Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»

ВЕБ-МОДУЛЬ АВТОМАТИЗАЦИИ ПРОЦЕССА ГЕНЕРАЦИИ ПАРОЛЕЙ ИНТЕРНЕТ-САЙТОВ «ACCOUNT PASSWORD CHANGER»

Выпускная квалификационная работа
по направлению подготовки 44.03.04 Профессиональное обучение
(по отраслям)
профилю подготовки «Информатика и вычислительная техника»
профилизации «Информационная безопасность»

Идентификационный номер ВКР: 501

Екатеринбург 2019

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»
Институт инженерно-педагогического образования
Кафедра информационных систем и технологий

К ЗАЩИТЕ ДОПУСКАЮ
Заведующий кафедрой ИС
_____ И. А. Суслова
«____» _____ 2019 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ВЕБ-МОДУЛЬ АВТОМАТИЗАЦИИ ПРОЦЕССА ГЕНЕРАЦИИ ПАРОЛЕЙ ИНТЕРНЕТ-САЙТОВ «ACCOUNT PASSWORD CHANGER»

Исполнитель:
обучающийся группы ИБ-402 А. С. Самойлов
Руководитель:
ст. преподаватель каф. ИС С. В. Ченушкина
Нормоконтролер:
ст. преподаватель каф. ИС Н. В. Хохлова

Екатеринбург 2019

АННОТАЦИЯ

Выпускная квалификационная работа состоит из веб-платформы и пояснительной записки на 61 странице, содержащей 29 рисунков, 31 источник литературы, а также 2 приложения на 3 страницах.

Ключевые слова: АУТЕНТИФИКАЦИЯ, ПАРОЛЬ, БЕЗОПАСНОСТЬ

Самойлов А. С., Веб-модуль автоматизации процесса генерации паролей интернет-сайтов «Account password changer»: выпускная квалификационная работа / А. С. Самойлов; Рос. гос. проф.-пед. ун-т, Ин-т инж.-пед. образования, Каф. информ. систем и технологий. — Екатеринбург, 2019. — 61 с.

Объектом исследования является процесс аутентификации как часть безопасности интернет-сайта.

Предметом исследования является повышение надежности аутентификации интернет-сайтов с использованием автоматизации процесса генерации паролей.

Цель работы — разработать веб-модуль автоматизации процесса генерации паролей интернет-сайтов «Account Password Changer». Для достижения цели были проанализированы особенности аутентификация на различных системах управления контента сайтов; проанализированы литературные и интернет-источники по теме аутентификации веб-сайтов, надежности паролей; написана электронная инструкция по использованию веб-модуля.

На момент разработки веб-модуля аналогов схожих с разработанным веб-модулем нет. Данный продукт является новым решением безопасности аккаунтов и их конфиденциальных данных, использующий в своей работе новейшие, скоростные, надежные технологии в сфере веб-разработки.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В современном мире информационные системы различной значимости и охвата стали неотъемлемой частью инфраструктуры государства, бизнеса, и общества в широком смысле этого слова. С каждым днем в информационные системы переносится все больше защищаемой информации. Современные технологии в информационном пространстве обеспечивают новые возможности ведения бизнеса, государственной и общественной деятельности, и вместе с тем создают значительные требования в обеспечении информационной безопасности с целью защиты этой информации. Известно, что более 25 % злоупотреблений информацией в информационных системах совершаются локальными пользователями, партнерами и поставщиками услуг, имеющими прямой доступ к информационной системе. До 70 % из них — случаи несанкционированного получения прав доступа, хищении и передачи конфиденциальной информации пользователей информационных систем. Это становится возможным в связи несовершенством технологий авторизации и аутентификации пользователей информационных систем. Совершенствование методов систем учета доступа и регистрации пользователей является одним из главных приоритетов совершенствования информационных систем. Основными процедурами регистрации пользователей в информационных системах являются процедура идентификации — получение ответа на вопрос «Кто Вы?» и аутентификации — доказательства того, что «Вы именно тот, кем представляетесь». Несанкционированное получение прав доступа злоумышленником к информационной системе связано, в первую очередь, с нарушением прохождения процедуры аутентификации.

Процесс регистрации пользователя в информационной системе состоит из трех взаимосвязанных, выполняемых последовательно процедур: идентификации, аутентификации и авторизации. Идентификация — это процедура распознавания субъекта по его идентификатору. В процессе регистрации

субъект предъявляет системе свой идентификатор и она проверяет его наличие в своей базе данных. Субъекты с известными системе идентификаторами считаются легальными, остальные субъекты относятся к нелегальным. Аутентификация (установление подлинности) — процедура проверки принадлежности субъекту доступа предъявленного им идентификатора и подтверждение его подлинности. Другими словами, аутентификация заключается в проверке: является ли подключающийся субъект тем, за кого он себя выдает. Авторизация — процедура предоставления определённому лицу или группе лиц прав на выполнение определённых действий; после прохождения им процедуры аутентификации. Часто можно услышать выражение, что какой-то человек «авторизован» для выполнения данной операции — это значит, что он имеет на неё право.

На сегодняшний момент аутентификация с помощью логина и пароля является наиболее распространенной, прежде всего, благодаря простоте использования. Однако данный вид аутентификации имеет некоторые недостатки. В отличие от случайно формируемых криптографических ключей, пароли пользователя бывает возможно подобрать из-за безответственного отношения подавляющего большинства пользователей к генерированию пароля. Выбор пользователями легко угадываемых паролей встречается чаще всего. Существуют и свободно доступны различные утилиты подбора паролей, в том числе, специализированные для конкретных широко распространенных программных средств. Пароль может быть подсмотрен или перехвачен при вводе. Начинающие пользователи персональных компьютеров используют один и тот же пароль для большинства своих аккаунтов в сети, для защиты учетной записи. Это одна из самых распространённых ошибок, так как злоумышленники могут скомпрометировать его и, разумеется, воспользоваться им. В связи с этим, пользователи должны задумываться о том, чтобы всегда устанавливать (или хотя бы стараться) разные пароли на учетные записи, аккаунты в Интернете. Кроме того, одна из теорий, подтвержденных на практике уже не один раз гласит, что пароль периодически нужно менять.

Периодическая смена паролей позволит более эффективно защитить аккаунт и конфиденциальную информацию на нем, а кроме того, это усложнит процесс взлома злоумышленникам, так как старая информация, которая возможно, была ими найдена уже будет неактуальной.

Объектом исследования — процесс аутентификации как часть безопасности интернет-сайта.

Предметом исследования является повышение надежности аутентификации интернет-сайтов с использованием автоматизации процесса генерации паролей.

Цель настоящего исследования — разработать веб-модуль автоматизации процесса генерации паролей интернет-сайтов «Account Password Changer».

В соответствии с поставленной целью в работе определены следующие задачи:

1. Проанализировать литературу и интернет-источники, посвященные аутентификации интернет-сайтов.
2. Подготовить проект веб-модуля с учетом специфики аутентификации различных системах управления контента (CMS) для создания интернет-сайтов.
3. Реализовать функционал веб-модуля в выбранных технологиях.
4. Подготовить интернет-сайт для демонстрации работы веб-модуля.
5. Разработать обучающий материал по организации процесса аутентификации с использованием веб-модуля.

1 РОЛЬ АУТЕНТИФИКАЦИИ В ЦИФРОВОМ ПРОСТРАНСТВЕ

1.1 Статистика взлома паролей сайтов

Sucuri опубликовала результаты анализа данных, собранных ею в ходе расследования случаев компрометации сайтов, произошедших в третьем квартале прошлого года [10].

За указанный период было изучено состояние более 8 тыс. атакованных ресурсов; 74 % из них используют Wordpress, 17 % — Joomla, 6 % — Magento, 2 % — Drupal. Эксперты подчеркивают, что в компрометации сайтов, как правило, повинно не основное приложение CMS, а неправильное его развертывание, ошибки в конфигурации или нарушение порядка обслуживания веб-мастером.

Согласно новой статистике Sucuri, на момент исследования устаревшие версии CMS или неактуальное окружение (при наличии патчей) использовали 61 % сайтов Wordpress, 84 % Joomla, 86 % Drupal и 94 % Magento. Для Wordpress- и Drupal-сайтов это ухудшение показателя по сравнению с предыдущим кварталом, в первом случае на 6 процентных пунктов, во втором — на 2.

Рейтинг плагинов, уязвимости которых хакеры используют в атаках на Wordpress, по-прежнему возглавляют Revslider (8,5 % случаев взлома), TimThumb (5,9 %) и GravityForms (4 %). Тем не менее показатели Revslider и GravityForms за квартал улучшились на 1,5 и 2 п.п. соответственно. Sucuri ожидает, что этот тренд сохранится и в дальнейшем, так как владельцы сайтов и операторы хостов стали уделять больше внимания патчингу такого программного обеспечения (ПО).

