Государственное бюджетное профессиональное образовательное учреждение

Ростовской области

|  |
| --- |
| «Таганрогский колледж морского приборостроения» |

|  |
| --- |
| «Аптека» |
|  |
| Курсовой проект |
| Пояснительная записка |
| ТКМП.МДК.02.01.10.20 ПЗ |

|  |  |
| --- | --- |
| Руководитель  *Малыхина О.В.*  Студент П-320  *Чекулаев В.А..* |  |
|  |

2024

**Содержание**

[Задание 3](#_Toc138779628)

[Аннотация 4](#_Toc138779629)

[Введение 5](#_Toc138779630)

[1. Общая часть 6](#_Toc138779631)

[1.1. Постановка задачи 6](#_Toc138779632)

[1.2. Анализ и исследование задачи, построение модели системы 8](#_Toc138779633)

[1.3. Разработка архитектуры решения 10](#_Toc138779634)

[1.4 Обоснование и выбор средств разработки решения 12](#_Toc138779635)

[1.5. Формализация расчетов 13](#_Toc138779636)

[1.5.1. Проектирование базы данных 15](#_Toc138779637)

[1.5.2. Проектирование интерфейса 17](#_Toc138779638)

[2. Описание программы 19](#_Toc138779639)

[3. Тестирование программы 20](#_Toc138779640)

[4 Программирование 22](#_Toc138779641)

[5 Отладка, оптимизация и тестирование программы 24](#_Toc138779642)

[Заключение 28](#_Toc138779643)

[Список используемых источников 29](#_Toc138779644)

[Приложение А. Результаты выполнения программы 30](#_Toc138779645)

**Задание**

Вашей задачей является разработка системы для производственной компании, которая ежедневно выпускает продукцию в большом объеме, а также реализует ее агентам, которые в дальнейшем продают эти товары конечным потребителям.

Для понимания объема разработки и планирования архитектуры приложения заказчик разделил весь проект на не сколько подсистем:

* подсистема для хранения информации о продукции,
* подсистема для работы со складом и материалами,
* подсистема для производства,
* подсистема по работе с сотрудниками,
* подсистема по работе с агентами.

Компания “Супер резинка” занимается производством банковских резинок, а также схожей эластичной продукции. Для нашей продукции мы используем множество вариантов сырья с различными свойствами для производства конечной продукции. Для понимания текущего состояния нашего склада материалов и его своевременного пополнения необходимо создать небольшую информационную систему.

Вам предстоит разработать подсистему для работы с материалами нашей компании, которая должна включать в себя следующий функционал:

* просмотр списка материалов,
* добавление/удаление/редактирование данных о материалах,
* управление списком возможных поставщиков материалов.

**Аннотация**

Курсовой проект представляет собой разработку информационной системы для автоматизации деятельности «МУЗ Городской клинической больницы г. Большие Кабаны» с основным фокусом на подсистеме «Аптека» Информационная система «Аптека» направлена на оптимизацию процессов управления медикаментами, их учета и распределения в рамках клинической больницы. Основная задача проекта заключается в создании интуитивно понятного интерфейса, обеспечивающего эффективное взаимодействие с системой для медицинского персонала. Пользователи будут иметь возможность осуществлять такие операции, как заказ, прием и выдача медикаментов, учет остатков, генерация отчетов о наличии лекарственных средств. Для реализации проекта будут использованы современные технологии веб-разработки, включая HTML, CSS и JavaScript для создания пользовательского интерфейса, а также серверная разработка на PHP для обработки бизнес-логики. База данных будет использоваться для хранения информации о медикаментах, заказах, поставках и других операциях в аптеке клинической больницы.

# **Введение**

В современном здравоохранении, где оперативность и точность играют ключевую роль, эффективная информационная система становится неотъемлемой частью обеспечения качественного медицинского обслуживания. В контексте данного курсового проекта представляем вам информационную систему «Аптека» для МУЗ Городской клинической больницы г. Большие Кабаны.

Цель данного проекта заключается в создании информационной системы, предназначенной для автоматизации работы аптеки больницы. Решение направлено на оптимизацию управления медикаментами, сокращение времени на выполнение заказов и учет остатков лекарственных средств.

