http.HttpServletResponse;
        import lab.course.model.ListProducts;


        @WebServlet(name = "CheckBarcode", urlPatterns = "/checkBarcode")
        public class CheckBarcode extends HttpServlet {


            protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                // получаем штрих код
                String barcode = request.getParameter("barcode");


                // создаем/получаем корзину
                ListProducts listProducts = (ListProducts)
request.getSession().getAttribute("list");
                if(listProducts == null) listProducts = new ListProducts();


                // добавляем товар по штрих коду + возвраем корзину или выводим ошибку
                if (listProducts.addByBarcode(barcode)) {
                    request.getSession().removeAttribute("errorCheck");
                    request.getSession().removeAttribute("errorRead");
                    request.getSession().setAttribute("list", listProducts);
                } else {
                    request.getSession().setAttribute("errorRead", "Товар не найден, либо
штрих код считан неверно");
                }


                // редирект на себя, для очередного товара
                response.sendRedirect("/checkBarcode");
```

```
            }

            protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                request.getRequestDispatcher("salesman/").forward(request, response);
            }
        }
```

HandleBuyer.java

```
        package lab.course.controller.sales;


        import java.io.IOException;
        import java.sql.Connection;
        import java.sql.DriverManager;
        import java.sql.PreparedStatement;
        import java.sql.ResultSet;
        import java.util.regex.Matcher;
        import java.util.regex.Pattern;


        import jakarta.servlet.ServletException;
        import jakarta.servlet.annotation.WebServlet;
        import jakarta.servlet.http.HttpServlet;
        import jakarta.servlet.http.HttpServletRequest;
        import jakarta.servlet.http.HttpServletResponse;
        import lab.course.model.ListProducts;


        @WebServlet(name = "HandleBuyer", urlPatterns = "/buy")
        public class HandleBuyer extends HttpServlet {
            protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {


                // получаем корзину, если не получили выводим ошибку
                ListProducts listProducts = (ListProducts)
request.getSession().getAttribute("list");
                if(listProducts == null || listProducts.getListProducts().isEmpty()) {
                    request.getSession().setAttribute("errorCheck", "Не выбран ни один
товар");
                    response.sendRedirect("/checkBarcode");
                    return;
                }
                // считываем кол-во товаров в корзине из формы
                for (var a :listProducts.getListProducts()){
                    String basket = request.getParameter("basket" + a.getId());
                    a.setCount(Integer.parseInt(basket));
                }


                // проверяем, хватает ли товара на складе
                if (!listProducts.checkCountInStok()) {
                    request.getSession().removeAttribute("errorCheck");
                    response.sendRedirect("/checkBarcode");
                    return;
                }




                request.getSession().setAttribute("errorCheck",null);
                response.sendRedirect("/buy");
            }



            protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                if (request.getSession().getAttribute("errorCheck") != null) {
```

```
                    request.getRequestDispatcher("/checkBarcode").forward(request, response);
                } else {
                    request.getRequestDispatcher("salesman/handleBuyer.jsp").forward(request,
response);
                }
            }
        }
```

OrdersSevlet.java

```
        package lab.course.controller.show;


        import jakarta.servlet.ServletException;
        import jakarta.servlet.annotation.WebServlet;
        import jakarta.servlet.http.HttpServlet;
        import jakarta.servlet.http.HttpServletRequest;
        import jakarta.servlet.http.HttpServletResponse;
        import lab.course.model.Orders;


        import java.io.IOException;


        @WebServlet(name = "OrdersServlet", urlPatterns = "/orders")
        public class OrdersServlet extends HttpServlet {



            protected void doGet(HttpServletRequest request, HttpServletResponse response)
                    throws ServletException, IOException {
                Orders orders = new Orders((Integer)
request.getSession().getAttribute("user_id"));
                request.getSession().setAttribute("orders", orders);
                request.getRequestDispatcher("user/orders.jsp").forward(request, response);
            }
        }
```

Файлы .jsp и .css

AddProduct.jsp