Ситуация с TimThumb, к сожалению, осталась практически неизменной; исследователи полагают, что многие веб-мастера просто не в курсе, что

этот скрипт присутствует на сайте. Аналогичная судьба уготована, видимо, и Revslider: этот плагин устанавливается как часть темы, о чем может не знать владелец сайта, и в дальнейшем атаки на уязвимости в Revslider могут участиться.

В черные списки занесены лишь около 15 % сайтов, обследованных Sucuri. При этом Google Safe Browsing распознает 69 % из них, Norton SafeWeb — 24 %, McAfee SiteAdvisor — 10 %. Приводя столь неутешительную статистику, эксперты подчеркивают: нельзя полагаться только на традиционные инструменты веб-мастера вроде Google или Bing, нужно постоянно контролировать свое веб-имущество.

Из неприятных находок на взломанных сайтах наиболее часто присутствовали скрытые hypertext preprocessor (PHP) бэкдоры, в 37 % случаев исследователи обнаружили search engine optimization (SEO) спам. На 12 % повысилась доля скриптов, используемых для рассылки спама, взломы с целью распространения вредоносного ПО, напротив, заметно сократились.

В квартальном отчете Sucuri появился также новый набор данных — по файлам, которые хакеры обычно модифицируют при взломе. Как оказалось, наиболее часто в ходе атаки изменениям подвергаются index.php (25 %), header.php (20 %) и .htaccess (10 %).

1.2 Технологии реализации аутентификации на интернет-сайтах

Аутентификация в Интернет требуется при доступе к таким интернет-сервисам как: электронная почта; веб-форум; социальные сети; интернет-банкинг; платежные системы; корпоративные сайты; интернет-магазины.

Положительным результатом аутентификации (кроме установления доверительных отношений и выработки сессионного ключа) является авторизация пользователя, то есть предоставление ему прав доступа к ресурсам, определенным для выполнения его задач.

Основные реализуемые методы:

1. По паролю:
    - HTTP authentication;
    - forms authentication;
    - URL query;
    - request body;
    - HTTP header;
2. По сертификатам.
3. По одноразовым паролям.
4. По ключам доступа.

Аутентификация по паролю — метод основанный на том, что пользователь должен предоставить username и password для успешной идентификации и аутентификации в системе. Пара username/password задается пользователем при его регистрации в системе, при этом в качестве username может выступать адрес электронной почты пользователя.

Применительно к веб-приложениям, существует несколько стандартных протоколов для аутентификации по паролю, которые мы рассмотрим ниже.

Hyper text transfer protocol (HTTP) authentication — протокол, описанный в стандартах HTTP 1.0/1.1, существует очень давно и работает следующим образом:

1. Сервер, при обращении неавторизованного клиента к защищенному ресурсу, отсылает HTTP статус «401 Unauthorized» и добавляет заголовок «WWW-Authenticate» с указанием схемы и параметров аутентификации.

2. Браузер, при получении такого ответа, автоматически показывает диалог ввода username и password. Пользователь вводит детали своей учетной записи.

3. Во всех последующих запросах к этому веб-сайту браузер автоматически добавляет HTTP заголовок «Authorization», в котором передаются данные пользователя для аутентификации сервером.

4. Сервер аутентифицирует пользователя по данным из этого заголовка. Решение о предоставлении доступа (авторизация) производится отдельно на основании роли пользователя, access control list (ACL) или других данных учетной записи.

Весь процесс стандартизирован и хорошо поддерживается всеми браузерами и веб-серверами. Существует несколько схем аутентификации, отличающихся по уровню безопасности:

Basic — наиболее простая схема, при которой username и password пользователя передаются в заголовке Authorization в незашифрованном виде (base64-encoded). Однако при использовании HTTPS (HTTP over SSL) протокола, является относительно безопасной.

Digest — challenge-response-схема, при которой сервер посылает уникальное значение nonce, а браузер передает MD5 хэш пароля пользователя, вычисленный с использованием указанного nonce. Более безопасная альтернативв Basic схемы при незащищенных соединениях, но подвержена man-in-the-middle attacks (с заменой схемы на basic). Кроме того, использование этой схемы не позволяет применить современные хэш-функции для хранения паролей пользователей на сервере. Network lan manager (NTLM) (известная как Windows authentication) — также основана на challenge-response подходе, при котором пароль не передается в чистом виде. Эта схема не является стандартом HTTP, но поддерживается большинством браузеров и веб-серверов. Преимущественно используется для аутентификации пользователей Windows Active Directory в веб-приложениях. Уязвима к pass-the-hash-атакам.

Negotiate — еще одна схема из семейства Windows authentication, которая позволяет клиенту выбрать между NTLM и Kerberos аутентификацией. Kerberos — более безопасный протокол, основанный на принципе Single Sign-On. Однако он может функционировать, только если и клиент, и сервер находятся в зоне intranet и являются частью домена Windows.

Стоит отметить, что при использовании HTTP-аутентификации у пользователя нет стандартной возможности выйти из веб-приложения, кроме как закрыть все окна браузера.

Forms authentication — для этого протокола нет определенного стандарта, поэтому все его реализации специфичны для конкретных систем, а точнее, для модулей аутентификации фреймворков разработки.

Работает это по следующему принципу: в веб-приложение включается HTML-форма, в которую пользователь должен ввести свои username/password и отправить их на сервер через HTTP POST для аутентификации. В случае успеха веб-приложение создает session token, который обычно помещается в browser cookies. При последующих веб-запросах session token автоматически передается на сервер и позволяет приложению получить информацию о текущем пользователе для авторизации запроса.

Приложение может создать session token двумя способами:

1. Как идентификатор аутентифицированной сессии пользователя, которая хранится в памяти сервера или в базе данных. Сессия должна содержать всю необходимую информацию о пользователе для возможности авторизации его запросов.

2. Как зашифрованный и/или подписанный объект, содержащий данные о пользователе, а также период действия. Этот подход позволяет реализовать stateless-архитектуру сервера, однако требует механизма обновления сессионного токена по истечении срока действия.

Необходимо понимать, что перехват session token зачастую дает аналогичный уровень доступа, что и знание username/password. Поэтому все коммуникации между клиентом и сервером в случае forms authentication должны производиться только по защищенному соединению HTTPS.

Аутентификация по сертификатам — представляет собой набор атрибутов, идентифицирующих владельца, подписанный certificate authority (CA). CA выступает в роли посредника, который гарантирует подлинность сертификатов (по аналогии с Федеральной миграционной службой (ФМС), выпус-

кающей паспорта). Также сертификат криптографически связан с закрытым ключом, которых хранится у владельца сертификата и позволяет однозначно подтвердить факт владения сертификатом.

На стороне клиента сертификат вместе с закрытым ключом могут храниться в операционной системе, в браузере, в файле, на отдельном физическом устройстве (smart card, USB token). Обычно закрытый ключ дополнительно защищен паролем или PIN-кодом.

В веб-приложениях традиционно используют сертификаты стандарта X.509. Аутентификация с помощью X.509-сертификата происходит в момент соединения с сервером и является частью протокола secure sockets layer (SSL) / transport layer security (TLS). Этот механизм также хорошо поддерживается браузерами, которые позволяют пользователю выбрать и применить сертификат, если веб-сайт допускает такой способ аутентификации.

Использование сертификатов для аутентификации — куда более надежный способ, чем аутентификация посредством паролей. Это достигается созданием в процессе аутентификации цифровой подписи, наличие которой доказывает факт применения закрытого ключа в конкретной ситуации (non-repudiation). Однако трудности с распространением и поддержкой сертификатов делает такой способ аутентификации малодоступным в широких кругах.

Аутентификация по одноразовым паролям — обычно применяется дополнительно к аутентификации по паролям для реализации two-factor authentication (2FA). В этой концепции пользователю необходимо предоставить данные двух типов для входа в систему: что-то, что он знает (например, пароль), и что-то, чем он владеет (например, устройство для генерации одноразовых паролей). Наличие двух факторов позволяет в значительной степени увеличить уровень безопасности, что может быть востребовано для определенных видов веб-приложений.

Аутентификация по ключам доступа — чаще всего используется для аутентификации устройств, сервисов или других приложений при обращении

к веб-сервисам. Здесь в качестве секрета применяются ключи доступа (access key, API key) — длинные уникальные строки, содержащие произвольный набор символов, по сути заменяющие собой комбинацию username/password.

В большинстве случаев, сервер генерирует ключи доступа по запросу пользователей, которые далее сохраняют эти ключи в клиентских приложениях. При создании ключа также возможно ограничить срок действия и уровень доступа, который получит клиентское приложение при аутентификации с помощью этого ключа.

Аутентификация по токенам — чаще всего применяется при построении распределенных систем Single Sign-On (SSO), где одно приложение (service provider или relying party) делегирует функцию аутентификации пользователей другому приложению (identity provider или authentication service). Типичный пример этого способа — вход в приложение через учетную запись в социальных сетях. Здесь социальные сети являются сервисами аутентификации, а приложение доверяет функцию аутентификации пользователей социальным сетям.

