

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук
Кафедра информационных и управляющих систем
Направление подготовки 09.04.04 - Программная инженерия
Направленность (профиль) образовательной программы Управление разработкой программного обеспечения

ДОПУСТИТЬ К ЗАЩИТЕ
Зав. кафедрой
_____ А.В. Бушманов
«__» _____ 2025 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему: Проектирование и реализация веб-сайта для обмена информацией
внутри кампуса на основе РНР

Исполнитель Студент группы 3105-ом2	_____	Чжан Ихэ
	(подпись, дата)	
Руководитель профессор, доктор техн. наук	_____	И.Е. Ерёмин
	(подпись, дата)	
Руководитель научного содержания программы магистратуры профессор, доктор техн. наук	_____	И.Е. Ерёмин
	(подпись, дата)	
Нормоконтроль инженер	_____	В.Н. Адаменко
	(подпись, дата)	
Рецензент доцент, канд. техн. наук	_____	С.Г. Самохвалова
	(подпись, дата)	

Благовещенск, 2025

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук
Кафедра информационных и управляющих систем

УТВЕРЖДАЮ
Зав. кафедрой
_____ А.В. Бушманов
«___» _____ 2025 г.

З А Д А Н И Е

К магистерской диссертации студента группы 3105–ом2

Чжан Ихэ

Тема магистерской диссертации: Проектирование и реализация веб-сайта для
обмена информацией внутри кампуса на острове РНР

(Утверждено приказом от 06.03.25 № 609-уч)

1. Срок сдачи студентом законченной работы (проекта): 10.06.2025
2. Исходные данные к магистерской диссертации: документация разработчиков,
интернет ресурсы, учебная литература
3. Содержание магистерской диссертации (перечень подлежащих разработке
вопросов): анализ осуществимости, выбор программного обеспечения, проекти-
рование алгоритма решения задачи
4. Перечень материалов приложения (наличие чертежей, таблиц, графиков,
схем, программных продуктов, иллюстративного материала и т.п.): _____
5. Рецензент магистерской диссертации: Самохвалова С.Г., доцент, канд. техн.
наук
6. Дата выдачи задания 29.01.25
7. Руководитель выпускной квалификационной работы : _____

И.Е. Ерёмин, профессор, доктор техн. наук
(фамилия, имя, отчество, должность, уч.степень, уч.звание)

Заявление принял к исполнению _____

РЕФЕРАТ

Магистерская диссертация содержит 88 страниц, 26 рисунков, 40 источников

ОБМЕН ИНФОРМАЦИЕЙ ОБ УЧАЩИХСЯХ, ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, АРХИТЕКТУРА СИСТЕМЫ, ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ, УПРАВЛЕНИЕ ИНФОРМАЦИЕЙ

С быстрым развитием информационных технологий управление и обмен информацией о студентах становятся все более важными, а информатизация сферы образования постоянно продвигается, особенно в управлении информацией о студентах, цифровые и сетевые решения стали важной тенденцией. Совместное использование информации не только повышает эффективность управления, но и предоставляет ученикам, учителям и школьной администрации более удобные и интеллектуальные услуги, обеспечивая при этом безопасность информации. В данной статье рассматривается процесс разработки и внедрения программного обеспечения под названием «Проектирование программного обеспечения для обмена информацией между учащимися». Вначале анализируются потребности в обмене информацией о студентах, затем предлагается схема проектирования системы, включая функциональные модули системы, техническую архитектуру, проектирование базы данных и другие аспекты, и, наконец, подробно рассматривается реализация системы. Разработка системы направлена на повышение эффективности и качества работы школы по управлению и обмену информацией об учениках.

СОДЕРЖАНИЕ

Введение	5
1 Характеристика системы обмена информацией в кампусе	7
1.1 Анализ текущей ситуации с веб-сайтами обмена данными в кампусах	7
1.2 Анализ осуществимости	13
1.3 Существующие решения	15
2 Алгоритмическое и программное обеспечение решения задачи	21
2.1 Предлагаемый алгоритм компьютеризированного решения задачи	21
2.2 Обзор возможностей профильного программного обеспечения	27
2.3 Характеристика выбранного программно-технического обеспечения	34
3 Программная реализация предлагаемого алгоритма решения задачи	45
3.1 Основные этапы практической разработки программного продукта	45
3.2 Примеры фактического тестирования программного продукта	72
Заключение	83
Библиографический список	84

ВВЕДЕНИЕ

В информационный век пользователи могут получить доступ к более значительному объему ресурсов через Интернет за короткий промежуток времени, а процесс получения доступа является относительно простым и быстрым. Для распространения и получения многих материалов и информации одноклассникам в кампусе необходима централизованная платформа, где они могут войти в зарегистрированный аккаунт, посмотреть, что происходит в кампусе, найти информацию, которую они хотят получить, или опубликовать информацию, которой они хотят поделиться. Обеспечить быстрое и эффективное распространение информации в школьной среде.

С быстрым развитием информационных технологий сфера образования претерпевает глубокие изменения. Традиционная модель образования постепенно трансформируется в цифровую и сетевую, особенно при поддержке Интернета, распространение и обмен знаниями становятся быстрее и эффективнее. В связи с этим растет и потребность в обмене информацией и коммуникации между школами и студентами. Сайты обмена информацией в кампусе стали важной платформой для связи между студентами, преподавателями и другими членами кампуса.

Основная цель веб-сайта по обмену информацией в кампусе предоставить удобную, эффективную и общую платформу знаний для членов кампуса. Он не только позволяет студентам, преподавателям и сотрудникам кампуса обмениваться различными академическими ресурсами, такими как учебные материалы, конспекты, курсовые и дипломные работы, но и способствует трансграничному обмену между различными дисциплинами, помогая студентам лучше справляться с академическими проблемами и улучшать результаты обучения.

В то же время создание сайта обмена данными в кампусе не ограничивается только обменом знаниями по учебникам, но и способно в определенной степени разрешить путаницу и проблемы, с которыми сталкиваются студенты в процессе.

Благодаря богатым учебным ресурсам на платформе студенты могут учиться и проводить исследования более самостоятельно, а преподаватели могут использовать эти ресурсы для предоставления большего количества учебных материалов в классе, обогащения содержания образования и повышения качества преподавания.

Кроме того, сайт обмена информацией в кампусе может также повысить интерактивность сообщества в кампусе. Благодаря таким интерактивным функциям, как комментирование, лайки и обсуждения, сайт не только способствует обмену ресурсами, но и стимулирует столкновение идей и дух сотрудничества между студентами. Такое взаимодействие не только способствует распространению и влиянию знаний, но и придает новый импульс формированию культуры кампуса.

Поэтому мы выбрали PHP в качестве языка фоновой разработки, использовали базу данных MySQL для создания системы фонового управления веб-сайтом совместного использования данных кампуса, реализовали управление информацией системного администратора, обычных пользователей и т.д., управление ресурсами системным администратором, управление правами, управление аудитом, управление пользователями, управление загруженными ресурсами и распределение прав суперадминистратором системного администратора. Пользователь проверяет, ищет и публикует данные. Система содержит базовый логин пользователя и управление ресурсами.

1 ХАРАКТЕРИСТИКА СИСТЕМЫ ОБМЕНА ИНФОРМАЦИЕЙ В КАМПУСЕ

1.1 Анализ текущей ситуации с веб-сайтами обмена данными в кампусах

В информационную эпоху пользователи могут получить доступ к большому количеству ресурсов через Интернет в течение короткого периода времени, а процесс получения доступа является более простым и быстрым. Для студентов в кампусе требуется централизованная платформа для распространения и получения многих данных и информации, где студенты могут войти в зарегистрированные учетные записи, посмотреть, что происходит в кампусе, найти информацию, которую они хотят получить, или опубликовать информацию, которой они хотят поделиться. Обеспечение быстрого и эффективного распространения информации в школьной среде.

Поэтому мы выбрали PHP в качестве языка разработки бэкенда и использовали базу данных MySQL для создания системы управления бэкендом сайта обмена информацией в кампусе, которая реализует управление информацией, такой как системные администраторы и обычные пользователи, осуществляемое супер администраторами системы, управление ресурсами системными администраторами, управление привилегиями системных администраторов супер администраторами, управление аудитами, управление пользователями, управление загруженными ресурсами, и распределение привилегий. Пользователи получают доступ, осуществляют поиск и публикацию информации, система содержит основные функции входа пользователей в систему, управления ресурсами.

В эпоху стремительного развития информационных технологий Интернет предоставляет пользователям беспрецедентные удобства, позволяя им получать доступ к огромному количеству ресурсов за короткий промежуток времени.

Особенно в условиях студенческого городка потребности студентов в информации становятся еще более разнообразными и частыми. В таких условиях

особенно важно создать централизованную платформу для быстрого распространения и доступа к информации.

Централизованная платформа может значительно повысить эффективность распространения информации. В кампусе платформа может служить интегрированным информационным центром, объединяющим различные уведомления, расписания мероприятий, академические ресурсы и другую важную информацию университета. Таким образом, студентам не нужно переключаться между несколькими каналами, а достаточно зайти на одну платформу, чтобы легко и быстро получить всю необходимую информацию. Такое централизованное распространение информации не только повышает ее доступность, но и снижает вероятность пропуска информации.

Благодаря единой системе учетных записей студенты могут настраивать поток информации в соответствии со своими индивидуальными потребностями. Каждый пользователь может зарегистрироваться и войти в свою учетную запись, чтобы получать персонализированные уведомления и рекомендации. Например, студенты могут сосредоточиться на интересующих их областях знаний и получать последние новости о научных исследованиях; те, кто участвует в мероприятиях на кампусе, могут получать информацию о них в режиме реального времени. Помимо доступа к информации, платформа также позволяет пользователям публиковать информацию и делиться ею. Это очень полезно для студентов, которые хотят распространять свои взгляды, делиться опытом или организовывать мероприятия в кампусе. С помощью функции публикации студенты могут создавать объявления, организовывать мероприятия, делиться учебными материалами и т. д., способствуя взаимодействию и сотрудничеству внутри кампуса. Такой двусторонний поток информации не только расширяет возможности общения и сотрудничества в кампусе, но и побуждает студентов к активному участию в жизни кампуса.

Чтобы обеспечить своевременность информации, платформа должна поддерживать функцию обновления в режиме реального времени. Все важные уведомления и обновления могут быть мгновенно отправлены соответствующим

пользователям, чтобы обеспечить своевременность и точность информации. В то же время платформа может предоставлять интерактивные функции, такие как комментарии, лайки и дискуссионные форумы, чтобы способствовать общению и обратной связи между пользователями. Этот интерактивный механизм помогает создать в кампусе атмосферу открытости и участия.

В централизованной информационной платформе безопасность и защита конфиденциальности пользовательских данных имеют не менее важное значение. Для обеспечения безопасности и конфиденциальности пользовательской информации платформа должна принять надежные меры безопасности, включая шифрование данных, управление правами и защиту конфиденциальности. Кроме того, платформа должна иметь функции резервного копирования и восстановления данных для предотвращения потери информации.

Во многих областях ИТ преодолели ограничения времени и пространства, обеспечив полный и быстрый доступ к информации со всего мира для всех, независимо от случая и времени суток. Они революционизируют среду, предмет, содержание и средства обучения, которые объединяют преподавание и обучение. Новое поколение в приобретении знаний и информации в методе отличается от предыдущей эпохи, студенты привыкли полагаться на сеть для обучения, использование сети может быть достигнуто путем совместного использования ресурсов, расширения доступа студентов к информационным каналам. Как в полной мере использовать ресурсы интернета для наших исследований и обучения, это совершенно новая проблема, будь то только в преподавании, идеологическом и моральном воспитании, или в управлении различными аспектами, будут новые события и ситуации, в которых происходят, что срочно заставит нас продолжать углубленное изучение, изучать исследования, чтобы решить.

Электронное обучение - это новый стиль обучения, новый способ обучения для студентов, чтобы создать больше возможностей, эта форма обучения для достижения свободы времени и места обучения, может повысить эффективность обучения, создать непринужденную среду обучения. В настоящее время большинство платформ основано на масштабном общении и обмене информацией,

что затрудняет студентам точное и быстрое получение информации и получение нужных материалов в кампусе. Поэтому мы проектируем и разрабатываем платформу обмена информацией в кампусе, основанную на преимуществах кампусной сети, чтобы предоставить студентам более удобные, быстрые и практичные каналы для получения и публикации информации, реализовать совместное использование ресурсов в кампусе, улучшить общение между студентами и лучше способствовать обучению и освоению знаний.

Стремительное развитие интернет-технологий оказало глубокое влияние на сферу образования. Современные технологии облегчили и ускорили доступ к образовательным ресурсам и изменили традиционный способ обучения. Ниже рассматриваются некоторые особенности этих изменений:

Интернет преодолел временные и пространственные ограничения традиционного обучения, позволив учащимся получать доступ к ресурсам знаний по всему миру в любое время и из любого места. Благодаря такому подходу обучение больше не ограничивается физическими аудиториями и фиксированным временем занятий. Например, благодаря таким платформам, как онлайн-библиотеки, электронные журналы и открытые курсы, студенты могут в любое время получать доступ к учебным материалам, смотреть образовательные видео и даже участвовать в интерактивных онлайн-курсах.

Еще один важный аспект веб-технологий заключается в том, что они облегчают обмен ресурсами и совместное обучение. Студенты могут делиться учебными материалами и обмениваться опытом обучения с помощью онлайн-оплат форм в кампусе или глобальных учебных сообществ. Например, студенты могут использовать онлайн-платформу кампуса для загрузки и скачивания материалов курса, обмена заметками, обсуждения и обратной связи. Такой подход не только повышает эффективность обучения, но и улучшает взаимодействие и сотрудничество между студентами.

Хотя интернет предоставляет огромное количество информации, получение из него точных и полезных сведений остается сложной задачей. В условиях

кампуса разработка и развитие специальной платформы для обмена информацией может помочь студентам более эффективно находить необходимые материалы и информацию. Такие платформы могут включать репозитории курсов, зоны обмена академическими ресурсами, доски объявлений в кампусе и т.д., обеспечивая централизованный канал для доступа к информации и ее распространения и сокращая время и усилия, затрачиваемые на поиск информации.

Существует ряд технических и управленческих проблем, которые могут возникнуть при разработке и использовании этих платформ для обмена информацией в кампусе. Например, как обеспечить точность и безопасность информации и как предотвратить злоупотребление или неправомерное использование информации. Для решения этих проблем можно принять следующие меры:

Механизм аудита информации: установить процесс аудита для загрузки и публикации информации, чтобы гарантировать, что информация на платформе является точной и надежной.

Обучение и поддержка пользователей: обеспечить соответствующее обучение и поддержку, чтобы помочь студентам и преподавателям эффективно использовать платформу и избежать злоупотреблений.

Меры по защите конфиденциальности: примите соответствующие меры по защите данных, чтобы обеспечить безопасность личной информации пользователей.

Применение интернет-технологий должно быть направлено на содействие обучению и приобретению знаний. Благодаря эффективной сетевой платформе кампуса студенты не только получают доступ к более богатым учебным ресурсам, но и могут участвовать в создании и обмене знаниями. Такая интерактивность и участие могут дополнительно стимулировать интерес студентов к учебе и повышать их академические способности.

Таким образом, интернет-технологии совершают революцию в сфере образования, и благодаря их рациональному использованию можно значительно повысить эффективность и результативность обучения. Веб-платформа кампуса, как ключевой инструмент, может играть важную роль в обмене ресурсами,

управлении информацией и содействии обучению.

Интернет в настоящее время известен как крупнейшая в мире всеобъемлющая компьютерная и интернет-сеть, которая охватила весь земной шар, различные масштабные интернет-сети мира соединены в единое целое. В настоящее время люди неотделимы от сети, чтобы принести нам удобство, будь то в жизни или работе и учебе, сеть, чтобы принести нам удобство является замечательным. Сетевое обучение имеет много преимуществ для студентов: удобство, свобода, масштабируемость и расширяемость. На мой взгляд, знаниями нужно делиться, и ими стоит делиться. Делиться знаниями положительно по сравнению с любым другим обменом материалами, будь то делитель или делимый, их запас знаний только увеличивается, и нет возможности уменьшения знаний ни, с одной стороны, и этот рост будет быстро увеличиваться по мере расширения сферы обмена.

Например, если у меня в кармане три конфеты, и я делюсь одной с кем-то еще, то моих конфет становится две, поэтому мои собственные конфеты уменьшаются, когда я делюсь с другими, в то время как обмен знаниями отличается от закона сохранения энергии, независимо от того, с каким количеством людей вы делитесь, и независимо от того, каким объемом знаний вы делитесь, это всегда приведет к беспроигрышному явлению, когда дело доходит до запаса знаний. Кампус сайт обмена данными является платформой для школьных студенческих групп, чтобы создать платформу для студентов в школе студенты могут легко и быстро использовать сеть для поиска и обмена данными информации, обмена онлайн-ресурсов, чтобы встретиться со студентами на обмен знаниями, чтобы облегчить студентам в школе обучения, повысить интерес студентов к обучению и повысить их эффективность обучения.

Реализуйте пользовательскую часть сайта обмена профилями в кампусе. После входа на сайт студентам необходимо зарегистрировать учетную запись, после успешного входа они могут редактировать свои личные данные. В центре пользователя они могут размещать акции, просматривать мои акции, размещать

белую стену, размещать потерянные и найденные предметы, менять пароль и выходить из системы, а на главной странице они могут просматривать информацию, которой поделились другие студенты, учебные материалы, белую стену, информацию о потере предметов и сообщения от других студентов.

Внедрите систему фоновое управление сайтом обмена данными в кампусе. Пользователи системы управления делятся на супер-администраторов и обычных администраторов. Супер-администраторы могут осуществлять общее управление, управление правами, управление членством, добавлять, удалять и изменять информацию, опубликованную пользователями, а обычные администраторы могут управлять пользователями.

1.2 Анализ осуществимости

Проектирование и разработка этого проекта, начиная от средств разработки и заканчивая языками программирования, осуществляется с открытым исходным кодом, бесплатно, не требует никаких экономических затрат, проектные идеи и конструкторские решения обсуждаются самими разработчиками, чтобы прийти к результатам. Вся система от анализа требований до разработки и обслуживания стоит очень мало, аппаратная конфигурация компьютера и интеграция программного обеспечения не требуют высоких требований, для пользователя можно использовать компьютер, вход в систему и регистрация и весь процесс использования системы не требует затрат, разработка системы сайта обмена данными в кампусе может принести знания пользователю, но также не требует от пользователя экономических затрат.

