

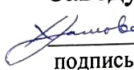
Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ЛЕСОСИБИРСКИЙ ПЕДАГОГИЧЕСКИЙ ИНСТИТУТ –
филиал Сибирского федерального университета

Высшей математики, информатики, экономики и естествознания
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

 Л.Н. Храмова
подпись инициалы, фамилия

« 11 » июня 2025 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 Информационные системы и технологии
код-наименование направления

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ
ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ СЕТИ КОМПЬЮТЕРНЫХ
МАГАЗИНОВ «КОМПЬЮТЕРНЫЙ МАГАЗИН»


Руководитель

 11.06.2025
подпись, дата

доцент, канд. пед. наук
должность, ученая степень

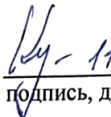
С.С. Ахтамова
инициалы, фамилия

Выпускник

 11.06.2025
подпись, дата

Д.А. Мошуренко
инициалы, фамилия

Нормоконтролер

 11.06.2025
подпись, дата

Е.В. Киргизова
инициалы, фамилия

Лесосибирск 2025

РЕФЕРАТ

Выпускная квалификационная работа по теме «Проектирование и разработка автоматизированной информационной системы для сети компьютерных магазинов «Компьютерный магазин» содержит 55 страниц текстового документа, 31 иллюстрация, 7 таблиц, 41 использованный источник.

АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА, ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ (MySQL), ВЕБ-ПРИЛОЖЕНИЕ, ЯЗЫК ПРОГРАММИРОВАНИЯ (PHP)

Цель исследования – теоретически обосновать и разработать автоматизированную информационную систему для сети компьютерных магазинов.

Объект исследования – автоматизированная информационная система для сети компьютерных магазинов.

Предмет исследования – процесс проектирования и разработки автоматизированной информационной системы для сети компьютерных магазинов.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Изучить и проанализировать существующие технологии процесса разработки автоматизированных информационных систем в виде веб-приложения.
2. Обосновать выбор архитектуры информационной системы с учетом требований безопасности, производительности и удобства.
3. Спроектировать и разработать автоматизированную информационную систему в виде веб-приложения с использованием современных инструментальных средств.

В выпускной квалификационной работе представлен процесс проектирования и разработки автоматизированной информационной системы для сети компьютерных магазинов.

Разработанная автоматизированная информационная система в виде веб-приложения включает в себя информацию о товарах и ценах, услугах и сотрудниках, способствует автоматизации указанной информации.

СОДЕРЖАНИЕ

Введение.....	4
1 Теоретические основы проектирования автоматизированной информационной системы для сети компьютерных магазинов	7
1.1 Архитектурные решения для сети компьютерных магазинов	7
1.2 Анализ используемого программного обеспечения и программных средств	13
1.3 Разработка технического задания на проектирование клиент-серверного приложения автоматизированной информационной системы.....	25
2 Разработка клиент-серверного приложения автоматизированной информационной системы	29
2.1 Установка программного обеспечения и проектирование автоматизированной информационной системы.....	29
2.2 Реализация регистрации и авторизации пользователей	34
2.3 Разработка основных страниц информационной системы	36
Заключение.....	48
Список использованных источников	49
Приложение А Код файлов reg. php, log.php и logout.php	53
Приложение Б Код скриптов на JavaScript	55

Введение

В наше время интернет стал неотъемлемой частью жизни почти каждого человека, предлагая доступ к неисчислимому количеству источников информации и видов услуг. Сегодня наиболее актуальными автоматизированными информационными системами считаются сервисы, позволяющие в реальном времени взаимодействовать пользователям с данными. Данная выпускная квалификационная работа посвящена разработке автоматизированной информационной системы с авторизацией и регистрацией пользователей пользователей, и поддержкой вывода информации в реальном времени.

Тема проекта была выбрана исходя из современной необходимости работать в сервисах, которые упростят жизнь пользователям и сделают их взаимодействие с данными комфортнее. Отображение важной информации в режиме реального времени, позволит не только просматривать данные, но и анализировать их в таких разных областях бизнеса и логистики.

Цель исследования – теоретически обосновать и разработать автоматизированную информационную систему для сети компьютерных магазинов.

Объект исследования – автоматизированная информационная система для сети компьютерных магазинов.

Предмет исследования – процесс проектирования и разработки автоматизированной информационной системы для сети компьютерных магазинов.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Изучить и проанализировать существующие технологии процесса разработки автоматизированных информационных систем в виде веб-приложения.

2. Обосновать выбор архитектуры информационной системы с учетом требований безопасности, производительности и удобства.

3. Спроектировать и разработать автоматизированную информационную систему в виде веб-приложения с использованием современных инструментальных средств.

Создание информационной системы с авторизацией добавит персонализацию пользователя и обеспечит его безопасность. Пользователи смогут зарегистрироваться, авторизоваться с использованием аккаунта, и взаимодействовать с автоматизированной информационной системой.

В рамках выполнения данной работы рассмотрены ключевые технологии, используемые для создания автоматизированных информационных систем, как для клиентской, так и серверной части приложения. Особую задачу в разработке автоматизированной информационной системы составит выбор системы управления базы данных, в которой будут храниться данные о пользователях и информация о товарах и ценах, услугах и сотрудниках.

В ходе работы также проведён анализ уже существующих решений в данной предметной области, для выявления положительных и отрицательных сторон реализации выявленных подходов к задаче проектирования и разработки автоматизированной информационной системы. А сложившаяся практика и собственные умозаключения помогут создавать инновационный продукт, который будет отвечать современным требованиям пользователей и рынка.

Практическая значимость исследования заключается в разработке автоматизированной информационной системы для сети компьютерных магазинов в виде веб-приложения будет способствовать не только дальнейшему развитию студента в области веб-разработки, но и внесут значительный вклад в существующие решения в данной области, обеспечив пользователей удобным инструментом для работы с данными.

По результатам исследования опубликованы статьи в сборник конференций:

1. Важность поддержки БД и ИС в образовательных организациях для обеспечения учебного процесса / Мошуренко Д.А., Ахтамова С.С. // Человек и язык в коммуникативном пространстве: сб. науч. ст. [Электронный ресурс] / отв. редактор М. В. Веккессер. – Электрон. дан. (1,67 Мб). – Красноярск: Сиб. федер. ун-т, 2025. Вып. 16 (25). – 208 с.

2. Роль баз данных и информационных систем в математическом образовании студентов / Д. А. Мошуренко, С. С. Ахтамова / Актуальные проблемы преподавания дисциплин естественнонаучного цикла: тез. докл. VIII Всерос. науч.-практ. конф. Лесосибирск, 14 – 15 ноября 2024 г. [Электронный ресурс] / отв. за вып. Л.Н. Храмова. – Электрон. дан. (2,42 Мб). – Красноярск: Сиб. федер. ун т, 2024. – 171 с.

Структура работы – работа состоит из введения, двух глав, заключения и списка использованных источников, включающего 41 источник. Результаты работы представлены в 7 таблицах, 31 рисунке. Общий объем работы – 55 печатных листов.

1 Теоретические основы проектирования автоматизированной информационной системы для сети компьютерных магазинов

Архитектура автоматизированных информационных систем в виде веб-приложения важна для определения принципа их работы, возможности расширения, а также их стабильности. Сейчас, когда цифровые технологии стремительно развиваются, правильный выбор архитектуры чрезвычайно важен для успешного осуществления проектов. Архитектура веб-приложений включает множество аспектов, таких как организация кода, связь между частями или модулями системы и управление данными. Знание этих моментов помогает создавать приложения, которые соответствуют текущим требованиям и готовы к будущим изменениям под нужды пользователей.

1.1 Архитектурные решения для сети компьютерных магазинов

Архитектура автоматизированных информационных систем определяет то, как различные компоненты системы взаимодействуют друг с другом для обеспечения функциональности, производительности и масштабируемости. В этом разделе мы рассмотрим существующие архитектурные решения разных уровней.

Клиент-серверная архитектура – это модель взаимодействия, где запросы на выполнение определенных операций инициируются клиентами, а серверы обрабатывают эти запросы и возвращают результаты [21]. Эта архитектура широко используется в веб-приложениях и других сетевых сервисах, обеспечивая четкое разделение обязанностей между клиентской и серверной частями системы.

Суть взаимодействия в том, что клиентские компьютеры отправляют запросы на сервер и используют полученные данные для выполнения своих задач. Обычно клиентские и серверные компьютеры находятся на разных физических уровнях и взаимодействуют друг с другом через сеть. Однако они

также могут располагаться на одном физическом уровне. Пример схемы взаимодействия клиентской и серверной части на рисунке 1.

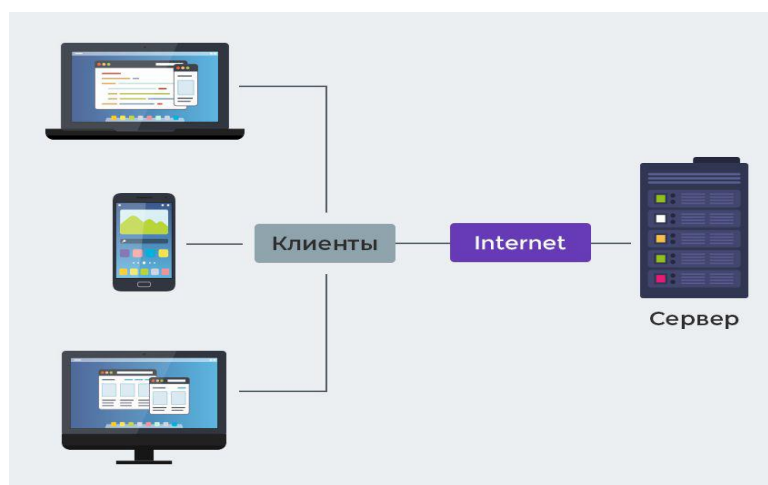


Рисунок 1 – Схема взаимодействия клиентской и серверной части

Эта модель предполагает разделение задач между клиентскими и серверными компьютерами, что позволяет более эффективно управлять ресурсами и данными. На сегодняшний день существует несколько различных типов клиент-серверной архитектуры, каждый из которых имеет свои особенности, преимущества и недостатки. Рассмотрим основные виды клиент-серверной архитектуры и их особенности применения, а также выделим наиболее подходящий вид для реализации.

Уровневые архитектуры различаются по количеству уровней, на которые разделена система. Каждый тип архитектуры имеет свои особенности, преимущества и недостатки.

Одноуровневая архитектура – это простейшая форма организации программного обеспечения, где вся система представлена одним уровнем. Все функции выполняются в одном модуле [12].

Преимущества:

- простота: одноуровневые системы просты в разработке и понимании, так как все функции находятся в одном месте;
- быстрота разработки: из-за своей простоты такие системы можно быстро создать и внедрить.

Недостатки:

- ограниченная масштабируемость: одноуровневые системы сложно масштабировать, так как все компоненты тесно связаны;
- сложность поддержки: с увеличением размера системы поддержка и обновление становятся сложнее.

Двухуровневая архитектура разделяет систему на два уровня: клиентский и серверный. Клиент отвечает за взаимодействие с пользователем, а сервер – за обработку данных и ответ клиенту.

Преимущества:

- разделение обязанностей: клиент и сервер выполняют свои функции, что упрощает управление системой;
- улучшенная производительность: сервер может быть оптимизирован для выполнения сложных вычислений, в то время как клиент сосредоточен на пользовательском интерфейсе и несложных операциях.

Недостатки:

- задержки сети: взаимодействие между клиентом и сервером может вызывать задержки, особенно при большом объеме данных;
- одна точка отказа: если сервер выходит из строя, клиенты не могут получить доступ к услугам.

На рисунке 2 изображена схема работы двухуровневой архитектуры.