Информационная система «Аптека» предлагает широкий спектр функциональных возможностей, обеспечивая оперативное выполнение заказов, контроль остатков медикаментов, а также генерацию отчетов о наличии лекарственных препаратов. Наш подход основан на использовании передовых технологий веб-разработки, а также применении языка программирования PHP для создания мощной и гибкой системы.

Основная особенность Подсистемы «Аптека» заключается в ее разработке на языке программирования PHP. PHP — это широко используемый язык программирования, особенно в веб-разработке, благодаря своей гибкости, простоте и мощности. Использование PHP позволяет нам создавать динамические веб-страницы, взаимодействовать с базами данных и разрабатывать функциональные модули для обработки заказов, управления запасами и многого другого.

**1. Общая часть**

**1.1. Постановка задачи**

Постановка задачи:

Целью данного курсового проекта является разработка веб-ресурса под названием «Аптека» - системы автоматизации работы аптеки больницы, используя язык программирования PHP. Веб-ресурс «Аптека» предназначен для автоматизации работы аптеки больницы.

Для достижения данной цели необходимо решить следующие задачи:

1. Разработка пользовательского интерфейса:

Создать интуитивно понятный и удобный пользовательский интерфейс, который позволит пользователям легко взаимодействовать с системой. Интерфейс должен быть отзывчивым и иметь интуитивно понятные элементы управления для выполнения различных задач, таких как учет препаратов, отслеживание запасов.

1. Учет и отслеживание препаратов:

Реализовать функционал, который позволит фармацевту принимать новые препараты, списывать существующие, а также отслеживать их количество и доступность на складе.

1. Управление запасами:

Создать механизм, позволяющий автоматически оповещать пользователя о низких уровнях запасов и предлагать возможности для пополнения запасов. Система должна предлагать оптимальные рекомендации по дозаказу препаратов, основываясь на уровне спроса и заданных параметрах запасов.

1. Оптимизация распределения товаров:

Реализовать функционал, который поможет оптимизировать распределение препаратов по различным зонам склада, учитывая их характеристики, требования к хранению и скорость продажи. Это позволит эффективно использовать пространство склада и снизить затраты на хранение.

**1.2. Анализ и исследование задачи, построение модели системы**

Анализ задачи:

Для начала необходимо проанализировать основные требования к системе «Аптека». Основной целью данного веб-ресурса должно быть обеспечение удобства и эффективности работы с продукцией.

Основные задачи, которые должны быть решены с помощью системы складского хранения, включают:

Учет и контроль за поступлением и отгрузкой препаратов;

Мониторинг остатков и движения препаратов на складе;

Поддержание актуальности информации о препаратах и их расположении на складе;

Управление заказами на выдачу препаратов.

Исследование технологий:

Для создания веб-ресурса системы «Аптека» можно использовать различные языки программирования и фреймворки. Наиболее распространенным языком программирования для разработки веб-приложений является PHP. Этот язык предоставляет широкие возможности для создания динамических веб-сайтов с использованием баз данных MySQL.

Построение модели системы:

Для построения модели системы необходимо определить ее структуру и функциональные возможности. Разработка системы «Аптека» должна включать следующие компоненты:

База данных для управления информацией о препаратах, клиентах и остатках;

Интерфейс пользователя для взаимодействия с системой;

Механизмы обработки заказов на выдачу препаратов;

Механизмы мониторинга остатков и движения препаратов на складе.

База данных должна содержать информацию о каждом препарате, включая его наименование, количество и расположение на складе. Кроме того, база данных должна содержать информацию о клиентах, заказах на выдачу препаратов и остатках на складе.

Интерфейс пользователя должен предоставлять возможность просмотра информации о товарах, оформления заказов на выдачу, а также мониторинга остатков товаров на складе.

Механизмы обработки заказов на доставку должны включать автоматическое формирование накладных и документов для отслеживания поставок товаров.

Механизмы мониторинга остатков и движения препаратов на складе должны обеспечивать точный и актуальный учет движения препаратов на складе, а также предоставлять отчеты для анализа эффективности работы системы.