```
        <%@ page contentType="text/html;charset=UTF-8" language="java" %>
        <html>
        <head>
            <title>Добавить товар</title>
            <link type="text/css" href="/style.css" rel="stylesheet">
            <script src="https://cdn.jsdelivr.net/npm/@ericblade/quagga2/dist/quagga.js"></
script>
            <script src="barcode.js"></script>
            <style>
                .block {
                    margin-left: auto;
                    margin-right: auto;
                    width: 30em;
                }
                input {
                    width: 30em;
                }
                .scanner {
                    position: relative;
                }
                /* In order to place the tracking correctly */
                canvas.drawing, canvas.drawingBuffer {
                    position: absolute;
                    padding-left: 0;
                    left: 0;
                    top: 0;
                }
```

```
                #scanner-container > video {
                    padding-left: 0;
                    width: 85%;
                }
                #scanner-container > canvas {
                    width: 85%;
                }
            </style>
        </head>
        <body>
        <jsp:include page="/header.jsp"/>



        <form action="/addProduct" method="post" class="block" >
            <h1>Новый товар</h1>

            <div> ${errorAddProduct} </div>
            <div> ${successAddProduct}</div>

            <p>Введите название</p>
            <input class="input" type="text" name="productName" placeholder="Алебра 10-11
класс" required>
            <p>Введите кол-во</p>
            <input class="input" type="number" name="count" placeholder="123" required>
            <p>Введите цену</p>
            <input class="input" type="number" name="price" placeholder="123" required>
            <p>Введите штрих-код</p>
            <input class="input" type="text" id="barcode" name="barcode" placeholder=""
required>

            <br>
            <label style="cursor:pointer" onclick="Quagga.stop()">Выключить</label><label
style="cursor:pointer" onclick="startScanner();">Включить</label>
            <div id="scanner-container" class="scanner" style="padding-left: 0;margin-left:
0;"></div>

            <button class="button-primary" type="submit" style="width:
14em;">Добавить</button>
            <button class="button-primary" type="reset" style="width: 14em;">Сброс</button>
        </form>

        <script>
            startScanner();
        </script>



        </body>
        </html>
```

EditUser.jsp

```
        <%@ page contentType="text/html;charset=UTF-8" language="java" %>
        <html>
        <head>
            <title>Изменить пользователя</title>
            <link type="text/css" href="/style.css" rel="stylesheet">
            <style>
                table {
                    border-spacing: 10px;
                    border-collapse: separate;
                    border: 1px solid black;
                }
                td {
                    border: 1px solid black;
                    text-align: center;
                }
            </style>
        </head>
        <body>
        <jsp:include page="/header.jsp"/>
        <h1>Изменяемый пользователь:</h1>
```

```
        <h3>Логин : ${user_login}</h3>
        <h3>Почта : ${user_email}</h3>
        <h3>Имя : ${user_name}</h3>
        <h3>Фамилия : ${user_surname}</h3>
        <h3>Номер телефона : ${user_phone}</h3>
        <h3>Привилегии: ${user_role}</h3>

        <div class="center" style="max-width: 60%; text-align: center">
        <form action="/editUser" method="post" >
            <h1>Форма</h1>

            <% if (request.getSession().getAttribute("errorEdit") != null) { %>
                <div class="error_message" > ${errorEdit}</div>
            <%} %>

            <% if (request.getSession().getAttribute("sucEdit") != null) { %>
                <div class="success_message" > ${sucEdit}</div>
            <%} %>

            <div class="table-container">
                <div class="table-row" style="margin: 0; padding: 0;">
                    <div class="row-sub-container">
                        <div class="row-item"> Введите новый пароль  </div>
                        <div class="row-item"> <input class="input" type="password"
name="passwordNew" placeholder="password">       </div>
                    </div>
                    <div class="row-sub-container">
                        <div class="row-item"> Повторите пароль      </div>
                        <div class="row-item"> <input class="input" type="password"
name="passwordNewSecond" placeholder="password"> </div>
                    </div>
                </div>