С точки зрения языка разработки, PHP - это динамический скриптовый язык с низким пороговым уровнем, который быстро и легко изучается. В то же время PHP довольно мощный, 81,7 % публичных веб-сайтов в мире используют PHP, например, довольно известная зарубежная социальная сеть Facebook разработана на языке PHP. Как самый популярный язык для веб-разработки, PHP разработан на языке C, поэтому вы можете программировать расширения PHP на C и C++, чтобы увеличить производительность языка. PHP поддерживает почти все

популярные базы данных и операционные системы. PHP может выполнять динамические веб-страницы намного быстрее, чем cgi или Perl. PHP может выполнять динамические веб-страницы быстрее, чем cgi или Perl. По сравнению с другими языками программирования, PHP встраивает программу в HTML-документ, эффективность выполнения намного выше, чем у cgi, который полностью генерирует HTML-разметку, и большинство функций cgi могут быть реализованы PHP. Что касается базы данных, сайт обмена информацией в кампусе использует MySQL для хранения базы данных. MySQL занимает второе место в мире по использованию, имеет небольшой размер, высокую скорость, низкую общую стоимость владения, открытый исходный код и широко используется, поэтому обычно малые и средние сайты выбирают mysql в качестве базы данных. mysql community edition имеет отличную производительность, а с серверами PHP и Apache можно создать хорошую среду разработки. mysql также поддерживает различные функции cgi, которые могут быть реализованы с помощью PHP. MySQL также поддерживает различные механизмы хранения данных, поддерживает многопоточность, полностью использует ресурсы процессора и предоставляет различные пути подключения к базе данных, такие как TCP/IP, ODBC и JDBC.

Весь процесс разработки проекта находится в строгом соответствии с законом, фронтэнд и бэкэнд на стадии разработки и эксплуатации, используются подлинные инструменты разработки, свободный open-source фреймворк, все технические данные находятся в самостоятельном поиске и нахождении, проект включает информацию и контент также в рамках разрешенного законом. Таким образом, разработка данной платформы полностью осуществима технически, экономически и юридически.

Анализ целесообразности в основном подразделяется на экономическую целесообразность, техническую целесообразность, эксплуатационную целесообразность, юридическую целесообразность, в основном для того, чтобы эти пункты, проект должен сначала соответствовать национальным законам, первый

пункт особенно важен, не через законы страны, невозможно стать квалифицированным дизайнером; и после точки, чтобы рассмотреть технические аспекты, достаточно технических условий на основе того, что вы хотите сделать, чтобы сделать фактические результаты, полагаться на свои собственные Сделай сам, в сочетании с фактическими, больше посещений, чтобы собрать больше информации, практика делает реальные знания, только своими руками, только осуществимо; то стоимость производства, экономическая целесообразность, если это не крупномасштабный трудно достичь проекта, для общей системы не проблема, не появится, чтобы беспокоиться об экономических условиях путаницы; операционные аспекты практических соображений пользователя, с точки зрения пользователя дизайн системы функций, не может быть слишком сложным, опыт пользователя, как начальник опыта пользователя. Сложность, опыт пользователя в качестве основного критерия, система также должна быть сделана в четкой и краткой, легко понять.

1.3 Существующие решения

Анализ потребностей пользователей является ключевым звеном в создании успешного сайта обмена кампусом. Он предназначен для тщательного проектирования взаимодействия интерфейса системы и конкретных функций использования с точки зрения пользователя. Для достижения этой цели необходимо провести практические исследования и собрать отзывы и потребности реальных пользователей, таких как учащиеся школ, чтобы в конечном итоге завершить разработку пользовательских функций сайта обмена кампусом.

Во - первых, разделение полномочий для различных ролей пользователей важно. Когда пользователь не входит в систему, он относится к роли туриста. В этом состоянии посетители могут зайти только на главную страницу веб - страницы и не могут увидеть конкретное содержание веб - сайта. Это ограничение призвано побудить пользователей регистрироваться и регистрироваться, чтобы получить более широкий функциональный опыт.

Когда студенты входят в систему, они становятся зарегистрированными

членами, и в это время они могут просматривать богатый контент на сайте. Зарегистрированные члены имеют ряд практических функций, одна из которых заключается в том, что они могут редактировать личные данные, включая, но не ограничиваясь, такие сведения, как аватары, прозвища, контактные данные, а также могут изменять пароли для обеспечения безопасности учетной записи.

Кроме того, зарегистрированные члены могут публиковать учебные материалы, будь то самостоятельно организованные заметки, курсовые задания или собранные учебные ресурсы, которые могут быть переданы другим одноклассникам. В то же время они также могут просматривать учебные материалы, которыми делятся другие пользователи, расширять свои горизонты обучения и осуществлять совместное использование и обмен ресурсами.

Функция сообщения также является одним из важных прав зарегистрированных членов. Они могут оставлять сообщения, взаимодействовать с другими пользователями, делиться опытом обучения, задавать вопросы или обсуждать различные темы. В то же время зарегистрированные члены также могут просматривать сообщения других пользователей, участвовать в дискуссиях сообщества и создавать позитивную атмосферу обучения.

Функция стены добавляет романтику и жизненную силу в жизнь кампуса. Зарегистрированные члены могут просматривать стену признания публикации и смело выражать свои эмоции, будь то комплименты одноклассникам, благодарность учителю или любовь к жизни в кампусе, которые могут быть переданы через стену признания.

Кроме того, зарегистрированные члены могут просматривать и публиковать объявления о пропаже или поиске. Если вы случайно потеряли предметы, вы можете своевременно опубликовать объявление о поиске предметов, надеясь, что другие одноклассники смогут предоставить подсказки; Если вы найдете предметы, потерянные другими людьми, вы также можете опубликовать информацию о находке, чтобы помочь владельцу вернуть вещи. Эта функция отражает дух взаимопомощи сообщества кампуса и облегчает жизнь студентов в кампусе.

Анализ потребностей администраторов играет решающую роль в дизайне

совместного веб-сайта кампуса, который тщательно разрабатывает системные интерфейсы взаимодействия и конкретные функции использования с точки зрения менеджера. Основная ответственность администратора заключается в управлении данными системы и поддержании безопасности системы, чтобы обеспечить стабильную и эффективную работу сайта обмена в кампусе и предоставить пользователям хороший опыт обслуживания.

Во - первых, регулярное управление является одной из основных обязанностей администратора. Что касается управления системой, администраторы должны контролировать и поддерживать всю систему, чтобы обеспечить стабильность и надежность системы. Это включает в себя регулярную проверку работоспособности системы, устранение системных сбоев, оптимизацию производительности системы и так далее. Для управления вложениями администратор должен убедиться, что загруженные вложения соответствуют установленным требованиям к формату и размеру и своевременно очищают ненужные вложения, чтобы сэкономить пространство для хранения. В то же время администраторы должны управлять личными данными, в том числе обновлять свои контактные данные, изменять пароли и т. д. Для обеспечения своевременного реагирования на различные вопросы и потребности.

Управление правами является уникальной и важной функцией суперадминистратора. Суперадминистратор отвечает за управление всеми учетными записями и информацией администратора, обеспечивая разумную конфигурацию и эффективную работу команды администраторов. Это включает в себя создание новых учетных записей администратора, присвоение различных уровней разрешений, изменение информации администратора и т.д.

Строгое управление правами эффективно предотвращает несанкционированный доступ и операции и гарантирует безопасность системы.

Управление членством также является одной из важных задач администратора. Администратор может редактировать права учетной записи пользователя и разумно корректировать уровень прав пользователя в соответствии с поведением

и потребностями пользователя. Например, для некоторых пользователей- нарушителей использование части их функций может быть ограничено; Пользователям, вносящим активный вклад, могут быть предоставлены дополнительные права и награды. Кроме того, администратор имеет право удалять учетные записи пользователей. Если пользователь нарушает правила веб - сайта или возникает аномалия, администратор может своевременно удалить учетную запись пользователя, чтобы поддерживать нормальный порядок на сайте.

Управление контентом является одной из основных обязанностей администратора. Администраторы должны полностью управлять информацией, сообщениями и т. д., публикуемыми пользователями. Что касается добавления, администраторы могут добавлять некоторые официальные учебные материалы, уведомления и другие материалы по мере необходимости, чтобы обогатить ресурсы веб - сайта. Что касается изменений, администратор может изменять неточный или неправильный контент, опубликованный пользователем, чтобы обеспечить точность и законность информации. Что касается удаления, то администраторы должны своевременно удалять некоторый контент с нарушениями, низким качеством или бесполезностью, чтобы поддерживать чистоту и высокое качество сайта. В то же время администраторы также должны управлять сообщениями, своевременно отвечать на вопросы и предложения пользователей, решать споры между пользователями и создавать гармоничную атмосферу общения.

Подводя итог вышесказанному, благодаря углубленному анализу потребностей администраторов и функциональному дизайну, сайт обмена кампусом может обеспечить безопасное управление данными и стабильную работу системы под эффективным управлением администраторов, предоставляя пользователям более качественные услуги.

Сайт обмена данными в кампусе действительно обладает высокой универсальностью и адаптируемостью к аппаратной среде. Веб - сайт не имеет особых требований к операционной среде, а это означает, что, будь то в компьютерной лаборатории школы, библиотеке или в различных местах, таких как студенче-

ское общежитие или кабинет учителя, обычные пользователи могут легко использовать веб - платформу обмена данными непосредственно после того, как различные клиенты браузера введут сайт. Будь то обычные браузеры, такие как Chrome, Firefox, Safari или другие, менее популярные, они лучше совместимы с веб - сайтом, обеспечивая пользователям последовательный опыт использования. В то же время фоновые системы проекта также не имеют специальных требований к оборудованию, что позволяет поддерживать и управлять веб - сайтом на более обычных аппаратных устройствах, снижая эксплуатационные расходы и технические пороги. Кроме того, этот проект адаптирован к ПК и полностью отвечает потребностям студентов и преподавателей в использовании компьютеров для поиска, обмена и обмена информацией в повседневной учебной и преподавательской работе, обеспечивая платформу для распространения знаний и обмена ресурсами в кампусе.

Что касается операционной системы, то выбор операционной системы Windows имеет множество преимуществ. Операционная система Windows имеет широкую пользовательскую базу по всему миру, удобный интерфейс, простую работу и легкую для большинства пользователей. В то же время система Windows обеспечивает богатую программную экосистему, которая хорошо совместима с различными инструментами разработки и системами управления базами данных. В кампусе многие школьные компьютерные устройства также широко оснащены операционной системой Windows, что позволяет сайтам обмена данными кампуса лучше интегрироваться с существующими школьными устройствами и облегчает их использование студентами и преподавателями.

Использование базы данных MySQL является разумным выбором. MySQL - это реляционная система управления базами данных с открытым исходным кодом, которая характеризуется стабильной производительностью, высокой надежностью и мощными функциями. Он может эффективно хранить и управлять большим количеством данных на вебсайте обмена данными кампуса, включая пользовательскую информацию, контент данных, комментарии и т. д. MySQL предлагает богатый язык запросов и функции управления данными для

удовлетворения различных потребностей веб - сайта в хранении и извлечении данных. В то же время возможности MySQL с открытым исходным кодом также позволяют настраивать и масштабировать его в соответствии с фактическими потребностями, обеспечивая большую гибкость в развитии веб - сайта.

Язык скриптов выбирает PHP версии 5.6 и выше, а PHP - язык скриптов на сервере, широко используемый в веб - разработке. Он имеет преимущества простой грамматики, высокой эффективности разработки и богатой функциональности. Версии PHP 5.6 и выше значительно улучшили производительность и безопасность, чтобы лучше соответствовать потребностям веб - сайта обмена данными в кампусе. PHP легко интегрируется с базами данных MySQL, что позволяет легко хранить и извлекать данные. В то же время PHP предлагает богатую библиотеку расширений и фреймворк, которые ускоряют разработку и улучшают обслуживание и масштабируемость кода. В разработке веб - сайта обмена данными в кампусе PHP может реализовать различные функции, такие как регистрация пользователей, загрузка и загрузка данных, управление комментариями сообщений и т. д. чтобы предоставить пользователям удобные услуги.

2 АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ РЕШЕНИЯ ЗАДАЧИ

2.1 Предлагаемый алгоритм компьютеризированного решения задачи

В данной дипломной работе «Разработка веб-сайта для обмена данными в кампусе на основе PHP» предлагаемый алгоритм решения компьютерных задач фокусируется на том, как эффективно решать такие ключевые проблемы, как загрузка файлов, управление пользователями, контроль прав и поиск информации. Ниже приводится подробное описание процесса разработки алгоритма:

Во-первых, для решения проблемы аутентификации пользователя разработан алгоритм проверки подлинности входа в систему на основе PHP и MySQL. Алгоритм проверяет личность пользователя, получая имя пользователя и пароль, введенные пользователем, и запрашивая базу данных. Процесс заключается в том, что, когда пользователь вводит имя пользователя и пароль, система запрашивает базу данных через SQL, чтобы проверить, существует ли имя пользователя, и сравнивает хэш-значение пароля. Если аутентификация пройдена, система создает сессию (session) или токен (token), чтобы пользователь мог войти в систему. Если аутентификация не прошла, возвращается сообщение об ошибке, в котором пользователю предлагается повторно войти или зарегистрировать учетную запись.

Во-вторых, для загрузки и выгрузки данных разработан поток обработки для загрузки и выгрузки файлов. В процессе загрузки система сначала проверяет формат и размер файла, загружаемого пользователем. Допустимыми форматами файлов могут быть PDF, Word и другие распространенные форматы документов, а размер устанавливается администратором системы в фоновом режиме. Файлы сохраняются в директории, указанной сервером с помощью функции загрузки файлов PHP, а метаданные файлов (например, имя файла, загрузчик, время загрузки и т. д.) записываются в базу данных. В процессе загрузки пользователь может выбрать файл для загрузки с помощью поиска или просмотра списка про-

файлей, а система предоставит ссылку для загрузки файла в соответствии с запросом пользователя, чтобы обеспечить нормальную передачу файла.

При разработке управления разрешениями алгоритм учитывает управление разрешениями для различных ролей пользователей, чтобы гарантировать, что только авторизованные пользователи могут получить доступ к определенным ресурсам. Каждый пользователь имеет различные права доступа в зависимости от своей роли (например, обычный пользователь, администратор или учитель). В системе, когда пользователь получает доступ к файлу, система проверяет, соответствует ли его роль правам доступа к файлу. Если совпадает, пользователь может загрузить или просмотреть файл; если нет, система укажет на недостаток прав и ограничит доступ. Администраторы обладают более высокими привилегиями и могут управлять правами доступа к файлам, а также выполнять операции по удалению или изменению файлов.

Управление базой данных - одна из основных частей системы. Чтобы обеспечить эффективную работу системы, база данных имеет разумную структуру, включая таблицу пользователей, таблицу файлов, таблицу типов файлов и т. д. Для оптимизации эффективности запросов в базе данных используется технология индексирования, которая позволяет значительно повысить скорость запросов, особенно при поиске данных. В то же время в системе предусмотрен механизм регулярного резервного копирования, позволяющий избежать риска потери данных. Учитывая масштабируемость системы, такие стратегии, как разделение таблиц и библиотек, позволяют справляться с растущим объемом данных.

При разработке веб-сайта для обмена информацией в кампусе на базе РНР основная проблема, которую должна решить система, заключается в обеспечении эффективной и надежной платформы для обмена информацией, которая позволяет пользователям загружать, скачивать и управлять различными типами ресурсов кампуса, такими как конспекты лекций, материалы курсов, внеклассное чтение и так далее. Цель системы - предоставить пользователям кампуса удобный способ распространения информации с помощью компьютерных технологий и сетевой архитектуры.

При разработке этого алгоритма особое внимание уделяется следующим аспектам:

- аутентификация и управление пользователями;
- механизм загрузки и скачивания файлов;
- управление файлами и индексирование;
- безопасность системы и контроль полномочий;
- поиск и рекомендация ресурсов;
- аутентификация пользователей и управление правами.

Система аутентификации пользователей - это первый этап создания веб-сайта для обмена данными в кампусе и основа всей системы. Чтобы обеспечить безопасность системы и простоту ее использования, необходимо разработать эффективный алгоритм аутентификации пользователей. Его основной процесс выглядит следующим образом.

Пользователь вводит необходимую регистрационную информацию (например, имя пользователя, адрес электронной почты, пароль и т. д.), а система проверяет, зарегистрирован ли адрес электронной почты. Пароль хранится в базе данных с использованием алгоритма шифрования (например, SHA256) для защиты пароля. После успешной регистрации информации система генерирует уникальный идентификатор пользователя и сохраняет его в базе данных (рисунок 1).

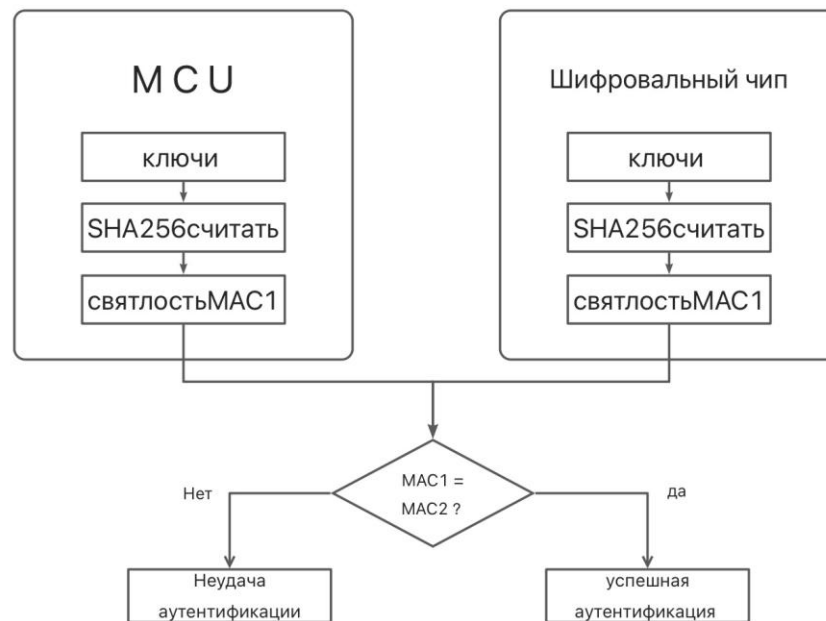


Рисунок 1 – Алгоритм шифрования SHA256

Вход пользователя в систему: после того как пользователь вводит идентификатор пользователя и пароль, система шифрует пароль и сопоставляет его с зашифрованным паролем в базе данных. Если имя пользователя и пароль успешно сопоставлены, система генерирует идентификатор сеанса пользователя с помощью механизма сеансов (например, реализованного с помощью технологии Session) и сохраняет статус входа пользователя.