Таким образом, создание веб-ресурса системы «Аптека» с использованием языка программирования PHP позволит обеспечить эффективное управление препаратов на складе, повысить его эффективность и сохранить актуальную информацию о препаратах.

## **1.3. Разработка архитектуры решения**

Для разработки подсистемы для работы с препаратами веб-ресурса «Аптека» необходимо определить ее архитектуру. Архитектура системы должна обеспечивать исключительную производительность, надежность и безопасность при работе с данными клиентов и препаратов сайта. В качестве архитектурного решения можно применить следующие подходы:

1. Микросервисная архитектура:

Этот подход предполагает разбиение приложения на независимые микросервисы, каждый из которых отвечает за выполнение отдельной функциональности. Для работы с препаратами можно выделить отдельный микросервис, который будет обеспечивать определенный функционал работы с препаратами. Каждый микросервис может иметь свой собственный индивидуальный набор взаимодействующих с ним процессов и элементов, что снижает вероятность возникновения сбоев и повышает отказоустойчивость системы.

1. Клиент-серверная архитектура:

Это классический подход в веб-разработке, который предполагает разделение приложения на клиентскую и серверную части. Основная логика и данные хранятся на сервере, а клиентская часть отвечает за отображение информации и взаимодействие с пользователем. В данной архитектуре в качестве серверной части может выступать отдельный сервер, на котором будет развернуто приложение для работы с товарами. Клиентская часть приложения будет предоставлять пользователю возможность просматривать, принимать, списывать препараты.

1. Эволюционная архитектура:

Она представляет собой подход к разработке ПО, основанный на постепенном приращении функционала системы. Компоненты и модули системы постепенно добавляются к уже имеющейся инфраструктуре. Для разработки подсистемы для работы с товарами можно начать с небольшого приложения, которое будет обеспечивать просмотр списка товаров и истории покупок. Дополнительный функционал, такой как принятие/списание препаратов. При выборе одного из этих подходов нужно учитывать требования к производительности, ограничения базы данных, требования к защите данных и возможности разработчиков. Комбинация различных архитектурных подходов может обеспечить наилучший баланс между функциональностью, производительностью и безопасностью системы.

## **1.4 Обоснование и выбор средств разработки решения**

Выбранное средство разработки решения для данного курсового проекта на тему Подсистема «Аптека» - веб-ресурс для учета препаратов нашего сайта «Аптека» - состоит из трех основных компонентов: HTML, CSS и JavaScript. Наиболее подходящим и распространенным инструментом, который обеспечивает все эти возможности является PHP. Выбор средства разработки веб-ресурса для данного проекта сделан, исходя из следующих причин:

1. HTML — это стандартный язык разметки веб-страниц, который является основным языком для создания всей структуры и функциональности веб-сайта.

2. CSS — это язык таблиц стилей, который позволяет определить стиль веб-страницы, изменять ее внешний вид и улучшать пользовательский опыт. 3. JavaScript - это язык программирования, который позволяет создавать интерактивные элементы веб-страниц, улучшать навигацию по сайту и предоставлять пользователю средства для взаимодействия со страницей. 4. PHP - это скриптовый язык программирования, который является очень распространенным веб-средством для создания и управления базами данных.

5. База данных сделана через XAMPP. Это кроссплатформенный дистрибутив, который может использоваться на Unix, Linux, Mac и Windows. XAMPP включает в себя Apache, PHP, MySQL и Perl, что делает его идеальным выбором для разработки веб-приложений.

6. PHP является наиболее распространенным языком программирования для создания веб-приложений с открытым исходным кодом. Он поддерживает работу с большинством баз данных, включая MySQL, SQLite, PostgreSQL и другие. В результате, выбор использования CSS, PHP и JavaScript оправдан, так как они позволяют создавать качественный и функциональный веб-ресурс для работы с товарами на сайте. Кроме того, использование XAMPP позволит обеспечить надежность и стабильность базы данных и приложения в целом.

