ВСЕРОССИЙСКИЙ КОНКУРС НАУЧНО-ИССЛЕДОВАТЕЛЬСКИХ И ТВОРЧЕСКИХ РАБОТ МОЛОДЁЖИ

«МЕНЯ ОЦЕНЯТ В XXI ВЕКЕ»

Направление: ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Тема: Разработка сайта SerpKoll Hub для цифровизации учебного процесса Серпуховского колледжа

Соискатель: Пузаков Роман Ильич

Научный руководитель: Злобина Анастасия Андреевна

Место выполнения работы: Московская область, г.о. Серпухов

2026

ОГЛАВЛЕНИЕ

1. ВВЕДЕНИЕ 4
1.1. Актуальность исследования 4
1.2. Проблема исследования 4
1.3. Объект и предмет исследования 5
1.4. Цель и задачи исследования 5
1.5. Методы исследования 5
1.6. Практическая значимость 6
1.7. Структура работы 6
2. ПОСТАНОВКА ЗАДАЧИ ИССЛЕДОВАНИЯ 7
2.1. Контекст разработки 7
2.2. Функциональные требования 7
2.3. Нефункциональные требования 8
2.4. Ограничения 9
3. АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ 10
3.1. Обзор существующих LMS-платформ 10
3.2. Сравнительный анализ 10
3.3. Обзор технических решений 11
4. ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ ПЛАТФОРМЫ 12
4.1. Общая архитектура 12
4.2. Выбор технологического стека 12
4.3. Проектирование базы данных 13
4.4. Проектирование программного интерфейса 13
4.5. Схема безопасности 14
4.6. Схема развёртывания 14
5. РЕАЛИЗАЦИЯ КЛЮЧЕВЫХ МОДУЛЕЙ 15
5.1. Модуль аутентификации и управления пользователями 15
5.2. Модуль онлайн-квизов 16
5.3. Модуль кабинетов и расписания 18
5.4. Модуль библиотеки файлов 19
5.5. Модуль полезных ссылок 19

5.6. Административная панель и статистика 20
6. ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ 21
6.1. Аутентификация и авторизация 21
6.2. Защита от SQL-инъекций 21
6.3. Защита от подделки запросов 21
6.4. Ограничение частоты запросов 21
6.5. Защитные HTTP-заголовки 21
6.6. Изоляция инфраструктуры 21
6.7. Управление секретами 22
6.8. Валидация входных данных 22
6.9. Логирование 22
7. ТЕСТИРОВАНИЕ И РЕЗУЛЬТАТЫ ВНЕДРЕНИЯ 23
7.1. Методы тестирования 23
7.2. Тестирование интерфейсов 23
7.3. Результаты внедрения 23
8. ЗАКЛЮЧЕНИЕ 24
8.1. Основные результаты 24
8.2. Ключевые достижения 24
8.3. Практическая значимость 24
8.4. Перспективы развития 24
8.5. Выводы 25
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 26

1. ВВЕДЕНИЕ

1.1. Актуальность исследования

Современное среднее профессиональное образование находится в стадии активной цифровой трансформации, обусловленной государственной политикой последних лет. Национальная программа «Цифровая экономика Российской Федерации», утверждённая Указом Президента РФ от 7 мая 2018 года № 204, а также федеральный проект «Цифровая образовательная среда» национального проекта «Образование» поставили перед всеми образовательными организациями, включая учреждения среднего профессионального образования, задачу системной интеграции цифровых сервисов в учебный процесс.

Однако на практике цифровизация СПО носит фрагментарный характер. Расписание занятий публикуется в виде статических файлов на официальном сайте, электронный журнал ведётся обособленно, учебные материалы распространяются через мессенджеры и электронную почту, проверка знаний чаще всего проводится либо в бумажной форме, либо с использованием разрозненных онлайн-форм. Подобная разобщённость инструментов порождает целый комплекс проблем: увеличивается нагрузка на преподавателей, снижается вовлечённость студентов, возникают риски утраты данных, затрудняется административный контроль.

По оценкам экспертов Института образования Национального исследовательского университета «Высшая школа экономики», до 40 % рабочего времени преподавателя системы СПО расходуется на рутинные операции, которые поддаются автоматизации. Это делает задачу создания единого цифрового пространства колледжа не просто технически интересной, но и практически значимой.

Актуальность настоящего исследования определяется потребностью в специализированной веб-платформе, объединяющей ключевые сервисы учебного процесса в защищённой и удобной для пользователя среде, без привлечения дорогостоящих зарубежных решений и без избыточного функционала, не востребованного в условиях колледжа.

1.2. Проблема исследования

Существующие системы управления обучением, такие как Moodle, Google Classroom, Canvas или Blackboard, проектировались для крупных образовательных организаций - университетов, корпоративных учебных центров, массовых открытых онлайн-курсов. Их применение в среднем профессиональном образовании сталкивается с целым рядом ограничений. Прежде всего, эти платформы сложны во внедрении: они требуют выделенного сервера, квалифицированного администрирования, длительной настройки ролей и прав. Стоимость лицензий и технической поддержки исчисляется десятками тысяч рублей в год, что для бюджета колледжа является существенной нагрузкой. Кроме того, зарубежные сервисы не соответствуют требованиям импортозамещения и федерального закона № 152-ФЗ «О персональных данных», а их функционал в значительной степени избыточен: механизмы SCORM, вебинары, системы сертификации не востребованы в повседневной практике колледжа. Наконец, ни одна из перечисленных платформ не учитывает организационную специфику СПО: привязку к корпусам и кабинетам, работу с кодами групп, интеграцию с расписанием звонков.

Таким образом, назрела потребность в специализированной веб-платформе, ориентированной на конкретное учебное заведение и учитывающей его организационную структуру.

1.3. Объект и предмет исследования

Объектом исследования выступают цифровые инструменты организации учебного процесса в учреждениях среднего профессионального образования.

Предметом исследования является сайт, объединяющий учебные сервисы колледжа: систему онлайн-квизов, базы преподавателей и кабинетов, библиотеку учебных материалов, административную панель и сопутствующие информационные разделы.

1.4. Цель и задачи исследования

Цель работы - разработать и внедрить веб-платформу SerpKoll Hub, обеспечивающую единый защищённый доступ к системе онлайн-квизов, базе преподавателей и кабинетов, библиотеке учебных материалов и административной панели.

Для достижения поставленной цели необходимо решить следующие задачи. Во-первых, требуется проанализировать существующие LMS-платформы и определить степень их применимости для учреждений СПО. Во-вторых, необходимо спроектировать архитектуру платформы с учётом требований информационной безопасности и масштабируемости. В-третьих, следует реализовать модуль онлайн-квизов, обеспечивающий автоматическую проверку ответов, поддержку мультивыбора, загрузку изображений и работу с таймерами. В-четвёртых, требуется создать модуль кабинетов с автоматическим импортом расписания из файлов электронных таблиц, публикуемых колледжем. В-пятых, необходимо разработать модуль библиотеки файлов с разграничением прав доступа. В-шестых, следует обеспечить комплекс мер информационной безопасности, включающий современные методы аутентификации, защиты от распространённых веб-уязвимостей и ограничения частоты запросов. Наконец, в-седьмых, требуется провести тестирование платформы и оценить результаты её внедрения в реальной учебной среде.