Управление правами пользователя: Система устанавливает различные разрешения в зависимости от роли пользователя (например, администратор, обычный пользователь, учитель и т. д.). Права администратора включают управление всеми пользователями и ресурсами; обычные пользователи могут только загружать и скачивать материалы; преподаватели могут иметь дополнительные права на редактирование или рецензирование.

Загрузка и скачивание файлов - это основная функция сайта обмена данными, и алгоритм ее выполнения должен обеспечивать эффективность и безопасность работы. Ниже приведена идея дизайна этой части:

Загрузка файлов: в процессе загрузки файла система сначала проверяет тип файла и ограничение на размер, чтобы избежать нелегальной загрузки. Загруженный файл сохраняется в указанном каталоге на сервере, а соответствующая

информация о файле (например, имя файла, размер, загрузчик, время загрузки и т. д.) записывается в базу данных. Система проверяет хэш-значение файла, чтобы убедиться, что файл не был подделан в процессе загрузки.

Загрузка файлов: когда пользователь нажимает на ссылку для скачивания, система определяет, разрешить ли скачивание файла в соответствии с правами пользователя. При загрузке файлов будет использоваться технология потоковой загрузки для снижения нагрузки на сервер и увеличения скорости загрузки. Система будет регистрировать количество загрузок каждого файла и информацию о загрузчике для последующей статистики и анализа.

Для того чтобы облегчить пользователям поиск и управление загруженными материалами, необходимо разработать эффективную систему управления файлами и индексации. Конкретные алгоритмы разработаны следующим образом:

Индексирование файлов: система будет автоматически классифицировать и помечать загруженные файлы, например, по типу файла (документ, PPT, видео и т. д.), теме, курсу и т. д. Каждому файлу будет присвоен уникальный идентификатор для облегчения последующих операций (например, удаление, загрузка, комментирование и т. д.).

Функция поиска файлов позволяет пользователям быстро находить нужную информацию по ключевым словам. Алгоритм поиска использует технологию инвертированного индексирования, чтобы проиндексировать название, описание, теги и т. д. файла для повышения эффективности поиска. Результаты поиска сортируются по релевантности, пользователи могут отфильтровать их по времени загрузки файла, количеству скачиваний и т. д.

Пользователи могут управлять своими загруженными файлами, например, редактировать описания, обновлять файлы, удалять их и выполнять другие операции. Система должна гарантировать, что пользователи могут управлять только своими файлами, и открывать администраторам более широкие права на управление.

Безопасность системы - важный момент при разработке любой веб-платформы, особенно когда речь идет о загрузке файлов и пользовательских данных. Ниже перечислены основные алгоритмы проектирования безопасности.

Предотвращение SQL-инъекций: все операции с базами данных в системе должны использовать подготовленные запросы (Prepared Statements, PS) для предотвращения атак SQL-инъекций. Строгая фильтрация и валидация пользовательского ввода во избежание внедрения вредоносного кода (рисунок 2).

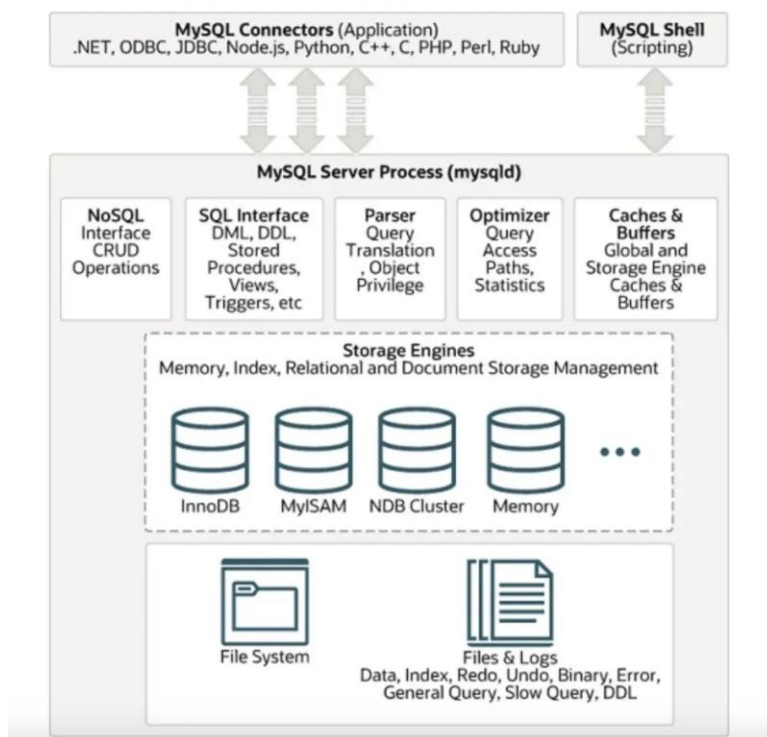


Рисунок 2 – Диаграмма рабочего процесса для индексов SQL

Безопасность загрузки файлов: загружаемые файлы проверяются на тип и содержание, чтобы убедиться, что файлы, загружаемые пользователями, не являются вредоносными скриптами. Файлы, хранящиеся на сервере, не должны быть напрямую доступны пользователям, и система должна использовать динамические ссылки для загрузки файлов, чтобы избежать прямого доступа.

Предотвращайте атаки CSRF и XSS: используйте механизмы токенов, такие как CSRF Token, для предотвращения атак подделки межсайтовых запросов. HTML-экранирование всех вводимых пользователем данных для предотвращения атак межсайтового скриптинга (XSS).

Шифрование и защита данных: протокол SSL/TLS используется для шифрования связи между пользователями и серверами. Все конфиденциальные данные (например, пароли, личная информация и т. д.) должны храниться с использованием шифрования.

Для улучшения пользовательского опыта система должна обладать определенным интеллектуальным рекомендательным механизмом, который поможет пользователям быстрее находить нужную информацию. Конкретные алгоритмы разработаны следующим образом:

Алгоритм рекомендаций: рекомендация на основе содержания: система рекомендует соответствующие материалы на основе содержания файлов, загруженных или просмотренных пользователем. Рекомендация на основе коллаборативной фильтрации: на основе поведения других пользователей (таких как загрузка, рейтинг) прогнозируются ресурсы, которые могут заинтересовать текущего пользователя.

Отображение «горячих» и новых файлов: система формирует список «Горячих файлов» на основе количества скачиваний, комментариев и других показателей для ознакомления пользователей. Недавно загруженные файлы отображаются на главной странице, чтобы побудить пользователей просмотреть последние ресурсы.

Персонализированные рекомендации: система анализирует историческое поведение пользователя (например, загрузку, скачивание, поиск записей), чтобы настроить список персонализированных рекомендаций и повысить удовлетворенность пользователей.

2.2 Обзор возможностей профильного программного обеспечения

Во-первых, основной функцией сайта является загрузка и скачивание материалов. Пользователи могут загружать свои учебные материалы на платформу сайта по мере необходимости, в форматах файлов, включая документы, изображения, аудио и видеоматериалы, чтобы обеспечить широкий спектр типов материалов. Для повышения удобства пользователей процесс загрузки разработан

простым и интуитивно понятным: пользователям достаточно выбрать файл и заполнить необходимую информацию о метке, чтобы завершить загрузку. Загрузка данных - еще одна важная функция сайта, которая позволяет пользователям просматривать и загружать данные, загруженные другими пользователями, чтобы получить необходимую информацию. При разработке функции загрузки данных учитывается размер файла и скорость загрузки, а также оптимизируется внутренний код, чтобы обеспечить пользователям быстрый и стабильный доступ к данным.

Во-вторых, платформа предоставляет пользователям мощную функцию поиска. Для того чтобы пользователи могли быстро находить нужные им ресурсы в огромном объеме информации, в системе поиска используются как сопоставление ключевых слов, так и фильтрация по тегам. Пользователям достаточно ввести ключевые слова или выбрать подходящие теги, и система быстро выдаст результаты поиска по соответствующим материалам. Кроме того, система поддерживает сортировку по типу файла, времени загрузки, релевантности и другим параметрам, что еще больше повышает эффективность поиска. Эта функция значительно сокращает временные затраты пользователей на поиск информации, позволяя им быстро находить ценные ресурсы.

Комментарии и интерактивные функции также являются одной из изюминок дизайна сайта. В нижней части страницы каждого ресурса есть раздел комментариев, который позволяет пользователям оценить содержание ресурса, высказать собственное мнение или задать вопросы. Благодаря функции комментариев платформа обеспечивает интерактивность и совместное использование академических ресурсов, а также способствует обмену знаниями между пользователями. Пользователи также могут просматривать комментарии других людей, что помогает им лучше понять содержание материалов или сделать выводы о качестве материалов. Для предотвращения спама или вредоносного поведения в системе предусмотрен механизм проверки содержания комментариев, обеспечивающий доброжелательное взаимодействие на платформе.

Управление разрешениями - ключевая часть безопасности сайта и управления пользователями. Чтобы гарантировать безопасность данных, в системе разработан строгий механизм управления разрешениями. Пользователи получают различные разрешения в зависимости от своего статуса. Обычные пользователи могут только просматривать и скачивать информацию, в то время как администраторы имеют право управлять информацией, удалять неподходящий контент, запрещать оскорбление пользователей и т. д. Система также предоставляет базовые функции, такие как регистрация учетной записи, вход в систему и смена пароля, что обеспечивает защиту конфиденциальности пользователей и безопасность данных. Кроме того, администраторы могут управлять статусом работы платформы через внутреннюю систему, просматривать статистику поведения пользователей, записи о загрузке данных и другую информацию, эффективно обеспечивая безопасность и стабильную работу платформы (рисунок 3).

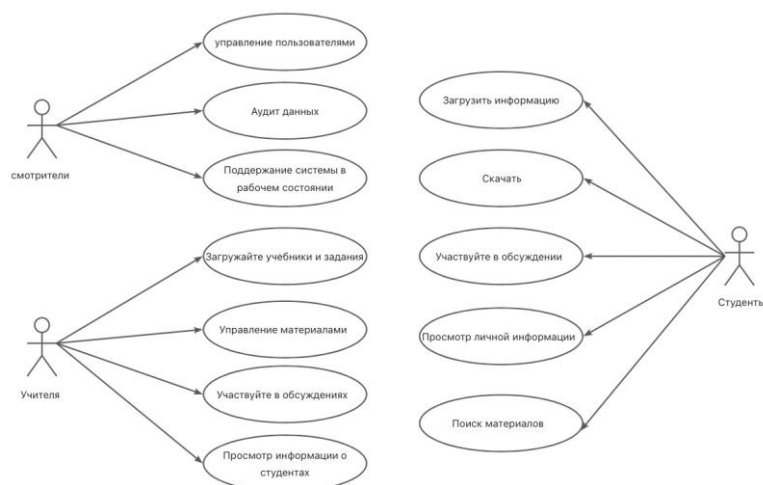


Рисунок 3 – Диаграмма вариантов использования

Управление пользователями - одна из основных функций сайта обмена профилями в кампусе. С помощью этого модуля администраторы могут управлять информацией о пользователях, включая регистрацию, вход, изменение пароля, отключение учетной записи и так далее. Различные типы пользователей, такие как студенты, преподаватели и администраторы, будут иметь разные разрешения, чтобы обеспечить безопасность системы и конфиденциальность информации.

Регистрация и вход в систему: пользователи могут зарегистрироваться, введя такую информацию, как номер студента, номер преподавателя, имя пользователя и т. д. После успешной регистрации они могут войти в систему. Система проверяет легитимность информации о пользователе с помощью PHP-скриптов для обеспечения уникальности каждой учетной записи пользователя.

Управление правами учетной записи: в зависимости от типа пользователя система автоматически назначает различные права. Например, пользователь-студент может только загружать и скачивать данные, а пользователь-администратор может выполнять операции по управлению другими пользователями, изменять информацию о данных, удалять несоответствующий контент и т. д.

Управление личной информацией: пользователи могут просматривать и редактировать личную информацию в своей учетной записи, например, имя пользователя, контактную информацию, специальность и т. д.

Модуль загрузки и скачивания информации - одна из основных функций сайта, которая поддерживает загрузку файлов различных форматов (например, PDF, Word, изображений и т. д.), и с помощью этого модуля пользователи могут скачивать общие материалы в соответствии с их правами.

Функция загрузки данных: пользователи загружают свои собственные учебные материалы, учебные программы, контрольные работы и т.д. через интерфейс, предоставляемый платформой, и загруженные файлы сохраняются в указанной директории на сервере. Для повышения эффективности и успешности загрузки файлов система может устанавливать ограничения на размер и формат файлов.

Классификация файлов и управление метками: загруженные материалы будут классифицированы по курсам, дисциплинам и т. д., что удобно для пользователей, которые могут получить их в соответствии с различными потребностями. В то же время пользователи могут устанавливать соответствующие метки (например, «Математика», «База данных») для файлов, чтобы повысить точность поиска данных.

Загрузка и просмотр: пользователи могут загружать необходимые данные

в соответствии со своими правами, в то время как администраторы имеют право просматривать и загружать все файлы, загруженные пользователями. При загрузке система будет генерировать статистику загрузки, чтобы помочь администраторам понять, как используется информация.

Чтобы облегчить пользователю поиск нужной информации, в системе реализованы мощные функции поиска и фильтрации, поддерживающие различные способы поиска, включая поиск, по ключевым словам, классификационную фильтрацию и фильтрацию по меткам.

Пользователи могут вводить ключевые слова (например, название курса, имя файла и т. д.), чтобы быстро найти необходимую информацию. Результаты поиска сортируются по релевантности, поэтому пользователям легче найти наиболее подходящие документы.

Фильтрация по категориям и тегам: система предоставляет множество критериев фильтрации, пользователи могут сузить поиск, выбрав категорию курса, категорию предмета или тег, чтобы помочь пользователям найти необходимые ресурсы более точно.

Функция расширенного поиска: для дальнейшего повышения эффективности поиска система также предоставляет функцию расширенного поиска, которая поддерживает комбинированные запросы по нескольким критериям, например, одновременный выбор нескольких дисциплин или типов файлов, что позволяет быстро находить необходимую информацию.

Этот модуль предоставляет пользователям функцию оценки и комментирования материалов, чтобы помочь другим пользователям лучше понять качество и полезность материалов. Благодаря этой функции студенты и преподаватели могут взаимодействовать друг с другом и делиться опытом использования материалов.

Каждый загруженный материал может быть оценен другими пользователями в диапазоне от 1 до 5. Пользователи могут оценивать качество материала, полноту содержания и его полезность.

Помимо выставления оценок, пользователи могут комментировать материалы и выражать свое мнение или опыт. Комментарии помогут другим пользователям понять качество материала, прежде чем скачивать его.

Отзывы и сообщения: если пользователи обнаружат, что у загруженных материалов есть проблемы, такие как нарушение авторских прав, неприменимость или низкое качество, они могут решить их с помощью функций отзывов и сообщений, предоставляемых платформой. Администратор рассмотрит сообщения о материалах и примет соответствующие меры.

Для улучшения взаимодействия и общения между пользователями в системе предусмотрены функции социального взаимодействия, включая обмен мгновенными сообщениями, форумы и т. д.

Функция мгновенного обмена сообщениями: пользователи могут общаться друг с другом через систему мгновенного обмена сообщениями платформы, чтобы обсудить, как использовать материалы или поделиться опытом обучения. Система обеспечивает отображение статуса онлайн и офлайн для облегчения контакта между пользователями.

Система имеет дискуссионный форум или форум, пользователи могут размещать вопросы, учебный опыт или обсуждение курса. С помощью этого модуля преподаватели и студенты могут обмениваться знаниями и взаимодействовать на платформе.

Пользователи могут добавлять других пользователей в качестве друзей и просматривать их динамику в любое время, чтобы повысить сплоченность сообщества.

Модуль управления бэк-офисом предоставляет администратору доступ к комплексным функциям системы управления, с помощью которых администратор может поддерживать систему, управлять пользователями и анализировать данные.

Управление пользователями: администратор может просматривать информацию обо всех зарегистрированных пользователях, управлять состоянием их учетных записей и устанавливать права пользователей.

Управление данными: администратор может управлять загруженными данными, в том числе просматривать загруженные файлы, удалять несоответствующее содержимое, изменять классификацию данных и т. д.

Настройки и конфигурация системы: администраторы могут настраивать систему, в том числе устанавливать ограничения на загрузку файлов, настраивать функции комментариев и оценок, изменять тему сайта и т. д.

Статистика и отчетность: система генерирует различные виды отчетов, включая статистику загрузки и скачивания материалов, анализ активности пользователей, анализ комментариев и оценок и т. д., чтобы помочь администраторам понять использование сайта и потребности пользователей (рисунок 4).

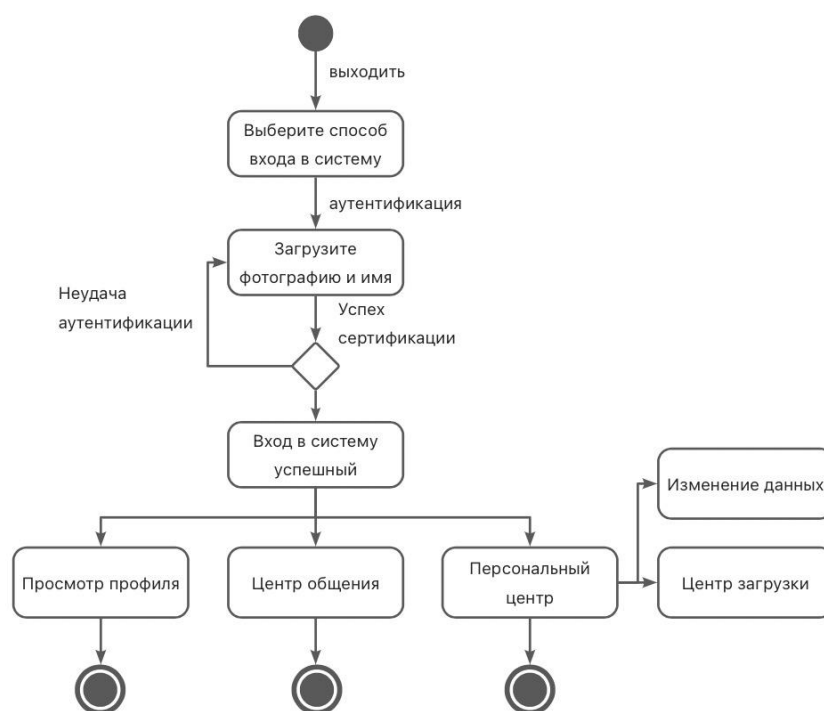


Рисунок 4 – Карта деятельности

Для обеспечения безопасности системы, особенно защиты конфиденциальности пользователей и их данных, модуль безопасности и контроля полномочий имеет решающее значение. Он обеспечивает стабильную работу платформы за счет разумного распределения прав и политики безопасности.