                <div class="table-row" style="margin: 0; padding: 0;">
                    <div class="row-item">Изменить логин : </div>
                    <div class="row-item"><input class="input" type="text" name="login"
placeholder="login"></div>
                </div>
                <div class="table-row" style="margin: 0; padding: 0;">
                    <div class="row-item">Изменить имя : </div>
                    <div class="row-item"><input class="input" type="text" name="name"
placeholder="Name"></div>
                </div>
                <div class="table-row" style="margin: 0; padding: 0;">
                    <div class="row-item">Изменить фамилию : </div>
                    <div class="row-item"><input class="input" type="text" name="surname"
placeholder="Surname"></div>
                </div>
                <div class="table-row" style="margin: 0; padding: 0;">
                    <div class="row-item">Изменить почту : </div>
                    <div class="row-item"><input class="input" type="email" name="email"
placeholder="you_mail@gmail.com"></div>
                </div>
                <div class="table-row" style="margin: 0; padding: 0;">
                    <div class="row-item">Изменить телефон : </div>
                    <div class="row-item"><input type="text" placeholder="Телефон"
name="phone" class="phone_mask input"></div>
                    <script src="https://cdn.jsdelivr.net/npm/jquery@3.2.1/dist/jquery.min.js"
type="text/javascript"></script>
                    <script
src="https://cdn.jsdelivr.net/npm/jquery.maskedinput@1.4.1/src/jquery.maskedinput.js"
                            type="text/javascript"></script>
                    <script src="js/jquery.maskedinput.min.js"></script>
                    <script> $(".phone_mask").mask("+7(999)999-99-99"); </script>
                </div>

                <div class="table-row" style="margin: 0; padding: 0;">
                    <div class="row-item">Выберите уровень доступа </div>
                    <div style="text-align:left;">
                        <p><input style="width:auto" type="radio" name="roleInput"
value="user" id="user"><label for="user">Рядовой
                            пользователь</label></p>
                        <p><input style="width:auto" type="radio" name="roleInput"
value="salesman" id="salesman"><label for="salesman">Продавец</label>
                        </p>
```

```
                        <p><input style="width:auto" type="radio" name="roleInput"
value="admin" id="admin"><label for="admin">Адимнистратор</label>
                        </p>
                    </div>
                </div>
            </div>
            <div style="margin-bottom: 30px;">
                <button class="button-primary" type="submit" style="width:45%">Обновить
данные</button>
                <button class="button-primary" type="reset" style="width:45%">Сброс</button>
            </div>
        </form>
        </div>
        </body>
        </html>
```

Index.jsp

```
        <%@ page contentType="text/html;charset=UTF-8" language="java" %>
        <html>
        <head>
            <title>Меню</title>
            <link type="text/css" href="/style.css" rel="stylesheet">
        </head>

        <body>

        <% if (request.getSession().getAttribute("noUser") != null &&
(boolean)request.getSession().getAttribute("noUser")) { %>
            <script>alert("Пользователь не найден")</script>
        <% } %>

        <jsp:include page="/header.jsp"/>
        <h1>Администрирование:</h1>

        <div style="display: flex; justify-content: center">

        <div style="border: 3px solid black; display: flex; flex-direction: column; align-
items: center;">
            <div><form action="func" method="post">
                <button class="button-primary" name="choose" value="addUser"><h2>Добавить
пользователя</h2></button>
            </form></div>
            <div><form action="func" method="post">
                <button  class="button-primary" name="choose" value="addProduct"><h2>Добавить
новый товар</h2></button>
            </form></div>
        </div>

        <div style="border: 3px solid black; display: flex; flex-direction: column; align-
items: center; ">
            <div><form action="func" method="post" style="margin-left:0; padding-left: 0;
display: flex; border-bottom: 3px solid black;">
                <div style=""><button  class="button-primary" name="choose"
value="editUser"><h2>Изменить данные пользователя</h2></button></div>
                <div>Введите логин пользователя:<br><input class="input" type="text"
name="login" placeholder="login" required></div>
            </form></div>
            <div><form action="func" method="post" style="margin-left:0; padding-left: 0;
display: flex">
                <div style=""><button  class="button-primary" name="choose"
value="editProduct"><h2>Изменить товар</h2></button></div>
                <div>Введите id товара:<br><input class="input" type="text" name="product"
placeholder=" 0" required></div>
            </form></div>
        </div>

        <div style="border: 3px solid black; display: flex; flex-direction: column; align-
items: center;">
            <div><form action="func" method="post">
                <button  class="button-primary" name="choose" value="listUser"><h2>Список
```