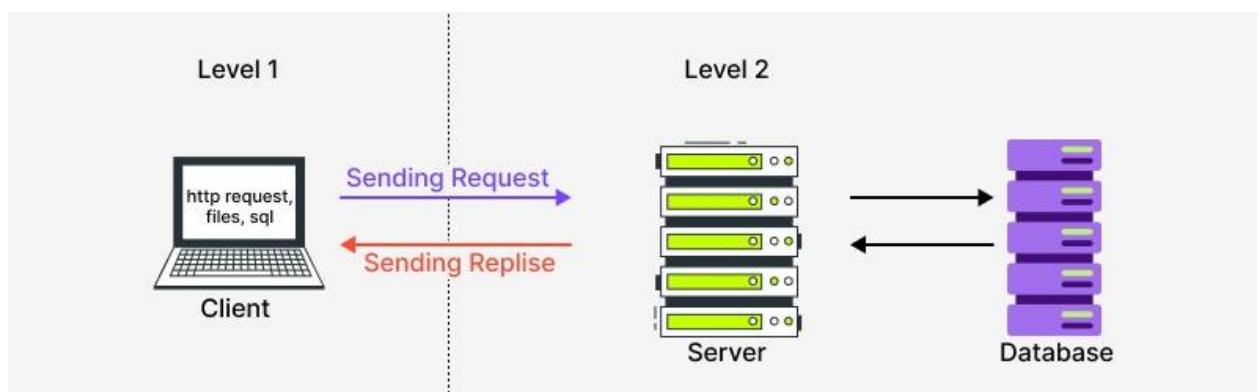


Рисунок 2 – Двухуровневая архитектура клиент-сервер

Трехуровневая архитектура добавляет промежуточный уровень между клиентом и сервером, который отвечает за бизнес-логику. Этот уровень называется уровнем приложений [13].

Преимущества:

- гибкость: легче вносить изменения в бизнес-логику без необходимости изменять клиентскую или серверную часть;

- масштабируемость: каждый уровень можно масштабировать независимо, что улучшает производительность системы.

Недостатки:

- сложность разработки: требуется больше усилий для проектирования и реализации;

- увеличенные затраты: разработка и поддержка трехуровневой архитектуры могут быть дороже.

На рисунке 3 изображена схема работы трехуровневой архитектуры.

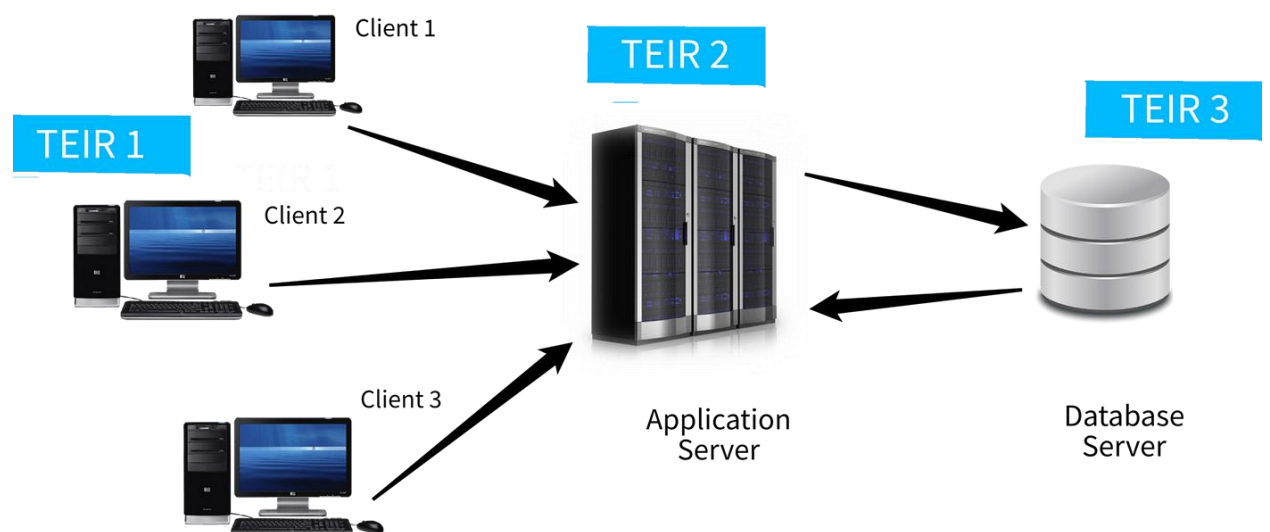


Рисунок 3 – Трехуровневая архитектура клиент-сервер

Многоуровневая архитектура разделяет систему на несколько уровней, каждый из которых выполняет специфические функции. Это может включать уровни безопасности, кэширования, интеграции и другие.

Преимущества:

- высокая масштабируемость: каждый уровень можно оптимизировать и масштабировать независимо;
- улучшенная безопасность: разделение на уровни позволяет лучше контролировать доступ к данным и функциям.

Недостатки:

- высокая сложность: управление множеством уровней требует тщательного планирования и координации;
- затраты на разработку: многоуровневые системы требуют значительных ресурсов для проектирования и внедрения.

На рисунке 4 изображена схема работы многоуровневой архитектуры.

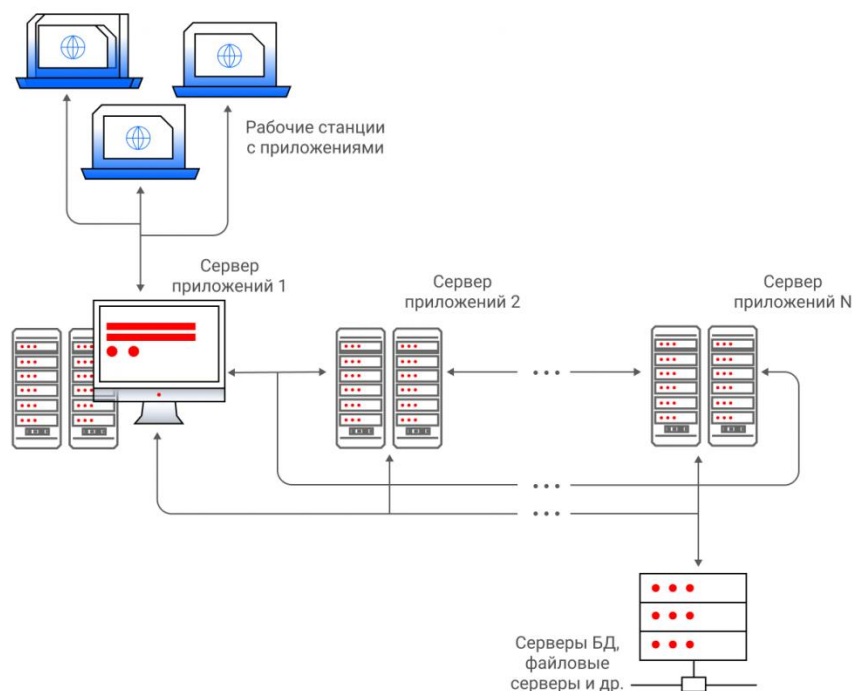


Рисунок 4 – Многоуровневая архитектура клиент-сервер

Каждый из этих типов имеет свои особенности и подходит для различных типов задач. Выбор архитектуры зависит от требований к системе, таких как масштабируемость, производительность и удобство поддержки.

Исходя из вышеперечисленных решений, наиболее подходящей для разработки приложения архитектурой является двухуровневая архитектура, так

как при ней хранение базы данных и развернутое приложение находятся на одной и той же машине.

Клиентская часть веб-приложения отвечает за взаимодействие с пользователем. Она включает в себя интерфейс, через который пользователи могут просматривать информацию и выполнять различные действия. Основные технологии, используемые для разработки клиентской части, включают HTML, CSS и JavaScript.

Серверная часть веб-приложения отвечает за обработку запросов клиентов, управление данными и обеспечение безопасности. Она включает в себя серверы, которые обрабатывают запросы, и базы данных, которые хранят информацию.

Базы данных используются для хранения и управления данными, которые необходимы для работы веб-приложения. API для карт позволяют интегрировать картографические функции в приложение.

Для разработки веб-приложений используются различные технологии и инструменты, которые упрощают процесс разработки и повышают качество конечного продукта.

Средства разработки:

Visual Studio Code: это популярная среда разработки, которая поддерживает множество языков программирования и имеет большое количество расширений. Она позволяет удобно редактировать код, отлаживать приложения и интегрироваться с системами контроля версий.

Инструменты для тестирования:

– selenium используется для автоматического тестирования веб-интерфейсов. Он позволяет создавать тесты, которые проверяют работу приложения в различных браузерах;

– pytest используется для написания и выполнения тестов на Python. Он обеспечивает высокую степень покрытия кода тестами и позволяет легко интегрироваться с другими инструментами.

1.2 Анализ используемого программного обеспечения и программных средств

Кроме выбора архитектуры, важно также выбрать язык программирования и систему управления базами данных (СУБД), для создания и редактирования базы данных (БД), а также дополнительные технологии для разработки веб-приложения. Эти решения определяют, как будет работать приложение, как оно будет взаимодействовать с пользователями и как будут обрабатываться данные. Для создания клиент-серверного веб-приложения выбраны следующие решения:

Для серверной части:

1. Язык программирования PHP

– PHP – это популярный язык для веб-разработки. Он прост в изучении и использовании, что делает его подходящим для создания динамических веб-страниц. PHP имеет большое сообщество разработчиков, что облегчает поиск решений и ресурсов. Он также хорошо интегрируется с различными базами данных, такими как MySQL [14].

– роль: PHP используется для создания динамических веб-страниц. Он позволяет взаимодействовать с базами данных, обрабатывать формы и управлять сессиями пользователей;

– преимущества: PHP прост в изучении и использовании, имеет большое количество встроенных функций и поддерживается большинством хостинг-провайдеров.

2. Система управления базами данных (СУБД) MySQL

– MySQL – это мощная система управления базами данных, которая позволяет эффективно хранить и управлять данными. Она поддерживает сложные запросы и может обрабатывать большие объемы информации. MySQL также обеспечивает высокую производительность и надежность, что важно для стабильной работы приложения [11].

– роль: MySQL используется для хранения и управления данными в веб-приложениях. Она позволяет выполнять сложные запросы и обрабатывать большие объемы данных;

– преимущества: MySQL оптимизирована для выполнения сложных запросов и обработки больших объемов данных. Она поддерживает репликацию и кластеризацию, что позволяет масштабировать базу данных для удовлетворения растущих потребностей.

3. Локальный веб-сервер: XAMPP

– XAMPP – это пакет программного обеспечения, который включает в себя Apache-сервер, MySQL, PHP и Perl. Он позволяет быстро и легко создать локальную среду разработки. XAMPP поддерживается на разных операционных системах, таких как Windows, Linux и macOS, что делает его универсальным инструментом для разработки.

– роль: XAMPP используется для создания локальной среды разработки. Он включает в себя Apache-сервер, MySQL, PHP и Perl;

– преимущества: XAMPP позволяет быстро и легко установить все необходимые компоненты для разработки веб-приложений. Он поддерживается на Windows, Linux и macOS, что обеспечивает гибкость в выборе операционной системы.

Для клиентской части:

Языки программирования: HTML, CSS, JS

1. HTML (HyperText Markup Language) используется для создания структуры веб-страниц. Он определяет, какие элементы будут отображаться на странице, такие как заголовки, параграфы и изображения;

2. CSS (Cascading Style Sheets):

– роль: CSS используется для описания внешнего вида и форматирования HTML-документов. Также позволяет управлять стилями тегов, такими как цвета, шрифты и расположение элементов на странице;

– преимущества: CSS позволяет создавать сложные и адаптивные дизайны, которые хорошо отображаются на различных устройствах. Он также позволяет отделить содержимое страницы от её внешнего вида, что упрощает поддержку и обновление.

3. JavaScript (JS):

– роль: JavaScript используется для создания интерактивных элементов на веб-страницах. Он позволяет добавлять динамическое поведение, такое как анимации, валидация форм и взаимодействие с пользователем в реальном времени;

– преимущества: JavaScript поддерживается всеми современными браузерами и имеет большое сообщество разработчиков. Он позволяет создавать динамические и интерактивные веб-приложения, реагирующие на действия пользователя.

PHP (Hypertext Preprocessor) – это серверный язык программирования, который был создан Расмусом Лердорфом в 1994 году. Изначально он назывался «Personal Home Page Tools» и использовался для отслеживания посещений онлайн-резюме Лердорфа [34]. В 1995 году PHP был выпущен как открытый проект, что позволило разработчикам со всего мира вносить свой вклад в его развитие. С тех пор PHP прошел долгий путь, превратившись из простого инструмента для создания динамичных веб-страниц в мощный язык программирования, поддерживающий сложные веб-приложения.