Другие протоколы аутентификации по паролю. Два протокола, описанных выше, успешно используются для аутентификации пользователей на веб-сайтах. Но при разработке клиент-серверных приложений с использованием веб-сервисов (например, iOS или Android), наряду с HTTP аутентификацией, часто применяются нестандартные протоколы, в которых данные для аутентификации передаются в других частях запроса.

Существует всего несколько мест, где можно передать username и password в HTTP запросах:

URL query — считается небезопасным вариантом, т. к. строки uniform resource locator (URL) могут запоминаться браузерами, прокси и веб-серверами.

Request body — безопасный вариант, но он применим только для запросов, содержащих тело сообщения (такие как POST, PUT, PATCH).

HTTP header — оптимальный вариант, при этом могут использоваться и стандартный заголовок Authorization (например, с Basic-схемой), и другие произвольные заголовки.

1.3 Технологии реализации систем аутентификации на различных системах управления контентом

Как известно, все CMS хранят пароли пользователей не в открытом виде, а в виде хэшей в базе данных. В основном для достижения данной цели используют такие алгоритмы хеширования как MD5 или Blowfish (bCrypt).

MD5 — алгоритм хеширования, разработанный профессором Р. Л. Ривестом в еще 1991 году. Алгоритм md5 шифрует любые данные в формате 128-bit hash (контрольную сумму). Алгоритм используется для проверки подлинности данных, когда происходит их передача в зашифрованном виде.

Blowsifh — криптографический алгоритм, реализующий блочное симметричное шифрование с переменной длиной ключа. Разработан Брюсом Шнайером в 1993 году. Blowfish был одним из первых защищенных блочных шифров, на которые не распространялись никакие патенты, и поэтому он был доступен для свободного использования. Это преимущество способствовало его популярности в криптографическом программном обеспечении. Bcrypt — это функция хеширования пароля, которая в сочетании с переменным числом итераций использует фазу настройки ключа Blowfish для увеличения рабочей нагрузки и продолжительности вычислений хэша, еще больше снижая угрозы от атак методом перебора [27].

Механизм CMS «Wordpress».

На сегодняшний момент Wordpress по умолчанию все еще использует алгоритм MD5 для хэшей с помощью функции wp_hash_password(). Данная функция шифрует переданный текст. Шифр всегда получается уникальный и для шифрования паролей. Хэш всегда получается разный, т.е. если 2 раза

одинаково вызвать функцию, результаты будут отличаться. В Wordpress имеется возможность обновить тип хеширования паролей до blowfish (bCrypt) или расширенного DES.

Механизм CMS «Joomla»

В более новых версиях ввели хэш пароля на базе встроенного в PHP BCrypt. Так же, в новых версиях Joomla можно не переживать по поводу, какой хэш генерировать, поскольку система определяет тип хэша на основе его длины и сможет сама понять, какой алгоритм использовался при создании хэша. Когда пользователь регистрируется на сайте, в базе данных joomla_users в таблице паролей, хранятся пароли в следующих форматах:

- PDo8QrURFT1r0NlWf0X/GrDF/aMqwqK/;
- PDH38Lch9z508gJiop3A6u0whTity390.

До версии 3.2 Joomla также использовала MD5 хэш. Более старый алгоритм также отлично работает в последней версии, только разница в более старой версии та, что создает пароль на 65 символов, а новый создает 34 символа.

Механизм CMS «UMI CMS»

В UMI.CMS встроен механизм разграничения полномочий по использованию функционала системы на основании разрешений, даваемых пользователям и группам пользователей на доступ к определенным методам модулей и элементам иерархии.

Если пользователь правильно передал системе идентификационные данные, UMI.CMS пытается произвести его аутентификацию. Прежде всего система проверяет наличие соответствующего внутреннего аккаунта, затем (если не удалось) — последовательно ищет аккаунты в AD, OpenId, PhpBB, IPB и пытается соотнести их со своими внутренними аккаунтами. В любом случае, в результате аутентификации CMS запоминает в сессии идентификатор одной из собственных учетных записей. Если системе не удается сопоставить запросу ни один из аккаунтов, в качестве текущей считается учетная запись «Гость». Во всех последующих запросах клиента при работе от лица

той же учетной записи идентификационные данные передавать не следует, поскольку аутентификация будет производиться средствами механизма сессий, при необходимости же «сменить пользователя» клиент должен просто передать новые идентификационные данные.

Если программа-клиент идентифицируется двумя способами одновременно — и посредством «прозрачной преавторизации», и при помощи /users/login_do, — то система выполнит оба этих запроса именно в такой последовательности: сначала «преавторизацию», затем /users/login_do. Таким образом, если будут переданы верные идентификационные данные двух существующих, но различных учетных записей, итоговая аутентификация будет осуществлена согласно данным, переданным в метод /users/login_do.

Механизм CMS «OpenCart CMS»

В CMS OpenCart используется Digest аутентификация. Digest-аутентификация представляет собой более продвинутый и сложный вид аутентификации, чем Basic-аутентификация. Главным отличием здесь является то, что логин-пароль пользователя пересылаются через сеть не в открытом виде, а шифруются по алгоритмуMD5. Настройка Digest-аутентификации похожа на настройку Basic-аутентификации. Один сайт может одновременно использовать несколько систем защиты, например Basic и Digest

Механизм CMS «Drupal CMS»

Первый пользователь, учетная запись которого будет создана в системе Drupal, является главным администратором сайта. Этот пользователь обладает максимальными правами доступа к сайту на Drupal. Для главного администратора Drupal никогда не выполняются проверки безопасности, поэтому, зайдя от имени главного администратора, легко можно нарушить работу сайта, удалив какие-либо необходимые данные. Поэтому лучше всего заходить от имени администратора на сайт только при необходимости, а для добавления и редактирования содержимого создать новую учетную запись с ограниченными правами доступа.

Любой создаваемой учетной записи на сайте автоматически присваивается роль «зарегистрированный пользователь», а любому посетителю, для которого еще не создана учетная запись (или который еще не прошел процедуру аутентификации со своим именем пользователя и паролем), присваивается роль «анонимный пользователь».

1.4 Анализ исходных данных по теме исследования

1.4.1 Анализ печатных и интернет-источников по аутентификации

Аутентификация: от паролей до открытых ключей. Автор — Ричард Э. Смит — Данная книга является первым в своем роде изданием, посвященным такому важному аспекту проблемы управления доступом к информации в компьютерных системах, как аутентификация пользователей [6].

«Аутентификация. Теория и практика обеспечения безопасного доступа к информационным ресурсам». Авторы — А. Шелупанов, С. Груздев, Ю. Нахаев. Книга посвящена одному из аспектов проблемы управления доступом к информации в компьютерных системах — аутентификации. Фактически защита информации начинается с аутентификации пользователей [5].

«Метод аутентификации с использованием динамических ключей». Автор — М. А. Стюгин. Многоразовые пароли — самый популярный способ аутентификации на сегодняшний день, однако при этом — самый небезопасный. В данной работе представлен метод аутентификации с использованием многоразовых паролей, существенно усложняющий реализацию атак, следствием которых является получение информации, достаточной для подбора паролей [17].

«Основы информационной безопасности автоматизированных систем: краткий курс». Автор — В. Л. Цирлов. В книге рассматриваются основные теоретические построения, лежащие в основе современных систем обеспечения информационной безопасности. Приводятся основы теории информаци-

онной безопасности, элементы формальной теории защиты информации, а также основные оценочные и управленческие стандарты в области информационной безопасности [22].

На сайте habr.ru в статье «Обзор способов и протоколов аутентификации в веб-приложениях» повествуется о применении различных способов аутентификации для веб-приложений, включая аутентификацию по паролю, по сертификатам, по одноразовым паролям, по ключам доступа и по токенам. Так же рассматриваются технологии единого входа (Single Sign-On), различные стандарты и протоколы аутентификации [20].

В статье «Аутентификация в HTTP. Проверка подлинности пользователей в HTTP» автор статьи дает информацию о протоколе HTTP в рубрике «Серверы и протоколы». Дает объяснение как реализована аутентификация клиентов на сервере в HTTP протоколе и как происходит проверка подлинности клиента, так же рассмотрено как происходит базовая аутентификации клиента на сервере [3].

В статье «Аутентификация пользователя. Веб» автор повествует о классификации методов аутентификации. Привод пример различных видов аутентификации, такие как: Базовая, Дайджест, HTTPS, по предъявлению цифрового сертификата [4].

В статье на habr.com «Как написать инструкцию так, чтобы тебя поняли» раскрываются рекомендации по написанию эффективной инструкции с пошаговыми рекомендациями и правилами [14].

В статье «Как написать инструкцию по эксплуатации товара?» описывается как нужно структурировать инструкцию, о позитивном и корректном настрое по ходу всей инструкции. Так же рассматриваются правила «Нет трем „С“» [10].

В статье «Как написать понятное руководство — советы для копирайтеров» повествуется о том, что важно учитывать особенности целевой аудитории. Так же о подборе стиля текста для оформления электронного пособия.