```
пользователей</h2></button>
            </form></div>
            <div><form action="func" method="post">
                <button  class="button-primary" name="choose" value="listProduct"><h2>Список
товаров</h2></button>
            </form></div>
        </div>

        </div>



        </body>
        </html>
```

HandleBuyer.jsp

```
        <%@ page import="lab.course.model.ListProducts" %>
        <%@ page contentType="text/html;charset=UTF-8" language="java" %>
        <html>
        <head>
            <title>Данные покупателя</title>
            <link type="text/css" href="/style.css" rel="stylesheet">
            <style>
                * {
                    margin: 5px;
                    padding: 5px;
                }

                table {
                    border-spacing: 10px;
                    border-collapse: separate;
                    border: 1px solid red;
                }

                td {
                    border: 1px solid red;
                    text-align: center;
                }

                .flex-container {
                    width: 100%;
                    height: 300px;
                    display: flex;
                    flex-flow: row nowrap;
                }

                .left {
                    width: 30%;
                    flex-grow: 3;
                }

                .right {
                    flex-grow: 7;
                }
            </style>
        </head>
        <body>
        <jsp:include page="/header.jsp"/>
        <% if (request.getSession().getAttribute("err") != null) { %>
        <script>alert("${err}")</script>
        <%} %>
        <div class="flex-container">
            <div class="left">
                <table>
                    <td>
                        <h4>По желанию введите телефон покупателя для привязки покупки </h4>

                        <form action="purchase" method="post">
                            <p><label>Телефон</label>
                                <script
src="https://cdn.jsdelivr.net/npm/jquery@3.2.1/dist/jquery.min.js"
                                        type="text/javascript"></script>
```

```
                                <script
src="https://cdn.jsdelivr.net/npm/jquery.maskedinput@1.4.1/src/jquery.maskedinput.js"
                                        type="text/javascript"></script>
                                <script src="js/jquery.maskedinput.min.js"></script>
                                <input type="text" placeholder="Телефон" name="buyer_phone"
class="phone_mask input">
                                <script> $(".phone_mask").mask("+7(999)999-99-99"); </script>
                            </p>
                            <div style="text-align:left;">
                                <h4> Способ оплаты :</h4>
                                <p><input type="radio" name="payment" value="card"
id="card"><label for="card">Банковской
                                    картой </label></p>
                                <p><input type="radio" name="payment" value="money" checked
id="money"><label for="money">Наличкой </label>
                                </p>
                                <p><input type="radio" name="payment" value="agreement"
id="agreement"><label for="agreement">Договор
                                    для юр. лиц </label></p>
                            </div>
                            <button class="button-primary" type="submit">Оплатить</button>
                        </form>
                    </td>
                </table>
            </div>

            <div class="right">
                <table style="width: 100%;align-content: center;">
                    <tr>
                        <td>№</td>
                        <td>Продукт</td>
                        <td>Цена</td>
                        <td>Кол-во</td>
                    </tr>
                    <% var list = (ListProducts) request.getSession().getAttribute("list");
                        if (list != null) {
                            int i = 1;
                            int sum = 0;
                            for (var a : list.getListProducts()) {
                                if (i % 2 == 0) {
                                    out.print("<tr><td>" + i++ + "</td>");
                                } else {
                                    out.print("<tr style=\"background: silver\"><td>" + i++ +
"</td>");
                                }
                                out.print("<td>" + a.getProduct_name() + "</td>");
                                out.print("<td>" + a.getPrice() + "</td>");
                                out.print("<td style=\"width:70px;\">" + a.getCount() +
"</td></tr>");
                                sum += a.getPrice() * a.getCount();
                            }
                            out.print("<tr><td colspan=\"4\">Общая цена : " + sum +
"</td></tr");
                        } %>
                </table>
            </div>
        </div>
        </body>
        </html>
```

Index.jsp

```
        <%@ page import="lab.course.model.ListProducts" %>
        <%@ page contentType="text/html;charset=UTF-8" language="java" %>
        <html>
        <head>
            <link type="text/css" href="/style.css" rel="stylesheet">
            <script src="https://cdn.jsdelivr.net/npm/@ericblade/quagga2/dist/quagga.js"></
script>
            <script src="barcode.js"></script>
            <title>Сканер</title>