Развитие PHP можно проследить через несколько ключевых версий. В 1997 году была выпущена версия PHP 3, которая включала поддержку баз данных и различных протоколов, что значительно расширило функциональность языка. В 2000 году появилась версия PHP 4, которая представила новый движок Zend Engine, существенно улучшивший производительность. Следующей важной вехой стала версия PHP 5, выпущенная в 2004 году, которая добавила поддержку объектно-ориентированного программирования и другие улучшения. Однако, наиболее значительные изменения принесла версия PHP 7, выпущенная в 2015

году, которая представила значительные улучшения производительности и новые функции, такие как строгая типизация и анонимные классы.

PHP был создан для упрощения разработки веб-приложений. Он позволяет легко встраивать динамический контент в HTML-страницы, что делает его идеальным для создания интерактивных веб-сайтов. PHP работает на стороне сервера, обрабатывая запросы пользователей и генерируя HTML-код, который затем отправляется клиенту для отображения в браузере. Это позволяет разделить логику приложения и представление данных, что упрощает разработку и поддержку кода.

Одной из сильных сторон PHP является его простота и доступность. Язык имеет понятный синтаксис, что делает его подходящим для начинающих разработчиков. Кроме того, PHP поддерживается большинством хостинг-провайдеров, что упрощает развертывание веб-приложений. PHP также обладает высокой производительностью и может обрабатывать большое количество запросов одновременно, что делает его подходящим для сайтов с высокой нагрузкой. Большое сообщество разработчиков PHP активно делится знаниями и решениями, что облегчает поиск помощи и ресурсов.

Однако, у PHP есть и свои слабые стороны. Одной из них является производительность, которая может уступать другим серверным языкам, таким как Java или C#, особенно в сложных вычислительных задачах. Кроме того, PHP может быть уязвим для атак, таких как SQL-инъекции, если не соблюдать меры безопасности при написании кода. Также, из-за своей простоты, PHP может поощрять плохие практики программирования, такие как дублирование кода и отсутствие структурированности, если не придерживаться рекомендаций по написанию качественного кода.

Несмотря на эти недостатки, PHP остается одним из самых популярных языков для веб-разработки благодаря своей гибкости, простоте и широкой поддержке. Он используется в различных проектах, от небольших личных сайтов до крупных корпоративных приложений, что подтверждает его универсальность и надежность.

MySQL – это система управления базами данных (СУБД), которая широко используется для хранения и управления данными в веб-приложениях [34]. Она была создана для обеспечения надежного и эффективного доступа к данным, и с тех пор стала одной из самых популярных СУБД в мире.

MySQL была разработана шведской компанией MySQL AB, и первая версия была выпущена в 1995 году. Основной целью создания MySQL было предоставление быстрой и надежной системы управления базами данных, которая могла бы работать с большими объемами данных. С тех пор MySQL прошла через множество версий, каждая из которых добавляла новые функции и улучшения. В 2008 году MySQL AB была приобретена компанией Sun Microsystems, а позже, в 2010 году, Sun Microsystems была куплена компанией Oracle Corporation. Несмотря на смену владельцев, MySQL продолжает развиваться и остается открытым программным обеспечением.

MySQL была создана для удовлетворения потребностей веб-разработчиков, которым требовалась мощная и гибкая система для управления данными. Она позволяет создавать, читать, обновлять и удалять данные с помощью языка запросов SQL (Structured Query Language) [39]. MySQL поддерживает различные типы данных и позволяет создавать сложные запросы для извлечения нужной информации из базы данных. Это делает её идеальной для использования в веб-приложениях, где требуется быстрый доступ к данным и их обработка.

Одной из сильных сторон MySQL является её высокая производительность. Она оптимизирована для выполнения сложных запросов и может обрабатывать большие объемы данных с высокой скоростью. MySQL также поддерживает репликацию и кластеризацию, что позволяет масштабировать базу данных для удовлетворения растущих потребностей приложения. Эти функции делают MySQL подходящей для использования в крупных проектах, где требуется высокая доступность и отказоустойчивость.

Еще одним преимуществом MySQL является её широкая поддержка и большое сообщество пользователей. MySQL поддерживается большинством

хостинг-провайдеров, что упрощает развертывание веб-приложений. Большое сообщество разработчиков активно делится знаниями и решениями, что облегчает поиск помощи и ресурсов. Кроме того, MySQL является открытым программным обеспечением, что позволяет использовать её без лицензионных затрат.

Однако, у MySQL есть и свои слабые стороны. Одна из них – это ограниченная поддержка NoSQL-решений. MySQL в первую очередь ориентирована на работу с реляционными данными, что может быть ограничением для приложений, требующих хранения неструктурированных данных [39]. Также, некоторые функции MySQL доступны только в коммерческих версиях, что может увеличить затраты на разработку.

Несмотря на эти недостатки, MySQL остается одной из самых популярных систем управления базами данных благодаря своей производительности, гибкости и широкой поддержке. Она используется в различных проектах, от небольших веб-сайтов до крупных корпоративных приложений, что подтверждает её универсальность и надежность.

Связка PHP и MySQL является популярным выбором для разработки веб-приложений, и многие успешные проекты использовали эту комбинацию, особенно на начальных этапах своего развития. Вот несколько примеров:

1. WordPress: это одна из самых популярных платформ для создания блогов и веб-сайтов. WordPress изначально был разработан на PHP и использовал MySQL в качестве системы управления базами данных. Благодаря своей гибкости и простоте использования, WordPress стал основой для миллионов сайтов по всему миру;

2. Facebook: на ранних этапах своего развития Facebook использовал PHP и MySQL для создания своей платформы. Это позволило быстро разрабатывать и масштабировать социальную сеть, которая впоследствии стала одной из крупнейших в мире. Хотя с течением времени Facebook перешел на другие технологии, PHP и MySQL сыграли ключевую роль в его начальном успехе;

3. Joomla: это еще одна популярная система управления контентом (CMS), которая использует PHP и MySQL. Joomla предоставляет мощные инструменты для создания и управления веб-сайтами, и её используют многие организации и компании для своих онлайн-присутствий;

4. Drupal: это мощная и гибкая CMS, которая также использует PHP и MySQL. Она известна своей масштабируемостью и возможностью создания сложных веб-приложений. Drupal используется многими крупными организациями и правительственными учреждениями для создания своих веб-сайтов;

5. MediaWiki: это программное обеспечение для вики, которое используется для создания Википедии. MediaWiki написан на PHP и использует MySQL для хранения данных. Благодаря своей мощной функциональности и масштабируемости, MediaWiki поддерживает одну из самых посещаемых веб-платформ в мире.

Эти примеры демонстрируют, как связка PHP и MySQL может быть использована для создания успешных и масштабируемых веб-приложений, которые обслуживают миллионы пользователей по всему миру [34].

Связка PHP и MySQL считается одним из самых популярных комплексов инструментов для создания веб-сайтов и клиент-серверных приложений по нескольким причинам. Эти технологии дополняют друг друга, обеспечивая удобство и эффективность в разработке. Давайте рассмотрим, почему эта связка так популярна [23].

1. Легкость изучения и использования:

– Язык PHP был специально разработан для веб-разработки, и его синтаксис достаточно прост для понимания. Это делает его доступным для начинающих разработчиков;

– MySQL эта система управления базами данных также имеет понятный интерфейс и простой язык запросов SQL, что облегчает работу с данными [28].

2. Большое сообщество разработчиков:

– У PHP есть огромное сообщество пользователей и разработчиков, которые активно делятся знаниями и решениями. Это означает, что можно легко найти помощь и ресурсы для решения проблем;

– MySQL также имеет большое сообщество, и многие разработчики делятся своими наработками и советами по оптимизации работы с базой данных.

3. Хорошая совместимость:

– PHP и MySQL. отлично работают вместе. PHP имеет встроенные функции для взаимодействия с MySQL, что упрощает процесс подключения к базе данных и выполнения запросов [31];

– поддержка хостинга: большинство хостинг-провайдеров поддерживают PHP и MySQL, что делает их удобным выбором для разработки и развертывания веб-приложений.

4. Высокая производительность:

– PHP может обрабатывать большое количество запросов одновременно, что делает его подходящим для сайтов с высокой нагрузкой;

– MySQL оптимизирована для выполнения сложных запросов и может эффективно управлять большими объемами данных.

5. Масштабируемость:

– MySQL поддерживает репликацию и кластеризацию, что позволяет масштабировать базу данных для удовлетворения растущих потребностей приложения.

6. Меры безопасности:

– PHP имеет встроенные механизмы безопасности, такие как защита от SQL-инъекций и XSS-атак [34];

– MySQL обеспечивает безопасность данных с помощью различных механизмов аутентификации и шифрования.

7. Бесплатное использование:

– PHP и MySQL являются открытым программным обеспечением, что позволяет использовать их без лицензионных затрат. Это делает их доступными для разработчиков с любым бюджетом.

Выбор языка программирования пал на PHP, так как он позволяет создавать приложения с такими характеристиками: поддержка разных платформ, относительно высокая скорость работы, широкие возможности и быстрая разработка программного обеспечения.

Детально рассмотрим технологии для реализации клиентской части приложения: язык разметки HTML, каскадные таблицы стилей CSS и язык программирования, выполняемый в среде браузера, JavaScript.

HTML (Hyper Text Markup Language) – это стандартный язык разметки для создания веб-страниц. Он был разработан в конце 1980-х годов Тим Бернерс-Ли, чтобы структурировать и отображать контент в интернете [1]. HTML позволяет определять элементы веб-страницы, такие как заголовки, параграфы, изображения и ссылки. С каждой новой версией HTML становился более функциональным, добавляя новые элементы и возможности для создания интерактивных и динамичных веб-страниц. HTML используется для определения структуры веб-страницы, в то время как стили и поведение элементов управляются с помощью CSS и JavaScript соответственно. Это разделение обязанностей упрощает разработку и поддержку веб-сайтов, позволяя разработчикам сосредоточиться на различных аспектах создания веб-страниц.

Одной из сильных сторон HTML является его универсальность. Он поддерживается всеми веб-браузерами, что делает его незаменимым инструментом для создания веб-страниц. HTML прост в изучении и использовании, что позволяет даже начинающим разработчикам быстро освоить основы создания веб-страниц. Однако, HTML сам по себе не обеспечивает динамическое взаимодействие с пользователем. Для создания интерактивных элементов на веб-странице необходимо использовать JavaScript. Кроме того, HTML не предоставляет средств для стилизации элементов – для этого

используется CSS. Это означает, что для создания современного веб-сайта необходимо использовать несколько технологий вместе с HTML. Несмотря на эти ограничения, HTML остается основой веб-разработки и постоянно развивается, добавляя новые возможности с каждой новой версией.

CSS (Cascading Style Sheets) – это язык стилей, используемый для описания внешнего вида и форматирования HTML-документов [1]. Он был разработан для разделения содержания веб-страницы и её представления, что упрощает управление дизайном и улучшает доступность. CSS позволяет разработчикам управлять цветами, шрифтами, расположением элементов и другими аспектами оформления веб-страниц. Это делает веб-сайты более привлекательными и удобными для пользователей [24].

Одной из ключевых особенностей CSS является его гибкость. С помощью CSS можно создавать сложные и адаптивные дизайны, которые хорошо отображаются на различных устройствах, от настольных компьютеров до мобильных телефонов. Это достигается за счет использования медиазапросов, которые позволяют применять разные стили в зависимости от размера экрана или других характеристик устройства. Кроме того, CSS поддерживает анимации и переходы, что позволяет добавлять динамические эффекты на веб-страницы без использования JavaScript. Это делает CSS мощным инструментом для создания интерактивных и визуально привлекательных веб-приложений.

JavaScript (JS) – это язык программирования, который используется для добавления интерактивности и динамического поведения на веб-страницы. Он был разработан в 1995 году Бренданом Эйхом и изначально назывался Mocha, а позже был переименован в LiveScript, и, наконец, в JavaScript [34]. Изначально JavaScript был создан для работы в браузере Netscape Navigator, но со временем стал поддерживаться всеми современными браузерами.

JavaScript позволяет разработчикам создавать динамические веб-страницы, которые могут реагировать на действия пользователей в реальном времени. Это достигается за счет возможности манипулировать элементами DOM (Document Object Model), что позволяет изменять содержимое и стиль веб-страницы без

необходимости её перезагрузки. Например, JavaScript может использоваться для проверки данных в формах, создания анимаций, обновления контента без перезагрузки страницы и взаимодействия с пользователем через события, такие как клики мыши или нажатия клавиш.