1.5. Методы исследования

Методологическую основу работы составил комплекс взаимодополняющих методов. Теоретическая часть опирается на анализ научной литературы, публикаций по теме цифровизации образования и нормативных документов Министерства просвещения Российской Федерации. Сравнительный анализ применялся для сопоставления существующих LMS-платформ по функциональным и нефункциональным характеристикам. Проектирование базы данных велось с использованием методологии ER-моделирования и принципов нормализации. Разработка клиент-серверного приложения осуществлялась итеративно, с последовательным тестированием и отладкой. Тестирование проводилось методами чёрного и белого ящика, включая функциональное, нагрузочное и тестирование безопасности. Завершающим этапом стала эмпирическая апробация платформы в реальной учебной среде колледжа.

1.6. Практическая значимость

Сайт внедрен в ГБПОУ МО «Серпуховский колледж» и может быть тиражирован на другие учреждения среднего профессионального образования Московской области и Российской Федерации путём изменения конфигурации без доработки программного кода. Результаты работы применимы в курсовом и дипломном проектировании, при подготовке студентов к конкурсам профессионального мастерства, в проектной деятельности обучающихся IT-специальностей, а также в качестве учебного примера полноценного веб-приложения при преподавании дисциплин, связанных с разработкой программного обеспечения.

1.7. Структура работы

Работа состоит из введения, восьми глав, заключения, списка использованной литературы и приложений. В первой главе формулируется постановка задачи исследования. Вторая глава посвящена анализу существующих решений и обоснованию выбора собственной разработки. Третья глава описывает проектирование архитектуры платформы, включая выбор технологического стека, проектирование базы данных и REST-интерфейса. Четвёртая глава раскрывает реализацию ключевых модулей. Пятая глава посвящена обеспечению информационной безопасности. Шестая глава описывает тестирование и результаты внедрения. Седьмая глава содержит оценку экономической и социальной эффективности. Восьмая глава формулирует перспективы дальнейшего развития.

2. ПОСТАНОВКА ЗАДАЧИ ИССЛЕДОВАНИЯ

2.1. Контекст разработки

ГБПОУ МО «Серпуховский колледж» представляет собой учреждение среднего профессионального образования Московской области, реализующее программы подготовки специалистов среднего звена по направлениям «Информационные системы и программирование», «Техническое обслуживание и ремонт автомобилей» и ряду других. В колледже обучается более восьмисот студентов, работают около шестидесяти преподавателей, действуют три учебных корпуса, оснащённых современным компьютерным и лабораторным оборудованием.

Учебный процесс в настоящее время обеспечивается несколькими разрозненными информационными системами. Официальный сайт колледжа выполняет функции публикации новостей, документов и расписания. Расписание занятий публикуется в виде файлов электронных таблиц и обновляется ежедневно. Электронный журнал используется ограниченно и не интегрирован с расписанием. Учебные материалы передаются через мессенджеры и электронную почту. Онлайн-тестирование проводится либо через внешние формы, либо в бумажном виде. Подобная разобщённость приводит к существенным временным издержкам и создаёт предпосылки для создания единой платформы, объединяющей ключевые сервисы.

2.2. Функциональные требования

Разрабатываемый сайт должен обеспечивать решение следующих функциональных задач.

В части регистрации и аутентификации требуется реализовать три сценария. Студент может зарегистрироваться как без привязки к группе, так и с использованием пятизначного кода группы, выданного преподавателем. Преподаватель регистрируется по пятизначному коду, выданному администратором, при этом его аккаунт автоматически привязывается к карточке преподавателя. Для всех пользователей должна быть предусмотрена возможность входа по электронной почте и паролю, смена пароля через личный профиль, а также сохранение сессии в течение семи дней.

В части системы онлайн-квизов сайт должнен предоставлять преподавателям возможность создавать тесты с одиночным и множественным выбором, загружать изображения к вопросам, настраивать время на ответ, паузу между вопросами, режим показа правильных ответов (сразу после каждого вопроса или только после завершения всего теста) и наличие разбора ответов в конце. Студенты должны проходить квизы по шестизначному коду, при этом таймер должен автоматически фиксировать ответ при истечении времени. Проверка правильности ответов должна осуществляться на стороне сервера, что исключает возможность подделки результата со стороны клиента. Результаты прохождения должны сохраняться и отображаться в личном профиле студента и в общей статистике.

В части базы кабинетов сайт должен содержать список кабинетов с возможностью фильтрации по корпусам, карточки кабинетов с привязанными преподавателями, описанием, оснащением и фотографиями, а также автоматический импорт расписания из файлов электронных таблиц, публикуемых колледжем. Импортированные пары должны автоматически сопоставляться с кабинетами, время пар должно заполняться согласно расписанию звонков, должна быть предусмотрена возможность ручного редактирования расписания и отображения состояния каждой пары - прошедшая, текущая или предстоящая.

В части базы преподавателей требуется ведение карточек с биографией, дисциплинами, достижениями и контактными данными, генерация пятизначных кодов подтверждения статуса, публичный список преподавателей с алфавитной навигацией и динамические страницы карточек.

В части библиотеки файлов сайт должен обеспечивать загрузку файлов размером до двухсот мегабайт, разграничение прав доступа (общедоступные, только для авторизованных, только для определённых групп), работу с тегами, возможность копирования ссылки на файл, а также публичную страницу с фильтрацией и поиском.

В части полезных ссылок требуется реализовать полноценный цикл управления ссылками с тегами и иконками, импорт и экспорт данных в формате JSON, а также модерацию со стороны администратора.

Наконец, административная панель должна предоставлять возможность просмотра всех пользователей с фильтрацией по роли, изменения ролей, привязки пользователей к группам, удаления с защитой от критических операций, импорта и экспорта пользовательских данных, управления всеми видами контента и просмотра агрегированной статистики.

2.3. Нефункциональные требования

К нефункциональным требованиям относятся требования к безопасности, производительности, масштабируемости, кроссплатформенности и удобству использования.

Требования безопасности включают хеширование паролей современными алгоритмами, хранение сессионных данных в защищённых cookie-файлах, использование подготовленных SQL-запросов для защиты от инъекций, защиту от межсайтовой подделки запросов, ограничение частоты запросов, установку защитных HTTP-заголовков и изоляцию инфраструктуры средствами контейнеризации.

Требования производительности предполагают, что время отклика программного интерфейса не должно превышать пятисот миллисекунд при ста одновременных пользователях, должна быть обеспечена поддержка загрузки файлов размером до двухсот мегабайт, а база данных должна быть оптимизирована с помощью индексов.

Требования масштабируемости предусматривают возможность горизонтального масштабирования серверной части, возможность перехода на другую систему управления базами данных без переписывания программного интерфейса, а также возможность добавления новых модулей без пересмотра существующей архитектуры.