            <style>
                .scanner {
```

```
                    position: relative;
                }

                /* In order to place the tracking correctly */
                canvas.drawing, canvas.drawingBuffer {
                    padding-left: 0;
                    position: absolute;
                    left: 0;
                    top: 0;
                }

                * {
                    margin: 5px;
                    padding: 5px;
                }

                table {
                    border-spacing: 10px;
                    border-collapse: separate;
                    border: 1px solid red;
                }

                td {
                    border: 1px solid red;
                    text-align: center;
                }

                .left {
                    width: 40%;
                    flex-grow: 3;
                }

                .right {
                    flex-grow: 7;
                }

                #scanner-container > video {
                    width: 100%;
                    height: 100%;
                    padding-left: 0;
                }

                #scanner-container > canvas {
                    width: 100%;
                    height: 100%;
                }

                .flex-container {
                    width: 100%;
                    height: 300px;
                    display: flex;
                    flex-flow: row nowrap;
                }
            </style>
        </head>
        <body>
        <jsp:include page="/header.jsp"/>
        <% if (request.getSession().getAttribute("errorCheck") != null) { %>
        <script>alert("${errorCheck}")</script>
        <%} %>
        <div class="flex-container">
            <div class="left">
                <table>
                    <tr>
                        <td>
                            <form action="checkBarcode" method="post">
                                <h3>Отсканируйте товар</h3>
                                <div>
                                    ${errorRead}
                                </div>
                                <p>
                                    <input class="input" type="text" name="barcode"
id="barcode" placeholder="Штрих код" required>
                                    <button class="button-primary" id="barcode-submit"
type="submit">Ввод</button>
```

```
                                </p>
                            </form>
                        </td>
                    </tr>
                    <tr>
                        <td>
                            <label style="cursor:pointer"
onclick="Quagga.stop()">Выключить</label><label style="cursor:pointer"


onclick="startScanner();">Включить</label>
                                <div id="scanner-container" class="scanner" style="padding-left:
0"></div>
                        </td>
                    </tr>
                </table>
            </div>

            <div class="right">
                <div style="display: flex; align-items: center; justify-content: center">
                    <form action="buy" method="post" id="basket_post"> <button class="button-
primary" type="submit">Перейти к оплате</button> </form>
                </div>
                <table style="width: 100%; align-content: center;">
                    <tr>
                        <td>№</td>
                        <td>На складе</td>
                        <td>Продукт</td>
                        <td>Цена</td>
                        <td>Кол-во</td>
                        <td>Флажок</td>
                    </tr>
                    <%
                        var list = (ListProducts) request.getSession().getAttribute("list");
                        if (list != null) {
                            int i = 1;
                            for (var a : list.getListProducts()) {
                                if (i % 2 == 0) {
                                    out.print("<tr><td>" + i++ + "</td>");
                                } else {
                                    out.print("<tr style=\"background: silver\"><td>" + i++ +
"</td>");
                                }
                                out.print("<td style=\"width:50px;\">" + a.getCount_in_stok()
+ "</td>");
                                out.print("<td>" + a.getProduct_name() + "</td>");
                                out.print("<td>" + a.getPrice() + "</td>");
                                out.print("<td style=\"width:50px;\"><input class=\"input\"
type=number form=\"basket_post\" required value=" + a.getCount() + " name=\"basket" + a.getId()
+ "\" min=\"1\" max=\"1000\" " +
                                        "onkeyup=\"if(this.value>999){this.value='999';}else
if(this.value<1){this.value='1';}\"></td>");
                                out.print("<td style=\"width:50px;\"><input class=\"input\"
type=\"checkbox\" form=\"check_del\" name=\"delete\" value=\"" + a.getId() + "\"></td></tr>");
                            }
                        }
                    %>
                </table>
                <table style="width: 100%; align-content: center">
                    <form action="remove" id="check_del" method="post">
                        <td>
                            <button class="button-primary" type="submit">Убрать выбранные
товары</button>
                        </td>
                    </form>
                </table>
            </div>
        </div>

        <script>
            startScanner();
        </script>

        </body>
        </html>
```

Orders.jsp

```
<%@ page import="lab.course.model.Orders" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>Заказы</title>

