СОДЕРЖАНИЕ

[1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 5](#_Toc199431844)

[1.1 История и предпосылки появления АИС 5](#_Toc199431845)

[1.2 Основные компоненты и структура АИС 7](#_Toc199431846)

[1.3 Анализ существующих решений для учета компьютеров и комплектующих. 9](#_Toc199431847)

[1.3 Эффективность АИС на предприятиях. 11](#_Toc199431848)

[1.4 Нормативно-правовая база, регламентирующая учет ИТ-оборудования 14](#_Toc199431849)

[1.3.4 Обеспечение безопасности и защиты АИС 17](#_Toc199431850)

[2 ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ УЧЕТА КОМПЬЮТЕРОВ И КОМПЛЕКТУЮЩИХ 20](#_Toc199431851)

[2.1 Сравнительный анализ сред разработки 22](#_Toc199431852)

[2.2 Анализ языков программирования 24](#_Toc199431853)

[2.3 Использование PHP в Backend разработке. 26](#_Toc199431854)

[2.4 Использование технологии AJAX c PHP 29](#_Toc199431855)

[2.5 Пояснительная характеристика веб-приложения 33](#_Toc199431856)

[3 ЭКОНОМИЧЕСКАЯ ЧАСТЬ 37](#_Toc199431857)

[3.1 Расчет затрат на разработку веб-приложения 37](#_Toc199431858)

[3.2 Расчет затрат на внедрение веб-приложения 38](#_Toc199431859)

[4 ОХРАНА ТРУДА ПРИ РАБОТЕ ЗА ПЕРСОНАЛЬНЫМ КОМПЬЮТЕРОМ 40](#_Toc199431860)

[ЗАКЛЮЧЕНИЕ 43](#_Toc199431864)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 45](#_Toc199431865)

[ПРИЛОЖЕНИЕ A 47](#_Toc199431866)

**ВВЕДЕНИЕ**

В современных условиях развития информационных технологий эффективное управление IT-инфраструктурой предприятия становится критически важным фактором обеспечения его конкурентоспособности. Автоматизация процессов учета и управления компьютерным парком и комплектующими позволяет существенно повысить производительность труда, оптимизировать затраты и обеспечить бесперебойную работу организации.

Актуальность разработки автоматизированной информационной системы (АИС) для учета компьютеров и комплектующих обусловлена рядом существенных факторов. В настоящее время на предприятиях часто встречаются серьезные проблемы в области учета IT-оборудования: высокая трудоемкость ведения документации вручную, вероятность ошибок при вводе данных, отсутствие оперативного доступа к необходимой информации, затруднения при проведении инвентаризации, сложность отслеживания перемещения техники и комплектующих. Все это приводит к задержкам в решении проблем с оборудованием, неэффективному использованию ресурсов и значительным финансовым потерям.

Отсутствие систематизированного подхода к учету IT-оборудования создает серьезные риски для предприятий. Невозможность оперативного получения информации о состоянии оборудования приводит к задержкам в ремонте компьютеров, что негативно сказывается на работе сотрудников и может приводить к срыву сроков выполнения задач. Кроме того, существует риск утраты оборудования и неэффективного использования имеющихся ресурсов из-за отсутствия должного контроля.

Объект исследования - процессы учета компьютеров и комплектующих на предприятии.

Предмет исследования - методика создания базы данных для учета IT-оборудования, включающая выбор модели данных, проектирование структуры таблиц и разработку SQL-запросов.

Цель - разработка автоматизированной информационной системы, обеспечивающей эффективное управление IT-инфраструктурой предприятия.

Задачи дипломного проекта:

1. Провести комплексный анализ существующих систем учета компьютеров и комплектующих с целью выявления их преимуществ и недостатков.
2. Разработать концептуальную и логическую модель базы данных для хранения информации об IT-оборудовании.
3. Выбрать оптимальные программные средства и технологии для реализации АИС.
4. Спроектировать удобный и интуитивно понятный пользовательский интерфейс системы.
5. Реализовать функциональность АИС, включая ввод, хранение, обработку данных и формирование отчетов.
6. Провести тестирование и отладку разработанной системы для обеспечения её стабильной работы.
7. Оценить экономическую эффективность внедрения разработанной АИС на предприятии.

1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 История и предпосылки появления АИС

История автоматизированных информационных систем (АИС) отражает эволюцию технологий и растущие потребности общества в эффективном управлении данными. Её истоки восходят к механизации вычислений в XIX веку, когда арифмометры и аналитическая машина Бэббиджа заложили основы автоматизации рутинных операций. Эти устройства, хотя и ограниченные в возможностях, стали прообразом будущих систем обработки информации, позволив стандартизировать учётные процессы в промышленности и торговле**.**

Начальный этап развития АИС охватывает период до середины XX века, когда были заложены фундаментальные основы автоматизации информационных процессов. Механизация вычислений началась с изобретения первых вычислительных устройств, таких как арифмометр и аналитическая машина Бэббиджа, которые существенно упростили и ускорили выполнение рутинных вычислительных операций. Развитие промышленности и торговли требовало стандартизации процессов учета и управления, что привело к появлению различных систем учета на основе карточек и книг. Однако эти системы оставались трудоемкими и были подвержены ошибкам. Значительным шагом вперед стало изобретение электромеханических счетных машин, в частности табуляторов Германа Холлерита, которые позволили автоматизировать обработку статистических данных.

Следующий этап развития АИС связан с появлением первых электронных вычислительных машин (ЭВМ) в 1940-1950-е годы. Создание таких систем, как ENIAC, EDVAC и UNIVAC, стало революционным прорывом в области обработки информации. ЭВМ значительно превосходили по производительности механические и электромеханические устройства, что позволило не только решать научные и инженерные задачи, но и приступить к автоматизации процессов учета на предприятиях. Важным достижением этого периода стало появление первых языков программирования, таких как Fortran и Cobol, которые существенно упростили разработку программного обеспечения для ЭВМ.

В 1960-1970-е годы произошло становление концепции автоматизированных систем управления (АСУ). Ключевым моментом стало появление систем управления базами данных (СУБД), таких как IMS и DB2, которые позволили организовать эффективное хранение и доступ к структурированным данным. АСУ представляли собой комплексные системы, включающие не только учетные функции, но и системы планирования, проектирования и контроля. Этот период также ознаменовался появлением мини-ЭВМ и персональных компьютеров, что сделало вычислительную технику более доступной для предприятий и организаций.

1980-1990-е годы стали периодом расцвета информационных технологий. Развитие локальных сетей (LAN) позволило объединить компьютеры в единые информационные системы, обеспечивая обмен данными и совместную работу. Значительным шагом вперед стало появление объектно-ориентированного программирования (ООП), графического пользовательского интерфейса (GUI) и CASE-технологий, которые автоматизировали процесс разработки программного обеспечения. Важным этапом развития АИС стало появление ERP-систем, интегрировавших все основные бизнес-процессы предприятия в единую информационную систему.

Современный этап развития АИС начался с 2000-х годов и характеризуется масштабной цифровизацией информационных процессов. Интернет стал основной платформой для разработки и развертывания АИС, появились веб-ориентированные системы, доступные через браузер из любой точки мира. Развитие облачных вычислений позволило существенно снизить затраты на создание и поддержку IT-инфраструктуры, обеспечив при этом масштабируемость и гибкость АИС. Интеграция с технологиями Big Data и аналитическими системами открыла новые возможности для получения ценной информации из больших объемов данных. Внедрение искусственного интеллекта и машинного обучения позволило автоматизировать рутинные задачи, осуществлять прогнозирование и оптимизировать бизнес-процессы.

Развитие АИС стало возможным благодаря ряду ключевых факторов: технологическому прогрессу в области вычислительной техники, растущей потребности в эффективной обработке больших объемов информации, развитию технологий хранения и доступа к данным, совершенствованию языков программирования и инструментов разработки ПО, развитию сетевых технологий, процессам глобализации и усилению конкуренции, а также стремлению к повышению эффективности бизнеса и оптимизации затрат.

Автоматизированные информационные системы (АИС) играют ключевую роль в повышении эффективности и конкурентоспособности современных предприятий. Они помогают автоматизировать различные бизнес-процессы, улучшить управление данными и принимать более обоснованные решения.

1.2 Основные компоненты и структура АИС

Автоматизированные информационные системы (АИС) представляют собой сложные человеко-машинные комплексы, структура которых определяется взаимодействием функциональных и обеспечивающих компонентов. Функциональная часть АИС ориентирована на решение конкретных задач управления и включает подсистемы, соответствующие ключевым направлениям деятельности организации: планирование, учет, контроль, анализ и регулирование. В промышленных АИС выделяют модули управления производством, логистикой, финансами и кадрами, каждый из которых реализует специализированные алгоритмы обработки данных. В корпоративных системах, таких как АИС ПАО «Транснефть», функциональные подсистемы интегрируются через единую информационную платформу, обеспечивая согласованность процессов на всех уровнях управления.

Обеспечивающая часть АИС формирует инфраструктуру для работы функциональных модулей и включает шесть ключевых компонентов. Информационное обеспечение представлено унифицированными системами документации, классификаторами, схемами информационных потоков и методологиями построения баз данных. Например, использование EAN-кодирования в логистических системах позволяет стандартизировать учет товародвижения. Техническое обеспечение представляет собой комплекс аппаратных средств (серверы, сети, устройства ввода-вывода) и их конфигурацию. Современные АИС часто используют гибридную архитектуру, сочетающую централизованное хранение данных и распределенную обработку на рабочих станциях.