## **1.5. Формализация расчетов**

Для формализации расчетов для разработки подсистемы учета препаратов на веб-ресурсе «Аптека» необходимо определить следующие шаги:

1. Определить ожидаемые результаты работы подсистемы: - Возможность просмотра списка препаратов - Возможность принятия/списания/перемещение препаратов.
2. Определить методы передачи данных: - Использование базы данных через xampp - Использование языков программирования: css, php, javascript.
3. Определить логику работы подсистемы: - Клиенты могут просматривать список препаратов - Фармацевт может принимать/списывать/перемещать препараты.
4. Определить использование инструментов программирования: - Для создания интерфейса можно использовать языки программирования HTML и CSS - Для обработки запросов и отображения данных необходимо использовать сценарии на языке программирования PHP - Для реализации динамической интерактивности можно использовать языки программирования AJAX и JavaScript.
5. Определить требования к базе данных: - Необходимо определить таблицы хранения информации о препаратах.
6. Определить и описать процессы и взаимодействия между компонентами системы: - Клиенты могут просматривать список препаратов. - Фармацевт может принимать/списывать/перемещать препараты. С учетом этих шагов необходимо разработать программный код и написать документацию по проекту.

### **1.5.1. Проектирование базы данных**

Для проектирования базы данных для веб-ресурса «Аптека» необходимо следовать определенной структуре. При этом можно использовать реляционную модель. Следующие таблицы могут быть использованы в данной базе данных:

1. Таблица "Пользователи"

- ID пользователя (уникальный идентификатор)

- ID группы пользователя (уникальный идентификатор группы)

- Логин пользователя

- Адрес электронной почты

- ФИО пользователя

- Пароль пользователя

2. Таблица "Склады"

- ID склада (уникальный идентификатор)

- Название склада

- Адрес склада

3. Таблица "Причины списания"

- ID причины (уникальный идентификатор)

- Название причины

4. Таблица "Препараты"

- ID (уникальный идентификатор)

- ID склада (уникальный идентификатор)

- Название препарата

- Количество препарата на складе

- Дата списания препарата

5. Таблица "Списанные препараты"

- ID (уникальный идентификатор)

- ID препарата (уникальный идентификатор)

- ID причины списания (уникальный идентификатор)

6. Таблица "Заказов"

- ID (уникальный идентификатор)

- ID препарата (уникальный идентификатор)

- Количество списания препарата со склада

- Срочность

- Статус

Для связи этих таблиц можно использовать сводные ключи:

- ID склада из таблицы «Склады» связывается с ID препарата из «Препараты»;

- ID препарат из таблицы «Препараты» связывается с ID препарата из «Списанные препараты»;

- ID причины списания из таблицы «Причины списания» связывается с ID списания из «Списанные препараты»;

- ID препарата из таблицы «Препараты» связывается с ID препарата из «Заказы».

С использованием PHP и JavaScript можно разработать интерфейс для взаимодействия с базой данных для выполнения вышеперечисленных функций. Просмотр списка препаратов может происходить путем запроса на получение всех записей из таблицы «Препараты» и отображения их на странице. Добавление, списание препаратов может осуществляться при помощи SQL-запросов на добавление, списание препаратов в таблице "Препараты". Для удобства учета препаратов фармацевты сервиса могут использовать специальную админ-панель, где будут представлены все вышеперечисленные функции. Таким образом, разработка подсистемы для работы с препаратами имеет несколько ключевых моментов: создание таблицы препаратов, добавление, списание препаратов. База данных может быть создана с помощью XAMPP, а интерфейс взаимодействия с базой данных может быть разработан с использованием PHP и JavaScript.

### **1.5.2. Проектирование интерфейса**

Для успешного проектирования интерфейса подсистемы по работе с препаратами необходимо учитывать следующие пункты:

1. Навигация по сайту должна быть простой и удобной. Для этого можно создать панель навигации, которая будет содержать ссылки на все подсистемы.

2. Главная страница должна содержать основную информацию о компании и ее услугах. Также на главной странице можно разместить список наиболее популярных товаров.

3. Страница со списком препаратов должна содержать таблицу, которая будет включать следующие поля: название препарата, наличие его на складах.