Одной из сильных сторон JavaScript является его универсальность. Он поддерживается всеми современными браузерами, что делает его незаменимым инструментом для создания интерактивных веб-приложений. JavaScript также работает на стороне клиента, что позволяет выполнять код непосредственно в браузере пользователя, снижая нагрузку на сервер. Это особенно важно для приложений, требующих мгновенного отклика, таких как онлайн-игры или чаты.

JavaScript также поддерживает объектно-ориентированное программирование (ООП), что позволяет создавать сложные и масштабируемые приложения. С помощью ООП разработчики могут создавать классы и объекты, которые можно повторно использовать и легко поддерживать. Это делает код более структурированным и удобным для чтения.

Кроме того, JavaScript имеет богатую экосистему библиотек и фреймворков, таких как React, Angular и Vue.js, которые упрощают разработку сложных веб-приложений. Эти инструменты предоставляют готовые компоненты и архитектурные решения, которые помогают разработчикам быстрее создавать интерактивные и отзывчивые интерфейсы.

Однако, у JavaScript есть и свои слабые стороны. Одной из них является безопасность. JavaScript-код может быть уязвим для атак, таких как межсайтовый скриптинг (XSS), если не соблюдать меры предосторожности при написании кода. Также, из-за своей гибкости и динамичности, JavaScript может поощрять плохие практики программирования, такие как использование глобальных переменных или отсутствие структурированности кода [34].

Несмотря на эти недостатки, JavaScript остается одним из самых популярных языков программирования для веб-разработки. Он используется не только для создания интерактивных веб-страниц, но и для разработки мобильных приложений с помощью таких технологий, как React Native, а также для

серверной разработки с помощью Node.js. Это делает JavaScript универсальным инструментом, который позволяет создавать полноценные приложения, работающие на различных платформах.

Для упрощения и ускорения процесса разработки важно настроить веб-сервер. Однако это требует от разработчика практического опыта и специальных знаний, которые не у каждого есть. Поэтому проще использовать готовые комплекты с предустановленными приложениями и настройками. Такие программные решения включают LAMPP, OpenServer, Denwer и XAMPP. Среди этих вариантов наиболее подходящим для работы в среде Windows является веб-сервер XAMPP. Рассмотрим поближе каждое решение и сравним по критериям бесплатного распространения, поддержки от разработчика, поддержка ОС Windows и размер дистрибутива для работы в таблице 1.

Таблица 1 – Анализ решений для заведения веб-сервера

Критерий	XAMPP	Openserver	LAMPP	Denwer
Бесплатное распространение	Да	Да	Да	Да
Поддержка от разработчика	Да	Да	Да	Нет
Поддержка ОС Windows	Да	Да	Нет	Да
Размер дистрибутива	500МБ	15ГБ	Зависит от стека	50МБ

Исходя из вышесказанного, приведем стек технологий и программных средств, который будет использован в ходе разработки в таблице 2.

Таблица 2 – Стек технологий и программных средств для разработки

Название	Описание
ОС Windows 11	Операционная система Windows 64-х разрядная, разработанная компанией Microsoft
XAMPP	Набор приложений, сконфигурированных для совместной работы: веб-сервер Apache, СУБД MySQL, язык программирования PHP
PHP7	Серверный скриптовый язык программирования, используемый в большинстве случаев для веб-программирования
MySQL 7	Система управления базами данных от компании Microsoft, предназначенная для не сильно нагруженных файл-серверных сетей
HTML 5	Язык для структурирования и представления содержимого веб-страниц в браузере
CSS	Формальный язык декодирования и описания внешнего вида документа или веб-страницы

На основании описанного ранее, делаем выбор в пользу двухуровневой архитектуры, языка программирования PHP, СУБД MySQL и использования комплекса решений XAMPP для запуска локального сервера. Данный стек соответствует основным требованиям к приложению и позволит работать с приложением любому, у кого есть браузер и доступ в интернет.

1.3 Разработка технического задания на проектирование клиент-серверного приложения автоматизированной информационной системы

Техническое задание:

1. Введение

Настоящее техническое задание (далее – ТЗ) описывает требования к созданию автоматизированной информационной системы в виде веб-приложения приложения с авторизацией пользователей и отображением интерактивных

данных. Цель проекта – разработать автоматизированную информационную систему в виде веб-приложения, которое позволяет системному администратору осуществлять: мониторинг заявок, остатков товара, информации о сотрудниках и доходах-расходах автоматизированный учет. На основе этого улучшиться качество оказываемых услуг, основываясь на статистике

2. Цели:

- мониторинг заявок, остатков товара, информации о сотрудниках и доходах-расходах;
- автоматизация учета.

Задачи:

- разработать клиентскую и серверную часть автоматизированной информационной системы для обработки запросов и управления данными;
- обеспечить безопасность данных и защиту от несанкционированного доступа.

3. Функциональные требования

Рассмотрим функциональные требования для клиентской части. Страницы должны включать:

- регистрация и авторизация пользователей;
- предоставлять доступ для работы с информацией.

Страницы для системного администратора включают:

- управление данными в автоматизированной информационной системе;
- просмотр данных о статусе заявок, товарах на складе, операций купли-продажи и списка пользователей.

Серверная часть разрабатываемой автоматизированной информационной системы включают:

- обработка запросов;
- управление базой данных;
- обеспечение безопасности и защиты данных;
- хранение всех данных на собственном сервере.

4. Нефункциональные требования

К нефункциональным требованиям относятся: технические требования, производительность, безопасность:

- локальная сеть с пропускной способностью не менее 100Мбит/с;
- сервер (процессор 3.5 ГГц, ОЗУ 16Гб, свободное место на накопителе не менее 500 Мб);
- рабочие станции (процессор 2.5 ГГц, ОЗУ 4Гб, свободное место на накопителе не менее 500Мб).
- время отклика сервера на запросы не должно превышать 2 секунд;
- система должна поддерживать одновременную работу не менее 1000 пользователей.
- использование протоколов шифрования данных для передачи данных;
- защита данных от изменения и перехвата.
- масштабируемость
- возможность масштабирования системы для поддержки увеличивающегося числа пользователей и данных.
- удобство использования
- поддержка различных устройств (ПК, планшеты, смартфоны);
- интуитивно понятный и удобный интерфейс для пользователей.

5. Технические требования

Отдельно выделим клиентскую часть разрабатываемой автоматизированной информационной системы:

- платформы: веб-браузеры (Chrome, Firefox, Edge);
- средства разработки: HTML, CSS, JS.

К техническим требованиям серверной части автоматизированной информационной системы относятся:

- Веб-сервер XAAMP;
- СУБД MySQL;
- языки программирования PHP;
- среда выполнения – Microsoft Windows 10-11.

6. Этапы разработки

- выбор общей технологии построения автоматизированной информационной системы;
- выбор интегрированной среды разработки;
- выбор СУБД;
- проектирование базы данных;
- проектирование автоматизированной информационной системы;
- реализация автоматизированной информационной системы.

7. Заключение

Создание автоматизированной информационной в виде клиент-серверного веб-приложения с авторизацией пользователей и отображением данных поможет автоматизировать процессы мониторинга и учёта заявок, товара на складе и списки сотрудников, улучшить взаимодействие с клиентами и повысить качество услуг. ТЗ описывает основные функциональные и технические требования к системе, а также этапы разработки и внедрения.

Таким образом, на основании вышеописанных технических и функциональных требований приступим к описанию процесса разработки автоматизированной информационной системы для сети компьютерных магазинов в формате веб-приложения.

Выводы по первой главе

Исходя из вышеизложенного, для проектирования автоматизированной информационной системы для сети компьютерных магазинов в формате веб-приложения определили выбор типа архитектуры, выбор набора технологий и ПО для разработки проекта, а также составили план и требования к разработке. Комплекс этих аспектов составляет важный базис, на который будет опираться следующий этап разработки.

2 Разработка клиент-серверного приложения автоматизированной информационной системы

Для начала разработки нужно подготовить и настроить рабочее ПО, установить необходимые пакеты расширений и подготовить проект, согласно которому будет идти процесс создания.

2.1 Установка программного обеспечения и проектирование автоматизированной информационной системы

Для начала работы следует запустить XAMPP и включить в интерфейсе программы модули Apache и MySQL.

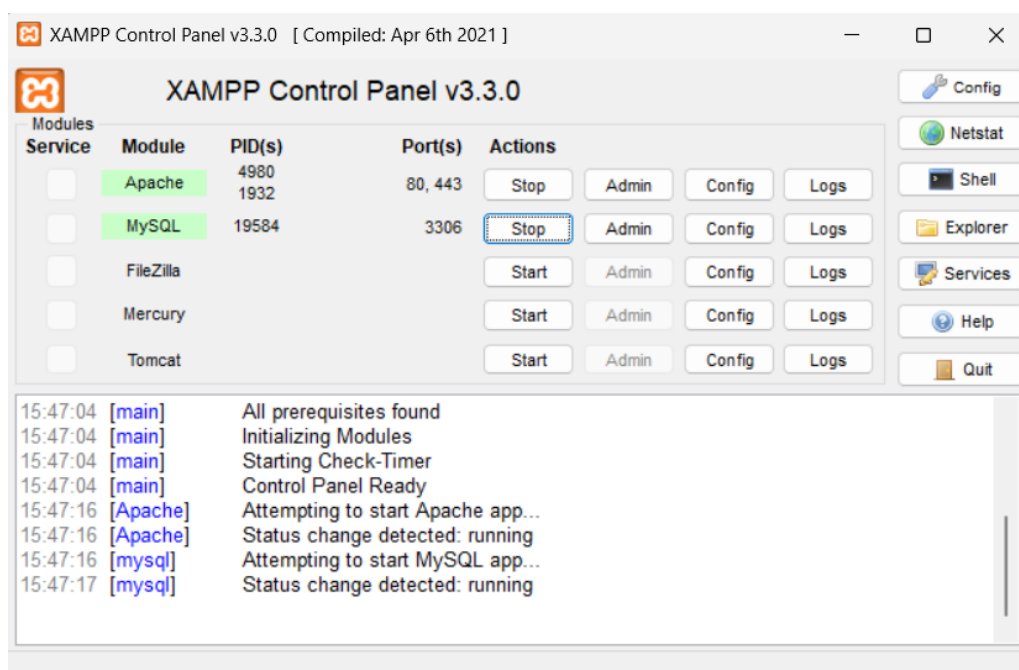


Рисунок 5 – Скриншот запущенных модулей Apache и MySQL

Далее заходим на локальную страницу администрирования phpMyAdmin. Для этого в любом браузере необходимо зайти на локальный сайт, URL которого: <http://localhost/phpmyadmin/>. На рисунке 6 представлен интерфейс страницы phpMyAdmin.

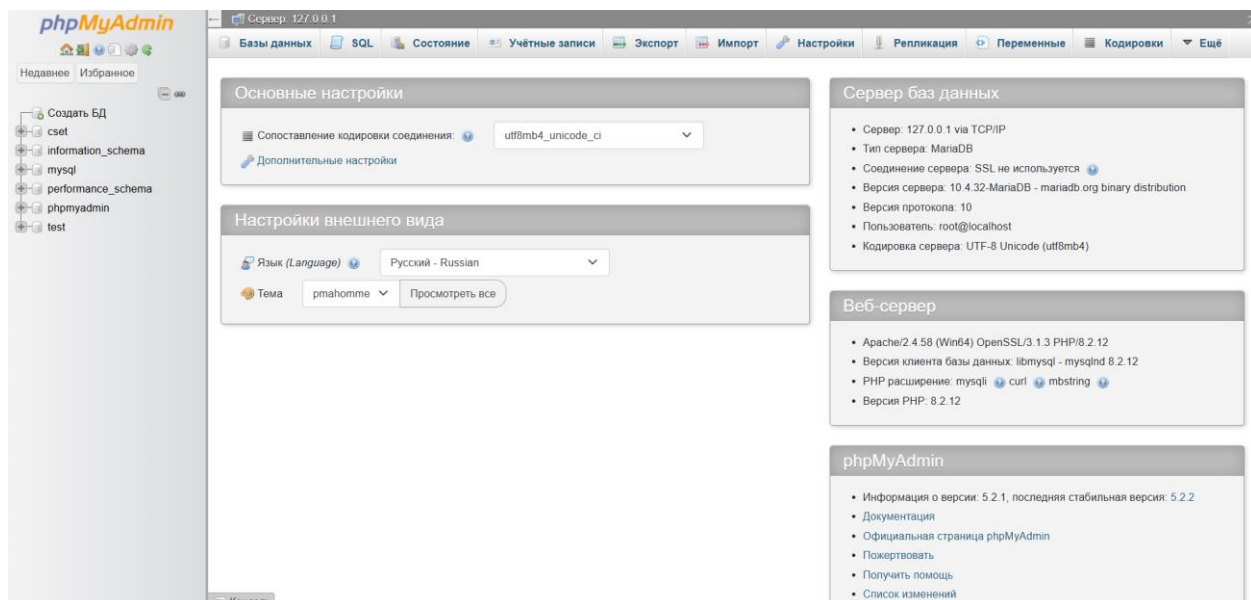


Рисунок 6 – Скриншот интерфейса страницы phpMyAdmin

Далее создадим новую базу данных, дадим ей имя и создадим пароль при необходимости.