Программное обеспечение включает системные и прикладные программы, включая СУБД, ERP-системы и аналитические инструменты. Применение облачных платформ, таких как Microsoft Azure, обеспечивает масштабируемость ресурсов. Математическое обеспечение базируется на методах оптимизации, статистического анализа и моделирования, например, использование нейросетей для прогнозирования спроса в розничных АИС. Организационное обеспечение охватывает регламенты взаимодействия пользователей с системой, включая должностные инструкции и методики обработки данных. Правовое обеспечение представляет собой нормативную базу, регулирующую электронный документооборот и защиту персональных данных.

Структура АИС строится на принципах системности (целостность компонентов), развития (адаптация к новым требованиям) и стандартизации (использование протоколов OPC UA для промышленных систем). Важную роль играет вертикальная и горизонтальная интеграция, позволяющая объединить разнородные подсистемы в единый контур. В АИС нефтебаз интеграция SCADA-систем с ERP обеспечивает сквозной учет от датчиков резервуаров до финансовой отчетности. Эффективность АИС напрямую зависит от соблюдения принципа однократности ввода данных, что достигается за счет централизованных баз данных и веб-сервисов. Современным трендом является внедрение цифровых двойников, объединяющих IoT-устройства и предиктивную аналитику для моделирования бизнес-процессов.

1.3 Анализ существующих решений для учета компьютеров и комплектующих.

На современном рынке представлен широкий спектр программных решений для учета компьютерной техники и комплектующих, каждое из которых обладает уникальными характеристиками и ориентировано на разные типы организаций. Severcart выделяется как надежная платформа для полного контроля жизненного цикла оборудования — от закупки и инвентаризации до списания и утилизации. Система автоматизирует сбор данных об аппаратной и программной конфигурации устройств через специальные агенты, что существенно упрощает мониторинг всего парка техники. Это решение особенно подходит для компаний, требующих детализированного учета и гибкой настройки отчетности, включая возможность создания пользовательских форм и отчетов, а также интеграции с корпоративными системами документооборота.

Для средних и крупных предприятий с разветвленной IT-инфраструктурой оптимальным выбором может стать Lansweeper. Этот инструмент обеспечивает глубокий анализ сетевых устройств, поддерживает управление мобильными девайсами и предоставляет расширенные возможности для формирования аналитических отчетов. Его функционал особенно ценен при работе с гетерогенными средами, где требуется централизованный контроль за разнородным оборудованием. Система позволяет проводить автоматическое обнаружение новых устройств в сети, вести учет лицензий на программное обеспечение и контролировать сроки их продления, а также предоставляет детальную статистику по использованию оборудования.

GLPI, будучи open-source решением, предлагает баланс между простотой использования и функциональностью. Система поддерживает полуавтоматический учет активов, обладает облачной редакцией и удобным интерфейсом для работы с мобильными устройствами. Однако рутинные операции в ней менее оптимизированы по сравнению с коммерческими аналогами, что делает ее предпочтительной для организаций с ограниченным бюджетом. GLPI также включает модуль управления инцидентами и запросами пользователей, систему управления конфигурациями и инструменты для планирования закупок оборудования.

Snipe-IT ориентирован на начинающих специалистов благодаря интуитивному веб-интерфейсу и упрощенному управлению лицензиями, расходниками и комплектующими. Система подходит для малого бизнеса, где не требуется детальный контроль за техническими характеристиками оборудования, но критически важен учет базовых активов. Snipe-IT позволяет создавать пользовательские поля для хранения дополнительной информации об активах, вести историю перемещений оборудования между подразделениями и формировать различные типы отчетов по остаточной стоимости активов.

SpiceWorks представляет собой комплексное решение для небольших предприятий, объединяющее инструменты мониторинга сети, управления ИТ-активами и клиентский портал. Его ключевое преимущество — интеграция функций учета картриджей, мониторов и других ресурсов в единую платформу, что снижает затраты на внедрение отдельных систем. Система также включает инструменты для управления проектами, ведения базы знаний и автоматизации рутинных задач технической поддержки.

При выборе системы необходимо учитывать не только текущие потребности, но и перспективы масштабирования. Для малых организаций оптимальны SpiceWorks или Snipe-IT благодаря низкому порогу входа и минимальным требованиям к инфраструктуре. Средние компании могут выбрать GLPI или Severcart для баланса стоимости и функциональности, тогда как крупные предприятия с сложными процессами оценят возможности Lansweeper.

Современные тенденции в развитии таких систем включают внедрение искусственного интеллекта для прогнозирования износа оборудования и оптимизации закупок, усиление интеграции с облачными сервисами для обеспечения удаленного доступа к данным, улучшение механизмов кибербезопасности с использованием современных криптографических методов защиты информации. Также активно развивается направление автоматизации процессов инвентаризации через интеграцию с системами компьютерного зрения и RFID-метками, что позволяет существенно сократить время проведения инвентаризации и минимизировать человеческий фактор при учете активов.

Важным направлением развития является интеграция с системами управления жизненным циклом оборудования (PLM), что позволяет создавать сквозные процессы от проектирования до утилизации IT-активов. Многие современные решения включают модули для анализа эффективности использования оборудования, позволяющие оптимизировать расходы на содержание IT-инфраструктуры и своевременно планировать обновление парка техники.

Особое внимание уделяется развитию мобильных приложений для учета активов, что позволяет сотрудникам проводить инвентаризацию и вносить данные непосредственно с места нахождения оборудования. Системы также расширяют возможности самообслуживания пользователей через веб-порталы, где сотрудники могут подавать заявки на обслуживание, отслеживать статус своих запросов и получать доступ к базе знаний по работе с оборудованием.

В условиях растущей цифровизации бизнеса все большее значение приобретает интеграция систем учета с платформами бизнес-аналитики, что позволяет получать детальную аналитику по использованию IT-активов, оценивать эффективность инвестиций в IT-инфраструктуру и принимать обоснованные управленческие решения.

1.3 Эффективность АИС на предприятиях.

Эффективность автоматизированных информационных систем (АИС) на предприятиях, особенно в контексте учета компьютеров и комплектующих, проявляется в комплексной оптимизации управленческих процессов. Внедрение таких систем позволяет перевести рутинные операции, такие как инвентаризация, списание оборудования или отслеживание перемещений, в цифровой формат, что минимизирует человеческие ошибки и существенно сокращает временные затраты. Централизованная база данных обеспечивает мгновенный доступ к информации о технических характеристиках устройств, их текущем состоянии, гарантийных сроках и истории обслуживания, что критически важно для планирования ремонтов и обновления парка техники.

Ключевым показателем эффективности АИС становится снижение операционных издержек. Автоматизация учета комплектующих исключает дублирование данных, сокращает потери из-за пересортицы и оптимизирует складские запасы. Так, в кейсе компании N внедрение системы WMS Manhattan позволило на 30% ускорить обработку заказов и на 25% снизить ошибки при инвентаризации за счет интеграции с устройствами сканирования штрихкодов. Аналогичные результаты демонстрируют предприятия, где автоматизация документооборота сократила время подготовки отчетов с нескольких часов до минут, высвобождая ресурсы для стратегических задач.

Дополнительным фактором эффективности становится возможность масштабирования системы под растущие потребности предприятия. Модульная архитектура современных АИС позволяет поэтапно внедрять новые функции, начиная с базового учета и заканчивая продвинутым анализом данных. Это особенно важно для компаний, планирующих расширение бизнеса или модернизацию производственных процессов.

Важным аспектом эффективности является повышение качества принимаемых управленческих решений. АИС предоставляют руководству инструменты для анализа больших данных, прогнозирования потребностей в комплектующих и оптимизации закупок. Например, использование предиктивной аналитики позволяет предвидеть выход оборудования из строя и планировать профилактические работы, что снижает простои и увеличивает срок службы техники.

Внедрение АИС также способствует улучшению контроля над активами предприятия. Система позволяет отслеживать перемещение каждого устройства, контролировать сроки обслуживания и ремонта, а также анализировать эффективность использования оборудования. Это особенно актуально для крупных компаний с распределенной структурой, где необходимо централизованно управлять ИТ-инфраструктурой.

Современные АИС интегрируются с другими корпоративными системами, что создает единую информационную среду предприятия. Например, интеграция с системами бухгалтерского учета позволяет автоматизировать процесс списания оборудования, а связь с CRM-системами обеспечивает более эффективное взаимодействие с поставщиками комплектующих.

Значительным преимуществом является повышение уровня информационной безопасности. АИС обеспечивают разграничение прав доступа к данным, ведение журналов аудита и защиту конфиденциальной информации. Это особенно важно при работе с дорогостоящим оборудованием и стратегически важными компонентами.

Экономический эффект от внедрения АИС проявляется не только в снижении прямых затрат, но и в увеличении производительности труда сотрудников. Работники освобождаются от рутинных операций и могут сосредоточиться на более сложных задачах, требующих творческого подхода.

Современные АИС также способствуют оптимизации процессов закупок и управления поставщиками. Система позволяет анализировать историю поставок, оценивать надежность поставщиков и оптимизировать условия контрактов. Это приводит к снижению закупочных цен и улучшению качества поставляемых комплектующих.

Важным аспектом является возможность проведения комплексного анализа эффективности использования ИТ-активов. АИС предоставляют инструменты для оценки окупаемости инвестиций в оборудование, анализа загрузки технических средств и оптимизации их распределения между подразделениями.

Внедрение АИС также способствует улучшению качества обслуживания конечных пользователей. Система позволяет оперативно реагировать на заявки, отслеживать сроки выполнения работ и оценивать удовлетворенность пользователей предоставляемыми услугами.