Контроль привилегий: привилегии каждого пользователя в системе строго

определены, например, обычные пользователи могут только загружать и скачивать данные, в то время как администраторы имеют все привилегии управления. Контроль прав позволяет эффективно предотвратить злоупотребления.

Шифрование данных: система шифрует личную информацию пользователей, загруженные файлы и другие важные данные, чтобы избежать незаконного доступа к конфиденциальной информации.

Механизм защиты от атак: система реализует меры защиты от распространенных уязвимостей безопасности, таких как SQL-инъекции и XSS-атаки, чтобы обеспечить безопасность веб-сайта и целостность пользовательских данных (рисунок 5).



Рисунок 5 – диаграмма последовательности

2.3 Характеристика выбранного программно-технического обеспечения

Для данного проекта основным выбором для разработки программного обеспечения является PHP - язык серверных сценариев, широко используемый в

веб-разработке, главным преимуществом которого является его простота, гибкость и высокая эффективность, особенно подходящая для разработки динамических веб-страниц и сложных веб-приложений. Будучи языком с открытым исходным кодом, PHP не только обладает высокой расширяемостью, но и имеет активное сообщество разработчиков, что позволяет им быстро получить доступ к богатой технической поддержке и разнообразным инструментам с открытым исходным кодом. Эта особенность не только значительно ускоряет процесс разработки, но и позволяет проекту идти в ногу с технологическим развитием, а также повышает стабильность и безопасность системы за счет ресурсов и решений, которыми делится сообщество.

Кроссплатформенность PHP - еще одно значительное преимущество. На таких операционных системах, как Windows, Linux или Mac OS, PHP работает эффективно, что очень важно для создания сайта обмена данными в кампусе. Поскольку при развертывании веб-сайты могут сталкиваться с требованиями различных операционных систем и сред, кроссплатформенная поддержка PHP позволяет проектам свободно переключаться между различными платформами, что снижает сложность разработки, эксплуатации и обслуживания. В то же время, кроссплатформенность позволяет гарантировать масштабируемость будущей системы, легко переносить ее на другие операционные системы или облачные платформы по мере необходимости (рисунок 6).

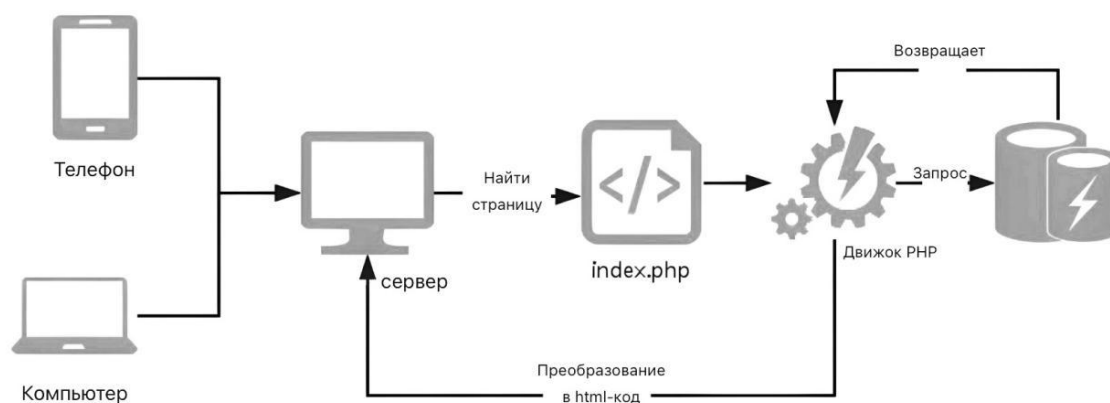


Рисунок 6 – Диаграмма рабочего процесса PHP

Кроме того, данный проект сочетает в себе мощную комбинацию PHP и базы данных MySQL, которая является очень распространенным и эффективным решением в веб-разработке. MySQL, как высокопроизводительная реляционная система управления базами данных, может эффективно хранить и управлять большим количеством пользовательских данных, информации о файлах, записей о загрузках и т.д., благодаря своим отличным возможностям обработки данных, что удовлетворяет высоким требованиям к хранению данных и доступу к ним на сайте обмена данными кампуса. Поддержка обработки транзакций, целостности данных, ограничений по внешним ключам и других расширенных функций MySQL позволяет обеспечить согласованность и безопасность данных в сложных бизнес-сценариях, избежать сбоев системы или ошибок, вызванных аномалиями данных.

Что касается безопасности данных, то сочетание PHP и MySQL обеспечивает надежное резервное копирование и восстановление данных. Когда пользователи посещают сайт, обработка данных в реальном времени и вычисления бизнес-логики выполняются с помощью PHP-скриптов, а MySQL отвечает за эффективное хранение и управление данными при масштабном доступе. Благодаря разумному проектированию структуры базы данных и индекса данных можно значительно повысить скорость поиска данных и гарантировать удобство работы пользователей. В то же время механизмы безопасности MySQL, такие как шифрованное хранение, управление правами, аутентификация пользователей и т. д., позволяют эффективно предотвращать утечку данных и неправомерный доступ, обеспечивая безопасность системы.

Сочетание PHP и MySQL обеспечивает стабильную, эффективную и гибкую техническую основу для данного проекта, что дает мощную поддержку для разработки и эксплуатации сайта обмена данными в кампусе. Такое сочетание технологий не только обеспечивает быструю разработку и гибкое развертывание, но и гарантирует стабильность и масштабируемость системы в долгосрочной эксплуатации, предоставляя широкое пространство для будущего расширения функциональности и оптимизации производительности.

Внешняя часть сайта была разработана с использованием HTML, CSS и JavaScript для обеспечения хорошего пользовательского опыта и эффективного взаимодействия с интерфейсом. В частности, применение HTML5 позволяет сайтам не только отображать структурированный контент, но и добиваться более мощной функциональной поддержки за счет новых тегов и API. Например, использование тегов `<video>` и `<audio>` для вставки мультимедийного контента напрямую, без использования сторонних плагинов, повышает скорость загрузки страниц и кроссплатформенную совместимость, а функции локального хранения (`localStorage` и `sessionStorage`) HTML5 также позволяют хранить просматриваемые пользователем данные локально, не обращаясь к серверу. Функции локального хранения (`localStorage` и `sessionStorage`) в HTML5 также позволяют сохранять просматриваемые пользователем данные локально, не полагаясь на сервер, что улучшает отзывчивость и доступность в автономном режиме.

Внедрение CSS3 сделало веб-дизайн более гибким и эстетически привлекательным, а отзывчивые макеты стали ключом к кросс-девайсной совместимости. Благодаря Media Queries веб-сайты могут автоматически подстраивать свой макет и стиль в соответствии с размером и разрешением экрана пользовательского устройства, обеспечивая пользователям оптимизированный просмотр как на настольных компьютерах, так и на планшетах и мобильных телефонах. Кроме того, эффекты анимации и переходов в CSS3 повышают динамичность сайта, делая взаимодействие пользователей более плавным и естественным. Например, эффекты наведения кнопок, градиентное отображение контента и т. д. могут быть реализованы без использования JavaScript, что повышает производительность страницы.

JavaScript, как основной язык фронтенд-разработки, отвечает за реализацию динамических интерактивных функций страницы. С помощью JavaScript сайт может реагировать на действия пользователя в режиме реального времени, например, динамически загружать контент, проверять формы, обновлять данные и создавать анимационные эффекты. Применение технологии AJAX (Asynchronous JavaScript and XML) позволяет веб-странице получать данные с

сервера и обновлять содержимое страницы без необходимости перезагрузки всей страницы, что значительно повышает плавность и отзывчивость сайта. Плавность и отзывчивость веб-сайта. Например, на сайте электронной коммерции, когда пользователь нажимает на категорию товара, список товаров динамически обновляется без перезагрузки всей страницы, что обеспечивает пользователю более слаженную и плавную работу.

Для эффективного взаимодействия клиент-серверных данных на сайте также используется формат JSON (JavaScript Object Notation). По сравнению с традиционным форматом XML, преимущества JSON заключаются в том, что он более лаконичен, его легче анализировать и передавать. Он способен обмениваться данными с сервером через AJAX, что позволяет веб-странице представлять обновленные данные без перезагрузки, например, отображать прогресс загрузки файлов пользователем, показывать результаты поиска или обновлять количество товаров в корзине. Сочетание JSON и AJAX не только повышает эффективность взаимодействия данных, но и снижает потребление пропускной способности сети, позволяя веб-сайту загружать большие объемы данных и при этом сохранять хорошую производительность. поддерживать хорошую производительность.

Кроме того, в современной веб-разработке широко используются и другие технологии, такие как JavaScript-фреймворки Vue.js, React и Angular, которые обеспечивают разработчикам более модульный и компонентный подход к разработке, что еще больше повышает эффективность разработки и удобство сопровождения кода. Эти фреймворки часто используются в сочетании с цепочками инструментов, такими как webpack и Babel, для оптимизации загрузки ресурсов и кроссбраузерной совместимости, обеспечивая стабильную работу веб-сайтов во всех основных браузерах и устройствах.

Благодаря совместному применению таких технологий, как HTML5, CSS3, JavaScript, AJAX и JSON, современные веб-сайты способны не только обеспечить эффективный и стабильный рендеринг страниц, но и добиться богатого динамического взаимодействия, что значительно повышает удобство пользования

и интерактивность сайта. В то же время сочетание этих технологий позволяет сайту адаптироваться к различным устройствам и экранам для удовлетворения потребностей разных пользователей, становясь важным инструментом повышения ценности бренда и лояльности пользователей.

При проектировании архитектуры системы используется паттерн проектирования MVC (Model-View-Controller), основная цель которого - разделить бизнес-логику, обработку данных и пользовательский интерфейс приложения, чтобы повысить ремонтопригодность, расширяемость и эффективность разработки системы. Благодаря такому разделенному дизайну код может управляться и обновляться более гибко, особенно при изменении бизнес-требований, а взаимодействие между front-end и back-end может быть сведено к минимуму.

Слой модели отвечает за обработку всей логики, связанной с данными, обычно включая операции с базой данных, проверку данных и взаимодействие с внешними сервисами. В данной системе PHP, как язык бэкенда, берет на себя ответственность за слой бизнес-логики, обрабатывая запросы пользователей и взаимодействуя с базой данных. Для хранения данных в системе в основном используется база данных MySQL, которая выполняет операции создания, удаления, изменения и исследования (CRUD) данных с помощью SQL-запросов. Для оптимизации производительности запросов могут использоваться SQL-индексы, хранимые процедуры, представления и т. д., ускоряющие поиск данных. В то же время слой модели отвечает за реализацию проверок легальности данных и некоторой сложной бизнес-логики, такой как регистрация пользователей, аутентификация входа и т. д.

Слой представления отвечает за отображение данных и непосредственное взаимодействие с пользователями. В системе внешняя страница строит структуру страницы с помощью HTML, а стиль страницы разрабатывается с помощью CSS, так что контент может быть представлен в эстетически приятном виде, отвечающем требованиям пользователей. В то же время, использование JavaScript значительно повышает интерактивность страницы, например, при отправке формы, динамической загрузке данных, обратной связи с пользователем и т.д.,

фронтальная страница может быть достигнута без обновления, так что система является более беглой и отзывчивой. Фронт-энд также может использовать технологию AJAX для асинхронного взаимодействия с данными бэк-энд для достижения мгновенных обновлений, чтобы улучшить пользовательский опыт.

Слой контроллера выступает в качестве посредника, соединяющего слой модели и слой представления, и отвечает за прием пользовательских запросов, вызов соответствующей обработки бизнес-логики и принятие решения о том, какое содержимое представления вернуть. В архитектуре MVC контроллер вызывает соответствующие методы слоя модели для обработки данных в соответствии с действиями пользователя (например, нажатие кнопки, отправка формы и т. д.), одновременно передавая результат слою представления и, наконец, отображая содержимое пользователю. PHP-контроллеры обычно направляют запрос в соответствии с URL-путем и методом запроса (GET, POST), и назначают запрос соответствующей функции контроллера для обработки.

Для обеспечения безопасности системы в ней предусмотрены механизмы управления правами и аутентификации пользователей, позволяющие разным пользователям иметь различные операционные привилегии в соответствии с их ролями. С помощью модуля аутентификации пользователей (например, вход в систему, регистрация, сброс пароля и т. д.) система может проверить личность пользователя и предотвратить доступ неавторизованных пользователей к конфиденциальному содержимому или выполнение незаконных операций. К распространенным методам аутентификации относятся аутентификация на основе сессий (Session) или аутентификация на основе маркеров. После того как пользователь успешно вошел в систему, ему выдается соответствующая информация о правах доступа, и система контролирует его доступ к страницам и ресурсам в соответствии с его ролью. Например, обычные пользователи могут просматривать только свою личную информацию и историю покупок, а также управлять масштабируемостью и ремонтопригодностью системы.

В процессе разработки системы особое внимание уделяется ее масштабируемости и удобству обслуживания, что позволяет гибко расширять ее по мере

роста бизнеса и изменения потребностей. Например, благодаря использованию модульного подхода к разработке, каждый функциональный модуль (например, управление пользователями, управление продуктами, управление заказами и т. д.) может разрабатываться и поддерживаться независимо, не затрагивая другие модули. Кроме того, архитектура системы учитывает возможность повторного использования кода за счет абстрагирования интерфейсов и сервисных слоев, что делает связь между различными модулями более низкой и облегчает расширение и поддержку будущей функциональности.

Помимо базового управления правами, в систему интегрированы и другие меры безопасности, обеспечивающие сохранность данных и конфиденциальность пользователей. Например, на уровне хранения данных конфиденциальная информация (например, пароли пользователей) хранится с помощью хэш-шифрования для предотвращения утечки паролей; на коммуникационном уровне для обмена данными между фронтальной и тыловой частями используется протокол шифрования SSL/TLS, который предотвращает атаки посредников на данные во время передачи. Кроме того, меры по предотвращению сетевых атак, таких как SQL-инъекции и XSS-атаки, также находятся в центре внимания при разработке системы, которая обеспечивает безопасность системы путем фильтрации входных данных и использования операторов предварительной обработки.

Эта система представляет собой гибкую, масштабируемую и эффективную структуру приложений, использующую архитектурный паттерн MVC и объединяющую технологии PHP и MySQL. Обеспечивая четкое разделение бизнес-логики, разработка front-end и back-end может осуществляться независимо друг от друга благодаря дизайну разделения front-end и back-end, что повышает эффективность разработки. Управление правами и безопасность системы гарантируют сохранность данных и пользователей, обеспечивая конечным пользователям хороший опыт и безопасность.

С точки зрения аппаратного обеспечения, оборудование, поддерживающее систему, включает в себя серверы и клиентские компьютеры, мобильные теле-

фоны и так далее. Сервер, как основная платформа поддержки веб-сайта, выполняет обработку большого количества запросов и задачи хранения данных, поэтому требования к его аппаратной конфигурации высоки, и он должен обладать достаточной вычислительной мощностью и объемом памяти, чтобы обеспечить эффективную и стабильную работу системы в случае резкого увеличения числа пользователей. В частности, конфигурация сервера должна рассматриваться с нескольких точек зрения.

Во-первых, решающее значение имеет производительность процессора (CPU). В случае высокого одновременного доступа сайт должен быстро реагировать на запросы пользователей, поэтому для сервера следует выбрать многоядерный процессор с высокой основной частотой, способный обрабатывать больше запросов за короткий промежуток времени. В то же время, чтобы повысить производительность обработки, многоядерные процессоры могут использоваться для поддержки большего количества потоков в параллельных вычислениях, чтобы избежать узких мест в производительности.

Во-вторых, память (RAM) также является одним из важных факторов, влияющих на производительность сервера. При работе веб-сайта, особенно когда речь идет об обработке больших объемов пользовательских данных и генерации динамического контента, требуется большой объем памяти. Достаточный объем памяти позволяет кэшировать данные и сократить количество запросов к базе данных, что повышает скорость отклика. При выборе конфигурации памяти следует исходить из предполагаемого количества пользователей, чтобы не допустить переполнения памяти или снижения производительности системы в условиях одновременной работы.

Пространство на жестком диске и производительность системы хранения данных - еще один ключевой аспект конфигурации сервера. Веб-сайт предполагает большой объем загрузки и выгрузки информации и хранения данных, поэтому емкость жесткого диска должна удовлетворять потребности в хранении и резервном копировании данных. Помимо емкости жесткого диска, не менее важна скорость его чтения и записи. Использование твердотельных накопителей

(SSD) по сравнению с традиционными механическими жесткими дисками (HDD) имеет более высокую скорость чтения и записи, позволяет значительно повысить скорость доступа к данным и общую производительность системы, особенно в случае масштабного доступа и обработки больших объемов данных, SSD имеет особое значение.

Что касается архитектуры сервера, то выбор веб-сервера также имеет решающее значение. К распространенным веб-серверам относятся Apache и Nginx - широко используемые решения с открытым исходным кодом, которые эффективно обрабатывают запросы клиентов и взаимодействуют с программным обеспечением внутреннего прикладного уровня, таким как PHP, MySQL и т. д. Apache подходит для сайтов любого размера благодаря своей мощности и масштабируемости, а Nginx известен своей легкостью и высокой параллельностью, высокой производительностью и широко используется для сайтов с высоким трафиком и запросами.

Для повышения масштабируемости и отказоустойчивости системы серверы более высокой конфигурации часто оснащаются технологией балансировки нагрузки. Балансировка нагрузки позволяет равномерно распределять запросы пользователей между несколькими серверами, что позволяет избежать чрезмерной нагрузки на один сервер и снизить риск простоя сервера. В то же время балансировка нагрузки повышает доступность системы, гарантируя, что в случае выхода из строя одного сервера другие смогут продолжать предоставлять услуги без ущерба для доступа пользователей.

На уровне базы данных, чтобы справиться с требованиями к хранению и запросам массивных данных, можно использовать распределенную архитектуру базы данных. Разбиение базы данных на несколько узлов данных и распределение их по разным физическим серверам позволяет эффективно повысить производительность доступа к базе данных, уменьшить задержку запросов и увеличить масштабируемость системы. Распределенная система баз данных не только повышает нагрузочную способность системы, но и обеспечивает высокую до-

ступность и целостность данных, позволяя другим узлам продолжать предоставлять услуги при выходе из строя одного из узлов базы данных.