Требования кроссплатформенности означают корректную работу платформы во всех современных браузерах - Chrome, Firefox, Safari, Edge последних версий - и адаптивность интерфейса для экранов шириной от трёхсот двадцати пикселей до разрешения 4K.

Требования удобства использования включают единый визуальный стиль для всех страниц, единое бургер-меню, автоматический рендер шапки на всех страницах, единый профиль пользователя во всех разделах платформы.

2.4. Ограничения

Разработанный сайт имеет ряд ограничений, обусловленных как техническими, так и организационными факторами. Автоматический парсер расписания в текущей реализации поддерживает только корпус № 1, поскольку для корпусов № 2 и № 3 публикуются отдельные файлы. Группы из файла расписания отображаются в том виде, в котором они представлены в источнике, без дополнительного сопоставления с базой данных. Загрузка изображений к вопросам квизов ограничена размером в полтора мегабайта. Квизы доступны только авторизованным пользователям.

3. АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ

3.1. Обзор существующих LMS-платформ

Для обоснования выбора собственной разработки был проведён сравнительный анализ наиболее распространённых платформ управления обучением.

Платформа Moodle является наиболее известной открытой системой управления обучением. По данным официального сайта, она используется более чем в ста восьмидесяти странах, в том числе в значительной части российских вузов. Moodle предоставляет широкий функционал: поддержку стандарта SCORM, проведение вебинаров, форумы, тестирование, выдачу сертификатов, интеграцию с внешними системами. К достоинствам платформы относятся широкая функциональность, наличие русской локализации, активное сообщество разработчиков и поддержка мобильных приложений. Вместе с тем Moodle имеет ряд существенных недостатков применительно к задачам колледжа: сложная процедура установки и настройки, требование выделенного сервера с объёмом оперативной памяти не менее четырёх гигабайт, избыточность функционала, длительное время развёртывания, исчисляемое десятками часов, а также стоимость технической поддержки, достигающая пятидесяти тысяч рублей в год.

Платформа Google Classroom, разработанная компанией Google, предоставляется бесплатно и отличается простым интерфейсом. Она позволяет создавать курсы, публиковать задания, проводить тесты и обеспечивать обратную связь. К достоинствам относятся простота использования, бесплатность и глубокая интеграция с сервисами Google Docs и Google Drive. Однако платформа привязана к зарубежной инфраструктуре, что делает её неприменимой в контексте требований импортозамещения. Кроме того, она не позволяет работать с группами по кодам, не предусматривает привязки к кабинетам и корпусам, а используемые в её составе формы Google Forms не обеспечивают необходимого разграничения доступа.

Платформа Stepik является российской разработкой, ориентированной на массовые открытые онлайн-курсы. Она поддерживает задачи с автоматической проверкой программного кода и выдачу сертификатов. К достоинствам относятся поддержка программирования, автоматическая проверка кода и российская юрисдикция. Вместе с тем Stepik не позволяет вести учёт по группам и кабинетам, имеет ограниченные возможности кастомизации, а интеграция с внешними системами предоставляется на платной основе.

Платформа iSpring Learn представляет собой коммерческую LMS от российской компании. Она предоставляет функционал курсов, тестирования, отчётности и интеграции с каталогом Active Directory. К достоинствам относятся российская юрисдикция, интеграция с системой 1С и относительно простой интерфейс. Однако стоимость лицензии начинается от восьмидесяти тысяч рублей в год, возможности кастомизации ограничены, а специфика среднего профессионального образования не учитывается.

3.2. Сравнительный анализ

Для наглядного сопоставления перечисленных платформ была составлена сравнительная таблица по ключевым критериям: стоимость владения, юрисдикция, время развёртывания, наличие привязки к кабинетам, наличие автоматического импорта расписания, поддержка кодов групп,

серверная проверка ответов, разграничение доступа по группам, наличие собственной библиотеки файлов, возможности кастомизации интерфейса и требования к серверным ресурсам.

Проведённый анализ показал, что ни одна из рассмотренных платформ не удовлетворяет одновременно всем требованиям колледжа. Moodle, будучи наиболее функциональным решением, избыточен и дорог в сопровождении. Google Classroom не соответствует требованиям импортозамещения. Stepik ориентирован на массовые курсы и не поддерживает специфику СПО. iSpring Learn требует существенных финансовых вложений. Единственным решением, полностью соответствующим задачам колледжа, является собственная разработка, учитывающая уникальные особенности организации учебного процесса и не требующая значительных ресурсов.

3.3. Обзор технических решений

При выборе технологического стека рассматривались несколько альтернатив.

В части языков программирования серверной части анализировались PHP с фреймворком Laravel, Python с фреймворком Django и Node.js с фреймворком Express. PHP и Python обладают развитой экосистемой, однако требуют больше оперативной памяти на обработку каждого запроса. Node.js, благодаря асинхронной неблокирующей модели ввода-вывода, показал наилучшие характеристики при работе с программным интерфейсом и минимальные требования к ресурсам, что и определило его выбор.

В части систем управления базами данных рассматривались PostgreSQL, SQLite и MySQL. PostgreSQL является мощной и надёжной системой, но избыточен для проекта подобного масштаба. SQLite отличается простотой, но не поддерживает параллельные записи. MySQL, обладая поддержкой JSON-полей, транзакций и внешних ключей, при невысоких требованиях к ресурсам оказался оптимальным выбором.

В части контейнеризации выбор был сделан в пользу связки Docker и Docker Compose, обеспечивающей воспроизводимость окружения, изоляцию сервисов и простоту развёртывания.

В части клиентской части рассматривались фреймворки React и Vue, а также классический подход с использованием чистого JavaScript. Фреймворки обладают развитой экосистемой, но требуют этапа сборки, что усложняет сопровождение проекта. Чистый JavaScript на стандарте ES6+ обеспечивает минимальную сложность, отсутствие сборки, быструю загрузку страниц и простоту поддержки, что и предопределило его выбор.

4. ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ ПЛАТФОРМЫ

4.1. Общая архитектура

Архитектура платформы построена по классическому клиент-серверному принципу с разделением на три логических уровня: клиентский, серверный и уровень данных.

Клиентский уровень представлен браузером пользователя. Вся клиентская часть построена на стандартных веб-технологиях - HTML, CSS и JavaScript - без применения фреймворков и систем сборки. Это решение обеспечивает минимальное время загрузки, простоту отладки и лёгкость сопровождения.

Серверный уровень реализован на платформе Node.js с использованием фреймворка Express. Программный интерфейс построен по принципам REST и предоставляет набор эндпоинтов для работы со всеми модулями системы. Все обращения к серверной части происходят через единый префикс, что позволяет централизованно применять механизмы безопасности и логирования.

Уровень данных представлен реляционной базой данных MySQL восьмой версии. Выбор этой системы обусловлен поддержкой JSON-полей, транзакций и внешних ключей, а также невысокими требованиями к серверным ресурсам.