Современные АИС обладают развитыми возможностями визуализации данных, что позволяет руководству получать наглядную картину состояния ИТ-инфраструктуры в режиме реального времени. Это способствует более быстрому принятию решений и повышению эффективности управления.

Таким образом, эффективность АИС на предприятиях проявляется в комплексной оптимизации всех аспектов управления ИТ-активами: от базового учета до стратегического планирования. Внедрение таких систем позволяет существенно повысить качество управления, снизить операционные издержки и обеспечить устойчивое развитие бизнеса в условиях цифровой трансформации.

1.4 Нормативно-правовая база, регламентирующая учет ИТ-оборудования

Настоящий раздел дипломного проекта посвящен критическому анализу нормативно-правовых актов, регулирующих учет ИТ-оборудования, и доказывает необходимость их безусловного учета при разработке автоматизированной информационной системы (АИС). В рамках данного исследования обосновывается, что игнорирование законодательных требований не просто снижает эффективность системы, но и создает серьезные риски для всего предприятия, включая возможность наложения административных штрафов и потери деловой репутации.

Успешность АИС для учета ИТ-оборудования определяется не только удобством интерфейса и автоматизацией процессов, но и её способностью обеспечивать полное соответствие требованиям бухгалтерского и налогового законодательства. Создание АИС для учета ИТ-оборудования – это не только вопрос автоматизации, но и, в первую очередь, вопрос обеспечения соответствия учетных процессов требованиям, установленным государством. Игнорирование нормативно-правовой базы приведет к тому, что АИС не сможет выполнять свою основную функцию – обеспечивать достоверное и своевременное отражение информации об активах предприятия в бухгалтерском и налоговом учете.

Квалификация ИТ-оборудования как актива (основного средства) требует от АИС наличия механизмов, позволяющих автоматически определять, соответствует ли конкретный объект критериям признания его основным средством согласно ПБУ 6/01, учитывая срок полезного использования и стоимость. Это критически важно для правильного отражения активов на балансе предприятия и формирования достоверной финансовой отчетности.

Определение стоимости ИТ-оборудования должно осуществляться с учетом всех затрат, связанных с его приобретением, доставкой, установкой и настройкой, в соответствии с требованиями ПБУ 6/01. Это необходимо для правильного расчета амортизации и налога на имущество, что напрямую влияет на финансовые показатели организации.

Амортизация ИТ-оборудования требует от АИС поддержки различных методов начисления амортизации, предусмотренных законодательством, и обеспечения автоматического расчета суммы амортизации за период, а также формирования необходимых бухгалтерских проводок. Это ключевой аспект для правильного расчета прибыли и налога на прибыль, что особенно важно в условиях рыночной конкуренции и необходимости оптимизации налоговых платежей.

Инвентаризация ИТ-оборудования должна поддерживаться АИС через автоматизацию формирования инвентаризационных описей, ввод результатов инвентаризации, выявление расхождений и формирование отчетов о недостачах и излишках. Это необходимо для обеспечения достоверности учетных данных и выявления случаев хищения или неправомерного использования имущества, что напрямую влияет на сохранность активов предприятия.

Документальное оформление операций с ИТ-оборудованием требует от АИС обеспечения автоматического формирования первичных учетных документов (актов приема-передачи, накладных, актов списания и т.д.) в соответствии с установленными форматами и требованиями законодательства. Это необходимо для подтверждения правомерности операций с ИТ-оборудованием и защиты интересов предприятия в случае споров с контролирующими органами.

Федеральный закон № 402-ФЗ «О бухгалтерском учете» определяет общие требования к ведению бухгалтерского учета и формированию бухгалтерской отчетности. АИС должна обеспечивать полноту, достоверность и своевременность информации об ИТ-оборудовании для целей бухгалтерского учета, что является основой для принятия управленческих решений и контроля за эффективностью использования активов.

ПБУ 6/01 «Учет основных средств» устанавливает правила учета ИТ-оборудования как основных средств. АИС должна обеспечивать автоматизацию процессов,

логообложения, связанные с ИТ-оборудованием (налог на имущество, налог на прибыль). АИС должна обеспечивать учет амортизации для целей налогообложения, а также формирование информации для расчета налога на имущество, что позволяет организации оптимизировать налоговые платежи и избежасвязанных с классификацией, оценкой, амортизацией и выбытием основных средств, что позволяет оптимизировать учетные процессы и снизить вероятность ошибок.

Приказ Минфина № 91н «Об утверждении Методических указаний по бухгалтерскому учету основных средств» детализирует положения ПБУ 6/01. АИС должна обеспечивать учет специфики различных видов ИТ-оборудования, а также автоматическое формирование необходимых первичных учетных документов, что упрощает документооборот и повышает его качество.

Общероссийский классификатор основных фондов (ОКОФ) классифицирует основные фонды по видам. АИС должна обеспечивать классификацию ИТ-оборудования в соответствии с ОКОФ для правильного определения срока полезного использования и начисления амортизации, что влияет на корректность налоговой отчетности и финансовое планирование. **А к**

Налоговый кодекс РФ устанавливает правила нать штрафов.

Основываясь на анализе нормативно-правовой базы, можно определить минимальный набор функций, которыми должна обладать АИС для учета ИТ-оборудования. Ведение подробных карточек учета объектов должно включать исчерпывающую информацию (инвентарный номер, серийный номер, дата приобретения, первоначальная стоимость, срок полезного использования, метод амортизации, местонахождение, ответственное лицо и т.д.), что обеспечивает полный контроль за движением активов.

Автоматизированный расчет амортизации должен осуществляться с учетом различных методов (линейный, уменьшаемого остатка, по сумме чисел лет полезного использования, пропорционально объему продукции), что позволяет гибко настраивать учет в соответствии с учетной политикой организации.

Формирование унифицированных первичных учетных документов должно происходить в соответствии с требованиями законодательства (акты приема-передачи ОС-1, акты списания ОС-4 и т.д.), что обеспечивает юридическую чистоту документооборота и защиту интересов организации.

1.3.4 Обеспечение безопасности и защиты АИС

В современных условиях разработка и внедрение автоматизированных информационных систем (АИС) неразрывно связаны с необходимостью обеспечения высокого уровня информационной безопасности. Особую актуальность данная проблема приобретает в связи с тем, что АИС содержат конфиденциальные данные об ИТ-активах предприятия, включающие сведения об их стоимости, местоположении, технических характеристиках, установленном программном обеспечении и пользовательской информации.

Несанкционированный доступ к подобным данным может повлечь за собой серьезные последствия: финансовые потери, репутационный ущерб и юридические риски. В связи с этим обеспечение защиты информации должно рассматриваться как критически важный аспект на всех этапах жизненного цикла системы, начиная с проектирования и заканчивая эксплуатацией.

Первоочередной задачей при построении системы защиты является проведение комплексного анализа потенциальных угроз и рисков. Спектр возможных атак варьируется от простейших форм несанкционированного доступа к учетным записям до сложных целенаправленных кибератак, направленных на кражу данных или дестабилизацию работы системы.

В качестве фундаментальной составляющей защиты выступает надежная система аутентификации, предусматривающая многоуровневую верификацию личности пользователя. Особое значение приобретает внедрение многофакторной аутентификации, сочетающей пароли с дополнительными методами подтверждения (одноразовые коды, биометрические данные). Не менее важным аспектом является строгая политика паролей, регламентирующая использование сложных и уникальных комбинаций.

Механизм авторизации, определяющий права доступа пользователей к функциональным возможностям и данным системы, базируется на принципе минимальных привилегий. Данный подход предполагает предоставление доступа исключительно к тем ресурсам, которые необходимы для выполнения должностных обязанностей, что минимизирует потенциальный ущерб при компрометации учетных данных.

Защита данных на уровне хранения и передачи обеспечивается применением современных криптографических методов. Конфиденциальная информация в базах данных подлежит обязательному шифрованию, а передача данных по сетевым каналам осуществляется посредством защищенных протоколов (HTTPS).

Важным элементом системы безопасности является реализация механизмов аудита и мониторинга. Ведение журналов событий позволяет фиксировать все значимые действия пользователей, включая попытки аутентификации, модификацию данных и доступ к файловым ресурсам. Внедрение систем обнаружения и предотвращения вторжений (IDS/IPS) обеспечивает автоматизированный контроль за попытками несанкционированного доступа.

Физическая безопасность оборудования достигается за счет размещения серверов и критически важного оборудования в специально оборудованных помещениях с ограниченным доступом, оснащенных современными системами видеонаблюдения и контроля доступа.

Ключевым аспектом обеспечения непрерывности функционирования системы является организация регулярного резервного копирования данных. Процедура включает создание копий базы данных, конфигурационных файлов и системного программного обеспечения. Наличие детально проработанного и протестированного плана восстановления данных гарантирует возможность оперативного восстановления работоспособности системы при возникновении нештатных ситуаций.

Эффективность системы информационной безопасности в значительной степени определяется уровнем подготовки персонала. Регулярное обучение сотрудников основам информационной безопасности, методам противодействия фишинговым атакам и другим видам угроз позволяет поддерживать необходимый уровень защищенности.

Систематический аудит безопасности представляет собой неотъемлемый элемент управления информационной безопасностью. Процедуры аудита включают анализ журналов событий, сканирование уязвимостей и тестирование на проникновение, что позволяет оценивать эффективность реализованных мер защиты и своевременно выявлять потенциальные риски.