На рисунке 7 показан процесс создания БД.

Базы данных

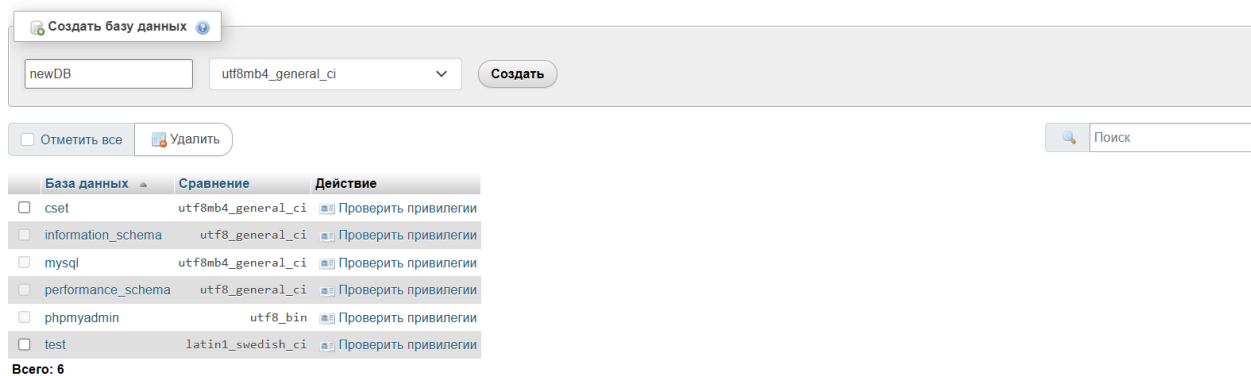


Рисунок 7 – Скриншот процесса создания БД

В новой базе данных создаём таблицы для хранения данных. Для этого составим схему таблиц: названия таблиц, столбцы, тип данных и что именно будет храниться. В таблицах 3-7 описана структура таблиц для БД cset.

Таблица 3 – Описание таблицы employes БД cset

Employes		
ID	Integer	Уникальный идентификатор
Name	Text	Фамилии и Имя
Position	Text	Должность
Salary	Integer	Заработная плата
Role	Text	Роль в системе

Таблица 4 – Описание таблицы sales БД cset

Sales		
ID	Integer	Уникальный идентификатор
Date	Date	Дата
Category	Text	Категория операции
Sum	Integer	Сумма
Who	Text	Кем проведена операция

Таблица 5 – Описание таблицы services БД cset

Services		
ID	Integer	Уникальный идентификатор
Name	Text	Имя заказчика
Type_serv	Text	Тип сервиса
Executor	Text	Исполнитель
State	Text	Статус заявки
Price	Integer	Стоимость Сервис

Таблица 6 – Описание таблицы sklad БД cset

Sklad		
ID	Integer	Уникальный идентификатор
Name	Text	Наименование товара
Category	Text	Категория товара
Proizv	Text	Производитель
Count	Integer	Количество
Price	Integer	Стоимость

Таблица 7 – Описание таблицы users БД cset

Users		
ID	Integer	Уникальный идентификатор
employee_id	Integer	ID пользователя
log	Text	Логин
pass	Text	Пароль
role	Text	Роль в системе

На рисунке 8 показана схема связей на основании вышеописанных таблиц в БД cset.



Рисунок 8 – Схема базы данных

После создания всех таблиц, приступим к написанию листинга программы для функционирования автоматизированной информационной системы с использованием редактора кода Visual Studio Code в котором установим дополнительные расширения из каталога: PHP Server, PHP Intelephense, HTML CSS Support. На рисунке 9 представлен интерфейс расширений.

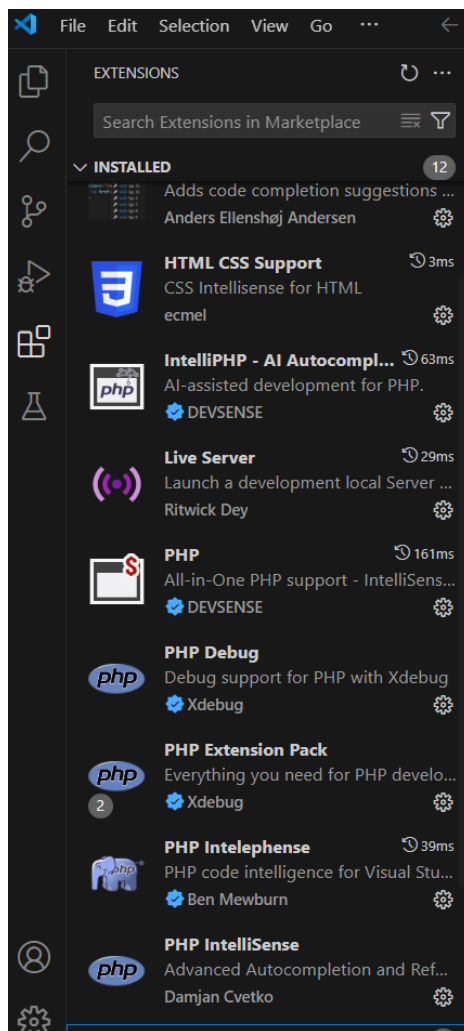


Рисунок 9 – Скриншот интерфейса вкладки расширений

PHP Server позволяет для запускать и отлаживать сервер на PHP, PHP Intelephense необходим для работы PHP Server, а также предоставляет удобный функционал для написания и редактирования кода на языке PHP. Расширение HTML CSS Support так же помогает с комфортом заниматься версткой HTML-документов и CSS-файлов.

После установки этих расширений можно приступать к написанию файлов для запуска их на сервере.

2.2 Реализация регистрации и авторизации пользователей

Для реализации регистрации и авторизации пользователей для автоматизированной информационной системы для сети компьютерных магазинов, создадим в папке проекта файлы log.php, reg.php, logout.php и logreg.html. Файлы log.php, reg.php будут содержать в себе функции для авторизации и регистрации соответственно, logout содержит функцию для выхода из сессии, документ logreg.html содержит страницу с формой авторизации и регистрации.

Разберем код для реализации функционала регистрации и авторизации. Для использования информации из БД создадим подключение к базе данных. Код для подключения к базе данных представлен на рисунке 10.

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "cset";

$conn = new mysqli(hostname: $servername, username: $username, password: $password, database: $dbname)

if ($conn->connect_error) {
    die("Ошибка подключения: " . $conn->connect_error);
}
```

Рисунок 10 – код подключения к базе данных

Этот код присутствует во всех файлах, где необходимо осуществлять взаимодействие с БД, поэтому он будет использован во всех файлах, взаимодействующих с БД.

Рассмотрим описание функций регистрации и авторизации.

В приложении А представлен код файлов reg.php, log.php и logout.php, процесс регистрации пользователя проходит с использованием хеширования пароля в БД. Для хранения пароля в защищённом виде, используются заранее

готовые sql-выражения во избежание возможных sql-инъекций, что позволит повысить надёжность передачи данных.

По умолчанию новому пользователю назначается роль «пользователь», а не «администратор», что разграничивает права доступа к информационной системе.

При авторизации запоминается ID пользователя и открывается сессия, это нужно для отображения в шапке сайта имя авторизованного пользователя, а также, чтобы при занесении в таблицу sales из этой сессии записывать в таблицу имя пользователя. Это помогает отказаться от связности таблиц и взаимодействия только лишь с пользователем в рамках сессии, что также обеспечит надёжность обработки данных.

Приступим к созданию формы регистрации и авторизации, с возможностью выбора процедуры авторизации либо регистрации.

Также был применен JavaScript, для открытия вкладки регистрации или авторизации в качестве функции openTab. На рисунке 11 представлен код функции openTab.

```
function openTab(evt, tabName) {  
    var i, tabcontent, tablinks;  
    tabcontent = document.getElementsByClassName("tabcontent");  
    for (i = 0; i < tabcontent.length; i++) {  
        tabcontent[i].classList.remove("active");  
    }  
    tablinks = document.getElementsByClassName("tablinks");  
    for (i = 0; i < tablinks.length; i++) {  
        tablinks[i].classList.remove("active");  
    }  
    document.getElementById(tabName).classList.add("active");  
    evt.currentTarget.classList.add("active");  
}  
  