    <style>
        body {
            background: bisque;
        }
        table {
            border: 1px solid red;
        }

        td {
            border: 1px solid red;
            text-align: center;
        }
        tr:hover {
            background-color: brown; color: white;
        }
    </style>
</head>
<body>
<jsp:include page="/header.jsp"/>
<h1>Список заказов</h1>
<table style="width: 100%; align-content: center;">
    <tr>
        <td>Дата заказа</td>
        <td>Номер заказа</td>
        <td>Сумма заказа</td>
    </tr>
    <% Orders orders = (Orders) request.getSession().getAttribute("orders");
        if (orders != null) {
            if (orders.getOrders() != null) {
                for (var a : orders.getOrders()) {
                    out.print("<tr style=\"cursor:pointer;\" method=\"post\" data-
href=\"/order?order_id="+ a.getPurchase_id() + "\">");
                    out.print("<td>" + a.getDate() + "</td>");
                    out.print("<td>" + a.getPurchase_id() + "</td>");
                    out.print("<td>" + a.getTotal_price() + "</td>");
                    out.print("</tr>");
                }
            } else {
                out.print("Вы пока не делали заказов<br>");
            }
        } else {
        }
    %>
    <script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
    <script>
        $(document).ready(function($) {
            $('*[data-href]').on('click', function() {
                window.location = $(this).data("href");
            });
        });
    </script>
</table>
</body>
</html>
```

Login.jsp

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
isELIgnored="false" %>
<!DOCTYPE html>
<html>
<head>
    <title>Вход</title>
```

```
            <link type="text/css" href="/style.css" rel="stylesheet">
            <style>
                .block {
                    margin-left: auto;
                    margin-right: auto;
                    width: 20em;
                }

                * {
                    margin: 5px;
                    padding: 5px;
                }
            </style>
        </head>
        <body>
        <% String id = (String) request.getSession().getAttribute("role");
            if (id != null) { %>
        <jsp:include page="header.jsp"/>
        <%} else {%>
        <form action="/">
            <button class="button-primary">На главную</button>
        </form>
        <%}%>

        <form action="login" method="post" class="block">
            <h1>Вход в систему</h1>

            <% if (request.getSession().getAttribute("errorMessage") != null) { %>
            <div class="error_message"> ${errorMessage} </div>
            <%}%>

            <p>Введите логин</p>
            <input class="input" style="width:95%;" type="text" name="login"
placeholder="login" required>
            <p>Введите пароль</p>
            <input class="input" style="width:95%;" type="password" name="password"
placeholder="password" required>
            <p>
                <button class="button-primary" style="width:40%;" type="submit">Вход</button>
                <input form="reg" class="button-primary" style="width:50%; float:right;"
type="submit" value="Регистрация"/></p>
        </form>

        <form id="reg" action="register.jsp" class="block"></form>
        </body>
        </html>
```