Выбор конфигурации и архитектуры серверного оборудования напрямую связан со стабильностью и производительностью сайта. Помимо основных ресурсов процессора, памяти, хранилища и других аппаратных средств, необходимо также сочетать балансировку нагрузки, распределенную архитектуру и другие технические средства для оптимизации производительности и масштабируемости системы, чтобы справиться с растущим пользовательским спросом и трафиком данных. Благодаря разумной конфигурации и оптимизации аппаратного обеспечения можно обеспечить пользователям более быстрое и стабильное обслуживание.

Кроме того, аппаратные средства включают сетевое оборудование, такое как маршрутизаторы и коммутаторы, которые обеспечивают эффективную передачу данных между серверами и клиентами. В кампусной сети также может потребоваться управление полосой пропускания в локальной сети, чтобы обеспечить пользователям высокую скорость доступа и загрузки информации. Между тем, аппаратные средства обеспечения безопасности, такие как брандмауэры и системы обнаружения вторжений, также играют важную роль в общей системе, защищая веб-сайт от внешних вредоносных атак и внутренней утечки данных.

Для обеспечения стабильности и надежности системы, помимо основных аппаратных средств, необходимо также оснастить ее оборудованием для резервного копирования и системой аварийного восстановления. Эти средства способны регулярно создавать резервные копии пользовательских данных и системной конфигурации сайта. В случае сбоя системы или потери данных сайт может быть быстро восстановлен с помощью резервных копий, что обеспечит его постоянную доступность.

3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРЕДЛАГАЕМОГО АЛГОРИТМА РЕШЕНИЯ ЗАДАЧИ

3.1 Основные этапы практической разработки программного продукта

При разработке системной архитектуры веб-сайта для обмена данными в кампусе основное внимание уделяется универсальности и простоте использования, чтобы обеспечить легкий доступ к платформе и ее использование обычными пользователями через обычные браузерные клиенты. Поскольку функции сайта в основном сосредоточены на загрузке, скачивании, категоризации, поиске и взаимодействии с пользователями, особых требований к аппаратному обеспечению нет. Пользователи могут получить доступ к платформе, просто введя URL-адрес в браузер на стороне ПК, без необходимости устанавливать дополнительное клиентское программное обеспечение.

На стороне сервера конфигурация оборудования может быть настроена в соответствии с предполагаемым количеством пользователей и объемом доступа. Для малых и средних кампусных веб-сайтов требованиям может соответствовать сервер с умеренной конфигурацией. Рекомендации по конфигурации включают в себя как минимум 4-ядерный процессор для поддержки одновременных запросов пользователей, 8 ГБ или более оперативной памяти для обеспечения стабильности системы в условиях высокой интенсивности работы, а также 500 ГБ или более места на жестком диске для хранения загруженных пользователями файлов данных и поддержки расширения. Рекомендуется пропускная способность сети не менее 100 Мбит/с, чтобы пользователи могли быстро загружать и скачивать информацию.

Что касается программного обеспечения, то сайт обмена данными в кампусе разработан с учетом стабильности, совместимости и простоты обслуживания. На стороне сервера используется операционная система Windows, которая проста в развертывании и управлении. Операционная система Windows, являющаяся одной из самых распространенных настольных операционных систем в

мире, имеет множество преимуществ. Во-первых, она обладает широким спектром совместимости с аппаратным обеспечением, может поддерживать различные марки компьютеров и устройств, пользователь может гибко выбирать конфигурацию оборудования в соответствии с потребностями. Во-вторых, Windows предоставляет богатую экосистему программного обеспечения, будь то офисные программы, средства разработки или развлекательные приложения, пользователи могут найти подходящую программу для себя, а многие основные программы имеют приоритет на поддержку платформы Windows. Кроме того, графический пользовательский интерфейс Windows разработан так, чтобы быть интуитивно понятным и простым в использовании, подходящим для пользователей разного уровня подготовки, от новичков до профессионалов, которые могут быстро приступить к работе. Для бизнес-пользователей Windows предоставляет мощные функции управления сетью и безопасности, которые защищают данные и поддерживают эффективную совместную работу. В то же время Windows постоянно обновляется, внедряя новые технологии, такие как ассистенты с искусственным интеллектом и интеграция облачных сервисов, чтобы помочь пользователям повысить эффективность работы. В целом, благодаря своей совместимости, простоте использования и функциональности ОС Windows стала одним из лучших вариантов для персональных и корпоративных пользователей.

В качестве языка сценариев на стороне сервера используется PHP 5.6 или выше. PHP - это широко используемый язык сценариев с открытым исходным кодом, который особенно подходит для веб-разработки и может быть легко интегрирован с базами данных MySQL. PHP имеет очень высокую степень адаптивности к MySQL, и эти два языка давно считаются классическим сочетанием в веб-разработке. PHP - это широко используемый язык сценариев на стороне сервера, который особенно подходит для веб-разработки, и MySQL - это широко используемый язык сценариев с открытым исходным кодом, который особенно подходит для веб-разработки, и MySQL - это широко используемый язык сценариев с открытым исходным кодом, который особенно подходит для веб-разра-

ботки. PHP - это широко используемый язык серверных сценариев, который особенно хорошо подходит для веб-разработки, а MySQL - популярная система управления реляционными базами данных с открытым исходным кодом. php предоставляет богатую библиотеку расширений и функций MySQL, которые позволяют разработчикам легко подключаться к базам данных MySQL, запрашивать их и манипулировать ими. Начиная с ранних расширений mysql, усовершенствованных расширений mysqli и заканчивая современным уровнем абстракции PDO (PHP Data Objects), PHP продолжает оптимизировать взаимодействие с MySQL для повышения производительности и безопасности.

Сочетание PHP и MySQL отлично подходит для динамической веб-разработки, эффективно справляясь с задачами хранения, поиска и управления данными. Будь то небольшая система ведения блогов или крупная платформа электронной коммерции, сочетание PHP и MySQL способно удовлетворить все потребности. Кроме того, многие популярные системы управления контентом (например, WordPress, Joomla и Drupal) и фреймворки (например, Laravel, Symfony) построены на PHP и MySQL, что еще раз доказывает их пригодность и стабильность.

Система состоит из нескольких модулей, каждый из которых имеет свои уникальные функции и обязанности, чтобы обеспечить эффективность и простоту использования системы. Ниже приведены подробные дополнения к дизайну каждого модуля:

Модуль управления информацией является одним из основных модулей системы, который включает в себя публикацию, обновление, удаление и управление всей информацией в системе. Этот модуль разделен на две части: фронтенд и бэкенд, каждая из которых имеет свои права и функции.

Просмотр учебных материалов и информации: пользователи могут просматривать все опубликованные учебные материалы и другую связанную с ними информацию в системе в соответствии со своими потребностями. Поиск информации может осуществляться по категориям, времени или ключевым словам, чтобы пользователи могли быстро найти необходимые им ресурсы.

Публикация учебных материалов и информации: пользователи могут публиковать новые учебные материалы и связанную с ними информацию в соответствии со своими полномочиями. При публикации пользователям необходимо выбрать категорию информации, загрузить файл или ввести текстовое содержимое, а также добавить к информации теги, чтобы другим было легче ее найти.

Управление личной информацией: пользователи могут просматривать и управлять собственным опубликованным контентом, в том числе редактировать и удалять опубликованные материалы и информацию. Информацию, которая больше не нужна, пользователи могут удалить.

Добавление, изменение и удаление информации: администраторы могут добавлять новую информацию в систему, изменять ошибочную информацию или обновлять устаревшую информацию по мере необходимости. Кроме того, администратор может удалять неквалифицированную информацию, например, контент, который не соответствует критериям аудита.

Управление списком информации: администраторы могут просматривать список всей информации и страницы, сортировать и фильтровать информацию, чтобы облегчить эффективное управление большим количеством информации в системе.

Просмотр и утверждение информации: администраторы могут просматривать предоставленную пользователями информацию, чтобы убедиться, что в открытом доступе находится только та информация, которая соответствует спецификациям, что предотвращает попадание в систему неподходящего контента.

Фронтальный модуль управления - это основной интерфейс для взаимодействия пользователей с системой. В основном он содержит функции входа в систему, просмотра информации, публикации и управления студентами.

Студенты входят в систему через страницу входа в систему и проходят аутентификацию в соответствии с паролем учетной записи. После успешного входа в систему пользователи получают доступ к различным функциям системы, таким как просмотр и публикация информации.

После того как студенты вошли в систему, система отображает понятное

навигационное меню, помогающее студентам быстро найти нужные им функциональные модули, такие как учебные материалы, информация о курсе, управление пользователями и т. д.

Просмотр и поиск материалов: студенты могут просматривать различные учебные материалы и информацию на главной странице и использовать функцию поиска в системе, чтобы найти интересующий их контент, по ключевым словам, или категориям.

Публикация и управление профилями: студенты могут публиковать свои собственные учебные материалы или информацию, редактировать содержание и предоставлять соответствующие файлы для загрузки. После публикации студенты могут управлять размещенным контентом, в том числе удалять или изменять его.

Управление информацией: администраторы могут утверждать, удалять и изменять информацию, размещенную пользователями фронтенда, через страницу управления бэкенда. Это обеспечивает качество и точность информации.

Управление правами пользователей: администраторы могут назначать различные права для разных пользователей через модуль управления фронтендом, гарантируя, что каждый пользователь в системе может получить доступ и работать с контентом только в рамках своих прав.

Статистика и отчетность: администраторы могут просматривать статистику информации, размещенной в системе, например, частоту загрузок, активность пользователей и т. д., чтобы обеспечить поддержку данных для последующего управления.

Основная роль модуля управления учетными записями заключается в управлении учетными записями всех пользователей в системе, чтобы гарантировать, что пользователи с различными ролями имеют четкие права и объем операций в системе.

Управление учетными записями пользователей: администратор может добавлять, изменять и удалять учетные записи пользователей. С помощью функции управления учетными записями администратор может управлять обычными

пользователями и учетными записями администраторов для обеспечения безопасности работы системы.

Добавление учетной записи: администратор может создавать новые учетные записи пользователей, назначать роли (например, обычных пользователей или администраторов), устанавливать начальный пароль и задавать соответствующие разрешения.

Изменить учетную запись: если пользователю необходимо изменить информацию о своей учетной записи (например, пароль, электронную почту и т. д.), администратор может выполнить операцию изменения.

Удалить учетную запись: если пользователь больше не нуждается в использовании системы, администратор может удалить его учетную запись и лишить всех его привилегий.

Назначение и управление разрешениями: администратор может назначать различные разрешения в соответствии с ролями пользователей. Супер-администраторы имеют все права управления, а обычные администраторы - определенные права, такие как просмотр информации и изменение данных. Эта функция обеспечивает четкую иерархическую структуру системы и четкие права на управление.

Помимо двух ролей - обычных пользователей и администраторов, администраторы также могут разграничивать права. Например, одни администраторы могут иметь право только на аудит информации, а другие - на удаление информации или изменение настроек системы. Риск операционных ошибок и утечки данных снижается благодаря уточненному управлению разрешениями.

Система записывает журналы операций каждой учетной записи, включая операции входа в систему, выдачи информации, изменения и удаления информации. Эти журналы играют важную роль в отслеживании поведения операций, устранении проблем и повышении безопасности системы на более поздних этапах.

Таблица `fa_message` (таблица базы данных стены исповеданий) - это таб-

лица базы данных, используемая для хранения и управления информацией, связанной со стеной исповеданий. В этой таблице хранится информация, размещенная пользователями на стене исповеди, включая основную информацию пользователей, объекты исповеди, загруженные файлы и так далее. С помощью этой таблицы пользователи могут размещать информацию об исповеди и загружать связанные с ней файлы, которые могут просматривать и взаимодействовать с ними другие пользователи.

Таблица 1 – Таблица fa_message (таблица базы данных tabula rasa)

Имя поля	Тип данных	Разрешено быть ненулевым	Примечания
id	int	yes	Первичный ключ
username	varchar	no	Имя пользователя
useremail	varchar	no	Электронная почта
usermobile	varchar	no	Номер мобильного телефона
create_time	char	no	Время
loveUser	varchar	no	Исповедник
loveUserEd	varchar	no	Исповедовался
sourceMark	varchar	no	Примечания
thumb	varchar	no	Миниатюра
file	varchar	no	Файл
status	int	no	Статус: 1=Существует, 0=Удален
filename	varchar	no	Имя файла

Таблица fa_source (таблица общих ресурсов) - это таблица базы данных, используемая для реализации обмена информацией и управления ресурсами между студентами. В таблице хранится информация об общих ресурсах, загруженных студентами, включая основную информацию о ресурсах, информацию о

загрузчике, информацию о файлах и статус ресурсов. С помощью этой таблицы студенты могут загружать материалы курса, учебные ресурсы или личные файлы, а другие студенты могут просматривать, скачивать и использовать эти ресурсы.

Таблица 2 – Таблица fa_source (таблица общих ресурсов)

Имя поля	Тип данных	Разрешено быть ненулевым	Примечания
id	int	yes	Первичный ключ
username	varchar	no	Имя пользователя
useremail	varchar	no	Электронная почта
usermobile	varchar	no	Номер мобильного телефона
create_time	char	no	Время
sourceTitle	varchar	no	Название
sourceDesc	varchar	no	Описание
sourceMark	varchar	no	Примечания
thumb	varchar	no	Миниатюра
file	varchar	no	Файл
status	int	no	Статус: 1=Существует, 0=Удален
filename	varchar	no	Имя файла

Таблица fa_user (таблица пользователей) - это таблица базы данных, используемая для хранения и управления информацией о пользователях. В этой таблице хранится основная информация, информация для входа в систему, состояние учетной записи и другие связанные атрибуты пользователей. С помощью этой таблицы система может осуществлять регистрацию пользователей,

вход в систему, управление разрешениями и другие функции, а также предоставлять пользователям персонализированные услуги.

Таблица 3 – Таблица fa_user (таблица пользователей)

Имя поля	Тип данных	Разрешено быть ненулевым	Примечания
1	2	3	4
id	int	yes	ID
group_id	int	yes	Группа ID
username	varchar	no	Имя пользователя
nickname	varchar	no	Псевдоним
password	varchar	no	Пароль
salt	varchar	no	Пароль Соль
email	varchar	no	Электронная почта
mobile	varchar	no	Номер мобильного телефона
avatar	varchar	no	Аватар
level	tinyint	yes	Класс
gender	tinyint	yes	Пол
birthday	date	no	День рождения
bio	varchar	no	Девиз
money	decimal	yes	Баланс
score	int	yes	Баллы
successions	int	yes	Дни подряд
	int	yes	Максимальное количество дней входа в систему

Продолжение таблицы 3

1	2	3	4
loginip	varchar	yes	IP-адрес входа в систему
loginfailure	tinyint	yes	Не удалось
joinip	varchar	no	Присоединиться IP
jointime	int	no	Время регистрации
createtime	int	no	Время создания
updatetime	int	no	Время обновления
token	varchar	no	Токен
status	varchar	no	Статус
verification	varchar	no	Подтверждено

Таблица fa_user_group (таблица разрешений) - это таблица базы данных, используемая для управления группами пользователей и их разрешениями. В таблице хранится такая информация, как имя группы пользователей, узел разрешения, время создания, время обновления и статус.

Таблица 4 – Таблица fa_user_group (таблица разрешений)

Имя поля	Тип данных	Разрешено быть ненулевым	Примечания
id	int	yes	
name	varchar	no	Имя группы
rules	text	no	Разрешение Узел
createtime	int	no	Время добавления
updatetime	int	no	Время обновления

Таблица fa_user_rule (таблица имен разрешений пользователей) - это таблица базы данных, используемая для определения и управления правилами разрешений пользователей. В таблице хранятся иерархические отношения, имя, название, примечания, отображается ли оно в виде меню, время создания, время обновления, вес и статус правила разрешения. С помощью этой таблицы система может реализовать классификацию, иерархическое управление и динамическую конфигурацию правил разрешений для обеспечения гибкости и масштабируемости разрешений.

Таблица 5 – Таблица fa_user_rule (таблица имен разрешений пользователей)

Имя поля	Тип данных	Разрешено быть ненулевым	Примечания
id	int	yes	
pid	int	no	Отец ID
name	varchar	no	Имя
title	varchar	no	Название
remark	varchar	no	Примечания
ismenu	tinyint	no	Меню или нет
createtime	int	no	Время создания
updatetime	int	no	Обновлено
weigh	int	no	Вес
status	enum	no	Статус

Таблица fa_lost (таблица находок) - это таблица базы данных, используемая для управления информацией о потерянных и найденных предметах. В этой таблице хранится информация о потерянных и найденных предметах, размещенная пользователями, включая основную информацию о потерянном предмете, контактную информацию пользователя, описание потерянного предмета, прикрепленную фотографию или файл, а также статус. Эта таблица позволяет пользователям размещать информацию о потерянных предметах, которую могут

просматривать и помогать другим пользователям.

Таблица 6 – Таблица fa_lost (таблица находок)

Имя поля	Тип данных	Разрешено быть ненулевым	Примечания
id	int	yes	Первичный ключ
username	varchar	no	Имя пользователя
useremail	varchar	no	Электронная почта
usermobile	varchar	no	Номер мобильного телефона
create_time	char	no	Время
LostColor	varchar	no	Цвет предмета
LostThings	varchar	no	Потерянный предмет Название
LostThingsDesc	varchar	no	Описание
thumb	varchar	no	Изображение
file	varchar	no	Файл
status	int	no	Статус: 1=Существует, 0=Удален
filename	varchar	no	Имя файла
type	varchar	no	Тип

Таблица fa_uan (таблица дизайна сообщений) - это таблица базы данных, используемая для управления информацией о сообщениях пользователей. В таблице хранится содержимое сообщений, размещенных пользователями, включая основную информацию об отправителе сообщения, содержание сообщения, описание, примечания, изображения или вложения файлов, а также статус. С помощью этой таблицы пользователи могут размещать сообщения, которые могут просматривать и взаимодействовать с ними другие пользователи.

Таблица 7 – Таблица fa_yan (таблица дизайна сообщений)

Имя поля	Тип данных	Разрешено быть ненулевым	Примечания
id	int	yes	Первичный ключ
username	varchar	no	Имя пользователя
useremail	varchar	no	Электронная почта
usermobile	varchar	no	Номер мобильного телефона
create_time	char	no	Время
content	varchar	no	Содержание
sourceDesc	varchar	no	Описание
sourceMark	varchar	no	Примечания
thumb	text	no	Изображение
file	varchar	no	Файл
status	int	no	Статус: 1=Существует, 0=Удален
filename	varchar	no	Имя файла

Таблица fa_admin_log (таблица журналов) - это таблица базы данных, используемая для записи журналов операций администратора. В таблице хранится информация об операциях администратора в системе, включая идентификатор администратора, имя администратора, страницу операции, заголовок журнала, содержание операции, IP-адрес операции, User-Agent и время операции. С помощью этой таблицы система может отслеживать и проверять поведение администратора при выполнении операций, чтобы обеспечить безопасность и обслуживаемость системы.