4. Для добавления нового препарата можно создать форму, которая будет содержать необходимые поля для заполнения: название, количество, склад, где будет находиться препарат, срок годности.

5. Для списания препарата из базы данных можно создать кнопку «Списать». При нажатии на данную кнопку препарат должен удалиться из базы данных.

6. Весь интерфейс должен быть адаптивным и подходить для использования на различных устройствах с разными разрешениями экранов.

7. Необходимо также придерживаться общих стандартов и требований к дизайну и интерфейсу веб-страниц, включая использование читабельных шрифтов, цветовую гамму, логотип, и т.д. Написание веб-ресурса на основе CSS, PHP, JavaScript и базы данных созданной в XAMPP, позволит создать профессиональный, удобный и функциональный интерфейс для подсистемы по работе с товарами нашей компании.

# **2. Описание программы**

Программа, разработанная для автоматизации деятельности подсистемы «Аптека» в МУЗ Городской клинической больницы г. Большие Кабаны, представляет собой веб-ресурс, специально адаптированный для управления лекарственными препаратами и складскими операциями.

Основное назначение этой программы заключается в обеспечении оперативного и эффективного учета, распределения и управления запасами медикаментов в рамках клинической больницы. Разработанная система включает в себя широкий спектр функциональных возможностей, предназначенных для облегчения работы персонала аптеки и оптимизации процессов управления лекарственными средствами.

Особое внимание уделено разработке подсистемы «Аптека», которая включает в себя следующий функционал:

- Просмотр списка лекарственных препаратов и их наличие на складах, представленных в удобном для пользователя виде, таком как таблица. Пользователи имеют возможность фильтрации по складам и просмотра информации о количестве конкретного препарата на всех доступных складах.

- Регистрация поступления лекарственных препаратов от поставщиков с обязательной проверкой срока годности партии.

- Ведение списка списания лекарственных препаратов с указанием причины по каждой позиции.

- Формирование списка заявок на выдачу лекарственных препаратов для отделений больницы.

- Подробный просмотр заявки и фиксация отпуска лекарственных препаратов со склада.

- Осуществление перемещения лекарственных препаратов между складами с автоматическим обновлением информации о наличии на каждом складе.

- Генерация отчета о необходимости заказа лекарственных препаратов в зависимости от фактического и оптимального количества, с подсветкой элементов по срочности заказа.

Для разработки веб-ресурса использовались передовые технологии веб-разработки, такие как CSS, PHP и JavaScript. База данных была создана с использованием XAMPP, обеспечивая надежное хранение и управление данными о лекарственных препаратах и операциях на складе. Эти технологии позволили создать простой в использовании и функциональный веб-ресурс, который значительно упрощает и автоматизирует процессы управления аптечным складом клинической больницы.

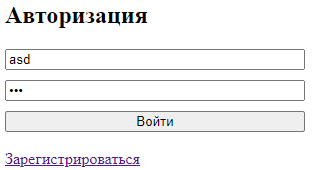
# **3. Тестирование программы**

Способ проведения тестирования:

1. Проверить работоспособности авторизации
2. Проверить работоспособность функций препарата
3. Проверить работоспособность функции выхода из аккаунта

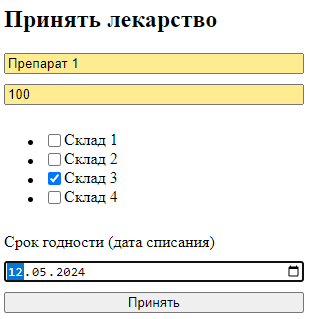
Тестирование:

1. Для проверки авторизации мы введём логин и пароль в соответствующие поля и нажмём кнопку «Войти». Пример авторизации в соответствии с рисунком 1.1.

****

**Рисунок 1.1 Авторизация**

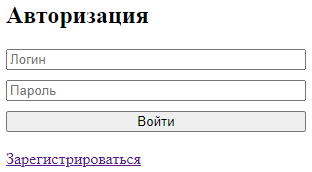
2) Для проверки работоспособности функций принятия препарата мы сначала должны выбрать вариант, который нам подходит.