«Инструкция без иллюстраций — плохая инструкция» — утверждается в данном руководстве [15].

В статье на habr.com «Написание инструкций пользовательского интерфейса» описываются такие рекомендации как особенности аудитории, баланс между «водой» и сутью, эстетике, эффективных и точных словах. Резюмируя, автор дает рекомендацию о том, что нужно проверять свои инструкции [16].

В статье «Организационно-распорядительные документы» подробно описаны правила составления и оформления инструкций, Регламент, правила подготовки и оформления. Так же приведены примеры шаблонов [16].

В электронном документе «Электронный документ. Правила его составления и оформления. Электронная цифровая подпись» приведены аргументы в пользу преимуществ использования электронных документов для пользователей [22].

В статье на habr.com «Примеры и рекомендации удобных инструкций» расписаны правила, которые помогут создать рабочий и удобный электронный документ [23].

1.4.2 Анализ нормативных источников

Для создания электронной инструкции были отобраны источники литературы и интернет-источники по созданию электронной инструкции.

ГОСТ 2.601-2013 был разработан Федеральным государственным унитарным предприятием «Всероссийский научно-исследовательский институт стандартизации и сертификации в машиностроении» (ВНИИНМАШ), Автономной некоммерческой организацией «Научно-исследовательский центр CALS-технологий "Прикладная логистика"» (АНО "НИЦ CALS-технологий" «Прикладная логистика»). Настоящий стандарт устанавливает общие требования к выполнению электронных конструкторских документов изделий всех отраслей промышленности. На основе настоящего стандарта могут быть раз-

работаны стандарты с учетом особенностей применения и обращения различных видов электронных конструкторских документов [1].

ГОСТ Р 1.0-2004 Стандартизация в Российской Федерации. Разработан Федеральным государственным унитарным предприятием Всероссийский научно-исследовательский институт стандартизации (ФГУП ВНИИ Стандарт). Положения основополагающих стандартов системы стандартизации в Российской Федерации разработаны для применения федеральными органами исполнительной власти, субъектами хозяйственной деятельности, техническими комитетами по стандартизации, общественными объединениями и заинтересованными лицами [2].

ГОСТ Р 1.5-2004 Стандартизация в Российской Федерации. Стандарты национальные Российской Федерации. Правила построения, изложения, оформления и обозначения. Настоящий стандарт устанавливает правила построения, изложения, оформления и обозначения национальных стандартов Российской Федерации, общие требования к их содержанию, а также правила оформления и изложения изменений к национальным стандартам Российской Федерации [3].

1.4.3 Анализ решений по хранению и управлению паролями

Перед началом проектирования веб-модуля безопасности были рассмотрены и проанализированы существующие решения по хранению и управлению паролями. Аналогичных сервисов разработанному веб-модулю безопасности не найдено, поэтому были рассмотрены аналогичные решения по хранению и управлению паролями.

Программа для хранения паролей «KeePass Password Safe» — это кроссплатформенная свободная программа для хранения паролей, распространяемая по лицензии general public license (GPL). Программа разработана Домиником Райхлом, изначально только для операционной системы

Windows. KeePass поддерживает алгоритмы Advanced Encryption Standard и Twofish для шифрования паролей своих баз данных (рисунок 1).

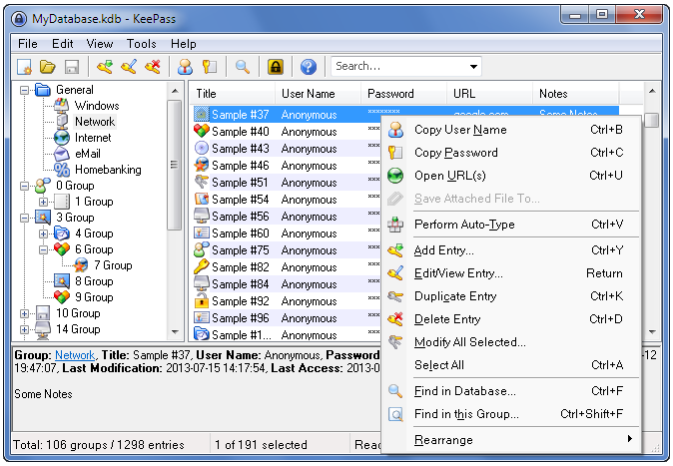

Рисунок 1 — KeePass Password Safe

Данное ПО обладает следующими функциями:

- создание записи;
- дублирование записи;
- сортировка записей;
- группировка записей;
- поиск по записям;
- копирование данных записи;
- autotype для автоматического ввода данных в браузерах и других программах;
- хранение дат;
- генератор паролей с заданными параметрами;
- кнопка блокировки;
- настройки базы и программы;
- смена мастер-пароля;
- триггеры;
- файлы экспорта: *.txt, *.html, *.xml, *.csv;
- 35 форматов импорта;
- перенос базы данных.

Менеджер паролей «eWallet» — существует в версиях для iPhone, iPad, Mac, Windows — не ниже 7. Windows-версия eWallet интегрируется с браузерами Internet Explorer, Firefox и Chrome, а версия для OS X — только с Safari. Программа платная, но есть тестовая версия на 30 дней.

Данное ПО обладает следующими функциями:

- дерево категорий;
- добавление карточки с кастомизацией;
- кастомизация полей в карточках;
- более 30 шаблонов — кредитные карты, пароли, банковские данные;
- статистика по дереву — счетчики по типам информации;
- список последних карточек;
- генератор паролей;
- «живые поля», например, звонок по клику на поле номера;
- auto pass — автоматическая подстановка логина и пароля в поле браузера;
- smart copy — быстрое копирование карточных номеров.

Менеджер паролей «RoboForm» — Windows и Mac OS X, приложения для мобильных платформ iOS, Android, BlackBerry, Windows Mobile, Palm OS и Symbian. При этом десктопная версия не предполагает импорта данных из Chrome (рисунок 2).

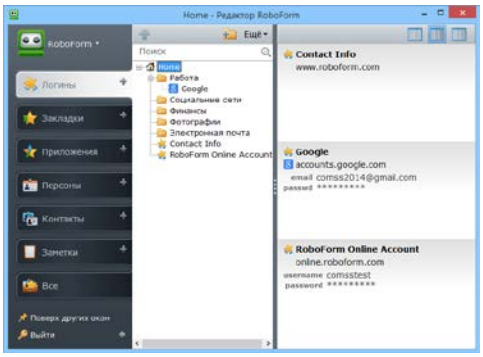

Рисунок 2 — RoboForm

Данное ПО обладает следующими функциями:

- создание записей;
- поиск;
- печать — логины, персоны, заметки;
- auto login;
- генератор паролей;
- дополнительная защита каждой записи. для открытия данных требуется ввод мастер-пароля;
- импорт введенных данных из браузера;
- отправка записей по e-mail — требуется ввод мастер-пароля;
- создание ярлыков на рабочем столе и в браузере;
- интеграция с windows login;
- возможность открытия нескольких окон программы;
- администрирование: создание, редактирование групп пользователей, шаринг записей на группы пользователей, синхронизация баз при редактировании пользователями, проверка, к каким записям обращался конкретный пользователь;
- профили разных пользователей на одной копии программы;
- бэкапы;
- портативная версия базы, которую можно хранить на флешке;
- управляющие символы юникода;
- заполнение длинных форм в интернет-магазинах одним кликом;
- экранная клавиатура.

Диспетчер паролей «LastPass» — это расширение для основных браузеров — Internet Explorer (IE), Firefox, Safari, Chrome, Opera, работает только в сети. Существуют версии для мобильных устройств iPhone, iPad, Android, BlackBerry и других. Возможно управление базой паролей через веб-интерфейс на сайте LastPass — это довольно удобно. Есть портативный клиент для Windows — загружаете базу, после чего можете использовать ее офлайн (рисунок 3).

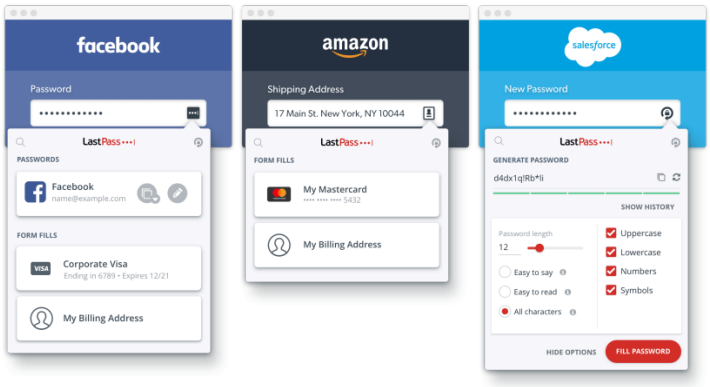

Рисунок 3 — LastPass

Данное ПО обладает следующими функциями:

- поиск;
- автозаполнение;
- многофакторная аутентификация;
- защита учетных данных от кражи;
- поиск незащищенных объектов на персональном компьютере (ПК);
- импорт из другого менеджера паролей;
- запрещенные адреса;
- обмен паролями;
- синхронизация;
- отчеты и управление пользователями;
- закладки;
- экранная клавиатура.