Таблица 8 – Таблица fa_admin_log (таблица журналов)

Имя поля	Тип данных	Разрешено быть ненулевым	Примечания
id	int	yes	ID
Admin_id	int	yes	Администратор ID
username	varchar	no	Имя администратора
url	varchar	no	Страница операции
title	varchar	no	Название журнала
Content	text	yes	Содержание
ip	varchar	no	IP
useragent	varchar	no	User-Agent
createtime	int	no	Время работы

Таблица fa_admin (таблица администраторов бэкенда) - это таблица базы данных, используемая для хранения и управления информацией об администраторах бэкенда. В таблицу записывается основная информация, информация для входа в систему, статус учетной записи и другие соответствующие атрибуты администратора. С помощью этой таблицы система может реализовать функции входа администратора в систему, управления полномочиями и аудита операций.

Таблица 9 – Таблица fa_admin (таблица администраторов бэкенда)

Имя поля	Тип данных	Разрешено быть ненулевым	Примечания
1	2	3	4
id	int	yes	ID
username	varchar	no	Имя пользователя
nickname	varchar	no	Прозвище

1	2	3	4
password	varchar	no	Пароль
salt	varchar	no	Пароль Соль
avalar	varchar	no	Аватар
email	varchar	no	Электронная почта
loginfailure	tinyint	yes	Время отказов
logintime	int	no	Время входа в систему
loginip	varchar	no	IP-адрес входа
createtime	int	no	Время создания
updatetime	int	no	Время обновления
token	varchar	no	Идентификатор сессии
status	varchar	yes	Статус

Таблица fa_admin (таблица групп привилегий администраторов бэкенда) - это таблица базы данных, используемая для управления группами привилегий администраторов бэкенда. В таблице хранятся иерархические отношения, имя группы, идентификатор правила, время создания, время обновления и статус группы привилегий. С помощью этой таблицы система может реализовать иерархическое управление группами привилегий и гарантировать, что разные группы привилегий имеют разные операционные привилегии. В таблицу записывается основная информация, информация для входа в систему, статус учетной записи и другие соответствующие атрибуты администратора.

Таблица 10 – Таблица fa_admin

Имя поля	Тип данных	Разрешено быть ненулевым	Примечания
id	int	yes	
pid	int	yes	Родительская группа
name	varchar	no	Группа
rules	text	yes	Идентификатор правила
createtime	int	no	Создано
updatetime	int	no	Обновлено
status	varchar	no	Статус

Таблица video (таблица видеоинформации) - это таблица базы данных, используемая для хранения и управления информацией о видеофайлах. В таблицу записывается основная информация о видео, информация о загрузчике, атрибуты файла и информация о хранении. С помощью этой таблицы система может реализовать функции загрузки, хранения, управления и отображения видеофайлов.

Таблица 11 – Таблица video (таблица видеоинформации)

Имя поля	Тип данных	Разрешено быть ненулевым	Примечания
1	2	3	4
id	int	yes	ID
admin_id	int	yes	ID администратора
user_id	int	yes	ID участник
url	varchar	no	Физический путь
imagewidth	varchar	no	Ширина
imageheight	varchar	no	Высота

1	2	3	4
imagetype	varchar	no	Тип изображения
imageframes	int	yes	Количество кадров
filename	varchar	no	Имя файла
filesize	int	yes	Размер файла
mimetype	varchar	no	Тип mine
extparam	varchar	no	Проходные данные
createtime	int	no	Дата создания
updatetime	int	no	Обновлено
uploadtime	int	no	Время загрузки
storage	varchar	yes	Место хранения
sha1	varchar	no	Кодировка файла sha1

Разработка этого сайта по обмену профилями в университетском городке основана на фреймворке ThinkPHP5.1, построенном по классическому шаблону MVC (Model-View-Controller), что значительно повышает эффективность разработки, особенно в версии 5.1, фреймворк представляет несколько новых мощных функций для дальнейшей оптимизации процесса разработки и управления проектом. ThinkPHP5.1 улучшает поддержку ThinkPHP5.1 имеет расширенную поддержку управления пакетами Composer, что делает управление зависимостями более эффективным и удобным, разработчики могут легче интегрировать сторонние библиотеки, чтобы улучшить масштабируемость и ремонтпригодность проекта. Кроме того, встроенный в фреймворк механизм статических прокси эффективно упрощает вызовы сервисов, уменьшая избыточность кода и делая его более лаконичным и читаемым; внедрение инструмента миграции базы данных

также делает контроль версий и миграцию базы данных более удобными, гарантируя, что изменения в структуре данных могут быть синхронизированы с обновлениями кода.

В плане фронтенд-разработки на сайте используются Bootstrap и jQuery и другие основные фреймворки, сочетание которых делает отзывчивый дизайн страницы более эффективным и позволяет быстро адаптироваться к различным размерам экранов и устройств. Bootstrap, как популярный фронтенд-фреймворк, предоставляет множество компонентов пользовательского интерфейса и гибких инструментов верстки, делая дизайн страницы более красивым и удобным в работе, в то время как jQuery обеспечивает простоту и удобство использования. В то время как jQuery предоставляет чистый и простой в использовании интерфейс JavaScript для облегчения реализации динамических эффектов и взаимодействия с пользователем. Внешние страницы были настроены командой разработчиков в соответствии с реальными потребностями, обеспечивая уникальный и работоспособный пользовательский опыт.

Для внутреннего раздела управления команда выбрала шаблонную разработку, чтобы быстрее внедрить различные функциональные модули и повысить эффективность разработки и сопровождения. Разработка шаблонов не только делает структуру кода более понятной, но и позволяет повторно использовать страницы и сократить объем повторяющейся работы, а также облегчает разделение front-end и back-end и повышает эффективность совместной работы команды.

Прежде всего, зайдите на главную страницу, с левой стороны мы видим, что сверху вниз расположены следующие функции: общее управление, управление разрешениями, управление членством, общий контент, стена исповедей, бюро находок и центр сообщений, а также различные подфункции под этими функциями. Все эти функции составляют бэкэнд-систему управления сайтом обмена данными в кампусе, которая проста и понятна (Рисунок 7).

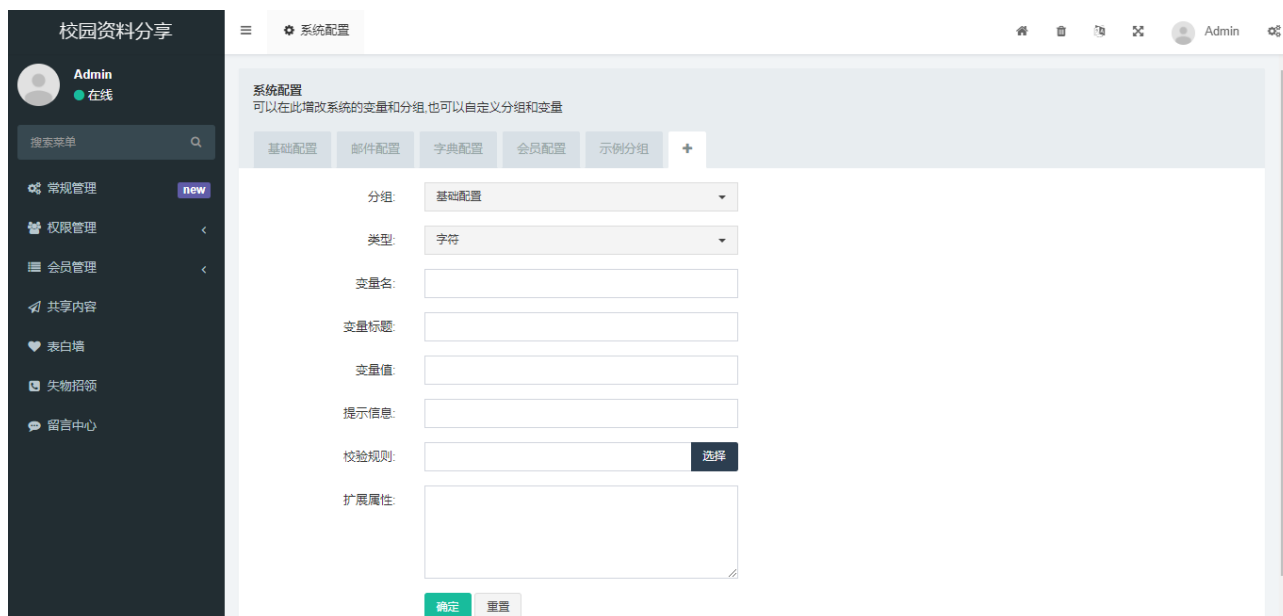


Рисунок 7 – Главный интерфейс Backstage

В общем модуле управления, содержит конфигурацию системы, управление вложениями и личные данные, текущий администратор может изменить конфигурацию здесь, разделенные на базовую конфигурацию, почтовая конфигурация, конфигурация словаря и конфигурация члена, вы можете просмотреть веб-страницу вложения конфигурации, такие как документы, изображения, аудио и видео и добавлять, удалять и изменять эти вложения также могут быть собствен-ные данные администратора, чтобы изменить и сбросить (Рисунок 8).



Рисунок 8 – рутинное управление

Список управления разрешениями основной записью является информация об администраторах. Здесь вы можете просмотреть все идентификаторы администраторов, имя пользователя, категорию, статус и время присоединения, запись входа администратора в систему и ролевую группу, а также добавить, удалить и изменить информацию об администраторах, а также удалить учетную запись администратора в то же время (Рисунок 9).



Рисунок 9 – Управление правами

Модуль управления участниками может просматривать все учетные записи пользователей, такие как ID, ролевая группа, имя пользователя, аватар, электронная почта, номер мобильного телефона, время входа, статус и т.д., а также добавлять, удалять и изменять информацию об учетной записи пользователя, вы также можете выйти из учетной записи пользователя, из учетной записи участников группы (Рисунок 9, 10).

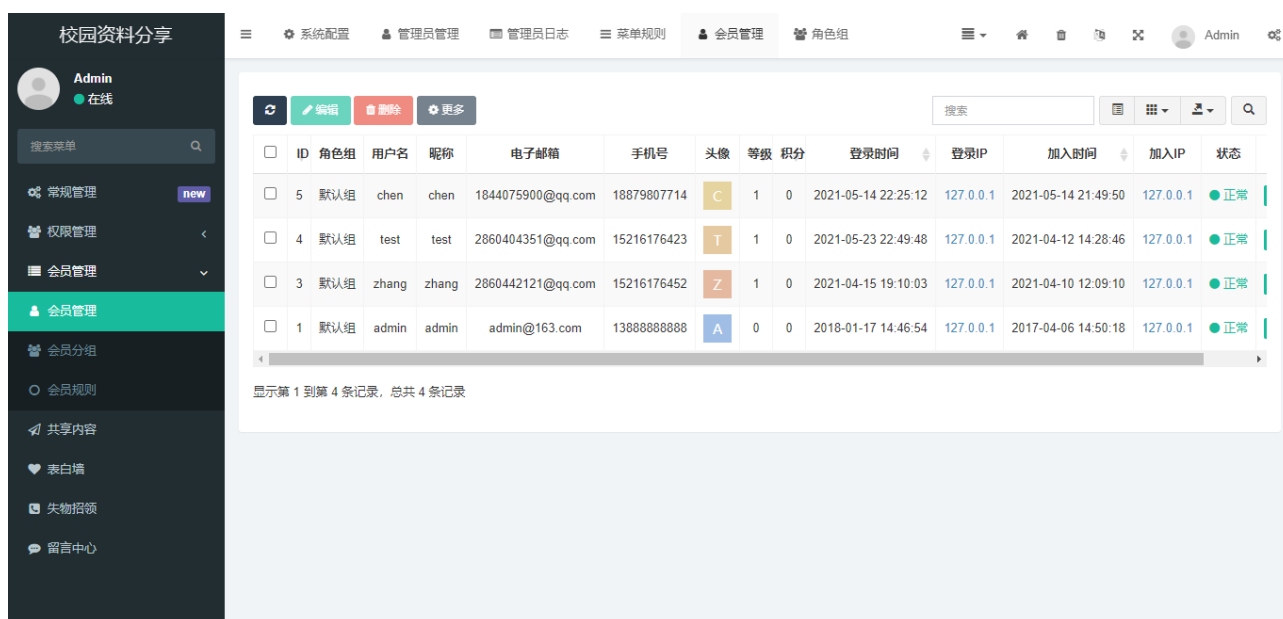


Рисунок 9 – Управление членством

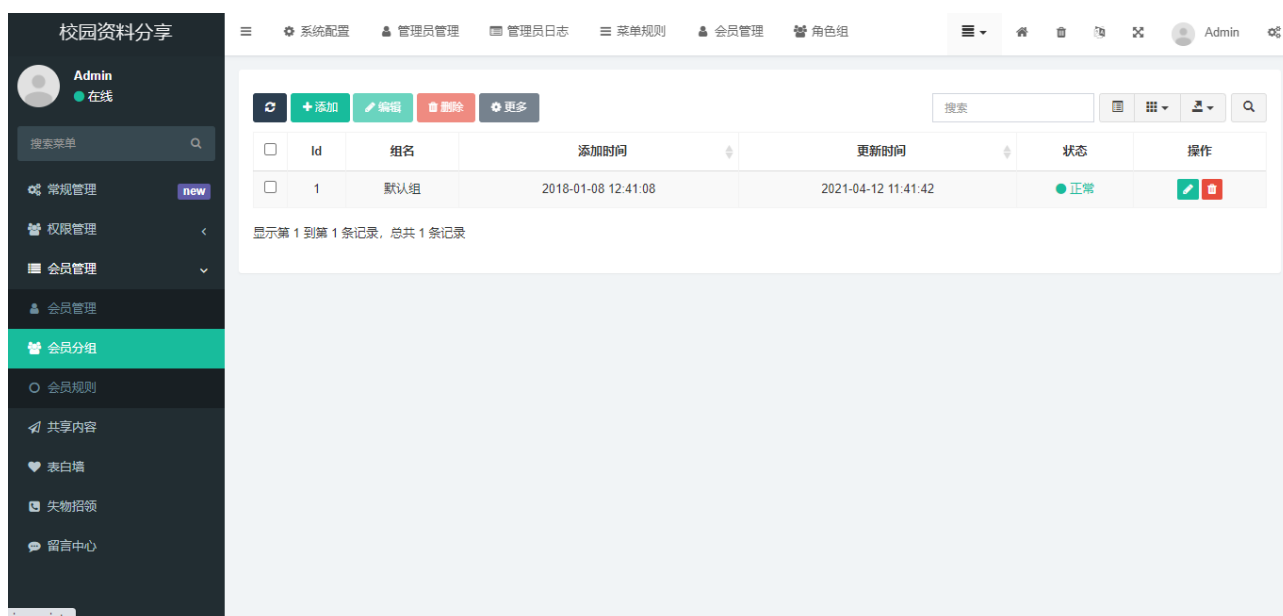


Рисунок 10 – Группировка участников

В модуле Shared Content администраторы могут просматривать содержимое учебных материалов, которыми пользователи поделились в центре совместного доступа, включая названия ресурсов, описания ресурсов, метки ресурсов, пути к ресурсам и т. д., а также добавлять, удалять и изменять их (Рисунок 11).



Рисунок 11 – Общий контент

В модуле «Исповедальная стена» администраторы могут просматривать содержание исповедальных стен, размещенных пользователями, добавлять, удалять и изменять их (Рисунок 7).



Рисунок 12 – Стена исповеди

В модуле «Находки» администраторы могут просматривать информацию о находках, размещенную студентами, а также добавлять, изменять и удалять их (Рисунок 13).



Рисунок 13 – Потерянный и найденный

Модуль «Центр сообщений», в котором администраторы могут просматривать содержимое всех сообщений пользователей, добавлять, изменять, удалять операции (Рисунок 14).