**Рисунок 1.2 Оформление заказа**

3) Для проверки работоспособности функции выхода из системы, мы должны нажать на кнопку «Выйти», после чего нас перенесут на страницу «Авторизации».



****

**Рисунок 1.3 Выход из системы**

# **4 Программирование**

Система автосервиса спроектирована на языке PHP и БД MySQL.

Описание запросов MySQL:

1. Создание таблицы «Пользователи».

CREATE TABLE `users` (

`id` int(11) NOT NULL,

`role` int(11) NOT NULL,

`login` text NOT NULL,

`email` text NOT NULL,

`fullname` text NOT NULL,

`password` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_general\_ci;

Создание таблицы «Склады».

CREATE TABLE `warehouses` (

`id` int(11) NOT NULL,

`name\_warehouse` varchar(100) NOT NULL,

`address` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_general\_ci;

Создание таблицы «Причины списания».

CREATE TABLE `reasons` (

`id` int(11) NOT NULL,

`name\_reason` varchar(100) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_general\_ci;

Создание таблицы «Препараты».

CREATE TABLE `medications` (

`id` int(11) NOT NULL,

`id\_warehouse` text NOT NULL,

`name\_medication` varchar(100) NOT NULL,

`quantity\_medication` text NOT NULL,

`expiration\_date` date NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_general\_ci;

Создание таблицы «Списанные Препараты».

CREATE TABLE `deleted\_medications` (

`id` int(11) NOT NULL,

`id\_medication` int(11) NOT NULL,

`id\_reason` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_general\_ci;

Создание таблицы «Заказы».

CREATE TABLE `requests` (

`id` int(11) NOT NULL,

`ids\_medical` text NOT NULL,

`quantityes` text NOT NULL,

`urgency` tinyint(1) NOT NULL DEFAULT 0,

`status` tinyint(1) NOT NULL DEFAULT 0

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_general\_ci;

Сценарии в коде программы:

1. login.php – Авторизация

Входные данные: логин и пароль  
Выходные данные: вход на сайт

1. delete\_medication.php – Списание препарата из таблицы

Входные данные: ID препарата, ID причины списания.  
Выходные данные: списанные препараты

1. giveout\_medications.php – Выдача препаратов в поликлинику.

Входные данные: ID заказа.  
Выходные данные: изменённая строка

1. db.php – Подключение к базе данных

Входные данные: Название БД, пароль, имя учётной записи phpMyAdmin  
Выходные данные: Данные в таблицах

# **5 Отладка, оптимизация и тестирование программы**

Способ проведения тестирования:

1. Проверить работоспособности авторизации.
2. Проверить работоспособность функций принятия и списания препаратов.
3. Проверить работоспособность функции перемещения препарата между складами.
4. Проверить работоспособность функции выдачи заказов поликлинике.
5. Проверить работоспособность функции выхода из аккаунта.

Тестирование:

1. Для проверки авторизации мы введём логин и пароль в соответствующие поля и нажмём кнопку «Войти». Пример авторизации в соответствии с рисунком 1.4.

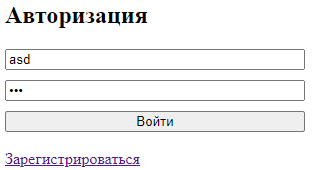
****

Рисунок 1.4 – Авторизация

1. Для проверки функций принятия и списания препарата введём данные в поля для добавления и списания препарата, и нажмём кнопку «Принять» / «Списать» в соответствии с рисунком 1.6 – рисунком 1.7.

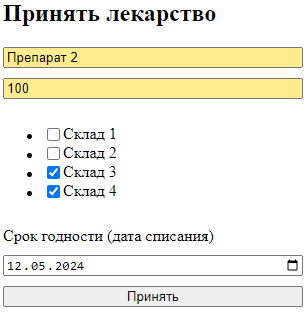


Рисунок 1.6 – Принятие препарата

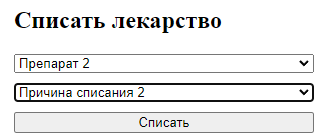


Рисунок 1.7 – Списание препарата

1. Для проверки возможности перемещения препарата между складами, заполним форму перемещения и нажмем кнопку «Переместить», в соответствии с рисунком 1.8.