Таким образом, обеспечение информационной безопасности АИС требует внедрения комплексного подхода, включающего аутентификацию, авторизацию, шифрование данных, мониторинг событий, резервное копирование, физическую защиту, обучение персонала и регулярный аудит безопасности. Только при соблюдении данного набора мер возможно гарантировать надежную защиту конфиденциальной информации и бесперебойную работу системы в условиях современных информационных угроз.

2 ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ УЧЕТА КОМПЬЮТЕРОВ И КОМПЛЕКТУЮЩИХ

**2.1 Что такое веб-приложение**

Веб-приложение – это программа, которая работает не на вашем компьютере, а на специальном сервере “в облаке” (то есть, просто в другом компьютере, который всегда включен и подключен к интернету). Доступ можно получить к этой программе через обычный веб-браузер – Chrome, Firefox, Safari, Edge. Это как смотреть видео на видеохостингах: видео хранится не на вашем компьютере, а на серверах видеохостинга, и оно проигрывается через браузер.

Данная дипломная работа была создана в виде веб-приложения, по следующим причинам.

В первую очередь, это доступность и универсальность. Все, что необходимо для работы с веб-приложением, – это подключение к интернету и веб-браузер. Отсутствие необходимости скачивания и установки делает их чрезвычайно удобными, позволяя получить доступ к нужным функциям с любого устройства – от настольного компьютера до смартфона. Эта кроссплатформенность обеспечивает бесшовный опыт работы независимо от операционной системы, что является огромным плюсом в условиях постоянно растущей мобильности.

Еще одним ключевым преимуществом является простота обновлений и централизованное управление. Все обновления происходят на стороне сервера, что избавляет пользователя от необходимости загружать и устанавливать новые версии программы. Эта функция не только экономит время, но и гарантирует, что пользователь всегда будет работать с самой актуальной версией приложения. Централизованное управление также облегчает поддержку и обслуживание, позволяя разработчикам оперативно реагировать на ошибки и внедрять новые функции.

Масштабируемость и возможности для совместной работы – еще два важных аспекта, которые делают веб-приложения привлекательным выбором для бизнеса и командных проектов. Облачная природа позволяет легко увеличивать ресурсы, выделенные для работы с приложением, в зависимости от нагрузки, что особенно важно для быстрорастущих компаний. Возможность предоставления совместного доступа к данным и функциям, – отличительная черта многих веб-приложений, – существенно упрощает командную работу, позволяя нескольким пользователям работать над одним проектом одновременно.

Экономичность также является существенным фактором. Зачастую, для работы с веб-приложением не требуется приобретение дорогостоящего программного обеспечения или мощного оборудования. Это особенно важно для малого и среднего бизнеса, где оптимизация расходов имеет первостепенное значение. Кроме того, простота поддержки и обслуживания также способствует снижению затрат.

Нельзя не отметить гибкость и адаптивность веб-приложений. Процесс разработки веб-приложений, как правило, более быстрый и гибкий, чем разработка традиционного программного обеспечения. Разработчики могут быстро реагировать на изменяющиеся потребности пользователей, внедряя новые функции и адаптируя приложение к новым условиям. Это позволяет веб-приложениям оставаться актуальными и соответствовать постоянно меняющимся требованиям рынка.

Веб-приложения представляют собой революционный подход к доступу и обработке информации. Их удобство, доступность, простота обновления, масштабируемость и экономичность делают их идеальным инструментом для решения самых разных задач, от личных до корпоративных.

2.1 Сравнительный анализ сред разработки

Выбор среды разработки (IDE) – один из важнейших шагов для любого программиста. IDE обеспечивает инструменты, необходимые для эффективной разработки, от написания и отладки кода до управления проектами и интеграции с системами контроля версий. Этот сравнительный анализ рассматривает наиболее популярные IDE, выделяя их сильные стороны, недостатки и области.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Инструмент | Платформы | Преимущества | Кому подходит | Поддерживаемые языки |
| **Visual Studio** | Windows, macOS (с огр.) | Мощный инструмент для .NET, развитая среда для C++, обширная поддержка, интеграция с системами контроля версий. | Разработчикам .NET, C++, Windows-приложений, Unity, профессиональным разработчикам, большим проектам | C#, C++, Visual Basic, JavaScript, TypeScript, Python и многие другие через расширения |
| Visual Studio Code | Windows, macOS, Linux | Богатые функциональные возможности, интеграция с Git, широкий выбор расширений, поддержка отладчика. | Веб-разработчикам, которым нужны мощные инструменты. | JavaScript, TypeScript, Python, Java, C++, PHP, HTML, CSS и др. |
| WebStorm | Windows, macOS, Linux | Интеграция с популярными веб-фреймворками, мощные инструменты для отладки и тестирования. | Профессиональным веб-разработчикам и фронтендерам. | JavaScript, TypeScript, HTML, CSS, Node.js, Angular, React и др. |
| IntelliJ IDEA | Windows, macOS, Linux | Интеллектуальные функции автозаполнения, мощные инструменты для рефакторинга. | Разработчикам на Java и других языках, работающим с большими проектами. | Java, Kotlin, Scala, Groovy, JavaScript, TypeScript, Python, Ruby, |

Таблица 1 – Анализ средств разработки

WebStorm – это специализированная IDE, созданная для эффективной работы с JavaScript, TypeScript, HTML, CSS и популярными веб-фреймворками, такими как React, Angular и Vue.js. Ее сильные стороны включают в себя умный редактор кода, передовую поддержку веб-технологий, отличную систему отладки и полную интеграцию с инструментами разработки, такими как Node.js, Git и системами сборки. Это идеальный выбор для профессиональных веб-разработчиков, стремящихся к максимальной эффективности в своей работе

IntelliJ IDEA – это мощная IDE, известная своей превосходной поддержкой Java, Kotlin, Scala и других JVM-языков. Она также предлагает отличные возможности для веб-разработки и создания приложений для Android. Её ключевые преимущества включают в себя умный редактор кода, отличную кроссплатформенную поддержку, интуитивно понятный интерфейс, множество плагинов и систему Smart Completion. IntelliJ IDEA – это выбор профессионалов, ищущих надежный и функциональный инструмент для разработки на Java, а также создания веб-приложений и приложений для Android.

Visual Studio Code – это легковесный, но мощный редактор кода, который завоевал популярность своей гибкостью и способностью поддерживать широкий спектр языков программирования. Он выделяется своей простотой использования, огромным количеством доступных расширений, бесплатностью и встроенной поддержкой Git. VS Code – это отличный выбор для разработчиков, работающих на разных языках, для быстрого редактирования кода, а также для тех, кто ищет легкий и настраиваемый инструмент для веб-разработки.

Visual Studio – это мощная IDE, разработанная Microsoft, предназначенная в первую очередь для разработки .NET-приложений и C++ проектов. Она обладает мощным редактором кода, продвинутой системой отладки и полной интеграцией с инструментами Microsoft. Visual Studio – это выбор профессионалов, работающих с .NET-платформой, а также разработчиков C++.

Для реализации данного проекта было решено использовать Visual Studio Code. Его гибкость, кроссплатформенность и поддержка огромного количества расширений делают его мощным и универсальным инструментом. Возможность кастомизации под нужды пользователя, быстрая работа и отличная поддержка со стороны сообщества стали решающими факторами при принятии решения.

2.2 Анализ языков программирования

Для создания качественных веб-приложений разработчики используют различные языки программирования, каждый из которых имеет свои особенности, преимущества и недостатки.

JavaScript занимает лидирующие позиции в сфере фронтенд-разработки и благодаря платформе Node.js стал полноценным языком для backend-разработки. Его ключевые преимущества включают кроссбраузерную совместимость и широкую поддержку со стороны всех современных браузеров. Большое и активное сообщество разработчиков постоянно создает новые библиотеки и фреймворки, что значительно ускоряет процесс разработки. Язык позволяет создавать динамичные и интерактивные веб-приложения, а возможность использования единого языка для фронтенда и бэкенда с помощью Node.js делает его особенно привлекательным для разработчиков.

Однако у JavaScript есть и определенные недостатки. До появления TypeScript управление большими проектами могло вызывать сложности. Ошибки в коде могут приводить к неожиданному поведению на стороне клиента, что требует тщательного тестирования. Кроме того, выполнение кода на стороне клиента создает потенциальные риски безопасности, которые необходимо учитывать при разработке.

PHP остается одним из самых популярных языков для веб-разработки, несмотря на свой солидный возраст. Его главные преимущества заключаются в простоте освоения и использования, широкой поддержке большинством хостингов, обширной документации и большом сообществе разработчиков. Существует множество готовых решений и фреймворков, что позволяет быстро запускать новые проекты.

Тем не менее, PHP имеет и определенные ограничения. Исторически сложившаяся структура кода не всегда соответствует современным стандартам элегантности. В некоторых случаях производительность может уступать другим языкам, а масштабирование крупных проектов может представлять определенные сложности.

Python зарекомендовал себя как популярный язык общего назначения с широким применением в веб-разработке. Его главные достоинства включают читабельный и простой синтаксис, мощные фреймворки Django и Flask, активное использование в области науки о данных и машинного обучения. Обширное сообщество разработчиков обеспечивает постоянную поддержку и развитие языка.

Несмотря на множество преимуществ, Python имеет некоторые ограничения. Кривая обучения может быть более сложной по сравнению с PHP, а производительность иногда уступает другим языкам, что необходимо учитывать при выборе технологии для конкретного проекта.