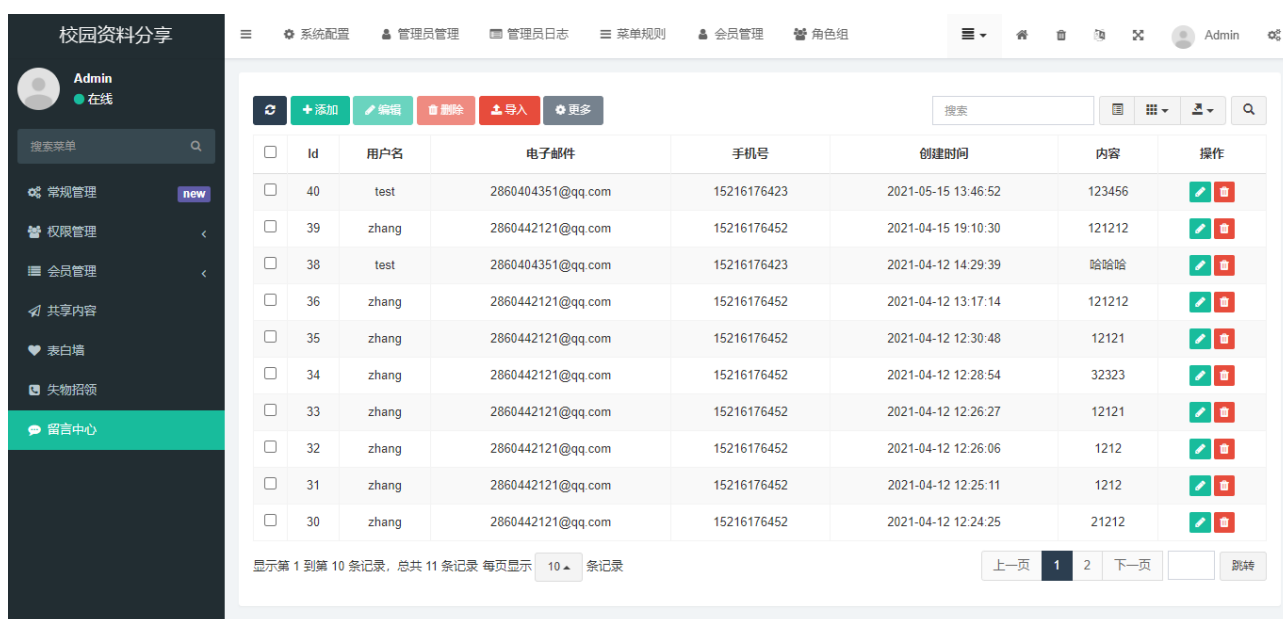


Рисунок 14 – Центр сообщений

Добавление, изменение и удаление информации - основные функции управления данными на сайте обмена профилями в кампусе, обеспечивающие динамическое обновление и поддержание содержимого профиля на сайте. PHP, как внутренний язык программирования, отвечает за обработку запросов пользователей, проверку вводимых данных, выполнение операций с базой данных и предоставление обратной связи пользователям в режиме реального времени. Эти

функции можно реализовать, взаимодействуя с базой данных, чтобы добиться комплексного управления содержимым профиля (Рисунок 15).

```
// 添加
public function insert($data){
    $res = Db::name($this->table)->insert($data);
    return $res;
}
// 修改
public function update($data){
    $res = Db::name($this->table)->where($this->where)-
>update($data);
    return $res;
}
// 删除
public function delete(){
    $res = Db::name($this->table)->where($this->where)->delete();
    return $res;
}
```

Рисунок 15 – Добавление, изменение и удаление кода

Функция загрузки файлов играет важную роль в веб-сайте обмена данными кампуса на базе РНР, который позволяет пользователям загружать на веб-сервер местные учебные материалы, файлы проектов и т.д. РНР обрабатывает загрузку файлов, принимая их, выполняя необходимую проверку, перемещая файлы в указанную директорию и регистрируя соответствующую информацию в базе данных. Кроме того, система должна обеспечивать соответствующую обратную связь с пользователем, чтобы процесс загрузки файлов проходил гладко и безопасно. Эта функция повышает интерактивность веб-сайта и облегчает обмен информацией и управление ею (Рисунок 16).

```

//文件上传接口
public function upload_img()
{
    $file = request()->file('file');
    if($file == null)
    {
        exit(json_encode(array('code'=>1,'msg'=>'没有文件上传')));
    }
    //选择文件的根目录
    $info = $file->move(ROOT_PATH.'public'.DS.'uploads');
    $ext = ($info->getExtension());
    if(!in_array($ext, array('jpg','jpeg','gif','png'))){
        //文件支持jpg','jpeg','gif','png'五种格式
        exit(json_encode(array('code'=>1,'msg'=>'文件格式不支持')));
    }
    $img = '/uploads/'.$info->getSaveName();
    exit(json_encode(array('code'=>0,'msg'=>$img)));
}

```

Рисунок 16 – Код загрузки файлов

При входе на сайт видно, что как только пользователь заходит на страницу системы, отображается главная страница сайта обмена профилями кампуса. Если пользователь не вошел в систему, мы предложим ему не входить в систему. В случае, если пользователь уже вошел в систему, он может нажать на Центр участника, чтобы просмотреть личные данные пользователя, а также может изменить личные данные пользователя, может просмотреть собственные совместные действия пользователя были выпущены, могут быть выпущены в учебные материалы, стена признания, потерянные и найденные или поиск вещей, сообщений и другой информации, но также может изменить пароль и выйти из логина (Рисунок 17).



Рисунок 17 – Основной интерфейс фронтенда

Основная цель функции входа в систему - обеспечить пользователям возможность входа на сайт с помощью уникального идентификатора (например, имени пользователя и пароля), получения определенных разрешений, доступа к ресурсам и обмена ими. Эта функция также обеспечивает аутентификацию личности пользователя и предотвращает доступ неавторизованных пользователей к страницам, требующим разрешений (Рисунок 18).

```

public function login()
{
    $url = $this->request->request('url', '', 'trim');
    // 已经登录, 不用再登录。
    if ($this->auth->id) {
        $this->
>success(__('You\'ve logged in, do not login again'), $url ? $url : url('user/
index'));
    }
    if ($this->request->isPost()) {
        $account = $this->request->post('account');
        $password = $this->request->post('password');
        $keeplogin = (int) $this->request->post('keeplogin');
        $token = $this->request->post('__token__');
        // 登录的规则 require : 不能是空 length:A,B 不得短于A, 也不能超过B
        $rule = [
            'account' => 'require|length:3,50',
            'password' => 'require|length:6,30',
            '__token__' => 'require|token',
        ];
        // 如果有错误, 弹窗报错
        $msg = [
            'account.require' => 'Account can not be empty',
            'account.length' => 'Account must be 3 to 50 characters',
            'password.require' => 'Password can not be empty',
            'password.length' => 'Password must be 6 to 30 characters',
        ];
        $data = [
            'account' => $account,
            'password' => $password,
            '__token__' => $token,
        ];
        $validate = new Validate($rule, $msg);
        $result = $validate->check($data);
        if (!$result) {
            $this->error(__($validate->getError()), null, ['token' => $this->request-
>token()]);
            return false;
        }
        if ($this->auth->login($account, $password)) {
            $this->success(__('Logged in successful'), $url ? $url : url('user/index'));
        } else {
            $this->error($this->auth->getError(), null, ['token' => $this->request-
>token()]);
        }
    }
}

```

Рисунок 18 – Код входа в систему

Суть функции Publish Lost and Found заключается в том, чтобы предоста-
вить пользователям возможность легко размещать информацию о найденных
или потерянных вещах, а также собирать, хранить и отображать данные с помо-
щью внешней формы, взаимодействующей с внутренним PHP-кодом. Эта функ-

ция помогает членам сообщества кампуса помогать друг другу находить потерянные вещи или возвращать найденные (Рисунок 19).

```
public function lost()
{
    $this->view->assign('title', __('Lost'));
    $user = $this->auth->getUser();
    $username = $user->username;

    $request = $this->request->post();
    $condition = [];
    $condition['username'] = $username;

    $search = isset($request['search']) ? $request['search'] : "";

    $condition['status'] = 1;
    if ($search != "") {
        $condition['LostThings'] = ['like', '%' . trim($search) . '%'];
    }
    $listData = Db::table("fa_lost")->where($condition)-
>Order('create_time DESC')->count();
    $data = Db::table("fa_lost")->where($condition)-
>Order('create_time DESC')->paginate(5, false);
    $page = $data->render();
    $sum = count($data);
    $score = $sum * 100;
    $this->assign("sum", $listData);
    $this->assign("score", $score);
    $this->assign("search", $search);
    $this->assign("page", $page);
    $this->assign("data", $data);
    return $this->view->fetch();
}
```

Рисунок 19 – Выдача кодов для находок

Кодовая реализация функции Delete Lost and Found заключается в удалении определенной записи в базе данных с помощью оператора DELETE посредством взаимодействия между PHP и базой данных MySQL. Параметр URL используется для передачи идентификатора удаляемой записи, и пользователь получает обратную связь после выполнения удаления. Эта функция повышает управляемость сайта и облегчает администраторам или пользователям работу с информацией о потерянных и найденных вещах.

3.2 Примеры фактического тестирования программного продукта

Тестирование системы - это процесс проведения и выполнения тестовых процедур с целью быстрого обнаружения ошибок в нашей системе. Успешное

обнаружение системных ошибок приводит к обнаружению любых системных ошибок, которые еще не были обнаружены нами. Конечная цель тестирования заключается в том, что мы хотим быстро найти все потенциальные ошибки и основные дефекты различной степени серьезности с наименьшими затратами времени и усилий. Тестирование должно основываться на этапах цикла разработки программного обеспечения, технических требованиях, проектной и другой технической документации или процедурах контроля конкретной внутренней структуры требований, тщательном изучении дизайна и анализе используемых тестовых примеров, а также должно в полной мере использовать дизайн тестовых примеров, использовать соответствующие примеры данных для контроля запуска программы, чтобы способствовать своевременному обнаружению важных ошибок. Управление работой тестера качества системы является предприятием для обеспечения качества проектирования инженерных систем и надежности системы важным ключевым шагом, является весь дизайн инженерной системы исследования и разработки продукта и рабочий процесс различных системных проблем в статистическом анализе, система для разработки и реализации проекта необходимо сделать окончательный контроль качества. Любая разработка программного обеспечения неотделима от этапа тестирования, иногда небольшая ошибка может привести к огромным потерям, поэтому тестирование системы чрезвычайно важно.

Программы в процессе разработки не могут гарантировать отсутствие ошибок, которых невозможно избежать, для любой программы в процессе разработки вероятны ошибки, поэтому для того, чтобы обнаружить ошибку, не продолжать существование при проектировании системы, необходимо тестирование. Ошибки могут иметь неизмеримые последствия, такие как отсутствие входа на веб-страницы, системные сбои, сообщения безопасности, неправильный запуск систем и множество других проблем. Роль тестирования заключается в поиске проблем и своевременном внесении изменений, хотя это и занимает много времени и сил, но это необходимо делать, чтобы гарантировать, что система мо-

жет быть использована правильно и в долгосрочной перспективе, это для будущего использования и прогресса, чтобы заложить прочный фундамент.

Перед официальным использованием системы, система должна пройти детальное и всестороннее тестирование. Работа по тестированию системы является обеспечением качества и надежности системы является важным шагом, является вся система исследования и разработки рабочего процесса всех систем для анализа, проектирования системы и реализации окончательного тестирования и оценки. Согласно теоретической концепции тестирования и его цели, когда мы проводим тестирование информационных систем сети электронной коммерции, мы должны строго следовать основным принципам.

Тестирование должно проводиться как можно раньше и без перерывов, а не после разработки прикладной системы. В силу сложности исходной задачи, разнообразия этапов разработки и других факторов, поэтому при валовой разработке на всех этапах возможны ошибки. Поэтому проект тестирования должен выполняться на всех этапах разработки, чтобы как можно раньше обнаружить ошибки в системе и устранить скрытые опасности.

При разработке тестовых примеров важно разрабатывать не только эффективные и разумные условия ввода, но и включать неразумные и недопустимые условия ввода. При тестировании люди склонны придерживаться обычного, не обратного образа мышления и пренебрегают тестированием аномальных, неразумных и неожиданных ситуаций, которые могут представлять собой скрытую опасность.

При тестировании приложения необходимо не только проверить, сделала ли программа то, что должна была сделать, но и тщательно изучить, не произошло ли с приложением чего-то лишнего, чего не должно было произойти. Дополнительное время и работа, скорее всего, окажут побочное влияние на общую производительность приложения, а иногда даже приведут к потенциальному вреду или ошибкам.

Важно разработать соответствующий план тестирования, а сам процесс тестирования должен проводиться строго в соответствии с планом, не порождая

произвола. Конкретное содержание тестового решения может включать в себя: конкретное содержание теста, составление графика, подбор персонала, условия тестирования, инструменты тестирования и другие необходимые для тестирования материалы. Для того чтобы эффективно и в полной мере обеспечить координацию работ по данному тестированию, организация графика тестирования должна строго соответствовать плану тестирования компании, чтобы следовать организационному порядку во избежание отклонений и ошибок процесса. Фаза тестирования - это отдельная, но очень важная фаза в процессе разработки программного обеспечения, и процесс тестирования в основном такой же, как и процесс разработки.

В системе сайта Campus Profile Sharing используется функциональное тестирование, которое проводится шаг за шагом по каждой функции.

После завершения работы над функцией важно своевременно протестировать ее, иначе потом будет очень сложно анализировать и тестировать все новые и новые итерации функции.

Тестирование заключается в предположении тестовых данных, имитации всех вводимых пользователем данных, проверке того, что система возвращает ожидаемые данные, если это не так, найдите место ошибки, измените код, чтобы продолжить тестирование до тех пор, пока система не будет возвращать ожидаемые данные.

Таблица 12 – Тестирование входа в бэкэнд

№	Тестовые элементы	Инструкции по вводу (работа)	Ожидаемые результаты	Результаты тестирования
1	Вход в внутреннюю систему	Введите: admin 123456	Доступ к системам управления бэк-офисом	Перейти на главную страницу системы управления бэкэндом
		Введите учетную запись разработчика	Переход в систему разработчика	Перейти на главную страницу системы управления разработкой
		Введите учетную запись тестировщика	Доступ к системе тестирования	Перейти на главную страницу системы тестирования
		Ошибка ввода имени пользователя	Невозможность входа в систему	Пожалуйста, введите ваше имя пользователя
		Неверный пароль	Сбой входа в систему	Пожалуйста, введите пароль

После проверки данных без ошибок, через json данные возвращаются пользователю, предлагая ему успешно войти в систему, чтобы облегчить дальнейшую работу пользователя (Рисунок 20).

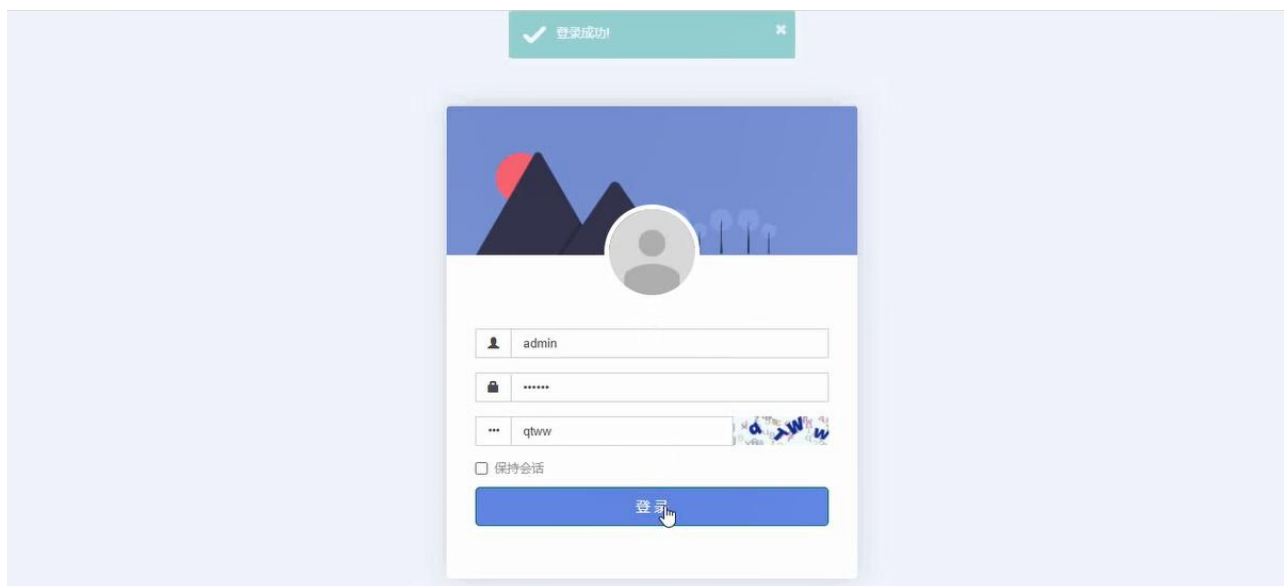


Рисунок 20 – Вход в систему Тест 1

Разные системные администраторы видят разные страницы, это связано с установкой разных прав, представляющих разное управление правами. Ниже представлен основной интерфейс для доступа системных администраторов и тестировщиков к внутренней системе (Рисунок 21, 22):

ID	用户名	昵称	所属组别	Email	状态	最后登录	操作
2	test	测试管理员	二级管理员2	2860442352@qq.com	正常	2021-05-19 21:58:34	[Edit] [Delete]
1	admin	Admin	超级管理员组	admin@admin.com	正常	2021-05-26 04:03:58	[Edit] [Delete]

显示第 1 到第 2 条记录, 总共 2 条记录

Рисунок 21 – Вход в систему Тест 2



Рисунок 22 – Вход в систему Тест 3

Если вы не введете имя пользователя, вам будет предложено указать, что имя пользователя не может быть пустым (Рисунок 23).

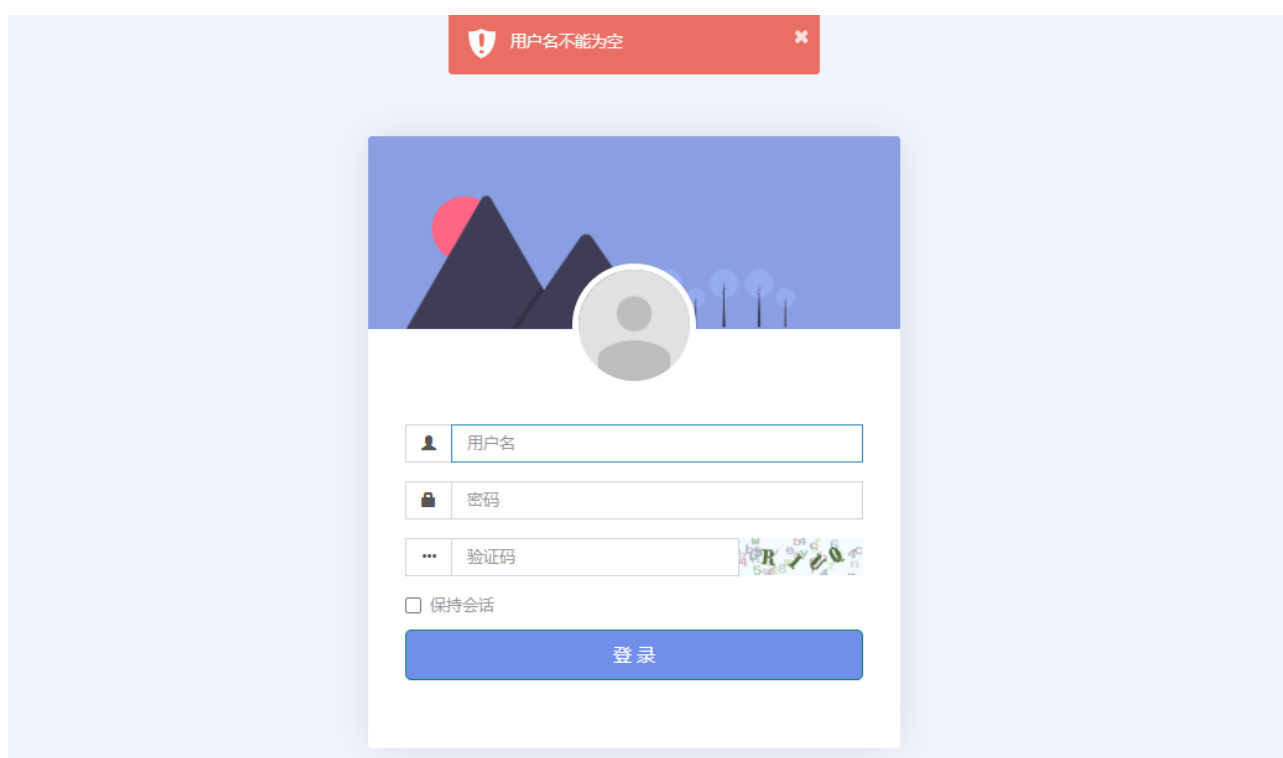


Рисунок 23 – Вход в систему Тест 4

После того как администратор успешно вошел в внутреннюю систему управления сайтом обмена профилями в кампусе, он выполняет соответствующие тесты в модуле пользователя. Тест пользовательского модуля включает в

себя, в основном, тест работы администратора и тест работы пользователей.