Программа бесплатна, но за дополнительные функции вроде выявления слабых паролей, экранной клавиатуры, защиты от фишинга придется платить.

Хранилище паролей «1Password» — эта программа доступна на всех популярных платформах: Mac, Windows, iOS, Android. С её помощью можно надёжно хранить пароли, данные кредитных карт, баз данных, паспортную информацию, лицензии сторонних приложений и защищённые заметки. Сервис может помочь сгенерировать сложный пароль и синхронизировать сохранённую информацию с другими вашими устройствами посредством Dropbox, iCloud или Wi-Fi Sync. В приложении есть встроенный браузер, ко-

торый позволит не переживать о сохранности данных ваших кредитных карточек при совершении покупок (рисунок 4).

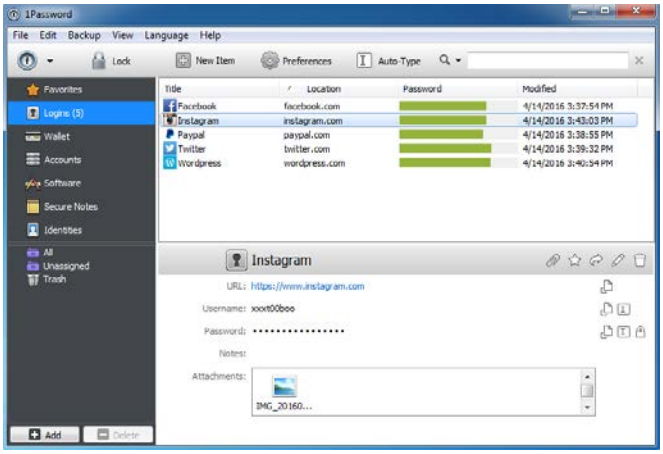

Рисунок 4 — 1Password

Данное ПО обладает следующими функциями:

- создание записей;
- навигационная колонка;
- поиск;
- избранное;
- генератор паролей;
- модуль для браузеров;
- security-аудит;
- очистка буфера обмена через определенный промежуток времени;
- корзина;
- импорт;
- экспорт.

Таким образом, мы рассмотрели программы для хранения паролей на компьютере и в браузере. Необходимо, ответственно подходить вопросу безопасности, иначе злоумышленники, попытаются воспользоваться Вашими конфиденциальными данными. Но, и здесь нет гарантии полной безопасности.

2 ОПИСАНИЕ МОДУЛЯ С ОБУЧАЮЩЕЙ ИНСТРУКЦИЕЙ

2.1 Постановка задачи и общие требования

Пароли нужно менять регулярно. Необходимо понимать, что абсолютно защищенных систем не бывает, и пароли нередко утекают с серверов компаний. Так же, после кражи базы данных она долгое время может висеть на хакерских биржах, а значит, своевременная замена пароля может предотвратить потерю доступа к аккаунту.

Самая распространенная ошибка — использование одного пароля на разных сервисах. Злоумышленники могут просто взломать сайт доставки пиццы, а в следствии получают пароль от вашей главной почты и банковского приложения. По данным компании Thycotic, даже в среде специалистов по информационной безопасности более половины респондентов не меняли пароли от социальных сетей более года, а каждый пятый — не менял вообще [16].

Разработанный веб-модуль «Account Password Changer» (APC) предназначен для администраторов веб-приложений использующих в своей работе политику аутентификации и авторизации аккаунтов пользователей с помощью логина и пароля. Суть данного модуля заключается в удаленной, регулярной, безопасной смене паролей аккаунтов пользователей и отправку нового пароля в электронном письме на почтовый ящик пользователя. Регулярная смена пароля аккаунта гарантирует безопасность аккаунта и в следствии конфиденциальных данных этого аккаунта.

Чем пароль будет сложнее, тем лучше. Пароль от 12 до 20 символов со строчными и прописными буквами, символами и цифрами должен стать нормой. При этом в его структуре не должно быть никакой логики: о надежной защите данных можно будет говорить только в том случае, если пароль

будет сложно запомнить. Веб-модуль «APC» генерирует сложные пароли, которые гарантируют защиту от метода перебором.

Использование специальных символов в пароле значительно увеличивает надежность самого пароля. Отправка сгенерированного пароля посредством SMS лишает пользователей возможности генерировать сложные и надежные пароли с использованием специальных символов, в связи с тем, что в SMS отсутствует поддержка специальных символов. К тому же, адреса почтовых ящиков, зачастую привязаны к аккаунту сайта, что нельзя сказать о номере телефона. В связи с этим, было принято решение отправлять новые и надёжные пароли на почтовый адрес. Хранить сгенерированные пароли в письме безопасно, если доступ к почтовому ящику надежно защищен. К безопасности почтового ящика всегда следует подходить ответственно, поскольку зачастую привязка аккаунта от любого сайта осуществляется через почтовый ящик. Несанкционированный доступ к почтовому ящику чреват потерей доступа ко всем привязанным сайтам и приложениям.

Разработанный веб-модуль «Account Password Changer» может быть использован в работе сайтов которых используется библиотека хеширования паролей «bCrypt» [27]. Данная библиотека имеет реализации на таких языках программирования как:

- java;
- php 5.3+;
- python;
- c#;
- objective c;
- ruby;
- perl;
- javascript.

В разработанном веб-модуле имеется поддержка таких баз данных как:

- mysql;
- postgressql;

- mssql;
- mongodb;
- sqlite3.

Исходя из списка язык программирования и поддерживаемых баз данных на которых могут быть написаны приложения, можно сделать вывод, что разработанный веб-модуль является универсальным решением для обеспечения безопасности аккаунтов пользователей. Так же следует отметить, что данный веб-модуль остается независимым от среды разработки, языка программирований, типа приложения и базы данных используемой в приложении.

Веб-модуль гарантирует безопасность данных от сайтов администраторов. Все данные от сайта хранятся в базе данных зашифрованном виде алгоритмом RSA (сокр. от фамилий разработчиков R. L. Rivest, A. Shamir и L. M. Adleman) 512-битным ключом. Веб-модуль работает на протоколе HTTP/2, что обеспечивает безопасность и высокую скорость соединения.

Целью разработки веб-модуля безопасности является осуществление безопасности аккаунтов пользователей при помощи веб-модуля безопасности «APC».

Для администраторов (пользователей) веб-модуля «Account Password Changer» была разработана электронная инструкция, в которой обеспечивается объяснение по использованию веб-модуля и пошаговые руководства по правильной настройке и подключению веб-сайта администратора к базе данных веб-модуля безопасности «Account Password Changer».

2.2 Выбор технологии реализации

Разработка данного веб-модуля безопасности требует современных, надежных, безопасных, скоростных решений. В связи с данными требованиями было принято решение использовать стек технологий представленный ниже.

Фреймворк Express — Express.js, или просто Express, фреймворк веб-приложений для Node.js, реализованный как свободное и открытое программное обеспечение под лицензией massachusetts institute of technology (MIT) [27]. Он спроектирован для создания веб-приложений и API. Де-факто является стандартным каркасом для NodeJS. Автор фреймворка, T. J. Holowaychuk, описывает его как созданный на основе написанного на языке Ruby каркаса Sinatra, подразумевая, что он минималистичен и включает большое число подключаемых плагинов. Express может являться backend'ом для программного стека MEAN, вместе с базой данных MongoDB и каркасом Vue.js, React или AngularJS для frontend'a.

Технология NodeJS — это серверная платформа для работы с JavaScript через движок V8. JavaScript выполняет действие на стороне клиента, а Node — на сервере. С помощью Node можно писать полноценные приложения. Node умеет работать с внешними библиотеками, вызывать команды из кода на JavaScript и выполнять роль веб-сервера [19].

Преимущество NodeJS заключается в том, что с Node проще масштабироваться. При одновременном подключении к серверу тысяч пользователей Node работает асинхронно, то есть ставит приоритеты и распределяет ресурсы эффективно. Java же, например, выделяет на каждое подключение отдельный поток.

Крупные компаний использующие NodeJS:

- walmart;
- e-bay / paypal;
- microsoft (azure);
- linkein;
- yahoo;
- google;
- netflix;
- uber.

Язык программирования Javascript — это клиентский язык программирования, разработанный, прежде всего, для интерактивности веб-страниц [25]. Клиентский означает, что он выполняется не на стороне сервера (как, скажем, PHP), а на стороне браузера. В этом огромный плюс JavaScript, поскольку ему не нужно специальное окружение, в любом современном браузере имеется интерпретатор JavaScript, поэтому для работы с JavaScript достаточно лишь браузера.