Ruby, особенно в сочетании с фреймворком Ruby on Rails, ориентирован на повышение продуктивности разработчиков. Его элегантный синтаксис и подход convention over configuration позволяют быстро создавать прототипы и разрабатывать веб-приложения. Высокая продуктивность разработчиков является одним из ключевых преимуществ этого языка.

Однако Ruby имеет и определенные ограничения. Сообщество разработчиков меньше, чем у PHP, Python или JavaScript, что может сказаться на доступности ресурсов и поддержки. Масштабирование крупных проектов на Ruby может представлять определенные сложности.

Java остается мощным и надежным языком для веб-разработки. Его ключевые преимущества включают высокую производительность, отличную масштабируемость и строгую типизацию, что обеспечивает безопасность приложений. Множество фреймворков, таких как Spring и Spring Boot, предоставляют разработчикам широкие возможности для создания сложных веб-приложений.

Тем не менее, Java имеет и определенные недостатки. Сложный синтаксис и высокая кривая обучения могут отпугивать начинающих разработчиков. Разработка на Java требует больше усилий по сравнению с некоторыми другими языками, что может увеличивать сроки и стоимость проектов.

Архитектура данного дипломного проекта построена на использовании нескольких ключевых технологий. Для реализации серверной логики выбран PHP, что обусловлено его простотой, широкими возможностями и отличной интеграцией с различными базами данных. HTML и CSS используются для структурирования и оформления пользовательского интерфейса, обеспечивая его корректное отображение в различных браузерах. JavaScript применяется для реализации динамических элементов и улучшения пользовательского опыта, что позволяет создавать интерактивные веб-приложения с современной функциональностью.

2.3 Использование PHP в Backend разработке.

Для разработки дипломного проекта был выбран PHP в качестве серверного языка программирования. Основной особенностью реализации стал процедурный подход без применения объектно-ориентированного программирования (ООП). Такой выбор был обусловлен несколькими важными факторами. Во-первых, процедурный подход обеспечивает простоту и скорость разработки, что особенно важно для небольшого проекта. Во-вторых, минимальные требования к инфраструктуре позволяют работать с чистым PHP и MySQL без необходимости установки дополнительных компонентов. В-третьих, прямолинейная логика проекта делает сложные архитектурные паттерны избыточными, что упрощает разработку и поддержку кода. Простота и скорость разработки для небольшого проекта.

PHP был выбран в качестве основного языка для backend-разработки по нескольким ключевым причинам, которые идеально соответствовали требованиям дипломной работы. Низкий порог входа позволяет быстро писать рабочий код даже разработчикам с минимальным опытом. Отсутствие сложной настройки является значительным преимуществом: в отличие от Node.js, который требует установки npm, или Python с его зависимостями через pip, PHP работает “из коробки” с Apache и MySQL. Быстрое прототипирование обеспечивается возможностью непосредственного встраивания логики в HTML-код с помощью PHP-тегов, что особенно удобно для небольших проектов.

В проекте реализовано безопасное подключение к базе данных с помощью PDO (PHP Data Objects), что обеспечивает надежную работу с данными.

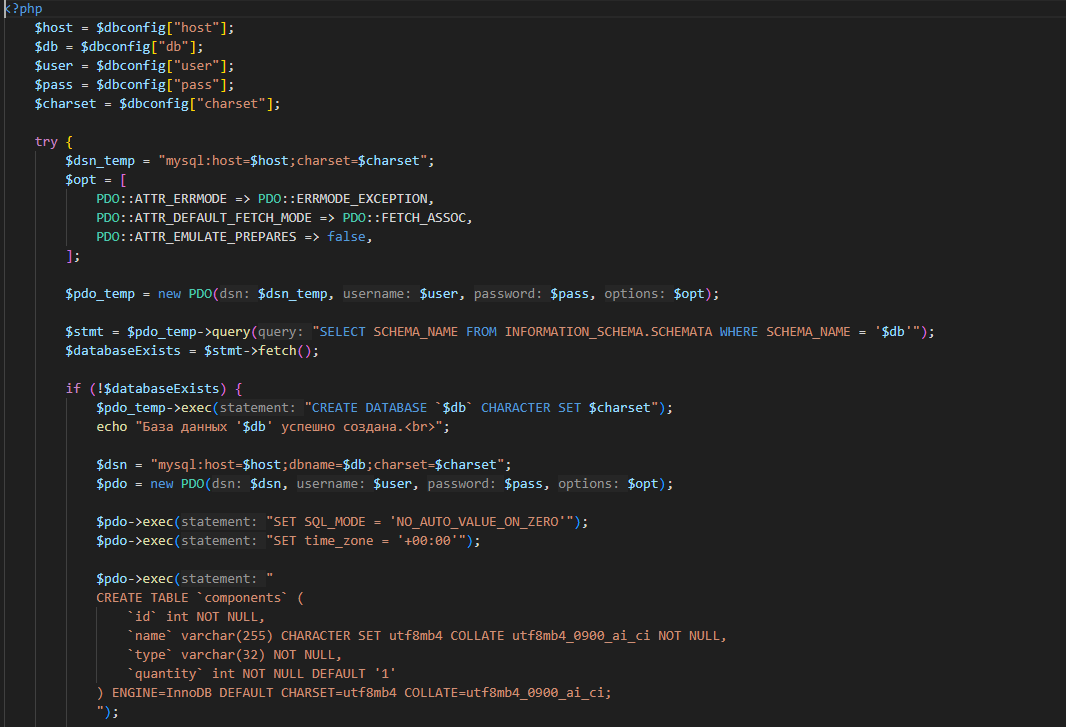


Рисунок 1 – Подключение к PDO

Особое внимание уделено вопросам безопасности кода. В частности, для защиты от XSS-атак (Cross-Site Scripting) используется функция htmlspecialchars.

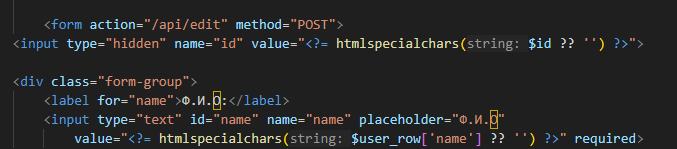


Рисунок 2 – Форма ввода пользователя.

Функция преобразует специальные символы HTML в их безопасные эквиваленты, предотвращая выполнение вредоносного кода.

Для безопасной работы с SQL-запросами применяется метод prepare.

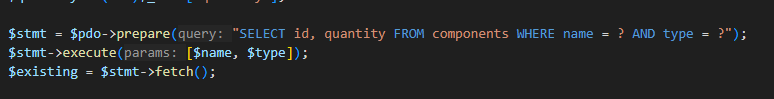


Рисунок 3 – Код выполняющий SQL запрос с prepare().

Функция prepare создает подготовленное выражение (prepared statement). Этот подход отделяет SQL-код от данных, что делает запросы к базе данных более безопасными и эффективными, защищая приложение от SQL-инъекций..

В проекте был реализован простой, но эффективный механизм роутинга через ассоциативный массив маршрутов. Этот механизм работает следующим образом: ключи массива представляют собой URL-пути (например, “/auth”), а значения указывают на файлы, которые должны быть подключены при совпадении пути.



Рисунок 4 – Массив роутера.

Это обычный ассоциативный массив, где:

Ключи — URL-пути (например, "/auth").

Значения — файлы, которые нужно подключить при совпадении пути.

Роутер проверяет текущий URL и подключает соответствующий файл:

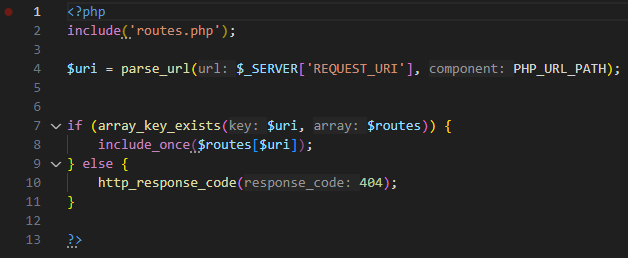


Рисунок 5 – Код роутера.

Как он работает по шагам:

1. Пользователь запрашивает URL

Например: https://site.com/auth

1. Сервер получает путь

parse\_url() извлекает /auth из полного URL.

1. Роутер ищет совпадение в массиве

Проверяет, есть ли ключ "/auth" в $routes.

1. Подключение файла
2. Если путь найден, подключается app/views/auth.php.

Такой подход к организации роутинга обеспечивает простую и эффективную маршрутизацию запросов в рамках проекта, не требуя использования сложных фреймворков или дополнительных библиотек. Это решение полностью соответствует выбранному процедурному подходу и позволяет гибко управлять запросами пользователей к различным разделам приложения.

2.4 Использование технологии AJAX c PHP

AJAX (Asynchronous JavaScript and XML) представляет собой не язык программирования, а передовую методологию, позволяющую веб-страницам обновлять содержимое без необходимости перезагрузки всей страницы. В сочетании с PHP, одним из наиболее популярных языков серверной разработки, AJAX открывает широкие возможности для создания динамичных и интерактивных веб-приложений.

Суть технологии AJAX заключается в асинхронном обмене данными между клиентом (браузером) и сервером. Вместо полной перезагрузки страницы для обновления информации, AJAX позволяет отправлять запросы на сервер в фоновом режиме, получать ответы и обновлять только те части страницы, которые требуют изменения. Такой подход обеспечивает более плавный и отзывчивый пользовательский интерфейс, что значительно повышает удовлетворенность конечных пользователей.