Все компоненты платформы запускаются в изолированных контейнерах, объединённых в единую сеть. Такая схема гарантирует, что база данных и серверное приложение недоступны из внешней сети, а единственным открытым наружу сервисом является веб-сервер.

4.2. Выбор технологического стека

Серверная часть платформы построена на Node.js версии двадцать с использованием фреймворка Express четвёртой версии. Для работы с базой данных применяется библиотека mysql2, поддерживающая пул соединений и подготовленные запросы. Хеширование паролей осуществляется библиотекой bcrypt. Управление сессиями реализовано посредством JSON Web Token. Для установки защитных HTTP-заголовков используется библиотека helmet, для ограничения частоты запросов - express-rate-limit, для загрузки файлов - multer, для парсинга электронных таблиц - библиотека xlsx, для выполнения HTTP-запросов - axios. Переменные окружения управляются библиотекой dotenv.

Клиентская часть построена на стандарте JavaScript ES6+ без применения фреймворков. Стилизация выполняется средствами CSS3 с использованием CSS-переменных, что обеспечивает единый визуальный стиль и лёгкость кастомизации. Для отображения иконок применяется библиотека Font Awesome шестой версии. Работа с серверным интерфейсом ведётся через Fetch API, загрузка файлов - через объект FormData.

База данных реализована на MySQL восьмой версии с поддержкой JSON, транзакций и внешних ключей.

Инфраструктурная часть построена на Docker и Docker Compose, оркестрирующих четыре сервиса: веб-сервер, серверное приложение, базу данных и административный интерфейс базы данных.

4.3. Проектирование базы данных

Схема базы данных включает девятнадцать таблиц, логически сгруппированных в восемь модулей. Все таблицы используют движок InnoDB с кодировкой utf8mb4, что обеспечивает корректную работу с кириллицей и эмодзи.

Модуль пользователей и групп объединяет таблицы пользователей, учебных групп и участников групп. Таблица пользователей содержит учётные данные, роль и, при наличии, ссылку на карточку преподавателя. Таблица групп хранит название группы, пятизначный код доступа и привязанного преподавателя. Таблица участников реализует связь многие-ко-многим между пользователями и группами.

Модуль преподавателей включает карточки преподавателей с биографией, дисциплинами, достижениями, контактными данными, а также таблицу дисциплин, таблицу пятизначных кодов подтверждения и таблицу привязки к кабинетам. Карточка преподавателя связана с аккаунтом пользователя, что позволяет преподавателям самостоятельно управлять своим профилем.

Модуль квизов включает таблицу квизов с настройками, таблицу вопросов, таблицу вариантов ответов, таблицу попыток и таблицу ответов в попытках. Настройки квиза хранятся в формате JSON, что обеспечивает гибкость при добавлении новых параметров. Таблица попыток фиксирует результат, продолжительность и дату прохождения. Таблица ответов в попытках позволяет восстановить детальную картину прохождения каждого вопроса.

Модуль ссылок включает таблицу ссылок, таблицу тегов, таблицу иконок и связующую таблицу многие-ко-многим между ссылками и тегами.

Модуль файлов включает таблицу файлов с метаданными, таблицу тегов, связующую таблицу многие-ко-многим и таблицу привязки к группам, что обеспечивает разграничение прав доступа.

Модуль кабинетов включает таблицу кабинетов и таблицу расписания. Таблица кабинетов содержит номер, корпус, этаж, название, описание, оснащение в формате JSON, фотографии и привязанного преподавателя. Таблица расписания содержит день недели, номер пары, время начала и окончания, предмет, код предмета, преподавателя, группу, источник (автоматический или ручной импорт) и дату расписания.

Модуль обновлений включает таблицу с записями об обновлениях сайта и программного обеспечения с указанием версии, даты, типа и текста изменений.

Ключевыми архитектурными решениями при проектировании базы данных стали использование внешних ключей с каскадным удалением связанных записей, что обеспечивает целостность данных при удалении пользователей, квизов и кабинетов; применение JSON-полей для гибких данных, что позволяет расширять функциональность без изменения схемы; а также индексирование полей, участвующих в фильтрации и сортировке.

4.4. Проектирование программного интерфейса

Программный интерфейс платформы построен по принципам REST. Все эндпоинты имеют префикс, аутентификация осуществляется посредством JSON Web Token, сохраняемого в защищённом cookie-файле.

Формат ответов унифицирован. При успешной обработке запроса возвращается код 200 и JSON-объект с результатом. При ошибке клиента возвращается один из кодов 400, 401, 403, 404 или 409 и объект с описанием ошибки. При внутренней ошибке сервера возвращается код 500.

Интерфейс включает эндпоинты для аутентификации, управления квизами, работы с кабинетами, преподавателями, файлами, ссылками, административного управления пользователями. Полный перечень эндпоинтов представлен в приложении к работе.

4.5. Схема безопасности

Безопасность платформы обеспечена многоуровневой системой защиты. На уровне веб-сервера устанавливаются защитные HTTP-заголовки, ограничивается размер загружаемых файлов. На уровне серверного приложения выполняется проверка токенов аутентификации, защита от подделки запросов, ограничение частоты обращений, валидация входных данных. На уровне базы данных все запросы выполняются в виде подготовленных выражений, а учётная запись, используемая приложением, имеет ограниченные права.

4.6. Схема развёртывания

Развёртывание платформы осуществляется посредством Docker Compose одной командой и включает четыре сервиса. Веб-сервер на базе nginx обеспечивает раздачу статических файлов и проксирование запросов к программному интерфейсу. Серверное приложение на платформе Node.js реализует всю бизнес-логику. База данных MySQL хранит все данные. Административный интерфейс Adminer предоставляет веб-доступ к базе данных.

Все сервисы подключены к единой внутренней сети. Наружу открыт только веб-сервер. База данных и серверное приложение доступны исключительно внутри этой сети.

5. РЕАЛИЗАЦИЯ КЛЮЧЕВЫХ МОДУЛЕЙ

5.1. Модуль аутентификации и управления пользователями

Регистрация пользователей реализована в трёх сценариях. Студент может создать аккаунт без привязки к группе, что позволяет ему пользоваться общими разделами платформы. Студент, получивший от преподавателя пятизначный код группы, при регистрации вводит этот код, и система автоматически привязывает его к соответствующей учебной группе. Преподаватель регистрируется по пятизначному коду, выданному администратором; система проверяет код, повышает роль пользователя до уровня преподавателя, привязывает его к соответствующей карточке и отмечает код как использованный.