document.getElementById("defaultOpen").click();
```

Рисунок 11 – Код функции openTab

На рисунке 12 представлена страница авторизации и регистрации.

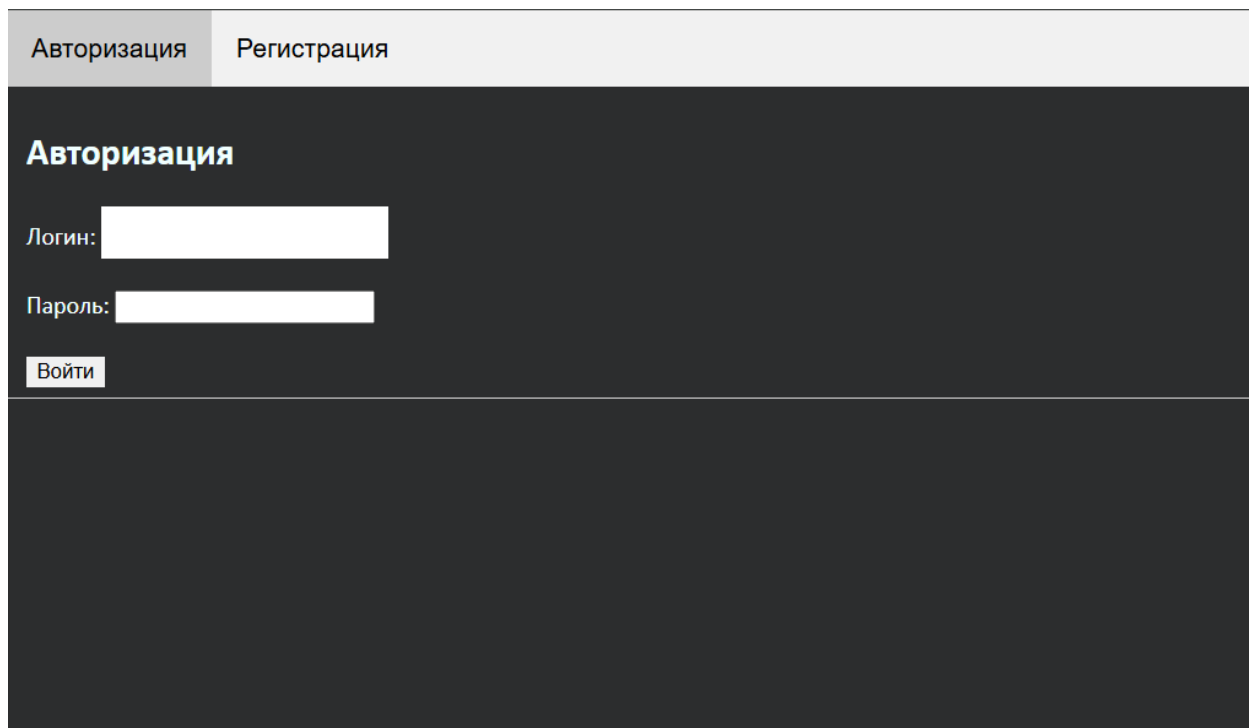


Рисунок 12 – Скриншот страницы авторизации и регистрации

Создадим основные страницы в автоматизированной информационной системе. Для пользователя в роли администратора их будет 4 «Склад», «Персонал», «Сервис» и «Отчеты», а для пользователя (без роли администратора) и будет 2 – «Главная» и «Сервис». Подразумевается, что пользователь без роли администратора – сотрудник, отвечающий за продажу и закупку товаров на склад, а также сотрудник сервисного центра.

2.3 Разработка основных страниц информационной системы

Создадим отдельные файлы `header_usual.php` и `header.php` для разных ролей пользователей, эти файлы будут подгружаться сами на разных страницах ИС, в зависимости от роли авторизованного пользователя.

Далее по тексту, содержание заголовков в файлах `header_usual.php` и `header.php` будем называть шапками страниц.

Для администратора доступны вкладки «Склад», «Персонал», «Сервис» и «Отчёты».

В шапке для неавторизованного пользователя есть всего 2 вкладки «Главная» и «Сервис». Подразумевается, что для работы с услугами и продажами-закупками этого функционала для персонала хватает.

На рисунках 13 – 14 показаны шапки сайта для пользователя с ролью администратора и без роли администратора.



Рисунок 13 – Скриншот шапки сайта для авторизованного пользователя с ролью администратора



Рисунок 14– Скриншот шапки сайта для авторизованного пользователя без роли администратора

Опишем функционал для страницы «Склад»: на странице показываться все записи таблицы «sklad», функционал открытия каждой конкретной записи с возможностью её редактирования или удаления, возможность добавления новой записи.

Рассмотрим реализацию вывода записей из таблицы «sklad». Для этого создадим файл stock.php, напишем код для подключения к базе данных и вызова всех записей из таблицы с помощью запросов SQL, разметим страницу, чтобы динамично отображалась соразмерная таблица с данными.

Для открытия конкретной записи таблицы, необходимо нажать на поле «Name» интересующей строки, для этого реализована функция openRecord.

Далее создадим файл record_staff.php для открытия выбранной записи таблицы. Код в файле содержит такие функции как deleteRecord – функция удаления записи, updateRecord – функция обновления записи в таблице и goBack – функция возвращения на страницу назад в истории.

Также были созданы вызываемые функциями файлы `update_record_stock.php` и `delete_record.php`.

Реализуем функционал добавления новых записей в таблицу, через создание файла `add_record_stock.php` и формы добавления новой записи `add_stock.html`.

Файл `add_record_stock.php` вызывается после заполнения формы в html-документе и записывает в таблицу данные из формы, на рисунке 15 представлена форма для добавления новой записи.

Рисунок 15 – Скриншот формы добавления новой записи

28	Rode NT-USB	микрофон	Rode	5	16000
29	WD 1000gb	CPU	Intel	2	80000

Рисунок 16 – Скриншот добавленной записи в таблицу

Теперь попробуем исправить эту запись и изменить столбец стоимости, для этого надо нажать на наименование товара, откроется карточка записи, на рисунке 17 продемонстрирована карточка записи.

[<<-Назад](#)

Позиция на складе

Наименование	WD 1000gb
Категория	CPU
Производитель	Intel
Количество	2
Цена	80000

Сохранить изменения

Рисунок 17 – Скриншот формы редактирования записи

В этой форме можно изменить наименование товара, категорию товара, производителя, количество и цену. В нашем случае изменим цену и уберем лишний ноль, после чего нажмём на кнопку «Сохранить изменения». Исправленная запись в таблице продемонстрирована на рисунке 18.

28	Rode NT-USB	микрофон	Rode	5	16000
29	WD 1000gb	CPU	Intel	2	8000

Рисунок 18– Отредактированная запись

Страница «Склад» создана и представлена на рисунке 19.

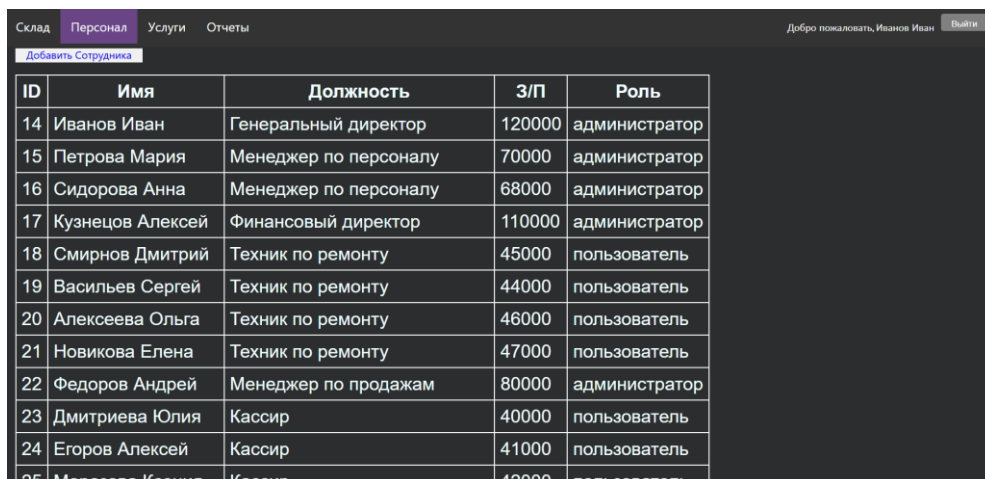
Склад Персонал Услуги Отчеты Добро пожаловать, Иван Иванович Выйти

Добавить товар

ID	Наименование	Категория	Производитель	Количество	Цена
1	Intel Core i7-11700K	CPU	Intel	1	35000
2	AMD Ryzen 9 5900X	ЦП	AMD	8	32000
3	Corsair RM750x	блок питания	Corsair	1	12000
4	EVGA SuperNOVA 850 G5	блок питания	EVGA	6	13000
5	ASUS ROG Strix B550-F	материнская плата	ASUS	7	18000
6	MSI MPG B550 Gaming Plus	материнская плата	MSI	5	17000
7	NVIDIA GeForce RTX 3080	ГПУ	NVIDIA	3	85000
8	AMD Radeon RX 6800 XT	ГПУ	AMD	4	80000
9	Corsair Vengeance LPX 16GB	оперативная память	Corsair	15	7000
10	G.Skill Trident Z RGB 16GB	оперативная память	G.Skill	12	7500
11	Samsung 970 EVO Plus 1TB	SSD	Samsung	8	12000
12	WD Black SN850 1TB	SSD	Western Digital	9	13000

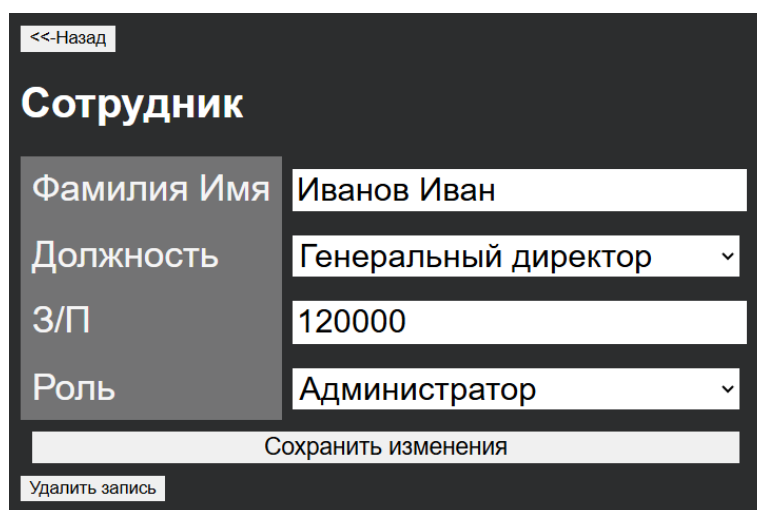
Рисунок 19 – Страница «Склад»

Далее создадим страницу «Персонал». Принцип реализации не сильно отличается от реализации страницы «Склад» с поправкой формы добавления записи и карточки записи в соответствии со структурой таблицы «employee». На Рисунках 20-22 представлены скриншоты страниц «Персонал», карточки записи и формы добавления записи.



ID	Имя	Должность	З/П	Роль
14	Иванов Иван	Генеральный директор	120000	администратор
15	Петрова Мария	Менеджер по персоналу	70000	администратор
16	Сидорова Анна	Менеджер по персоналу	68000	администратор
17	Кузнецов Алексей	Финансовый директор	110000	администратор
18	Смирнов Дмитрий	Техник по ремонту	45000	пользователь
19	Васильев Сергей	Техник по ремонту	44000	пользователь
20	Алексеева Ольга	Техник по ремонту	46000	пользователь
21	Новикова Елена	Техник по ремонту	47000	пользователь
22	Федоров Андрей	Менеджер по продажам	80000	администратор
23	Дмитриева Юлия	Кассир	40000	пользователь
24	Егоров Алексей	Кассир	41000	пользователь
25	Морозова Ксения	Кассир	42000	пользователь

Рисунок 20– Страница «Персонал»



<< Назад

Сотрудник

Фамилия Имя: Иванов Иван

Должность: Генеральный директор

З/П: 120000

Роль: Администратор

Сохранить изменения

Удалить запись

Рисунок 21 – Карточка записи таблицы «employee»

The image shows a web form titled "Добавить нового сотрудника" (Add new employee). It contains four input fields: "Фамилия Имя" (Surname Name) with the value "Аксенов Петр", "Должность:" (Position:) with a dropdown menu showing "Кассир", "Оклад:" (Salary:) with the value "40000", and "Роль:" (Role:) with a dropdown menu showing "Сотрудник". A "Добавить запись" (Add record) button is located at the bottom right of the form.

Рисунок 22 – Форма добавления записи в таблицу «employees»

Создадим страницу «Сервис», она будет отличаться наличием фильтра записей таблицы по признаку статуса заявки. Для этого predeterminedены все состояния заявок: принята, в процессе, отказ и выполнена. Очевидно, что функционал вывода таблицы, добавления, редактирования и удаления записей в таблице будет схожим с предыдущими примерами.

Для реализации фильтрации используется метод GET, с помощью которого при выборе активного варианта фильтра на сервер передаётся переменная со значением параметра, это значение подставляется в SQL-выражение, создавая выборку, которая уже потом выводится тем же методом, что и таблица на страницу. На рисунках 23-24 показана страница «Сервис» и результат работы фильтра записей.

<div> <div>Все</div> <div>Принята</div> <div>В процессе</div> <div>Отказ</div> <div>Выполнена</div> </div>					
№	Имя/Организация	Тип услуги	Исполнитель	Статус	Цена
1	Поликлиника №56	диагностика	Смирнов Дмитрий	принята	1150
2	Ватугин И.А	сборка	Новикова Елена	в процессе	2000
3	"Гемотест"	ремонт	Васильев Сергей	отказ	1500
4	Коргина А.Ю	диагностика	Смирнов Дмитрий	выполнена	1200
5	Организация А	диагностика	Смирнов Дмитрий	принята	1000
6	Клиент Б	сборка	Новикова Елена	в процессе	2000
7	Организация В	ремонт	Васильев Сергей	отказ	1500
8	Клиент Г	диагностика	Смирнов Дмитрий	выполнена	1200

Рисунок 23 – Страница «Сервис»

<div> <div>Все</div> <div>Принята</div> <div>В процессе</div> <div>Отказ</div> <div>Выполнена</div> </div>					
№	Имя/Организация	Тип услуги	Исполнитель	Статус	Цена
1	Поликлиника №56	диагностика	Смирнов Дмитрий	принята	1150
5	Организация А	диагностика	Смирнов Дмитрий	принята	1000

Рисунок 24 – Страница «Сервис» с фильтром записей только в статусе принята

По аналогии с предыдущими страницами были реализованы карточки записи и формы добавления новой записи для таблицы «services» страницы «Сервис». Ниже на рисунках 25-26 представлены карточки записи услуг и форма добавления новой записи.

<< Назад

Услуга

Имя/Организация Поликлиника №56

Тип услуги диагностика ▾

Исполнитель Смирнов Дмитрий ▾

Статус принята ▾

Стоимость 1150

Сохранить изменения

Удалить запись

Рисунок 25 – Карточка записи таблицы «services»

Добавить новую заявку

Имя/Организация: Магазин "Для вас"

Тип услуги диагностика ▾

Исполнитель Смирнов Дмитрий ▾

Статус В процессе ▾

Стоимость 2000

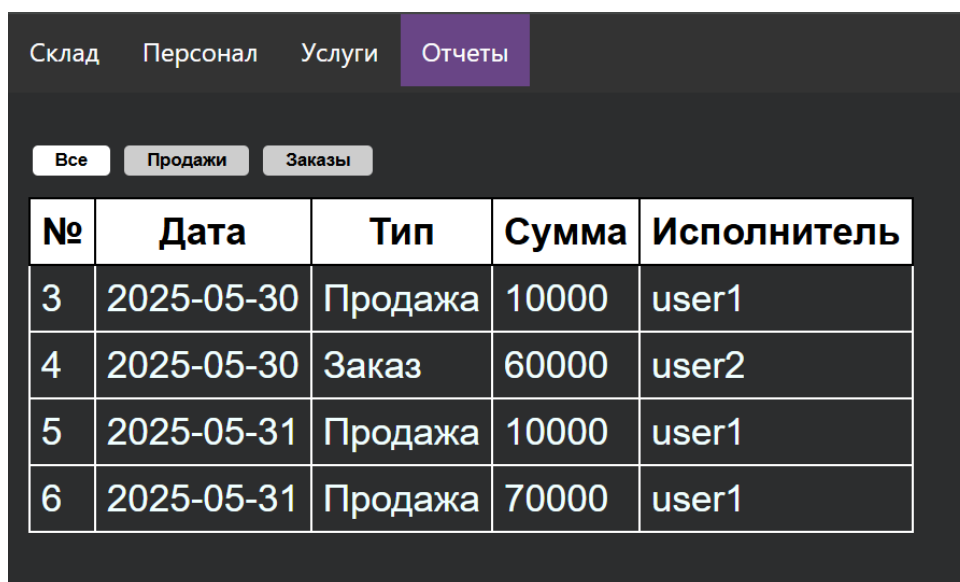
Добавить запись

Рисунок 26 – Форма добавления записи

Приступим к созданию страницы «Отчеты», суть заключается в том, чтобы записывать дату, тип операции, сумму и исполнителя. Когда пользователь без роли администратора проводит закупку или продажу под своим аккаунтом. Создаётся запись в таблице «sales», куда записывается настоящая дата проведения операции, тип и сумма операции, а исполнитель берется из сессии пользователя, которая создаётся при авторизации, так же благодаря этой сессии в шапке отображается имя пользователя.

Следовательно, добавлять новые записи от имени администратора не имеет смысла во избежание фальсификации или искажения данных, данная страница служит для мониторинга отчетности.

Для таблицы «sales», так же, как и для таблицы «services», внедрен фильтр записей по признаку типа операции. Ниже на рисунке 27 показана страница «Отчеты».



№	Дата	Тип	Сумма	Исполнитель
3	2025-05-30	Продажа	10000	user1
4	2025-05-30	Заказ	60000	user2
5	2025-05-31	Продажа	10000	user1
6	2025-05-31	Продажа	70000	user1

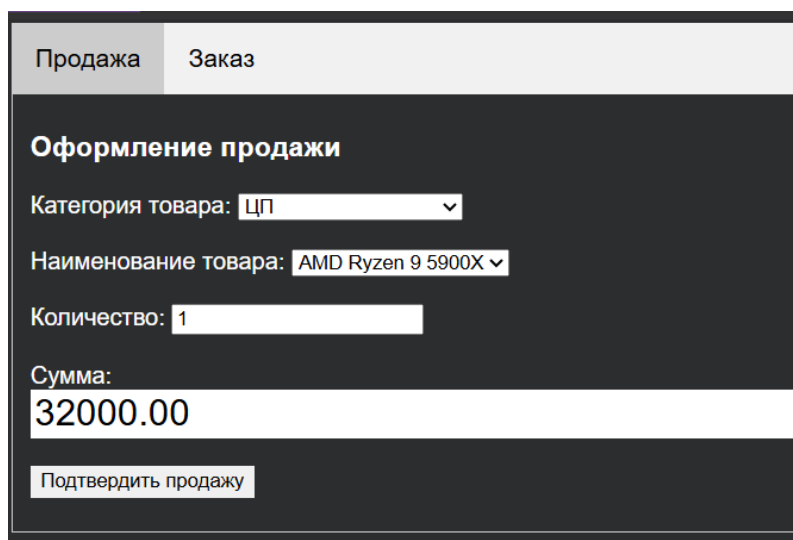
Рисунок 27 – Страница «Отчеты»

Приступим к созданию страниц, которые будут доступны не администраторам. Для этого создадим страницу для продажи или закупки товара, а также зеркало страницы с услугами.

Начнем с главной. Для удобства возможность на одной странице закупки и продажи будет реализована с помощью вкладок на странице с разными формами и скриптами для своих действий. Первая вкладка будет для продажи товаров, а вторая будет для заказа новых товаров на склад.

Здесь встаёт задача, чтобы при продаже у нас показывались только товары в наличии на складах, следовательно, необходимо чтобы в форме продажи подгружались актуальные данные с таблицы «sklad». Для подгрузки записей о товарах на продажу используется функция loadProducts, которая передает в данные в виде списка, после выбора категории.

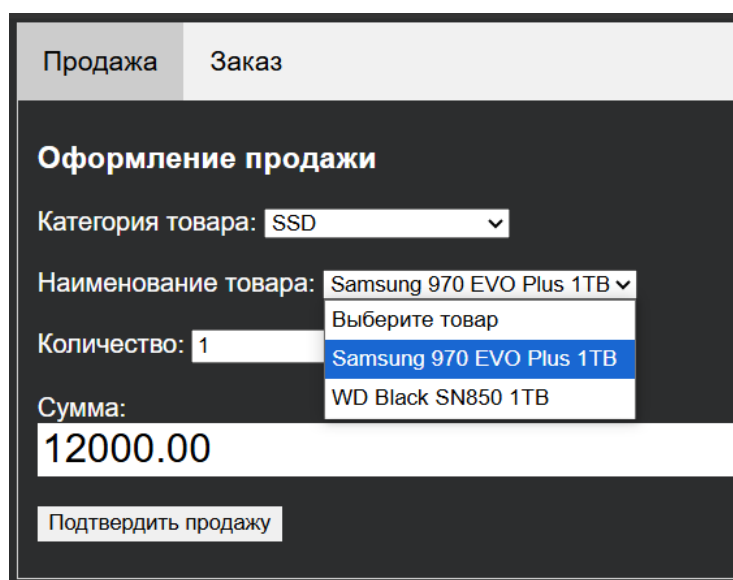
При выборе категории, будут предлагаться товары этой категории для продажи, стоимость будет сама подгружаться при выборе товара, а после проведения операции, в таблице «sklad» уменьшается количество товара в соответствии с количеством проданных. На рисунках 28-29 представлен вид формы при выборе разных категорий товара.