User_profile.jsp

```
        <%@ page contentType="text/html;charset=UTF-8" language="java" %>
        <html>
        <head>
            <title>Смена данных</title>
            <link type="text/css" href="/style.css" rel="stylesheet">
        </head>
        <body>
        <jsp:include page="/header.jsp"/>
        <h1>Текущий профиль:</h1>
        <h3>Логин : ${login}</h3>
        <h3>Почта : ${email}</h3>
        <h3>Имя : ${name}</h3>
        <h3>Фамилия : ${surname}</h3>
        <h3>Номер телефона : ${phone}</h3>

        <div class="center" style="max-width: 80%; text-align: center">
        <form action="/account" method="post">
            <h1>Форма для изменения данных</h1>

            <% if (request.getSession().getAttribute("error") != null) { %>
                <div class="error_message" > ${error}</div>
            <%} %>

            <% if (request.getSession().getAttribute("suc") != null) { %>
```

```
                <div class="success_message" > ${suc}</div>
            <%} %>


            <div class="table-container" >
                <div class="table-row" style="margin: 0; padding: 0;">
                    <div class="row-sub-container">
                        <div class="row-item"> Введите старый пароль </div>
                        <div class="row-item"> <input class="input" type="password"
name="passwordOld" placeholder="password">       </div>
                    </div>
                    <div class="row-sub-container">
                        <div class="row-item"> Введите новый пароль  </div>
                        <div class="row-item"> <input class="input" type="password"
name="passwordNew" placeholder="password">       </div>
                    </div>
                    <div class="row-sub-container">
                        <div class="row-item"> Повторите пароль      </div>
                        <div class="row-item"> <input class="input" type="password"
name="passwordNewSecond" placeholder="password"> </div>
                    </div>
                </div>

                <div class="table-row" style="margin: 0; padding: 0;">
                    <div class="row-item">Изменить логин : </div>
                    <div class="row-item"><input class="input" type="text" name="login"
placeholder="login"></div>
                </div>
                <div class="table-row" style="margin: 0; padding: 0;">
                    <div class="row-item">Изменить имя : </div>
                    <div class="row-item"><input class="input" type="text" name="name"
placeholder="Name"></div>
                </div>
                <div class="table-row" style="margin: 0; padding: 0;">
                    <div class="row-item">Изменить фамилию : </div>
                    <div class="row-item"><input class="input" type="text" name="surname"
placeholder="Surname"></div>
                </div>
                <div class="table-row" style="margin: 0; padding: 0;">
                    <div class="row-item">Изменить почту : </div>
                    <div class="row-item"><input class="input" type="email" name="email"
placeholder="you_mail@gmail.com"></div>
                </div>
                <div class="table-row" style="margin: 0; padding: 0;">
                    <div class="row-item">Изменить телефон : </div>
                    <div class="row-item"><input type="text" placeholder="Телефон"
name="phone" class="phone_mask input"></div>
                    <script src="https://cdn.jsdelivr.net/npm/jquery@3.2.1/dist/jquery.min.js"
type="text/javascript"></script>
                    <script
src="https://cdn.jsdelivr.net/npm/jquery.maskedinput@1.4.1/src/jquery.maskedinput.js"
                            type="text/javascript"></script>
                    <script src="js/jquery.maskedinput.min.js"></script>
                    <script> $(".phone_mask").mask("+7(999)999-99-99"); </script>
                </div>
            </div>
            <p>
                <button type="submit" class="button-primary">Обновить данные</button>
                <button type="reset" class="button-primary">Сброс</button>
            </p>
        </div>
        </body>
        </html>
```

Barcode.js

```
        function order_by_occurrence(arr) {
            var counts = {};
            arr.forEach(function(value){
                if(!counts[value]) {
                    counts[value] = 0;
                }
                counts[value]++;
            });
```

```
            return Object.keys(counts).sort(function(curKey,nextKey) {
                return counts[curKey] < counts[nextKey];
            });
        }


        function startScanner() {
            Quagga.init({
                inputStream: {
                    name: "Live",
                    type: "LiveStream",
                    target: document.querySelector('#scanner-container'),
                    constraints: {
                        aspectRatio: 4/3,
                        facingMode: "environment"
                    }
                },
                numOfWorkers: 8,
                frequency: 'full',
                locator: {
                    patchSize: 'medium',
                    halfSample: true,
                },
                decoder: {
                    readers: [
                        //"code_128_reader",
                        "ean_reader",
                        "ean_8_reader"
                        //"code_39_reader",
                        //"code_39_vin_reader",
                        //"codabar_reader",
                        //"upc_reader",
                        //"upc_e_reader",
                        //"i2of5_reader"
                    ],
                    debug: {
                        showCanvas: true,
                        showPatches: true,
                        showFoundPatches: true,
                        showSkeleton: true,
                        showLabels: true,
                        showPatchLabels: true,
                        showRemainingPatchLabels: true,
                        boxFromPatches: {
                            showTransformed: true,
                            showTransformedBox: true,
                            showBB: true
                        }
                    }
                },


            }, function (err) {
                if (err) { console.log(err); return }
                Quagga.initialized = true;
                Quagga.start();
            });


            Quagga.onProcessed(function (result) {
                var drawingCtx = Quagga.canvas.ctx.overlay,
                    drawingCanvas = Quagga.canvas.dom.overlay;


                if (result) {
                    if (result.boxes) {
                        drawingCtx.clearRect(0, 0,
parseInt(drawingCanvas.getAttribute("width")), parseInt(drawingCanvas.getAttribute("height")));
                        result.boxes.filter(function (box) {
                            return box !== result.box;
                        }).forEach(function (box) {
                            Quagga.ImageDebug.drawPath(box, {x: 0, y: 1}, drawingCtx, {color:
"green", lineWidth: 2});
                        });
                    }
```

```
                    if (result.box) {
                        Quagga.ImageDebug.drawPath(result.box, {x: 0, y: 1}, drawingCtx,
{color: "#00F", lineWidth: 2});
                    }


                    if (result.codeResult && result.codeResult.code) {
                        Quagga.ImageDebug.drawPath(result.line, {x: 'x', y: 'y'}, drawingCtx,
{color: 'red', lineWidth: 3});
                    }
                }
            });


            var last_result = [];
            if (Quagga.initialized == undefined) {
                Quagga.onDetected(function(result) {
                    var last_code = result.codeResult.code;
                    last_result.push(last_code);
                    if (last_result.length > 30) {
                        code = order_by_occurrence(last_result)[0];
                        last_result = [];
                        // Quagga.stop();
                        document.getElementById("barcode").value = code;
                        document.getElementById("barcode-submit").click();
                    }
                });
            }
        }
```