PHP, как один из наиболее распространенных языков серверной разработки, демонстрирует отличную совместимость с AJAX. Он предоставляет мощные инструменты для обработки запросов, работы с базами данных и генерации динамического контента. PHP-скрипты на сервере получают входящие запросы, обрабатывают их, выполняют необходимые операции (например, извлекают данные из базы данных или производят вычисления) и формируют ответ, который обычно передается в формате JSON (JavaScript Object Notation) или XML (Extensible Markup Language). Этот ответ затем обрабатывается JavaScript-кодом на стороне клиента, который обновляет соответствующие элементы веб-страницы.

Тандем AJAX и PHP обладает рядом существенных преимуществ. Во-первых, обеспечивается высокая динамичность и интерактивность интерфейса благодаря обновлению только части страницы, что создает мгновенную обратную связь для пользователя. Во-вторых, улучшается общий пользовательский опыт за счет более плавного и отзывчивого интерфейса. В-третьих, снижается нагрузка на сервер благодаря передаче только необходимых данных, что положительно влияет на производительность системы. Наконец, появляется возможность реализации сложных функций, таких как автозаполнение форм, валидация в реальном времени, динамическое отображение контента и многое другое.

Практическое применение AJAX охватывает множество сценариев. Например, автозаполнение форм позволяет мгновенно отображать предлагаемые варианты по мере ввода данных пользователем. Система обновления комментариев обеспечивает отображение новых сообщений без перезагрузки страницы. Динамическое отображение новостей позволяет автоматически обновлять ленту при появлении новых публикаций. Системы поиска демонстрируют результаты по мере ввода поискового запроса, что значительно улучшает пользовательский опыт.

Современные JavaScript-библиотеки и фреймворки, такие как jQuery, Axios и Fetch API, существенно упрощают работу с AJAX. Они предоставляют более лаконичный и удобный синтаксис для отправки запросов и обработки ответов, что делает процесс разработки более быстрым и эффективным. Благодаря этим инструментам, разработчики могут создавать динамичные веб-приложения с минимальными затратами времени и усилий, сохраняя при этом высокое качество и производительность конечного продукта.



Рисунок 1 – AJAX проверка авторизации

Рассмотрим принцип работы JavaScript-кода, который обрабатывает форму авторизации (Auth.php) с использованием технологии AJAX. Данный механизм обеспечивает плавную и удобную авторизацию пользователей без перезагрузки страницы, что значительно улучшает пользовательский опыт.

Процесс обработки формы авторизации начинается с момента загрузки веб-страницы. JavaScript-скрипт инициализирует обработчик событий, который постоянно отслеживает действия пользователя в форме входа. Для корректной работы скрипта форма должна иметь уникальный идентификатор (ID), что позволяет точно определить элемент на странице и связать его с обработчиком событий.

Когда пользователь нажимает кнопку “Войти”, скрипт мгновенно перехватывает стандартное поведение браузера, предотвращая полную перезагрузку страницы. Это достигается путем использования метода preventDefault() для объекта события, что позволяет сохранить текущее состояние страницы и обеспечить более плавный пользовательский интерфейс.

Перед отправкой новых данных на сервер, скрипт выполняет очистку предыдущего состояния формы. Это включает в себя удаление всех ранее отображаемых сообщений об ошибках, что обеспечивает чистоту интерфейса и удобство восприятия информации пользователем. Такой подход предотвращает путаницу при повторных попытках авторизации.

После успешной валидации данных на стороне клиента, скрипт собирает введенные пользователем данные (логин и пароль) и отправляет их на сервер посредством AJAX-запроса. Данные передаются по указанному URL (/api/auth), что позволяет отделить логику авторизации от основной страницы и обеспечить более структурированную архитектуру приложения.

После отправки запроса скрипт переходит в режим ожидания ответа от серверного скрипта, написанного на PHP. На сервере происходит проверка введенных пользователем данных: выполняется валидация формата, сверка с данными в базе данных и формирование соответствующего ответа.

Получив ответ от сервера, JavaScript-скрипт обрабатывает его и принимает соответствующие действия. В случае успешной авторизации пользователь автоматически перенаправляется в личный кабинет, где может воспользоваться всеми доступными функциями приложения. Если же данные для входа оказались неверными, скрипт отображает информативное сообщение об ошибке непосредственно под формой авторизации, указывая на причину неудачной попытки входа.

2.5 Пояснительная характеристика веб-приложения

Главная страница веб-приложения служит отправной точкой для пользователей, стремящихся получить доступ к его функциональности. Одним из ключевых элементов главной страницы является кнопка авторизации, интуитивно понятным образом призывающая пользователей к входу в систему. Этот элемент дизайна играет важную роль, направляя пользователей к безопасному и персонализированному взаимодействию с приложением.

Веб-приложение имеет продуманную структуру пользовательского интерфейса, где каждый элемент выполняет определенную функциональную роль. **Главная страница** выступает в качестве стартовой точки взаимодействия пользователя с системой. Ключевым элементом интерфейса является кнопка авторизации, которая интуитивно направляет пользователей к процессу входа в систему и обеспечивает безопасный доступ к персонализированным функциям приложения.

При активации кнопки авторизации пользователь перенаправляется на специализированную страницу входа, где требуется ввести персональные учетные данные – уникальный логин и пароль. Эти идентификаторы выдаются системным администратором в процессе регистрации, что гарантирует защиту системы от несанкционированного доступа. Только пользователи с подтвержденной идентификацией могут получить доступ к конфиденциальной информации и функционалу приложения.

После успешной авторизации пользователь попадает в **личный кабинет** – персонализированное рабочее пространство, адаптированное под индивидуальные потребности и уровень доступа. Важно отметить, что функциональность личного кабинета формируется динамически на основе прав доступа пользователя, что обеспечивает многоуровневую систему безопасности.

В зависимости от уровня привилегий, пользователи получают доступ к различным функциям системы.

Администраторы имеют расширенные возможности, включающие управление пользователями, просмотр статистики и настройку системных параметров.

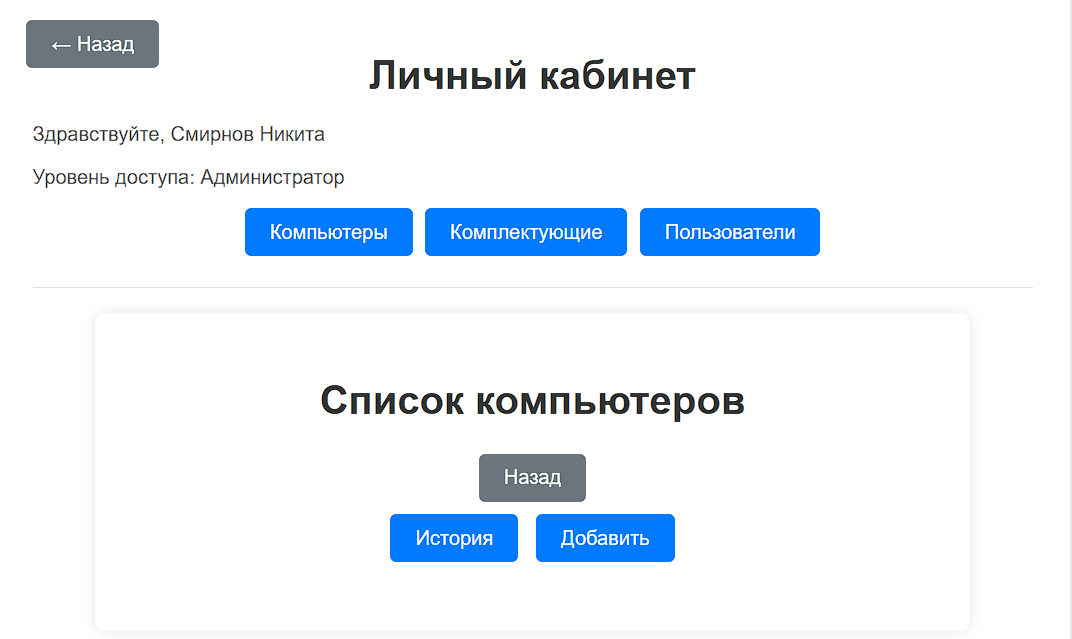


Рисунок 2 – Личный кабинет (Администратор)

Обычные пользователи могут просматривать списки, выполнять определенные операции и получать необходимую отчетность.

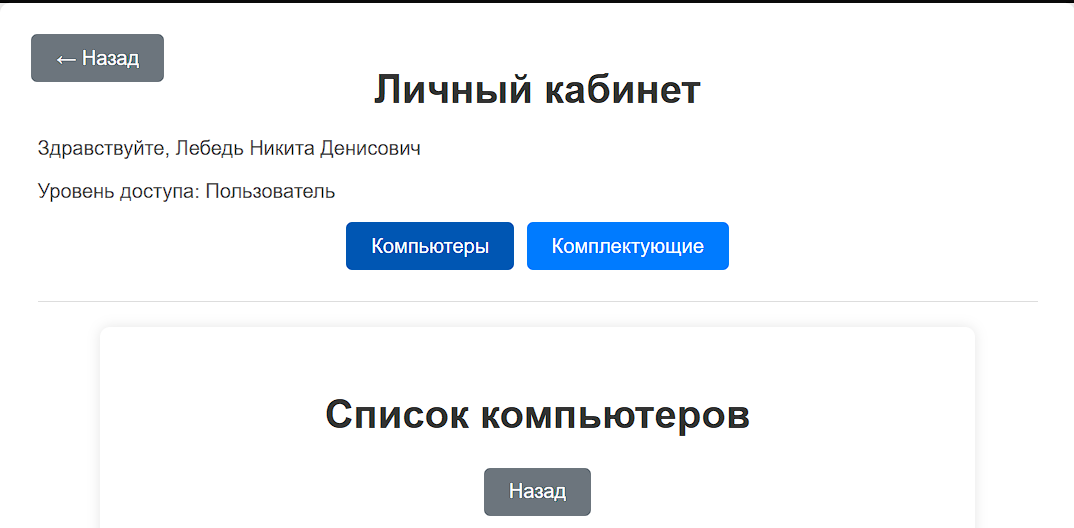


Рисунок 4 – Личный кабинет (Пользователь)

Так видит личный кабинет пользователь с уровнем доступа обычного пользователя.