The screenshot shows a web form titled 'Оформление продажи' (Sale Form) with two tabs: 'Продажа' (Sale) and 'Заказ' (Order). The 'Продажа' tab is active. The form contains the following fields and values:

- Категория товара: ЦП (dropdown menu)
- Наименование товара: AMD Ryzen 9 5900X (dropdown menu)
- Количество: 1 (text input)
- Сумма: 32000.00 (text input)
- Подтвердить продажу (button)

Рисунок 28 – Форма продажи с товаром категории ЦП

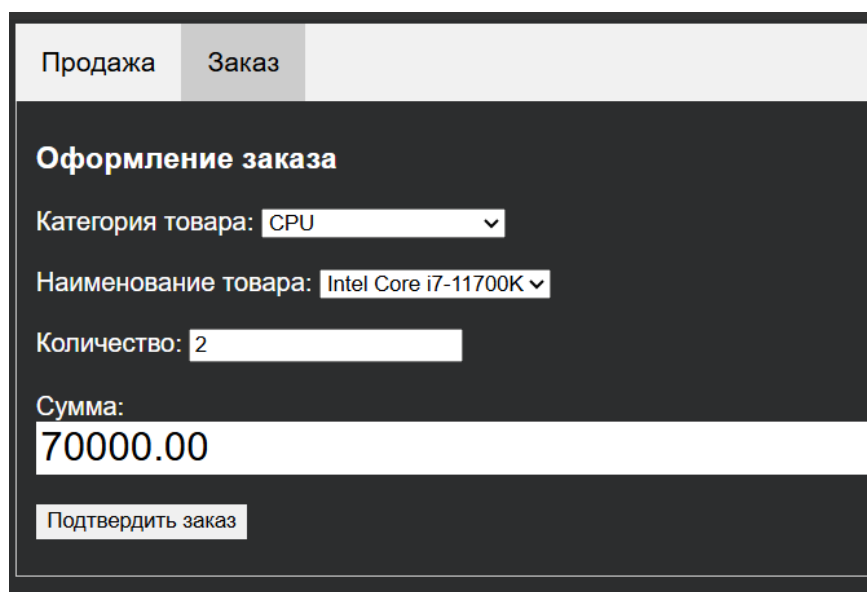


The screenshot shows the same 'Оформление продажи' form, but with the 'Наименование товара' dropdown menu open. The 'Категория товара' is now 'SSD'. The dropdown menu lists three options: 'Выберите товар' (Select item), 'Samsung 970 EVO Plus 1TB' (highlighted in blue), and 'WD Black SN850 1TB'. The other fields and the 'Подтвердить продажу' button remain the same.

Рисунок 29 – Форма продажи с товаром категории SSD

При нажатии на кнопку подтверждения продажи активируется скрипт process_sale.php, который собирает данные с формы и добавляет запись в таблицу «sales» с категорией «продажа».

Теперь перейдём к вкладке закупки новых товаров на склад. На рисунке 30 представлен вид формы заказа товаров.



Продажа **Заказ**

Оформление заказа

Категория товара: CPU ▾

Наименование товара: Intel Core i7-11700K ▾

Количество: 2

Сумма:
70000.00

Подтвердить заказ

Рисунок 30 – Форма заказа

При подтверждении заказа, вызывается скрипт `process_order.php`, который так же добавляет запись в таблице «sales» с категорией «заказ».

Так же для удобства работы были написаны скрипты, которые автоматически считают стоимость продажи или закупки в зависимости от количества товара (функция `calculateTotal`). Все функции представлены по вложению Б.

В завершении работы, проведем 2 операции – продажа и закупка и потом проверим появились ли новые записи в отчетах. На рисунке 31 представлена страница «Отчеты», где появились новые записи со свежей датой.

<div> <div>Все</div> <div>Продажи</div> <div>Заказы</div> </div>				
№	Дата	Тип	Сумма	Исполнитель
3	2025-05-30	Продажа	10000	user1
4	2025-05-30	Заказ	60000	user2
5	2025-05-31	Продажа	10000	user1
6	2025-05-31	Продажа	70000	user1
7	2025-06-11	Заказ	70000	Смирнов Дмитрий
8	2025-06-11	Продажа	18000	Смирнов Дмитрий

Рисунок 31 – Обновленная таблица отчетов

Выводы по второй главе

Во второй главе реализован важный прогресс по нескольким этапам разработки.

Во-первых, установлено все необходимое ПО, настроены инструменты разработки с обозначенными технологиями, которые составили основу для разработки и отладки ИС, что позволило вести разработку эффективно и комфортно.

Во-вторых, выполнен основной этап разработки ИС. Основываясь на выбранной архитектуре и выбранном списке технологий составлены схемы таблиц БД, утверждён и реализован весь функционал автоматизированной информационной системы для сети компьютерных магазинов в формате веб-приложения.

Итогом разработки является разработанная автоматизированная информационная система для сети компьютерный магазин в формате веб-приложения, которое отвечает ожиданиям пользователей. Созданное для информационной системы веб-приложение имеет преимущество в виде простого и понятного дизайна для пользователей, наличием ролей пользователей и функционалом для проведения продаж и закупок, а также рядом возможностей для администраторов системы в виде мониторинга и анализа информации.

Заключение

Подводя итоги данной выпускной квалификационной работы, можно сделать следующие выводы.

В современных реалиях навык использования браузера на устройствах есть у преобладающей части населения, что становится ценным инструментом для решения большого спектра задач, а также оптимизации трудового процесса. Принимая во внимание современные требования бизнеса, внедрение автоматизированных информационных систем помогает бизнесу расширяться и быть более эффективным, оптимизируя процесс работы.

В ходе выполнения данной выпускной квалификационной работы спроектирована и разработана автоматизированная информационная система для компьютерного магазина, которая позволяет пользователям взаимодействовать с данными в реальном времени.

Для достижения цели решены поставленные задачи.

В процессе работы изучены и проанализированы существующие технологии создания автоматизированной информационной системы, выбрана, спроектирована и разработана архитектура с учетом обозначенных требований к информационной системе, реализован функционал авторизации и регистрации. По мере продвижения в создании проекта мы столкнулись с новыми технологиями, преодолевали проблемы реализации и познакомились с новыми принципами разработки, так же изучили выбранные технологии и получили опыт разработки легко управляемой автоматизированной информационной системы.

Данная выпускная квалификационная работа демонстрирует реализацию проекта, который может быть использован для улучшения взаимодействия пользователей с данными и автоматизацию процессов в компьютерных магазинах. Нами приобретен навык работы с технологиями, которые широко используются в индустрии информационных, что позволит развиваться дальше, как профессионалам.

Список использованных источников

1. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов. – Санкт-Петербург : БХВ-Петербург, 2021. – 416 с.
2. HTML5BOOK: сайт. – URL: <https://html5book.ru/> (дата обращения: 03.02.2025)
3. PHP: официальный сайт. – URL: <https://www.php.net/> (дата обращения: 15.02.2025)
4. w3schools // PHP tutorial: сайт. – URL: <https://www.w3schools.com/php/default.asp> (дата обращения: 08.02.2025)
5. Аквино, К. Front-end. Клиентская разработка для профессионалов / К. Аквино. – Санкт-Петербург : Питер, 2017. – 512 с.
6. Андерсон, С. Приманка для пользователей. Создаем привлекательный сайт / С. Андерсон. – Москва : Питер, 2021. – 793 с.
7. Бакулев, В. А. Использование субд MySQL в разработке мобильных приложений / В. А. Бакулев // Современные информационные технологии и ИТ-образование. – 2019. – Т. 15. – № 1. – С. 37 – 43.
8. Безручкин В.Т., Информатика курс лекций / В.Т. Безручкин. – Москва : ИД «Форум»: ИНФРА–М. 2015 – 432 с.
9. Браун, Э. Изучаем JavaScript. Руководство по созданию современных веб-сайтов / Э. Браун. – Москва : Альфа-книга, 2017. – 368 с.
10. Бройдо, В. Л. Архитектура ЭВМ и систем: учебник для вузов / В. Л. Бройдо; научный редактор О. П. Ильина. – Санкт-Петербург : Питер, 2019. – 704 с.
11. Веллинг, Л. В. Разработка веб-приложений с помощью PHP и MySQL / Л. В. Веллинг, Л. Т. Томпсон. – Москва : Вильямс, 2020. – 768 с.
12. Вора, П. В. Шаблоны проектирования веб-приложений / П. В. Вора. – Москва : Эксмо, 2011. – 870 с.

13. Гагарина, Л. Г. Разработка и эксплуатация автоматизированных информационных систем: учебное пособие / Л. Г. Гагарина, Д. В. Киселев, Е. Л. Федотова. – Москва : ФОРУМ: ИНФРА-М, 2019. – 400 с.
14. Губин, М. С. PHP 8. Новинки языка и программы для работы с ним / М. С. Губин, Д. Котеров. – Екатеринбург : Издательские Решения, 2020. – 19 с. – ISBN 9785005138330.
15. Дакетт, Д. Основы веб-программирования с использованием HTML / Джон Дакетт. – Москва : Эксмо, 2020. – 239 с.
16. Дакетт, Д. Разработка и дизайн веб-сайтов / Джон Дакетт. – Москва : Эксмо, 2018. – 250 с.
17. Дронов, В.А. HTML5, CSS3 и web2.0. Разработка современных Web сайтов / В.А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2021. – 416 с.
18. Егармин, П.А. Базы данных: методические указания к проведению практических занятий для студентов направления 09.03.01 Информатика и вычислительная техника очной, заочной и очно-заочной форм обучения / П.А. Егармин. – Лесосибирск : СибГУ, 2017. – 21 с.
19. Емелин, Н.В. Способы оценки и анализа конкурентоспособности / Н. В. Емелин. – Москва : Форсайт, 2018. – 355 с.
20. Журавлев, А. Е. Корпоративные информационные системы. Администрирование сетевого домена. Учебное пособие для СПО / А. Е. Журавлев, А. В. Макшанов, Л. Н. Тындыкарь. – Санкт-Петербург : Лань, 2021. – 172 с.
21. Интернет-технологии в экономике: учебник под ред. Н. М. Абдиксева – Москва: ИД «Форум» : ИНФРА. – 2014 – 448 с.
22. Карпов, Александр. Создание и продвижение сайтов. Непрофессионал для Непрофессионала / Александр Карпов, Джи Ким. – Москва : 2016. – 280 с.
23. Кевин, Янк. PHP и MySQL. От новичка к профессионалу / Янк Кевин. – Москва : Эксмо, 2018. – 384 с
24. Кириченко, А. В. Web на практике. CSS, HTML, JavaScript, MySQL, PHP для fullstack-разработчиков / А. В. Кириченко, А. П. Никольский,

Е. В. Дубовик. – Санкт-Петербург : Наука и техника, 2021. – 432 с. – ISBN 978 5-94387-271-6.

25. Климов, А. JavaScript на примерах / А. Климов. – Санкт-Петербург : Эксмо, 2018. – 336 с.

26. Котеров, Д. PHP 8 / Д.Котеров, И.Симдянов. – Москва : БХВ, 2023. – 992 с. – ISBN 978-5-9775-1692-1.

27. Коэн, Исси. Полный справочник по HTML, CSS и JavaScript / Исси Коэн. – Паблишерз : Эксмо, 2017. – 246 с.

28. Кузнецов, С. Д. Система управления базами данных // Большая российская энциклопедия. Электронная версия (2017); https://bigenc.ru/technology_and_technique/text/3666497 Дата обращения: 29.11.2024.

29. Локхарт, Д. Современный PHP. Новые возможности и передовой опыт / Д.Локхарт. – Москва : ДМК Пресс, 2016. – 304 с.

30. Обучение работе в Office 2007/ Компания MS Office [Электронный ресурс]. – Режим доступа: <http://www.office.microsoft.com/ru-ru/training>

31. Пирогова, В.Ю. Информационные системы и базы данных: организация и проектирование: учебное пособие / В.Ю. Пирогова. – Санкт-Петербург : БХВ-Петербург, 2014. – 528 с.

32. Пьюривал, С. Основы разработки веб-приложений / С.Пьюривал. – Москва : Питер, 2015. – 272 с.

33. Раневский, М.А. Компьютерные технологии в дизайне / М.А. Раневский. – Москва : ИД. «Форум» : ИНФРА– М. 2024 – 304 с.

34. Робин Никсон, Learning PHP, MySQL & JavaScript: A Step-by-Step Guide to Creating Dynamic Websites (2021), 826 с.

35. Рубальский, О.Н. Информатика. Самоучитель на CD : учеб.пособие / О.Н. Рубальский.– Москва : ИД. «Форум» : ИНФРА– М. – 224 с.

36. Светлова Н.М. Информационные технологии управления проектами: учеб.пособие / Н.М. Светлова. – Москва : ИД. «Форум» : ИНФРА–М. – 2014

37. Симонова В.А. Проектирование баз данных в СУБД MySQL: учебное пособие / В.А. Симонова. – Москва : ИД. «Форум» : ИНФРА-М. 2014 – 160 с.
38. Ульман Дж. Введение в системы баз данных / Дж. Ульман, Дж. Уидом. – Москва : Лори, 2000. – 374 с
39. Шварц Б. MySQL по максимуму. 3-е издание / П.Зайцев, В. Ткаченко, Б.Шварц MySQL по максимуму.– Санкт-Петербург: Питер – 2018 – 864с.
40. Эспозито, Д. Разработка современных веб-приложений. Анализ предметных областей и технологий / Д. Эспозито – Москва : Вильямс, 2017. – 464 с. – ISBN 978-5-9908910-3-6.
41. Яхьяева, Г. Э. Основы программирования: учебник / Г. Э. Яхьяева. – я2-е изд., испр. – Москва : БИНОМ. Лаборатория знаний, 2020. – 431 с.

Приложение А

Код файлов reg. php, log.php и logout.php

Файл reg.php