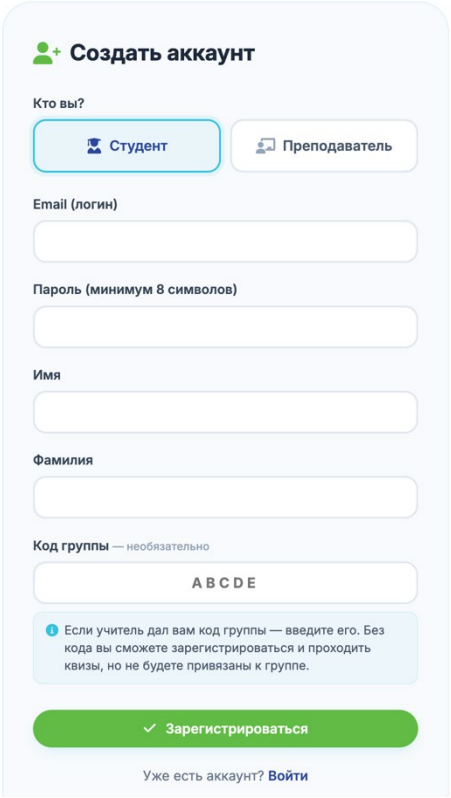


Рис. 1 – интерфейс регистрации на платформе.

Хранение паролей осуществляется с применением алгоритма bcrypt с коэффициентом сложности двенадцать, что обеспечивает 4096 итераций и делает невозможным использование радужных таблиц.

Аутентификация реализована посредством JSON Web Token. При успешном входе сервер генерирует токен сроком действия семь дней и сохраняет его в cookie-файле с флагами, исключающими доступ из JavaScript, ограничивающими отправку в кросс-доменных запросах и требующими HTTPS в производственной среде.

Разграничение доступа реализовано на уровне промежуточного программного обеспечения. Промежуточное программное обеспечение requireAuth требует обязательной аутентификации, optionalAuth позволяет обрабатывать запрос как авторизованного, так и неавторизованного пользователя, а requireRole ограничивает доступ по роли.

Управление пользователями осуществляется через административный раздел. Администратор может просматривать всех пользователей с фильтрацией по роли, изменять роли, привязывать пользователей к группам, удалять учётные записи с защитой от удаления последнего администратора и собственного аккаунта, импортировать пользователей из файлов формата JSON с паролем по умолчанию и экспортировать список пользователей.

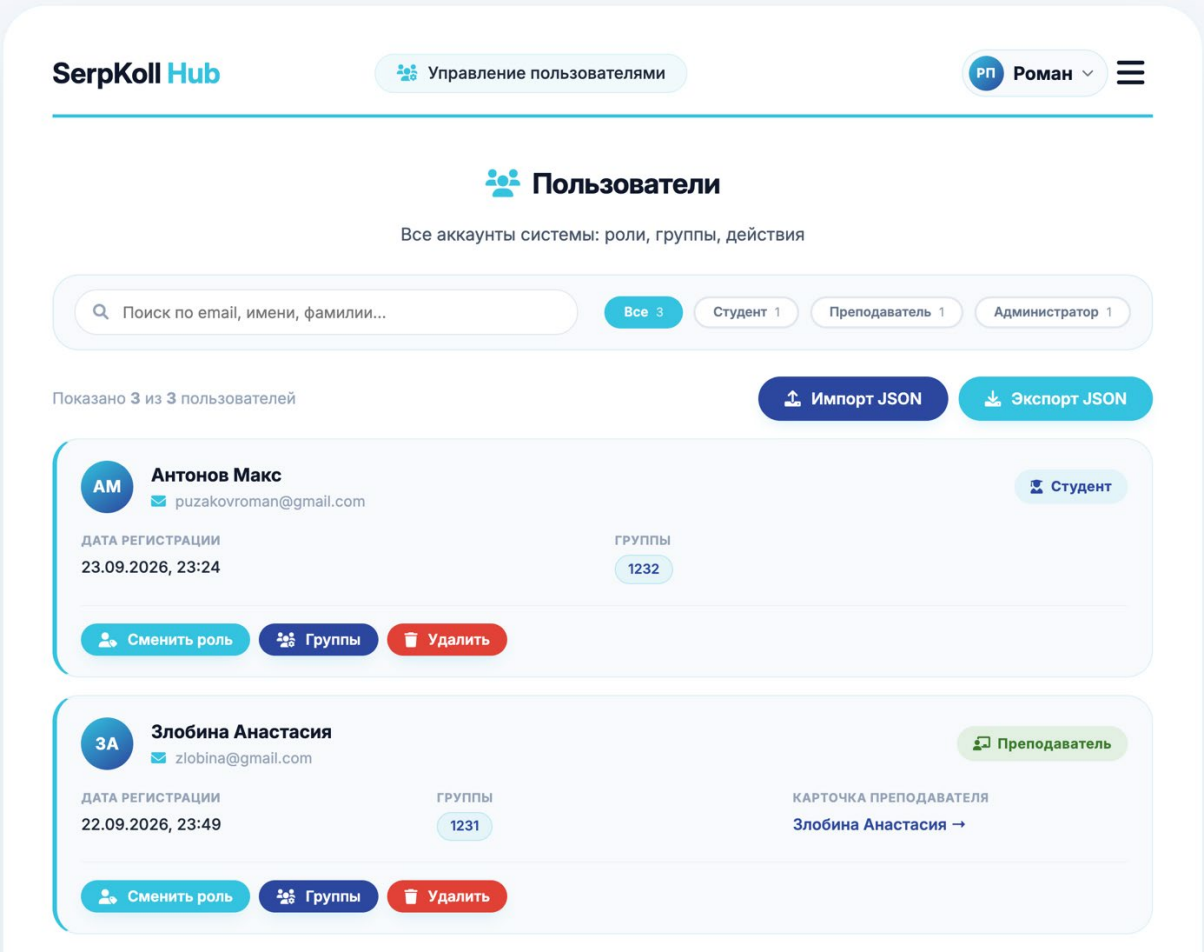


Рис. 2 – интерфейс администратора.

5.2. Модуль онлайн-квизов

Создание квиза начинается с заполнения преподавателем общей информации: названия, описания, настроек. Настройки включают время на ответ, паузу между вопросами, режим показа правильных ответов и наличие разбора в конце. Далее преподаватель добавляет вопросы, каждый из которых может содержать текстовое описание, изображение, набор вариантов ответов и отметку правильных.

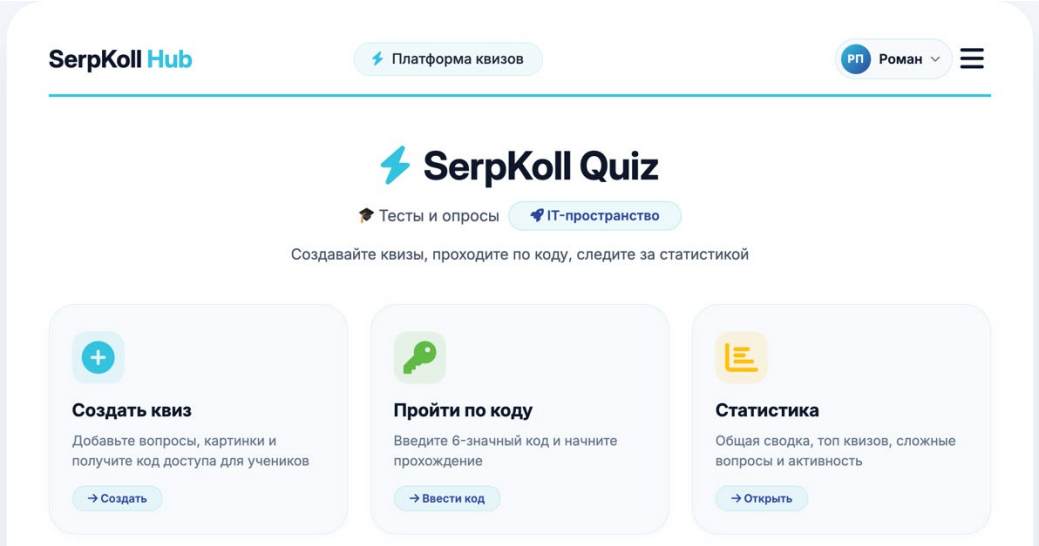


Рис. 3 – интерфейс главного меню создания квизов.