Администратор может добавлять компоненты, компьютеры и пользователей.

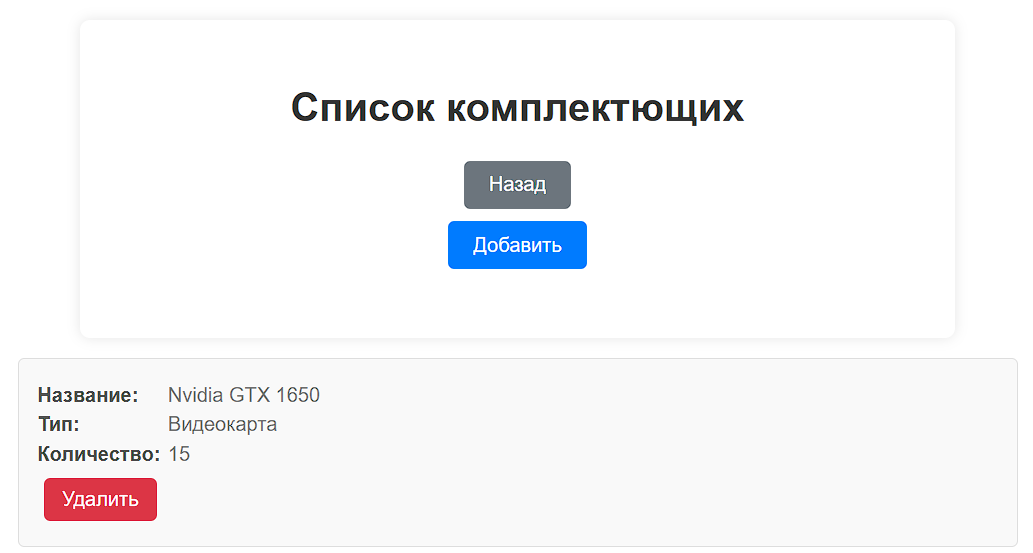


Рисунок 5 – Личный кабинет (Список комплектующих)

При нажатии на кнопку добавить “Добавить” открывается соответствующая форма, в которой администратор, заполнив обязательные поля сможет добавить элемент в базу данных.

Система управления компонентами демонстрирует гибкость интерфейса. В личном кабинете пользователи могут просматривать список комплектующих и при необходимости добавлять новые элементы через специальную форму. При нажатии кнопки “Добавить” открывается форма, где администратор, заполнив обязательные поля, может внести новый элемент в базу данных.

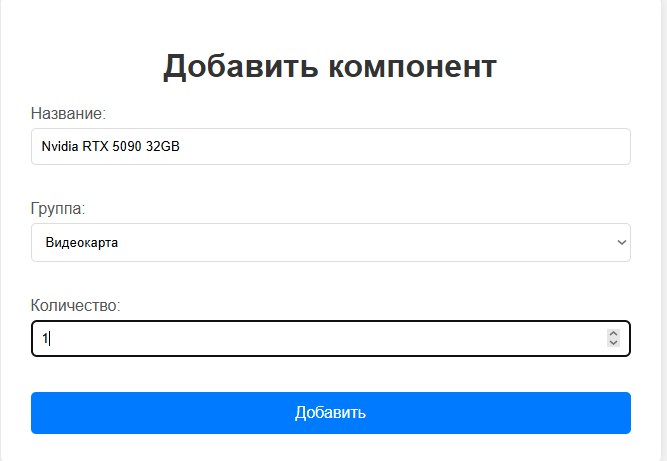


Рисунок 6 – Форма добавления компонентов

Таким образом, структура веб-приложения позволяет создать надежную и удобную среду взаимодействия, где каждый пользователь получает доступ только к тем функциям и данным, которые соответствуют его уровню полномочий, что обеспечивает оптимальное сочетание безопасности и функциональности.

3 ЭКОНОМИЧЕСКАЯ ЧАСТЬ

3.1 Расчет затрат на разработку веб-приложения

Затраты на разработку программного продукта рассчитываются по следующей формуле:

ЗРМУ = ОФОТР + НАПРВФ + ЗЭТ + ЗСПП + ЗХОН + ОСФ

ФОТР – Общий фонд оплаты труда разработчиков:

Формула: ОФОТР = ∑\_(i=1)^m〖OР〗\_i ∙ ТРПРj∙ (1+kд )(1+kу)

Расчет: ОФОТР = 20000 \* 4 \* (1 + 0,1) \* (1 + 1,2) = 176000 руб. (За счет большего времени разработки)

НАПРВФ – Начисления на заработную плату во внебюджетные фонды:

Формула: НАПРВФ = ОФОТР \* НЗАП

Расчет: НАПРВФ = 176000 \* 0,302 = 53203,2 руб.

ЗЭТ – Затраты, связанные с эксплуатацией техники:

Сначала необходимо рассчитать машинное время в часах (Т\_час):

Формула: Т\_час = Т\_мес \* ЧРД \* Т\_см \* К\_см

Расчет: Т\_час = 3 \* 20 \* 8 \* 1 = 480 часов (Увеличилось из-за большего машинного времени)

Формула расчета ЗЭТ: ЗЭТ = Т\_час \* k\_r \* n \* С\_(м-ч)

Расчет: ЗЭТ = 480 \* 0,95 \* 1 \* 10 = 4560 руб. (Увеличилось из-за большего машинного времени)

ЗСПП – Затраты на специальные программные продукты

Формула: ЗСПП = ∑\_(р=1)^n▒Ц\_р

Расчет: ЗСПП = 4500 руб.

ЗХОН – Затраты на хозяйственно-операционные нужды:

Расчет: ЗХОН = 150 руб.

ОСФ – Отчисления в социальные фонды:

Формула: ОСФ = ОФОТР \* КОСФ

Расчет: ОСФ = 176000 \* 0,3 = 52800 руб. (Зависит от ОФОТР)

ЗРМУ = 176000 + 53203,2 + 4560 + 4500 + 150 + 52800 = 291213,2 руб.

Предполагаемые затраты на разработку веб-приложения с АИС по учету компьютеров и комплектующих составляют 291213,2 руб. Данная сумма является оценочной и может варьироваться в зависимости от фактических сроков разработки, сложности функциональности.

3.2 Расчет затрат на внедрение веб-приложения

Затраты на внедрение программного продукта (КВПР) рассчитываются по формуле:

ЗВПР=ЗМ+ЗКТСХ(1+КТУН) + ЗПО+ЗФОТВ+НАПРВФ+ЗЭТ+РКОМ+РН

Где:

ЗВПР – Затраты на внедрение программного продукта, руб.

ЗМ – Затраты на приобретение материалов, руб.

ЗКТСХ – Затраты на приобретение комплекса технических средств, руб.

КТУН – Коэффициент транспортирования, установки и наладки комплекса технических средств.

ЗПО – Затраты на приобретение программного обеспечения (включают стоимость разработанного ПП, а также других существующих ПП, необходимых для функционирования системы), руб.

ЗФОТВ – Затраты на оплату труда работников, занятых внедрением проекта, руб.

НАПРВФ – Отчисления во внебюджетные фонды с заработной платы работников, занятых внедрением проекта, руб.

ЗЭТ – Затраты, связанные с эксплуатацией ЭВМ при внедрении проектного решения, руб.

РКОМ – Командировочные расходы, руб.

РН – Накладные расходы, руб.

Затраты на приобретение материалов (ЗМ):

Приобретение продуктов не требуется следовательно ЗМ = 0 руб. Дополнительного приобретения компьютеров или других КТС так же не требуется, следовательно, ЗКТС = 0

ЗПО = 120627,2 руб. (затраты на приобретение программного обеспечения);

ЗФОТВ = 20000 руб. (оклад) · 0,4 мес. · (1 + 0,1) · (1 + 1,2) = 12342 руб;

НАПРВФ = 20000 руб. · 0,302 = 6040 руб;

ЗЭТ = 0,4 · 20 · 8 · 8 = 512 руб;

РКОМ = 0 (командировочные расходы отсутствуют);

РН = (ЗФОТВ · коэффициент накладных расходов) = 12342 · 0,3 = 3702,6 руб;

Подставим значения:

ЗВПР=0 + 0 ⋅ (1 + 0) + 120627,2 + 12342 + 6040 + 512 + 0 + 3702,6

ЗВПР=120627,2 + 12342 + 6040 + 512 + 3702,6З

ЗВПР=143223,8 руб.

Заключение.

Суммарные затраты на внедрение проекта составляют 143223,8 руб.

4 ОХРАНА ТРУДА ПРИ РАБОТЕ ЗА ПЕРСОНАЛЬНЫМ КОМПЬЮТЕРОМ

Основными вредными факторами при осуществлении трудовой деятельности за персональными компьютерами выступают:

* высокая степень электромагнитного воздействия;
* высокий уровень наличия статического электричества;
* низкая степень ионизации воздуха;
* нагрузки, связанные с длительным сидячим положением тела;
* крайне высокая нагрузка на органы зрения;
* сопутствующие длительной сидячей работе факторы: болевые симптомы в пояснице и позвоночнике, венозная недостаточность, стресс и депрессии.