Преимущества JavaScript:

- быстрый для конечного пользователя, язык не нуждается в компиляции на стороне клиента, что дает ему определенные преимущества скорости. поскольку сценарий выполняется на компьютере пользователя, в зависимости от задачи, результаты выполняются почти мгновенно. например, вы можете проверить любой пользовательский ввод перед отправкой запроса на сервер. это снижает нагрузку на сервер;
- простота. javascript относительно прост в освоении и реализации. он использует модель dom, которая обеспечивает множество предустановленных функций для различных объектов на страницах, что делает его легким для разработки сценария для решения пользовательской цели;
- универсальность: javascript отлично работает с другими языками и может использоваться в самых разных приложениях. в настоящее время существует множество способов использования javascript через серверы node.js. если вы загрузили node.js с помощью express, используйте базу данных документов, такую как mongodb, и используйте javascript в интерфейсе для клиентов, вы можете создать приложение javascript полностью из одного окна вперед, используя только javascript.

Протокол Speedy (SPDY) — это протокол прикладного уровня для передачи веб-контента. Протокол разработан корпорацией Google. По замыслу разработчиков, данный протокол позиционируется как замена некоторых частей протокола HTTP — таких, как управление соединениями и форматы передачи данных [27].

Преимущества SPDY:

- снижение времени загрузки страницы. В одном соединении можно отправлять неограниченное количество запросов, что существенно повышает эффективность и скорость передачи данных;
- простая установка. Модификация требуется лишь в браузере и веб-сервере, а сам протокол умеет использовать старую сетевую инфраструктуру, поскольку на базовом уровне SPDY по-прежнему работает поверх TCP. изменения в коде страниц также не нужны;
- неограниченное количество запросов по ограниченному каналу приводит к тому, что в результате ни одно приложение не получает необходимые данные вовремя. Поэтому в SPDY реализована система приоритетов, позволяющая передавать критически важные данные в первую очередь;
- снижение веб-трафика достигается за счет использования компрессии заголовков и удаления ненужных данных. Это позволяет уменьшать размер заголовков запросов на 88 %, а заголовков ответов — на 85 %, чего уже достаточно для значительного ускорения загрузки страниц на медленных соединениях;
- добавление прослойки в виде SSL-протокола обеспечивает надежную защиту передаваемых данных. Использование безопасного соединения немного снижает эффективность SPDY, однако даже в этом случае он оказывается экономнее и быстрее HTTP;
- учитывая то, что SPDY отправляет на 40 % меньше пакетов по сравнению с HTTP и использует меньше TCP-соединений, вероятность потери и повторной отправки пакетов также снижается. В случаях неудачной доставки пакетов SPDY быстрее HTTP на 27 % при прочих равных;
- возможность передачи данных сервером даже до поступления запроса от клиента также приводит к снижению задержек.

СУБД MongoDB — это документоориентированная система управления базами данных (СУБД) с открытым исходным кодом, не требующая опи-

сания схемы таблиц. Классифицирована как NoSQL, использует JSON-подобные документы и схему базы данных. Написана на языке C++ [11].

Преимущества MongoDB:

- отсутствие схемы;
- эта база данных (БД) основана на коллекциях различных документов. количество полей, содержание и размер этих документов может отличаться. т.е. различные сущности не должны быть идентичны по структуре;
- крайне понятная структура каждого объекта;
- легко масштабируется;
- для хранения используемых в данный момент данных используется внутренняя память, что позволяет получать более быстрый доступ;
- данные хранятся в виде JSON документов;
- поддерживает динамические запросы документов;
- отсутствие сложных JOIN запросов;
- нет необходимости маппинга объектов приложения в объекты БД.

2.3 Проектирование веб-модуля

MongoDB реализует новый подход к построению баз данных, где нет таблиц, схем, запросов SQL, внешних ключей и многих других вещей, которые присущи объектно-реляционным базам данных.

В отличие от реляционных баз данных MongoDB предлагает документо-ориентированную модель данных, благодаря чему MongoDB работает быстрее, обладает лучшей масштабируемостью, ее легче использовать.

Описание моделей

Документ «App». В данном документе описаны данные которые будут храниться в объекте приложения, которое зарегистрирует пользователь.

1) owner — владелец приложения:
    - тип: ссылка на пользователя;
    - обязательное поле;

2) domain — имя домена приложения:
   - тип: строка;
   - обязательное поле;
3) dbhost — адрес приложения:
   - тип: строка;
   - обязательное поле;
4) dbname — имя базы данных:
   - тип: строка;
   - обязательное поле;
5) dbuser — логин пользователя базы даных:
   - тип: строка;
   - обязательное поле;
6) dbpassword — пароль пользователя:
   - тип: строка;
   - обязательное поле;
7) dbtable — таблица, где хранятся пользователи:
   - тип: строка;
   - обязательное поле;
8) dbport — порт соединения с БД:
   - тип: число;
   - не обязательное поле;
9) dbtype — тип базы данных:
   - тип: строка;
   - обязательное поле;
10) coluserid — имя поля, в котором храниться идентификатор пользователя:
   - тип: строка;
   - обязательное поле;

11) coluserpassword — имя поля, в котором храниться пароль пользователя:

- тип: строка;
- обязательное поле;

12) coluseremail — имя поля, в котором храниться email пользователя:

- тип: строка;
- обязательное поле;

13) coluserphone — имя поля, в котором храниться номер телефона пользователя:

- тип: строка;
- не обязательное поле;

14) isactive — активна ли функция изменения пароля:

- тип: бинарное значение;
- по умолчанию: ложь;

15) createat — дата создания:

- тип: дата;
- по умолчанию: текущая дата.

Документ «User». В данном документе описаны данные которые будут храниться в объекте зарегистрированного пользователя.

1) email — email пользователя:

- тип: строка;
- обязательное поле;

2) password — пароль пользователя:

- тип: строка;
- обязательное поле;

3) security — активна ли функция изменения пароля аккаунта:

- тип: бинарное значение;
- по умолчанию: ложь;

4) verifyemail — подтвержден ли email пользователя:
   - тип: бинарное значение;
   - по умолчанию: ложь.

Документ «Log». В данном документе описаны данные, которые будут храниться в журналах событий модуля для конкретного пользователя и приложения.

1) owner — владелец данного события:
   - тип: ссылка на объект пользователя;
2) appid — приложение к которому относится данный лог:
   - тип: ссылка на объект приложения;
3) category — категория события:
   - тип: бинарное значение;
   - обязательное поле;
4) type — тип события:
   - тип: бинарное значение;
   - обязательное поле;
5) rectext — текст события:
   - тип: объект;
   - обязательное поле;
6) createat — дата создания события:
   - тип: дата;
   - по умолчанию: текущая дата.

2.4 Описание основных библиотек и используемых функций

Библиотека bcrypt-nodejs — Адаптивная криптографическая хеш функция формирования ключа, используемая для защищенного хранения паролей. Разработчики: Нильс Провос и David Mazières. Функция основана на шифре Blowfish, впервые представлена на USENIX в 1999 году. Для защиты от атак с помощью радужных таблиц bCrypt использует соль (salt); кроме то-

го, функция является адаптивной, время её работы легко настраивается и её можно замедлить, чтобы усложнить атаку перебором.

Этот npm пакет использует UNIX-библиотеку bcrypt, написанную в 1999 году. Это позволяет хэшировать и шифровать конфиденциальные данные, такие как пароли пользователей, перед их сохранением в базу данных.

Generate-password — библиотека предназначена для генерирования паролей из командной строки, NodeJS или браузера. Библиотека имеет гибкие настройки генерирования паролей. Список параметров:

- длина пароля;
- использование цифр;
- использование специальных символов;
- использование заглавных букв;
- использование похожих символов;
- Исключение из пароля символов.

Node-rsa — NodeJS RSA библиотека. В проекте требуется для шифрования конфиденциальных данных пользователей.

Особенности Node-rsa:

- чистый javascript;
- не требуется openssl;
- генерация ключей;
- поддерживает длинные сообщения для шифрования / дешифрования;
- подписание и проверка.

Nodemailer — это модуль для приложений NodeJS, позволяющий легко отправлять электронные письма. Проект начался в 2010 году, когда не было никакой разумной возможности отправлять электронные письма, сегодня это решение, к которому большинство пользователей NodeJS обращаются по умолчанию. Nodemailer лицензируется по лицензии MIT.

2.5 Описание интерфейса веб-модуля

2.5.1 Страница регистрации аккаунта

Для начала использования веб-модуля безопасности необходимо зарегистрировать аккаунт в системе веб-модуля. Что бы это сделать, необходимо перейти на страницу регистрации (рисунок 5).

На странице регистрации расположена форма, которую необходимо заполнить. Нужно заполнить три поля:

- email;
- пароль;
- повторение пароля;
- чекбокс который устанавливают активным после ознакомления с правилами.