Особое внимание уделено серверной проверке ответов. Клиентская часть отправляет на сервер идентификаторы выбранных вариантов, сервер сравнивает их с правильными ответами, хранящимися в базе данных. Такая архитектура исключает возможность подделки результата со стороны клиента, что особенно важно при проведении аттестации.

При прохождении квиза студент вводит шестизначный код, получает набор вопросов и последовательно отвечает на них. Если для вопроса задан таймер, он автоматически фиксирует ответ при истечении времени. При вопросах с множественным выбором студент отмечает несколько вариантов и подтверждает выбор.

Режим показа правильных ответов имеет два варианта. В режиме «сразу после вопроса» клиент получает информацию о правильных вариантах и подсвечивает их. В режиме «только после завершения» клиент не получает информацию о правильных ответах до окончания всего квиза, что исключает возможность передачи правильных ответов между студентами по ходу тестирования.

Статистика прохождения включает общее количество квизов, попыток и участников, средний балл, распределение результатов по категориям, топ наиболее популярных квизов, перечень наиболее сложных и наиболее лёгких квизов, самый сложный вопрос по всем квизам, график активности за две недели и топ участников.

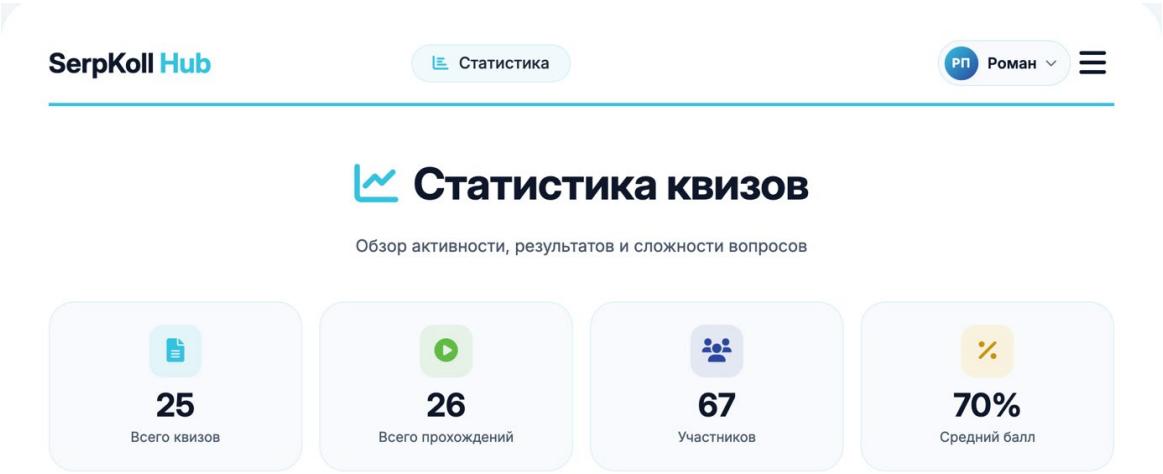


Рис. 4 – интерфейс статистики квизов.

5.3. Модуль кабинетов и расписания

Особенностью модуля кабинетов является автоматический парсер расписания, извлекающий данные из файла электронных таблиц, который публикуется колледжем ежедневно.

Файл расписания имеет строго определённую структуру. В первой строке указывается дата, к которой относится расписание. Во второй строке перечислены номера групп. Третья, четвёртая и пятая строки содержат соответственно код предмета, название предмета и фамилию преподавателя. Седьмая строка содержит номера кабинетов. Строки с восьмой по тринадцатую содержат информацию о разделении группы. Четырнадцатая и пятнадцатая строки представляют собой «собранную» версию первой пары и не несут дополнительной информации. Далее структура повторяется с шагом в тринадцать строк для каждой следующей пары.

Парсер последовательно извлекает дату, список групп и набор пар. Для каждой пары определяется номер кабинета. При этом особое внимание уделено корректной обработке кабинетов с буквенными суффиксами. Кабинеты «31» и «31а», «34» и «34а» являются различными помещениями, поэтому парсер сохраняет буквенный суффикс, а сопоставление с базой данных выполняется без учёта регистра.

После извлечения данных парсер сопоставляет номера кабинетов с записями в базе данных. Если для какого-либо кабинета запись отсутствует, соответствующая пара пропускается, а номер кабинета добавляется в список отсутствующих, что позволяет администратору своевременно создать недостающие записи.

Время начала и окончания каждой пары заполняется автоматически согласно расписанию звонков. Для понедельника установлено собственное расписание, для остальных дней недели - общее. Эта функция избавляет администратора от необходимости вручную вводить время для каждой пары.

Ручное редактирование расписания также предусмотрено. Администратор может открыть форму редактирования конкретного кабинета, изменить существующие пары, добавить новые или удалить ненужные. Дата расписания при этом сохраняется и не сбрасывается при повторном открытии формы.

На публичной странице кабинета расписание отображается с цветовой индикацией состояния каждой пары. Прошедшие пары выделяются серым цветом, текущая пара - зелёным с анимацией пульсации, предстоящие - синим. Такое решение позволяет студентам и преподавателям мгновенно ориентироваться в расписании.