Охрана труда при работе за компьютером предусматривает следующую структуру соблюдения правил:

* по организации и оснащению трудового места;
* по освещенности;
* по регламентации перерывов в работе.

Организация и общие правила работы.

Нормативные положения СанПиНа предъявляют определенные требования к площади рабочего места при работе за ПК в совокупной мере более 4 часов за смену:

* для мониторов ЭЛТ – от 6 м2 и более;
* для мониторов ЖК – от 4,5 м2 и более.

Общие правила работы за ПК:

1. в помещении с компьютерами рекомендовано, чтобы окна выходили на север или северо-восток;
2. если в офисе отсутствует естественный солнечный свет, должно быть организовано искусственное освещение в соответствии с нормами и правилами освещенности рабочих мест;
3. если мониторы расположены в ряд, люминесцентные лампы следует исполнять в виде сплошных или прерывистых линий;
4. при расположении ПК по периметру, источники освещения должны находиться непосредственно над рабочим столом. Иные требования указаны в Таблице 4.

Таблица 2 – Дополнительные требования

|  |  |
| --- | --- |
| Требования | Расстояние |
| Освещенность рабочего места | От 300 до 500 люкс |
| Освещенность экрана видеомонитора | Не более 30 люкс |
| Нормативное требование к чистке окон и источников света | Не реже двух раз в год |
| Минимальное расстояние между рабочими местами | От 2 метров |
| Минимальное расстояние между видеомониторами | От 1,2 метров |

Данные по оснащению рабочего места представлены в Таблице 5.

Таблица 3 – Оснащение рабочего места

|  |  |
| --- | --- |
| Требования | Расстояние |
| Высота перегородок, разделяющих рабочие места | Не менее 1,5 метров |
| Ширина рабочего стола | От 80 до 140 см |
| Глубина рабочего стола | От 80 до 100 см |
| Высота рабочего стола | 7,25 см |
| Расстояние от глаз до монитора | От 60 до 70 см |
| Расстояние клавиатуры от края стола | От 10 до 30 см |
| Сидение | Должно позволять регулировку по высоте, повороту и углу наклона спинки (регулировки должны быть независимыми друг от друга) |
| Подставка для ног | Ширина – от 30 см, глубина – от 40 см, с углом наклона до 20 градусов |

На рисунке 12 представлена правильная посадка за персональным компьютер

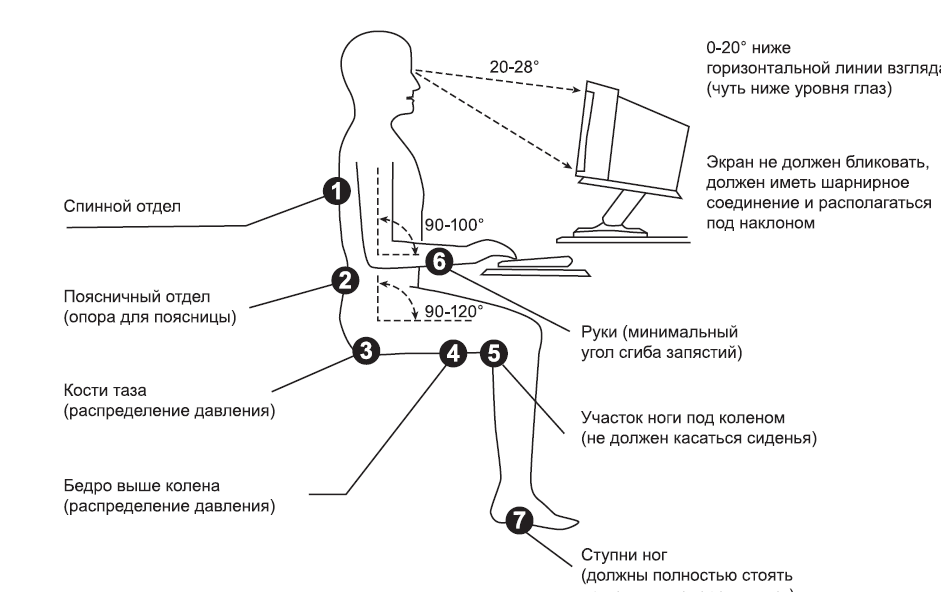


Рисунок 6 - Схема правильной посадки за компьютером

Вышеуказанные нормы установлены в СанПиНе. Кроме того, данный нормативный акт предусматривает:

* осуществление влажной уборки в ежедневном порядке;
* проветривание помещений с компьютерами через каждый час;
* обязательное оборудование офисов системами вентиляции, отопления и кондиционирования воздуха.

СанПиН регламентирует следующие требования к атмосферным показателям в помещениях, оборудованных рабочими местами с ПК:

* температура воздуха – от 19 до 21 градусов;
* влажность – от 55 до 62 %;
* скорость движения воздуха – не больше 0,1 м/с.

Режим работы и перерывы:

* осуществить организацию перерывов в трудовой деятельности продолжительностью от 10 до 15 минут;
* проводить такие перерывы через каждый час работы.

Совокупная продолжительность перерывов: при рабочем дне продолжительностью 8 часов – от 50 минут до 1,5 часа; при 12-часовом – от 80 до 140 минут.

ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломной работы была успешно разработана автоматизированная информационная система (АИС) «Учет компьютеров и комплектующих». Поставленная цель – создание эффективного инструмента для управления IT-инфраструктурой предприятия – была успешно достигнута.

В процессе работы над проектом были последовательно решены следующие задачи: проектирование структуры базы данных, разработка backend-логики на PHP, создание пользовательского интерфейса и проведение комплексного тестирования с последующей отладкой системы.

Результаты проведенной работы имеют практическую значимость и могут быть успешно применены при внедрении АИС в организациях малого и среднего бизнеса, а также в образовательных учреждениях. Внедрение разработанной системы позволит существенно повысить эффективность управления IT-инфраструктурой, сократить затраты на обслуживание оборудования и обеспечить надежную защиту IT-активов предприятия.

Перспективы дальнейшего развития системы включают несколько ключевых направлений. Прежде всего, планируется расширение функциональности системы путем добавления модуля управления лицензиями на программное обеспечение и интеграции с системами мониторинга IT-инфраструктуры. Также перспективным направлением является разработка мобильного приложения для управления АИС с мобильных устройств. Особое внимание планируется уделить внедрению технологий искусственного интеллекта и машинного обучения, что позволит автоматизировать рутинные задачи и реализовать функции прогнозирования отказов оборудования.

Таким образом, результаты данной дипломной работы могут внести существенный вклад в повышение эффективности бизнес-процессов и оптимизацию затрат предприятий. Разработанное приложение представляет собой современный и эффективный инструмент, значительно повышающий удобство и результативность работы организации. Внедрение данной системы позволит предприятиям существенно улучшить качество управления IT-активами и повысить общую эффективность бизнес-процессов.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Григорьев, В.В. Основы веб-программирования: учебник для студ. учреждений высш. проф. образования / В.В. Григорьев. – М.: Издательство, 2020. – 320 с. : ил.
2. Visual Studio Code: Официальная документация / Майкрософт. – 2021. – URL: https://code.visualstudio.com/docs (дата обращения: 05.05.2025)
3. MySQL : MySQL 8.0 Reference Manual / Oracle. – 2021. – URL: https://dev.mysql.com/doc/refman/8.0/en (дата обращения: 08.05.2025).
4. HTML и CSS: Руководство по адаптивной верстке – URL: https://developer.mozilla.org/ru/docs/Web (дата обращения: 06.05.2025)
5. JavaScript: Руководство / Mozilla Developer Network. – 2022. – URL: https://developer.mozilla.org/ru/docs/Web/JavaScript/Guide (дата обращения: 06.05.2025).
6. PHP: Руководство / PHP.net. – 2021. – URL: https://www.php.net/manual/ru/ (дата обращения: 06.05.2025)
7. Веллинга, Л. PHP и MySQL. Разработка Web-приложений / Л. Веллинга, К. Томсон. – М.: Вильямс, 2017. – 864 с. : ил.
8. Крокфорд, Д. JavaScript. Сильные стороны / Д. Крокфорд. – СПб.: Питер, 2024. – 176 с. : ил. (JavaScript: The Good Parts)
9. Симдянов, И.В. HTML5, CSS3 и JavaScript для начинающих. Практическое руководство / И.В. Симдянов. – СПб.: БХВ-Петербург, 2023. – 448 с. : ил.
10. Робачевский, И. PHP 8. Практическое руководство / И. Робачевский. – СПб.: БХВ-Петербург, 2023. – 400 с. : ил.
11. Web Security Guidelines / W3C. – 2022. – URL: https://www.w3.org/Security/ (дата обращения: 15.05.2025).
12. JetBrains. WebStorm: Официальная документация / JetBrains. – 2021. – URL: https://www.jetbrains.com/webstorm/documentation (дата обращения: 15.05.2025).
13. Жуков, В.А. Интернет-безопасность и защита данных: учебник для студ. учреждений высш. проф. образования / В.А. Жуков. – М.: Издательство, 2021. – 340 с. ил.
14. Введение в AJAX с PHP / W3Schools. – 2022. – URL: https://w3schools.tech/ru/tutorial/php/php\_ajax (дата обращения: 10.05.2025).
15. Основной учебник PHP8+ / codemu. – 2021. – URL: https://code.mu/ru/php/book/prime/ (дата обращения: 10.05.2025).

ПРИЛОЖЕНИЕ A