Index.jsp

```
        <%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
isELIgnored="false" %>
        <!DOCTYPE html>
        <html>
        <head>
            <title>Главная страница</title>
            <link type="text/css" href="/style.css" rel="stylesheet">
        </head>
        <body>
            <% String id = (String) request.getSession().getAttribute("role");
                if (id != null) { %>
                    <jsp:include page="header.jsp"/>
                <%}%>
            <center>
                <h1>Главная страница</h1>

                <h2>Информационная система для управления<br>
                    клиентской базой данных и автоматизаци процесса продаж</h2>
                <h3>Разработал студент группы ИКПИ-11 Дунаев В.Е.</h3>

                <form action="/login">
                    <button class="button-primary" >Войти</button>
                </form>
            </center>
        </body>
        </html>
```

Style.css

```
        .error_message{
            background: darkred;
            border: 0.125rem solid;
            display:flex;
            align-items: center;
            justify-content: center;
```

```
    color: red;
    text-align: center;
    padding: 8px;
    margin-left: auto;
    margin-right: auto;
    width: fit-content;
}

.success_message{
    background: darkgreen;
    border: 0.125rem solid;
    display:flex;
    align-items: center;
    justify-content: center;
    color: lawngreen;
    text-align: center;
    padding: 8px;
    margin-left: auto;
    margin-right: auto;
    width: fit-content ;
}

body {
    background: bisque;
}

.button-primary {
    min-height: 40px;
    padding: 0 15px;
    color: #FFF;
    text-align: center;
    font-weight: 600;
    letter-spacing: .1rem;
    text-transform: uppercase;
    border-radius: 4px;
    border: 1px solid #bbb;
    cursor: pointer;
    background: #505050;
    border-color: #000;
    box-shadow: 2px 2px 1px black;
}

.button-primary:hover{
    color: #FFF;
    background: brown;
    border-color: brown
}

.table-container {
    display: flex;
    flex-flow: column nowrap;
    width: 80%;
    margin: 0 auto;
    border-radius: 4px;
    border: 1px solid #DADADA;
    box-shadow: 0px 1px 4px rgba(0, 0, 0, .08);
}

.table-row {
    display: flex;
    flex-flow: row nowrap;
    width: 100%;
    border-bottom: 1px solid #dadada;
}

.heading {
    background-color: #ececec;
    color: #3e3e3e;
    font-weight: bold;
}

.row-item {
    display: flex;
    flex: 1;
    font-size: 14px;
```

```
    justify-content: center;
    align-items: center;
    transition: all 0.15s ease-in-out;
}

.row-sub-container {
    display: flex;
    flex-flow: column nowrap;
    flex: 1;
}

.row-sub-container .row-item {
    border-bottom: 1px solid #dadada;
}

.table-row:last-child,
.row-sub-container .row-item:last-child {
    border-bottom: 0;
}
.input {
    height: 30px;
    padding: 6px 10px;
    background-color: #fff;
    border: 1px solid #D1D1D1;
    border-radius: 4px;
}
.center {
    margin-left: auto;
    margin-right: auto;
    width: auto;
}
```