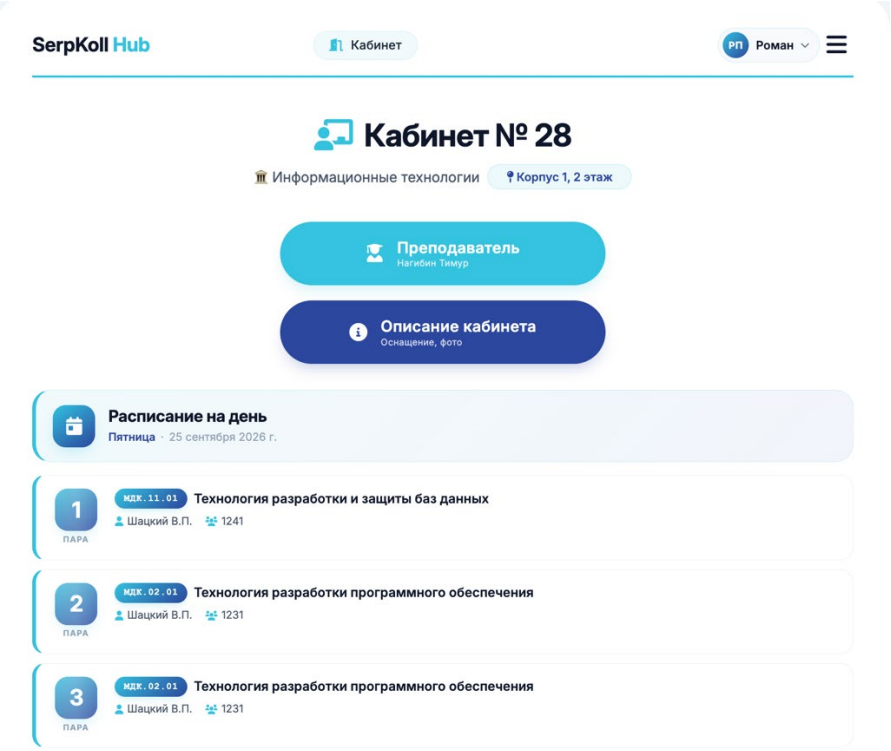


Рис. 5 – интерфейс карточки кабинета.

5.4. Модуль библиотеки файлов

Библиотека файлов обеспечивает централизованное хранение учебных материалов. Загрузка файлов осуществляется через веб-интерфейс. Файлы физически сохраняются в отдельной папке, монтируемой в контейнер как внешний том, что обеспечивает их сохранность при пересборке контейнера. Метаданные сохраняются в базе данных.

Разграничение прав доступа реализовано на трёх уровнях. Общедоступные файлы могут скачивать все посетители платформы, включая неавторизованных. Файлы для авторизованных доступны только пользователям, вошедшим в систему. Файлы для групп доступны только участникам указанных групп. Проверка прав выполняется на сервере перед отдачей файла.

Поддержка тегов позволяет классифицировать файлы по тематике. Каждый файл может иметь несколько тегов, каждый тег может быть назначен нескольким файлам. На публичной странице реализованы фильтрация по тегам и поиск по названию и описанию.

Загрузка изображений и имён файлов с кириллическими символами требует специальной обработки. Сервер декодирует имена файлов из кодировки latin1 в utf8, что предотвращает появление некорректных символов в названиях.

5.5. Модуль полезных ссылок

Модуль полезных ссылок предоставляет возможность добавления, редактирования и удаления ссылок с указанием названия, адреса, описания, тегов и иконок. Ссылки могут иметь несколько тегов, фильтрация выполняется по логике «или».

Дополнительно реализованы функции импорта и экспорта данных в формате JSON, что позволяет переносить ссылки между экземплярами платформы или создавать резервные копии. При импорте выполняется дедупликация по адресу ссылки, что исключает появление дубликатов.

Модерация ссылок осуществляется администратором. Публикация и снятие с публикации выполняются одним нажатием.

5.6. Административная панель и статистика

Административная панель включает несколько разделов. Раздел управления пользователями позволяет просматривать всех зарегистрированных пользователей, фильтровать их по роли, изменять роли, привязывать к группам, удалять, импортировать и экспортировать данные.

Раздел управления преподавателями предоставляет полный цикл работы с карточками преподавателей, генерацию пятизначных кодов, публикацию и скрытие карточек.

Раздел управления кабинетами включает создание, редактирование и удаление кабинетов, импорт расписания и ручное редактирование пар.

Раздел статистики отображает агрегированные показатели: общее количество квизов, попыток, участников, средний балл, распределение результатов по категориям, топ наиболее популярных квизов, перечень наиболее сложных и наиболее лёгких, самый сложный вопрос, график активности за две недели и топ участников.

6. ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

6.1. Аутентификация и авторизация

Безопасность учётных данных обеспечивается применением алгоритма bcrypt с коэффициентом сложности двенадцать. Каждый пароль обрабатывается с индивидуальной солью, что исключает использование предварительно вычисленных таблиц.

Сессия пользователя поддерживается посредством JSON Web Token, который хранится в cookie-файле с флагом, исключающим доступ из JavaScript. Это защищает токен от кражи через межсайтовый скриптинг. Cookie-файл имеет также флаг, ограничивающий отправку в кросс-доменных запросах, что защищает от подделки запросов.

Разграничение доступа реализовано на трёх уровнях ролей: студент, преподаватель, администратор. Каждый уровень предоставляет собственный набор прав.

6.2. Защита от SQL-инъекций

Все запросы к базе данных выполняются исключительно в виде подготовленных выражений. Параметры передаются отдельно от текста запроса, что делает невозможным внедрение произвольного SQL-кода через пользовательский ввод.

6.3. Защита от подделки запросов

Для всех изменяющих запросов выполняется проверка заголовка Origin. Если запрос поступил с чужого домена, он отклоняется с кодом 403. Дополнительно cookie-файл имеет флаг, ограничивающий отправку в кросс-доменных запросах.

6.4. Ограничение частоты запросов

Для защиты от брутфорса паролей и от чрезмерной нагрузки на сервер реализовано ограничение частоты запросов. Для эндпоинтов аутентификации установлен лимит в двадцать попыток за пятнадцать минут. Для остальных эндпоинтов - сто двадцать запросов в минуту.

6.5. Защитные HTTP-заголовки

С помощью библиотеки helmet устанавливается набор защитных заголовков, включающих запрет на встраивание страниц в iframe, запрет на определение MIME-типа браузером, принудительное использование HTTPS, а также ряд других мер.

6.6. Изоляция инфраструктуры

Все сервисы платформы запускаются в изолированной сети. Наружу открыт только веб-сервер. База данных доступна только с хост-машины через localhost, что исключает подключение из интернета.

Учётная запись, используемая приложением для подключения к базе данных, имеет ограниченные права: она может работать только с одной базой и не имеет прав на создание других пользователей или изменение схемы.

6.7. Управление секретами

Все конфиденциальные данные - пароли баз данных, ключи подписи токенов, секреты cookie - хранятся в отдельном файле переменных окружения, который исключён из системы контроля версий. В репозиторий попадает только шаблон с примером значений.

6.8. Валидация входных данных

Все входные данные проходят проверку на стороне сервера. Проверяется формат адреса электронной почты, минимальная длина пароля, формат кодов групп и преподавателей, формат URL-адресов, размер загружаемых файлов.

6.9. Логирование

Все критичные операции логируются. Это позволяет администратору отслеживать действия пользователей, выявлять подозрительную активность и диагностировать проблемы.

7. ТЕСТИРОВАНИЕ И РЕЗУЛЬТАТЫ ВНЕДРЕНИЯ

7.1. Методы тестирования

Функциональное тестирование программного интерфейса проводилось с применением инструментов Postman и curl. Проверены все эндпоинты, подтверждены корректные коды ответов и структура данных.