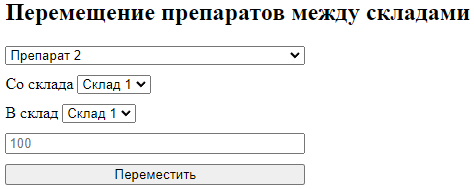


Рисунок 1.8 – Перемещение препарата

1. Для проверки возможности выдачи заказа поликлинике, в списке заказов нажмем на кнопку «Выдать», в соответствии с рисунком 1.9.

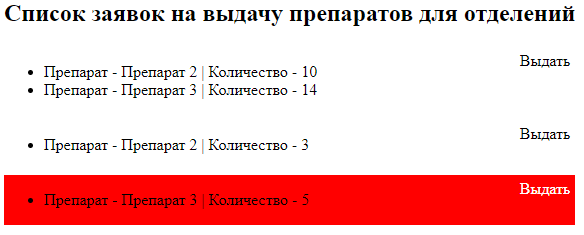


Рисунок 1.9 – Выдача препарата поликлинике

Для проверки возможности выхода из аккаунта, нажмем на кнопку «Выйти», после чего нас должно перенести на страницу авторизации, в соответствии с рисунком 2.0 – рисунок 2.1.



Рисунок 2.0 – Кнопка выхода из системы

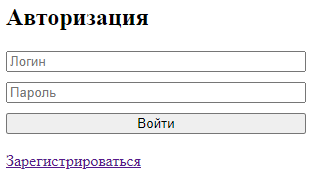


Рисунок 2.1 – Страница авторизации после выхода из системы

# **Заключение**

В ходе разработки информационной системы "Аптека" для МУЗ Городской клинической больницы г. Большие Кабаны была поставлена задача создать эффективный инструмент для автоматизации управления лекарственными препаратами и складскими операциями. Данная задача была анализирована, спроектирована, реализована и успешно выполнена.

В ходе работы над проектом были изучены и применены передовые технологии веб-разработки, включая CSS, PHP и JavaScript, а также базы данных с использованием XAMPP. Программа предоставляет широкий спектр функциональных возможностей, позволяющих управлять процессами поступления, хранения и выдачи лекарственных препаратов, а также обеспечивает генерацию необходимых отчетов для оптимизации складских операций.

В результате работы над проектом были выявлены некоторые недостатки, однако система остается весьма перспективной и может быть доработана и улучшена в дальнейшем. В целом, разработанная информационная система "Аптека" является значимым шагом в направлении современной автоматизации складских операций клинической аптеки, обеспечивая более эффективное управление запасами лекарственных препаратов и повышение качества медицинского обслуживания.

# **Список используемых источников**

1. <https://metanit.com/php/tutorial/>
2. <https://metanit.com/web/html5/>
3. <https://metanit.com/sql/>
4. <https://metanit.com/sql/mysql/>

# **Приложение А. Результаты выполнения программы**

При входе на сайт нас встречает окно авторизации, предназначенное для Пользователя сайта в соответствии с рисунком 2.2.

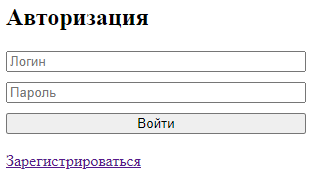


Рисунок 2.2 – Окно авторизации

После ввода логина и пароля нужно нажать кнопку «Войти», клиента переносит на страницу с препаратами в соответствии с рисунком 2.3.

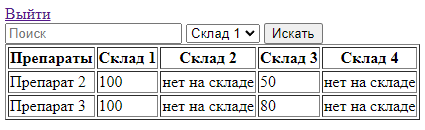


Рисунок 2.3 - Страница с препаратами

Далее есть возможность поиска препарата как на отдельных складах, так и во всех складах соответствии с рисунком 2.4.

Рисунок 2.4 - Страница оформления заказа