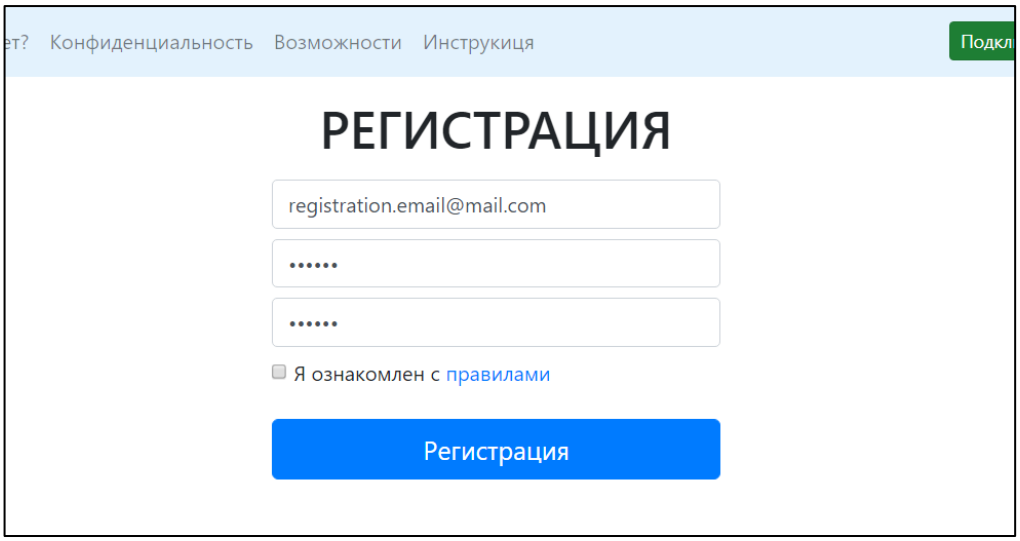

Рисунок 5 — Регистрация аккаунта

После завершения регистрации пользователь получить уведомление о статусе регистрации. После успешной регистрации пользователь должен подтвердить свой аккаунт по ссылке в письме отправленную ему на электронную почту.

2.5.2 Страница аккаунта

Страница «Аккаунт» разделена на две основных области. Слева — область навигации. В данном блоке расположены разделы страницы «Аккаунт». Страница «Аккаунт» разбита на такие разделы как:

- общее;
- мои сайты;
- безопасность.

Под блоком навигации располагается кнопка «Добавить сайт». По нажатии на данную кнопку пользователь будет перенаправлен на страницу регистрации сайта. По клику на пункт раздела в область контента загружается содержимое данного раздела.

В разделе «Общее» отображается email пользователя и дата последнего изменения пароля аккаунта. Так же имеется возможность изменить пароль от аккаунта кликнув по кнопке «Изменить» находящуюся в строке «Пароль» (рисунок 6).

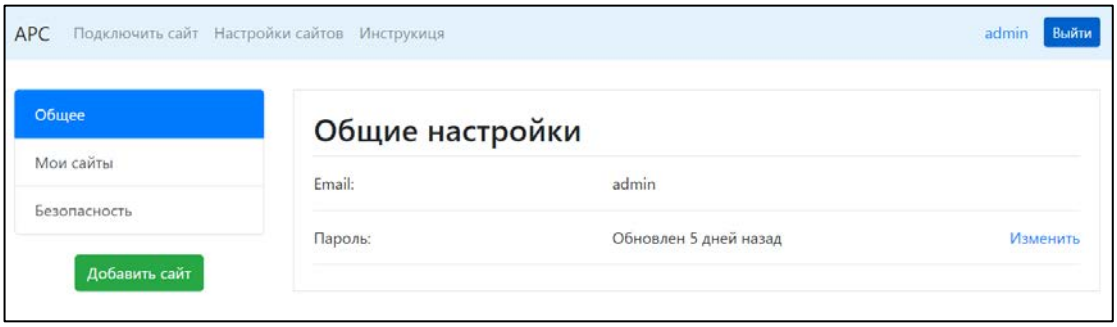

Рисунок 6 — Раздел «Общее»

В разделе «Мои сайты» выводится таблица зарегистрированных в веб-модуле-безопасности сайтов, которые принадлежат владельцу аккаунта (рисунок 7). Шапка таблица разделена на три колонки:

- домен;
- статус;
- действия.

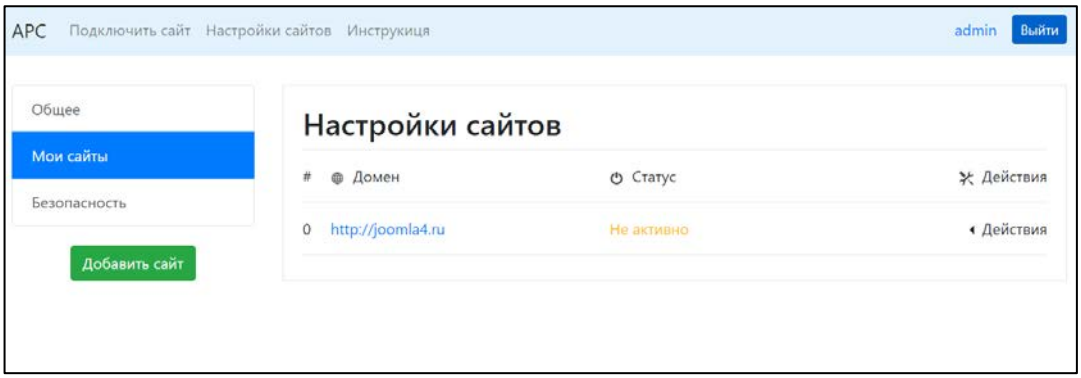

Рисунок 7 — Раздел «Мои сайты»

В колонке «Домен» выводится адрес добавленного сайта. Пользователь имеет возможность попасть на свой сайт просто кликнув по адресу.

В колонке «Статус» отображается статус действия веб-модуля безопасности для конкретного сайта. Статус «Не активно» свидетельствует о том, что работа веб-модуля безопасности на данном сайте приостановлена. Статус «Активно» говорит об обратном. Веб-модуль произведет необходимые действия с заданной пользователем периодичностью.

В колонке «Действия» расположена кнопка, по нажатии которой будет открыто меню с пунктами «Приостановить», «Редактировать», «Логи» и «Удалить» (рисунок 8).

По нажатии на кнопку «Приостановить» работа веб-модуля безопасности для данного сайта будет приостановлена. Статус в колонке «Статус» изменится на «Не активно».

По нажатии на кнопку «Редактировать» пользователь будет перенаправлен на страницу редактирования данного сайта. На странице «Редактирование сайта» пользователь сможет осуществить настройку веб-модуля безопасности и произвести его конфигурацию.

По нажатии кнопки «Логи» пользователь так же будет перенаправлен на страницу «Редактирование сайта» в раздел «Логи».

По нажатии на кнопку «Удалить» удаляет все данные о сайте из базы данных модуля веб-безопасности.

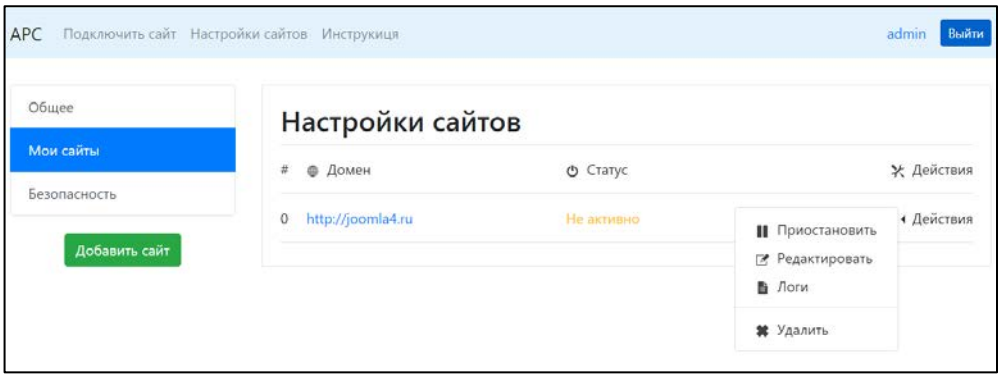

Рисунок 8 — Нажатая кнопка «Действия»

В разделе «Безопасность» располагаются настройки функции регулярной смены пароля (рисунок 9). Данная функция используется для регулярной смены пароля пользователям зарегистрированного сайта. Эта функция доступна и для владельцев аккаунтов. Активирование данной функции обеспечит эффективную безопасность аккаунта администратора сайтов. В параметрах данной функции есть возможность конфигурирования таких параметров как:

- длина пароля;
- использование цифр;
- использование специальных символов;
- использование заглавных букв;
- использование похожих символов;
- исключение из пароля символов.