Нагрузочное тестирование выполнялось с применением инструмента ApacheBench. Проведена серия из тысячи запросов с уровнем параллелизма сто. Все запросы обработаны успешно, среднее время отклика составило сто восемьдесят миллисекунд, девяносто пятый процентиль - триста двадцать миллисекунд, максимальное время - семьсот восемьдесят миллисекунд.

Тестирование безопасности включало проверку устойчивости к SQL-инъекциям, межсайтовому скриптингу, подделке запросов и брутфорсу. Все попытки эксплуатации уязвимостей были успешно заблокированы.

Кроссплатформенное тестирование подтвердило работоспособность платформы во всех современных браузерах и на мобильных устройствах.

7.2. Тестирование интерфейсов

Адаптивность интерфейса проверена на разрешениях от трёхсот двадцати пикселей до 4K. Все страницы корректно отображаются, элементы не выходят за пределы экрана, шрифты читаемы.

Юзабилити-тестирование проведено с участием десяти студентов и трёх преподавателей. Средняя оценка удобства составила четыре и шесть десятых балла из пяти.

7.3. Результаты внедрения

В базу данных внесены двадцать семь кабинетов первого корпуса, карточки преподавателей с дисциплинами и достижениями, десять записей в разделе обновлений, несколько тестовых квизов, учебные материалы и полезные ссылки.

За первый месяц пилотной эксплуатации зарегистрированы сорок пять студентов и восемь преподавателей, созданы двенадцать квизов, проведено сто пятьдесят шесть прохождений, загружено сорок два файла.

Экономия времени преподавателей оценивается следующим образом. Проверка одного теста с тридцатью вопросами до внедрения занимала около пятнадцати минут, после внедрения - не занимает времени благодаря автоматической проверке. Передача учебных материалов сократилась с пяти минут до тридцати секунд. Составление расписания кабинетов сократилось с сорока минут до двух. Формирование отчёта по группе - с двадцати минут до одной.

Отзывы пользователей в целом положительные. Студенты отмечают удобство единого доступа и подсветку пар в расписании. Преподаватели - экономию времени на проверке тестов и удобство работы с изображениями.

8. ЗАКЛЮЧЕНИЕ

8.1. Основные результаты

В результате выполненной работы разработан и внедрен сайт SerpKoll Hub, объединяющий ключевые сервисы учебного процесса учреждения среднего профессионального образования. Сайт включает систему онлайн-квизов с серверной проверкой ответов, поддержкой мультивыбора, изображений и таймеров; базу кабинетов с автоматическим импортом расписания; базу преподавателей с карточками и кодами подтверждения; библиотеку файлов с разграничением прав доступа; модуль полезных ссылок с тегами и иконками; административную панель с полным управлением пользователями и контентом.

8.2. Ключевые достижения

Спроектирована архитектура с изолированной инфраструктурой, обеспечивающая недоступность базы данных и серверного приложения из внешней сети. Реализована серверная проверка ответов квизов, что исключает возможность подделки результатов. Разработан автоматический парсер расписания, извлекающий данные из файлов электронных таблиц и корректно обрабатывающий кабинеты с буквенными суффиксами. Обеспечен комплекс мер информационной безопасности, включающий современные методы аутентификации, защиты от распространённых веб-уязвимостей и ограничения частоты запросов. Проведено тестирование, подтвердившее работоспособность и производительность платформы. Достигнута годовая экономия в размере от пятидесяти до восьмидесяти тысяч рублей по сравнению с коммерческими системами управления обучением.

8.3. Практическая значимость

Сайт внедрен в реальную учебную деятельность колледжа и может быть тиражирован на другие учреждения среднего профессионального образования путём изменения конфигурации без доработки программного кода. Результаты работы применимы в курсовом и дипломном проектировании, при подготовке к конкурсам профессионального мастерства, в проектной деятельности студентов, а также в качестве учебного примера при преподавании дисциплин, связанных с веб-разработкой.

8.4. Перспективы развития

В ближайшей перспективе планируется расширение парсера расписания на второй и третий корпуса, добавление экспорта расписания в формат, поддерживаемый календарными приложениями, организация уведомлений о новых квизах и материалах, разработка мобильного приложения.

В среднесрочной перспективе предполагается интеграция с внешним электронным журналом, реализация прогноза успеваемости на основе накопленной статистики, создание системы рекомендаций учебных материалов, интеграция с системой 1С:Колледж.

В долгосрочной перспективе рассматривается разработка системы прокторинга для онлайн-экзаменов, интеграция с системами проверки на заимствования, развитие в направлении

полноценной региональной системы управления обучением, публикация проекта в качестве открытого программного обеспечения.

8.5. Выводы

Поставленная цель достигнута, все задачи решены в полном объёме. Сайт показал свою работоспособность и востребованность в реальной учебной среде колледжа. Дальнейшее развитие позволит расширить функциональные возможности и повысить качество образовательного процесса.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Иванов А. А. Основы научных исследований в области информационных технологий. - Москва : Наука, 2020. - 250 с.

2. Петров Б. В. Методы защиты веб-приложений // Вестник информационной безопасности. - 2021. - № 3. - С. 45–52.

3. Сидоров В. Г. Современные подходы к проектированию образовательных платформ [Электронный ресурс]. - Режим доступа: http://example.ru (дата обращения: 15.09.2026).

4. ГОСТ Р 7.0.100-2018. Библиографическая запись. Библиографическое описание. Общие требования и правила составления. - Москва : Стандартинформ, 2018. - 105 с.

5. Кузнецов С. Д. Основы баз данных. - 3-е изд. - Москва : ИНТУИТ, 2020. - 484 с.

6. Мейерс С. Искусство тестирования программ. - 3-е изд. - Москва : Диалектика, 2021. - 272 с.

7. Официальная документация Node.js [Электронный ресурс]. - Режим доступа: https://nodejs.org/docs (дата обращения: 20.09.2026).

8. Официальная документация MySQL 8.0 [Электронный ресурс]. - Режим доступа: https://dev.mysql.com/doc/refman/8.0/en/ (дата обращения: 22.09.2026).

9. OWASP Top 10 - 2021: The Ten Most Critical Web Application Security Risks [Электронный ресурс]. - Режим доступа: https://owasp.org/Top10/ (дата обращения: 25.09.2026).

10. Приказ Министерства просвещения Российской Федерации от 17.05.2021 № 253 «О внесении изменений в ФГОС СПО» // СПС «КонсультантПлюс».

11. Федеральный государственный образовательный стандарт среднего профессионального образования по специальности 09.02.07 «Информационные системы и программирование», утверждённый приказом Минпросвещения России от 09.12.2016 № 1547.

12. Национальная программа «Цифровая экономика Российской Федерации», утверждённая Указом Президента Российской Федерации от 07.05.2018 № 204.

13. Федеральный закон от 27.07.2006 № 152-ФЗ «О персональных данных».

14. Дронов В. А. JavaScript и Node.js: разработка серверных приложений. - Санкт-Петербург : Питер, 2022. - 512 с.

15. Фленов М. Е. PHP и MySQL. Библия программиста. - 3-е изд. - Санкт-Петербург : БХВ-Петербург, 2021. - 768 с.