```
$login = $conn->real_escape_string($_POST['login']);
$pass = $conn->real_escape_string($_POST['password']);
$hashed_pass = password_hash($pass, PASSWORD_DEFAULT);
$sql_employees = "INSERT INTO employees (Name, Position, Salary, Role) VALUES ('$login',
'Position', '0', 'пользователь')";
if ($conn->query($sql_employees) === TRUE) {
    $last_employee_id = $conn->insert_id;
    $sql_users = "INSERT INTO users (log, pass, employee_id) VALUES ('$login', '$hashed_pass',
$last_employee_id)";

    if ($conn->query($sql_users) === TRUE) {
        echo "Регистрация прошла успешно!";
        header("Location: logreg.html");
        exit();
    } else {
        echo "Ошибка при добавлении пользователя: " . $sql_users . "<br>" . $conn->error;
    }
} else {
    echo "Ошибка при добавлении сотрудника: " . $sql_employees . "<br>" . $conn->error;
} echo "Ошибка: " . $sql . "<br>" . $conn->error;
$conn->close();
```

Файл log.php

```
$login = $conn->real_escape_string($_POST['login']);
$pass = $conn->real_escape_string($_POST['password']);

$sql = "SELECT ID, log, pass, role, employee_id FROM users WHERE log = '$login'";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    $row = $result->fetch_assoc();
    if (password_verify($pass, $row['pass'])) {
        $_SESSION['user_id'] = $row['employee_id'];
        $_SESSION['user_role'] = $row['role'];

        $employee_id = $row['employee_id'];
        $sql_employee = "SELECT Name FROM employees WHERE ID = '$employee_id'";
        $result_employee = $conn->query($sql_employee);

        $sql_role = "SELECT Role FROM employees WHERE ID = ?";
        $stmt_role = $conn->prepare($sql_role);
        $stmt_role->bind_param("i", $employee_id);
        $stmt_role->execute();
```

```

$result_role = $stmt_role->get_result();

if ($result_role->num_rows == 1) {
    $row_role = $result_role->fetch_assoc();
    $role = $row_role['Role'];

    if ($result_employee->num_rows > 0) {
        $row_employee = $result_employee->fetch_assoc();
        $_SESSION['user_name'] = $row_employee['Name'];
        $_SESSION['role'] = $role;
    }
}

if ($row['role'] == ('Администратор')) {
    header("Location: staff.php");
} else {
    header("Location: buhg_usual.php");
}

exit();
} else {
    echo "Неверный пароль!";
}
} else {
    echo "Пользователь не найден!";
}
}
$conn->close();

```

Файл logout.php

```

<?php
session_start();

session_unset();
session_destroy();

header("Location: logreg.html");
exit();
?>

```

Приложение Б

Код скриптов на JavaScript

```
function openRecord(recordId) {
    window.open('record_stock.php?id=' + recordId, '_self');
}

var clicked = false;

function deleteRecord(recordId) {
    if (confirm('Вы уверены, что хотите удалить эту запись?')) {
        // Отправить запрос на сервер для удаления записи
        window.location.href = 'delete_record.php?id=' + recordId;
    }
}

function updateRecord(recordId) {
    window.location.href = 'update_record_stock.php?id=' + recordId;
}

function goBack() {
    window.history.back();
}

function openTab(evt, tabName) {
    var i, tabcontent, tablinks;
    tabcontent = document.getElementsByClassName("tabcontent");
    for (i = 0; i < tabcontent.length; i++) {
        tabcontent[i].classList.remove("active");
    }
    tablinks = document.getElementsByClassName("tablinks");
    for (i = 0; i < tablinks.length; i++) {
        tablinks[i].classList.remove("active");
    }
    document.getElementById(tabName).classList.add("active");
    evt.currentTarget.classList.add("active");
}

document.getElementById("defaultOpen").click();
```