Таблица 13 – Тестирование входа в бэкэнд

№	Функция Название теста	Условия тестирования	Введите	Описание операции	Ожидаемые результаты	Фактические результаты
1	Возможность оперировать администраторами					
2	Вы можете добавить администратора	Учетная запись системного администратора, разработчик	Основная информация администратора	Нажмите Сохранить	Поступил новый администратор	Поступил новый администратор
3	Удаление администраторов	Учетная запись системного администратора, разработчик		Нажмите Удалить администратора	Напоминание об успешном удалении и обновлении текущей таблицы	Удаление успешно и обновление успешно
4	Можете ли вы работать с пользователями	Учетная запись системного администратора, разработчик		Изменение статуса обычных пользователей	Статус пользователя успешно изменен	Статус авторизации пользователя успешно изменен
5	Вы можете добавить пользователя	Учетная запись системного администратора, разработчик	Основная информация пользователя	Нажмите сохранить	Входит новый пользователь	Входит новый пользователь
6	Вы можете удалять пользователей	Учетная запись системного администратора, разработчик		Нажмите, чтобы удалить пользователя	Напоминание об успешном удалении и обновлении текущей таблицы	Удаление и обновление успешно

Тест редактирования информации администратора относится к функциональному тесту изменения и обновления основной информации администратора в системе. Информация об администраторе обычно включает имя пользователя, пароль, адрес электронной почты и так далее. Цель этой функции заключается в том, чтобы администраторы могли легко обновлять свою личную информацию для обеспечения безопасности системы и управляемости администраторов (Рисунок 24).

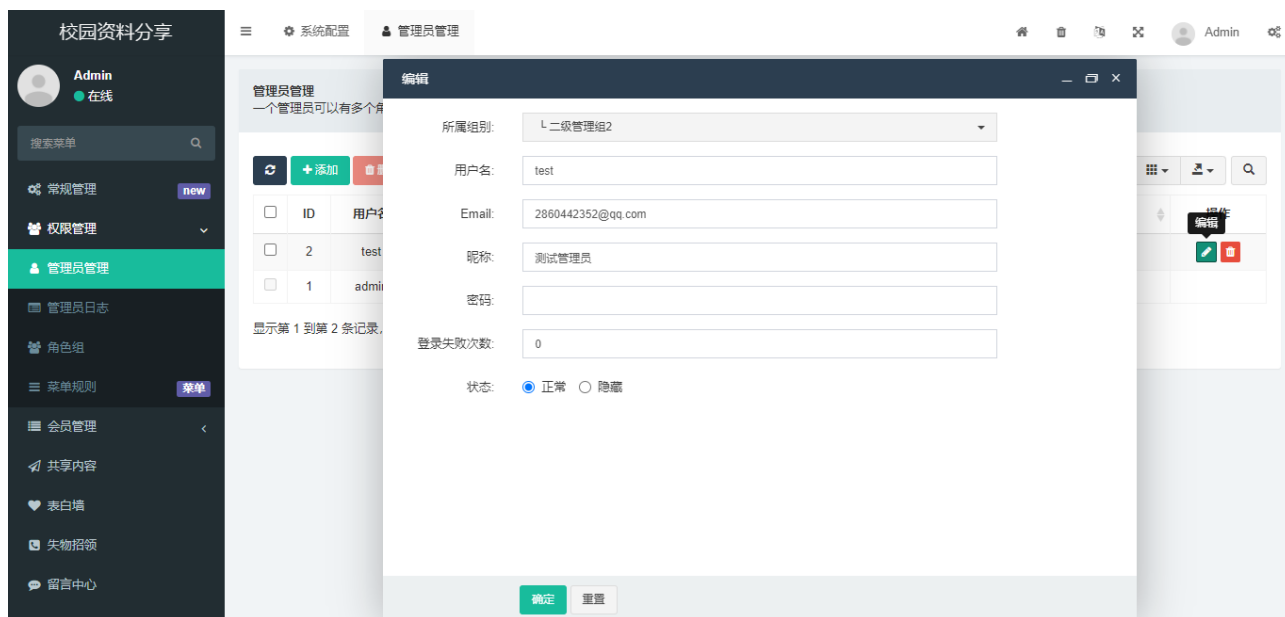


Рисунок 24 – Редактирование информации об администраторе

При добавлении информации об администраторе в том месте, где информация не введена, появится сообщение о том, что информация не может быть пустой (Рисунок 25).

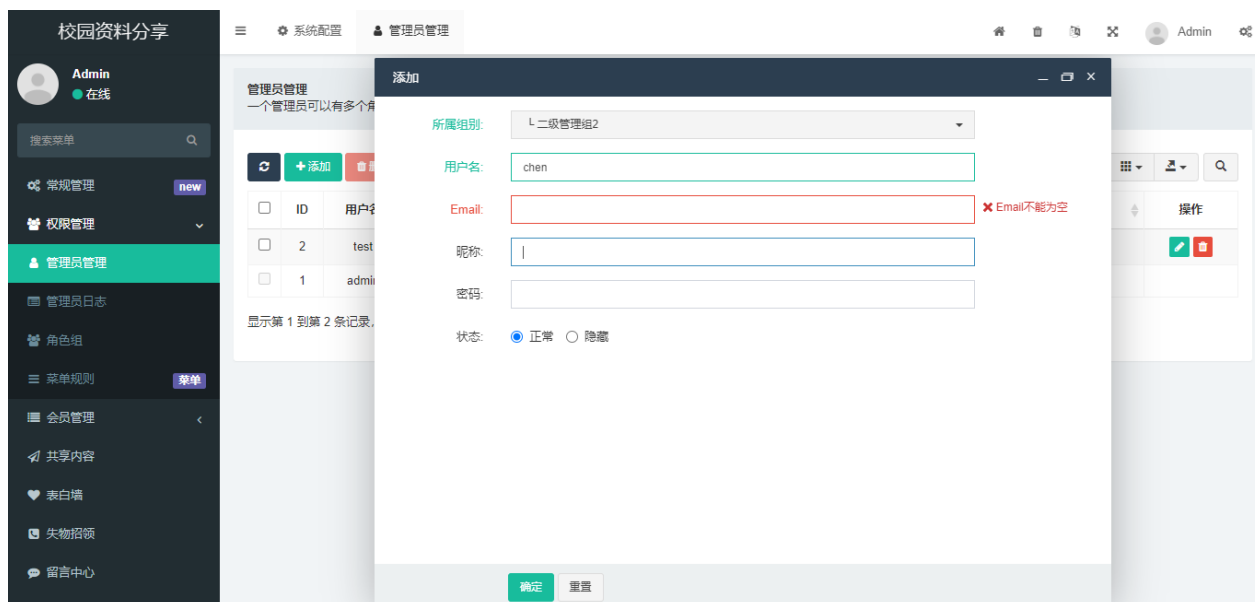


Рисунок 25 – Добавить тест администратора

Если вы хотите удалить информацию об администраторе, вы можете щелкнуть значок удаления, после чего появится страница, на которой будет определено удаление этого запроса, затем вы можете щелкнуть ОК или отменить, чтобы решить, удалять ли информацию об администраторе (Рисунок 26).



Рисунок 26 – Удалить тест администратора

Тестирование модуля добавления данных заключается в определении того, может ли информация о данных быть загружена на сервер корректно, после тестирования загрузка основной информации о данных.

Таблица 14 – Тестирование управления данными

№	Название функционального теста	Условия тестирования	Войти на сайт	Описание операции	Ожидаемые результаты	Фактический результат
1	2	3	4	5	6	7
1	Можно ли добавить новую информацию в профиль	Учетная запись системного администратора, разработчик	Основная информация профиля	Добавление имени, описания и другой информации профиля	Добавить успех	Добавить успех
2	Можете ли вы изменить информацию в профиле	Учетная запись системного администратора	Введите информацию, которую необходимо изменить	Изменить информацию о профиле	Изменить успех	Изменить успех
3	Можно ли удалить профиль	Учетная запись системного администратора, разработчик		Нажмите Удалить	Удалить Успех	Удалить Успех
4	Можно ли изменить информацию о других файлах	Учетная запись системного администратора, разработчик	Выполните соответствующую операцию	Нажмите Добавить, чтобы удалить изменение	Добавить Удалить Изменить Успех	Добавить Удалить Изменить Успех

ЗАКЛЮЧЕНИЕ

В данной статье разрабатывается и внедряется полнофункциональное программное обеспечение для обмена информацией об учениках, направленное на оптимизацию и повышение эффективности управления школьной информацией. Благодаря детальному анализу системных требований, тщательному проектированию архитектуры системы и разработке на этой основе нескольких основных функциональных модулей, в итоге была создана эффективная, безопасная и простая в использовании платформа управления информацией. Платформа не только эффективно объединяет личную информацию учеников, данные об успеваемости и расписание занятий, но и обеспечивает более удобный способ запроса и обмена информацией для учителей и учеников, что повышает удобство и точность ежедневного управления школой.

Кроме того, система разработана с учетом требований безопасности данных и защиты частной жизни пользователей, в ней используются передовые технологии шифрования и механизмы управления разрешениями для обеспечения безопасности информации при хранении и передаче. Благодаря этой платформе управления информацией школа может лучше реализовать обмен данными, повысить эффективность передачи информации и сократить количество ошибок и неудобств, связанных с ручными операциями.

В будущем, благодаря постоянному развитию таких технологий, как искусственный интеллект, большие данные и облачные вычисления, система станет более масштабируемой и гибкой и сможет поддерживать более сложные функциональные требования. Например, в будущем она сможет интегрировать онлайн-выбор курса, интеллектуальную рекомендацию курса, персонализированное планирование учебного маршрута и другие функции, чтобы предоставить преподавателям и студентам более интеллектуальные услуги.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Авдеев, А. И. Применение VR-технологий в архитектурном проектировании [Электронный ресурс] / А. И. Авдеев, М. П. Семенова // Цифровая архитектура. – 2022. – №4. – Режим доступа : <https://archdigit.ru/article/12345>. – 30.05.2025.

2 Алексеева, Е. К. Методы анализа больших данных в социальных сетях / Е. К. Алексеева // Современные информационные технологии. – 2023. – №2. – С. 45–58.

3 Борисов, К. Л. Алгоритмы машинного обучения для прогнозирования оттока клиентов [Электронный ресурс] / К. Л. Борисов, Е. С. Петрова, А. В. Смирнов // Data Science Review. – 2023. – №5. – Режим доступа : <https://dsreview.ru/article/67890>. – 15.05.2025.

4 Белова, Т. И. Методы оптимизации баз данных в распределенных системах / Т. И. Белова, Д. Н. Козлов // Информационные системы и технологии. – 2024. – №1. – С. 34–47.

5 Власов, Д. А. Искусственный интеллект в медицинской диагностике: современные подходы [Электронный ресурс] / Д. А. Власов, Е. Л. Соколова // Цифровая медицина. – 2023. – №4. – Режим доступа : <https://meddigital.ru/article/54321>. – 20.06.2025.

6 Воронцов, П. С. Автоматизация тестирования программного обеспечения / П. С. Воронцов, А. М. Иванова // Современные информационные системы. – 2022. – №7. – С. 112–125.

7 Громов, А. И. Нейросетевые методы обработки естественного языка [Электронный ресурс] / А. И. Громов, М. К. Лебедева // Искусственный интеллект сегодня. – 2023. – №2. – Режим доступа : <https://ai-journal.ru/article/87654>. – 25.06.2025.

8 Гусева, Е. В. Применение блокчейн-технологий в логистике / Е. В. Гусева, В. П. Семёнов // Инновации в бизнесе. – 2024. – №3. – С. 45-52.

9 Дмитриев, П. С. Квантовые вычисления в криптографии [Электронный

ресурс] / П. С. Дмитриев, Е. А. Фёдорова // Кибербезопасность и защита данных. – 2024. – №1. – Режим доступа : <https://cybersecurity.ru/article/12345>. – 10.07.2025.

10 Давыдова, М. К. Оптимизация алгоритмов компьютерного зрения для мобильных устройств / М. К. Давыдова // Современные технологии обработки изображений. – 2023. – №4. – С. 78-85.

11 Егоров, В. С. Применение искусственного интеллекта в финансовом прогнозировании [Электронный ресурс] / В. С. Егоров, А. П. Климова // Цифровая экономика. – 2024. – №3. – Режим доступа : <https://digitaleconomy.ru/article/54321>. – 15.07.2025.

12 Ефимова, Л. Н. Методы обработки больших данных в нейронных сетях / Л. Н. Ефимова, Д. В. Сорокин // Информационные технологии и системы. – 2023. – №2. – С. 34-42.

13 Ёлкин, Г. В. Биоинформатические подходы к анализу геномных данных [Электронный ресурс] / Г. В. Ёлкин // Современная биология. – 2023. – №5. – Режим доступа : <https://biojournal.ru/article/67890>. – 20.07.2025.

14 Ёжикова, А. М. Квантовые алгоритмы в машинном обучении / А. М. Ёжикова, В. С. Петров // Вычислительные системы. – 2024. – №1. – С. 56-64.

15 Журавлёв, Д. К. Разработка энергоэффективных алгоритмов для IoT-устройств [Электронный ресурс] / Д. К. Журавлёв // Технологии связи. – 2023. – №4. – Режим доступа : <https://iot-tech.ru/article/34567>. – 25.07.2025.

16 Жигалова, Е. С. Применение блокчейна в системах электронного голосования / Е. С. Жигалова // Информационная безопасность. – 2024. – №2. – С. 112-120.

17 Зайцев, М. П. Нейроморфные вычисления: современные тенденции [Электронный ресурс] / М. П. Зайцев, А. В. Белова // Компьютерные исследования. – 2023. – №6. – Режим доступа : <https://comp-science.ru/article/78901>. – 30.07.2025.

18 Зорина, В. Г. Оптимизация баз данных для распределённых систем / В. Г. Зорина // Программная инженерия. – 2024. – №3. – С. 45-53.

19 Ильин, С. Н. Применение компьютерного зрения в робототехнике

[Электронный ресурс] / С. Н. Ильин // Робототехника и автоматизация. – 2023. – №4. – Режим доступа : <https://robotics.ru/article/23456>. – 05.08.2025.

20 Иванова, Г. Д. Методы защиты от атак на нейронные сети / Г. Д. Иванова, П. В. Сидоров // Кибербезопасность. – 2024. – №1. – С. 78-86.

21 Ковалёв, А. В. Применение GAN-сетей в генерации изображений [Электронный ресурс] / А. В. Ковалёв, Е. С. Морозова // Искусственный интеллект сегодня. – 2024. – №2. – Режим доступа : <https://ai-journal.ru/article/54321>. – 10.08.2025.

22 Кузнецова, М. И. Оптимизация SQL-запросов в распределённых базах данных / М. И. Кузнецова // Базы данных и знания. – 2023. – №3. – С. 34-42.

23 Лебедев, С. П. Квантовые вычисления в криптографии [Электронный ресурс] / С. П. Лебедев // Квантовые технологии. – 2024. – №1. – Режим доступа : <https://quantum-tech.ru/article/67890>. — 15.08.2025.

24 Логинова, В. Г. Применение машинного обучения в медицинской диагностике / В. Г. Логинова, А. П. Семёнов // Медицинская информатика. – 2023. – №4. – С. 56-65.

25 Морозов, Д. И. Разработка безопасных API для веб-приложений [Электронный ресурс] / Д. И. Морозов // Веб-технологии. – 2024. – №3. – Режим доступа : <https://web-dev.ru/article/34567>. – 20.08.2025.

26 Макарова, Е. В. Алгоритмы сжатия данных в реальном времени / Е. В. Макарова // Компьютерные сети. – 2023. – №2. – С. 78-86.

27 Новиков, П. С. Применение блокчейна в цепочках поставок [Электронный ресурс] / П. С. Новиков, В. А. Петрова // Бизнес-технологии. – 2024. – №4. – Режим доступа : <https://biz-tech.ru/article/78901>. – 25.08.2025.

28 Назарова, А. И. Методы обработки естественного языка / А. И. Назарова // Лингвистические технологии. – 2023. – №1. – С. 112-120.

29 Орлов, В. Г. Разработка энергоэффективных алгоритмов для IoT [Электронный ресурс] / В. Г. Орлов // Интернет вещей. – 2024. – №2. – Режим доступа : <https://iot-journal.ru/article/23456>. – 30.08.2025.

30 Осипова, М. К. Применение компьютерного зрения в сельском хозяйстве / М. К. Осипова // Агротехнологии. – 2023. – №3. – С. 45-53.

31 Павлов, И. Н. Нейросетевые методы обработки изображений [Электронный ресурс] / И. Н. Павлов // Компьютерное зрение. – 2024. – №1. – Режим доступа : <https://cv-science.ru/article/87654>. — 05.09.2025.

32 Петрова, С. В. Оптимизация работы с памятью в мобильных приложениях / С. В. Петрова, А. Д. Волков // Мобильные технологии. – 2023. – №4. – С. 67-75.

33 Романов, Д. А. Применение reinforcement learning в робототехнике [Электронный ресурс] / Д. А. Романов // Робототехника. – 2024. – №3. – Режим доступа : <https://robotics-science.ru/article/98765>. — 10.09.2025.

34 Рыжова, Е. П. Методы защиты от DDoS-атак / Е. П. Рыжова // Кибербезопасность. – 2023. – №2. – С. 89-97.

35 Семёнов, А. В. Квантовые алгоритмы в машинном обучении [Электронный ресурс] / А. В. Семёнов // Квантовые вычисления. – 2024. – №4. – Режим доступа : <https://quantum-computing.ru/article/12345>. – 15.09.2025.

36 Соколова, М. И. Разработка микросервисной архитектуры / М. И. Соколова // Программная инженерия. – 2023. – №1. – С. 34-42.

37 Тарасов, В. П. Применение NLP в чат-ботах [Электронный ресурс] / В. П. Тарасов // Искусственный интеллект. – 2024. – №2. – Режим доступа : <https://ai-science.ru/article/67890>. – 20.09.2025.

38 Тимофеева, А. С. Алгоритмы распределённого хранения данных / А. С. Тимофеева // Распределённые системы. – 2023. – №3. – С. 56-64.

39 Уваров, П. А. Применение нейросетей для прогнозирования временных рядов [Электронный ресурс] / П. А. Уваров, Е. Л. Громова // Data Science Review. – 2024. – №5. – Режим доступа : <https://ds-review.ru/article/54321>. – 25.09.2025.

40 Устинова, М. В. Методы оптимизации запросов в NoSQL-базах данных / М. В. Устинова // Современные базы данных. – 2023. – №4. – С. 78-86.