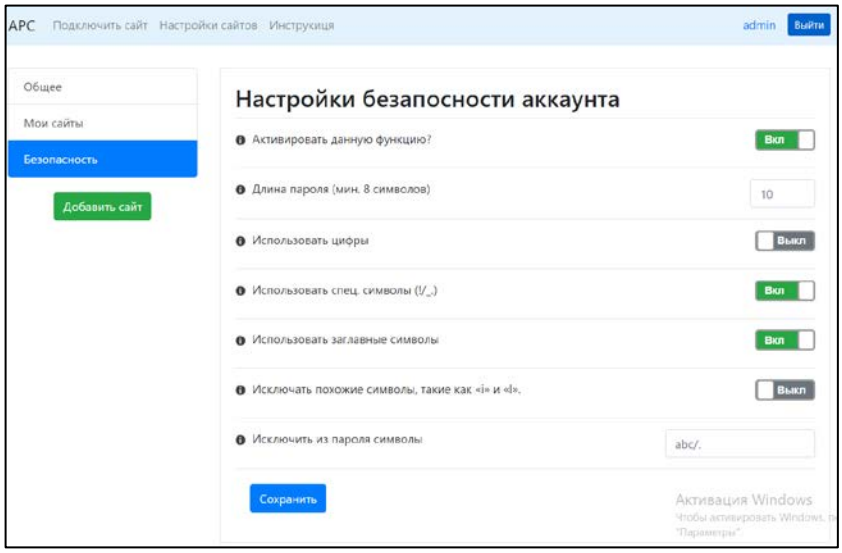

Рисунок 9 — Раздел «Безопасность»

Пользователь по своему усмотрению может сконфигурировать данные параметры так, как ему будет удобно.

2.5.3 Страница регистрации сайта в системе

Для того чтобы добавить сайт в систему веб-модуля безопасности необходимо перейти на страницу регистрации сайта, кликнув в шапке сайта «Подключить сайт» или перейти из страницы «Аккаунт» кликнув на кнопку под блоком навигации «Добавить сайт». Регистрация сайта разделена на 5 шагов.

- шаг 1 — «домен»;
- шаг 2 — «права»;
- шаг 3 — «база данных»;
- шаг 4 — «поля базы данных»;
- шаг 5 — «готово».

На первом шаге необходимо ввести корректный адрес сайта. После нажатия на кнопку «Отправить» веб-модуль отправляет запрос на данный адрес и в зависимости от кода состояния в ответе от данного адреса пропускает или не пропускает на следующий шаг пользователя (рисунок 10). Если код ответа «2xx— успешно», то пользователю станет доступен шаг 2. Если же код ответа иной, то пользователь получит соответствующее уведомление об ошибке и шаг 2 останется недоступен. Данная проверка позволяет отбросить недействительные или неактивные сайты.

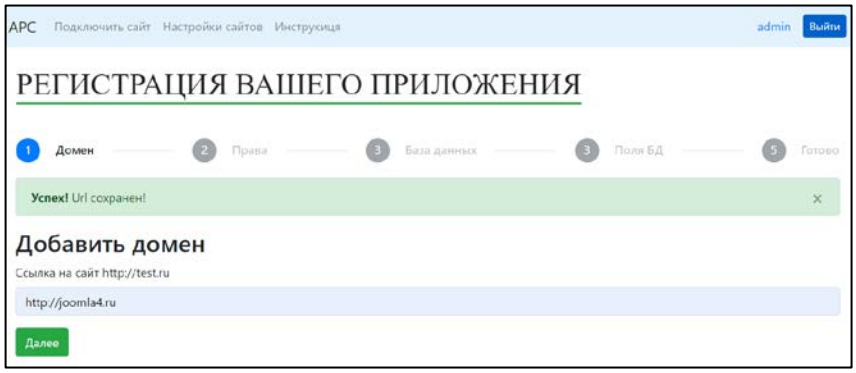

Рисунок 10 — Шаг 1

На втором шаге пользователю необходимо подтвердить права на владение тем сайтом, адрес которого он указал на шаге 1. Пользователю на выбор предоставляется два варианта подтверждения прав:

1. Вставить мета-тег на главную страницу сайта с индивидуальным содержимым (рисунок 11).

2. Добавить HTML файл в корневую директорию сайта с индивидуальным содержимым (рисунок 12).

Содержимое файла или мета-тега генерируется для каждого приложения индивидуально. После того, как пользователь произведет действия по внедрению мета-тега или загрузке файла в корневой каталог ему необходимо нажать кнопку «Проверить». Далее веб-модуль отправляет запрос на файл или парсит содержимое главной страницы на наличие мета-тега и возвращает ответ. Если статус ответа «истина» то проверка прошла успешно и пользователь может приступить к заполнению шага 3. Если статус ответа «ложь» то шаг 3 останется заблокирован. В обоих случаях пользователь получит соответствующее уведомление о провале или успехе проверки.

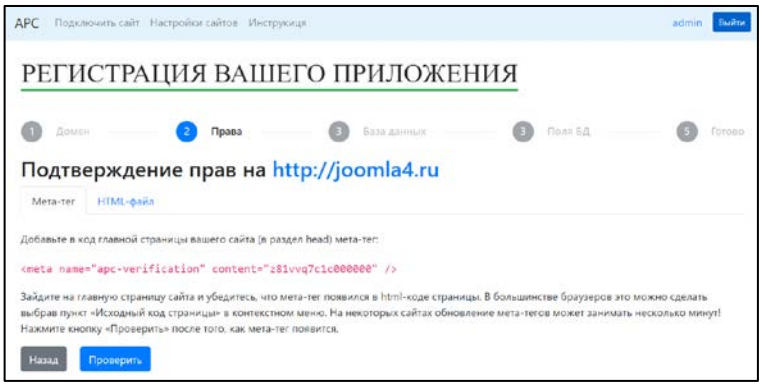

Рисунок 11 — Шаг 2. Проверка мета -тегом

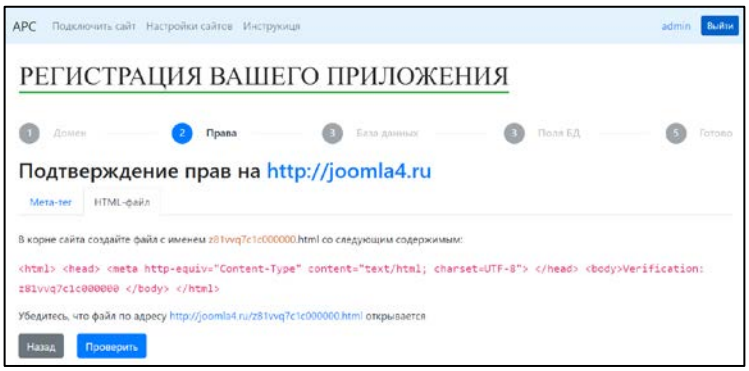

Рисунок 12 — Шаг 2. Проверка файлом гипертекстовой разметки

На шаге 3 пользователь должен заполнить данные от базы данных сайта, который он регистрирует (рисунок 13). Это нужно для того, чтобы веб-модуль имел возможность перезаписи новых паролей пользователям зарегистрированного сайта. Поля которые пользователь должен заполнить пользователь:

- Адрес БД;
- база данных;
- пользователь БД;
- пароль пользователя БД;
- имя таблицы, в которой хранятся пользователи;
- порт;
- тип базы данных.

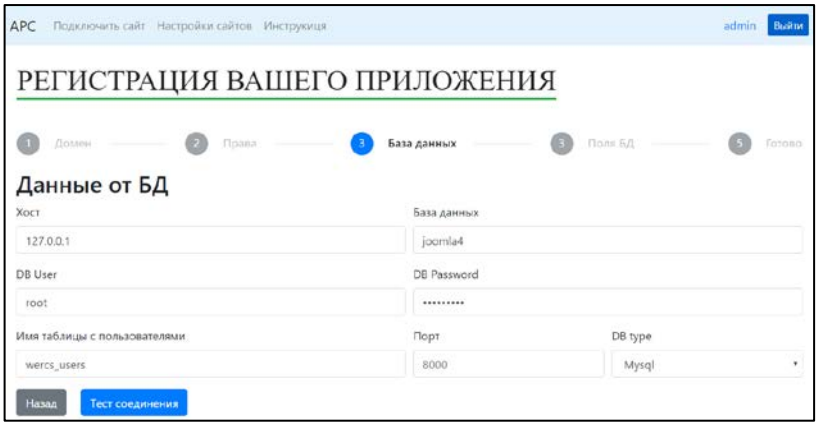

Рисунок 13 — Шаг 3

Все данные будут переданы по безопасному соединению протокола http2 и записаны в базу данных в зашифрованном виде, в связи с этим веб-модуль «APC» гарантирует конфиденциальность данных.

После того, как данные были заполнены, необходимо нажать кнопку «Тест соединения». Веб-модуль произведет тестовый запрос к базе данных сайта, после чего завершит соединение и вернет ответ «истина» — в случае успеха и «ложь» — в случае провала. После успешного соединения пользователю доступен шаг 4 — «Поля БД».

На шаге 3 веб-модуль парсит все поля из таблицы, которую указал пользователь и на шаге 4 необходимо сопоставить эти поля с названием ко-

торое находится выше (рисунок 14). Это необходимо для того, чтобы веб-модуль корректно использовал данные из соответствующих полей. После сопоставления, пользователю необходимо нажать кнопку «Сохранить» и перейти к завершающему шагу.

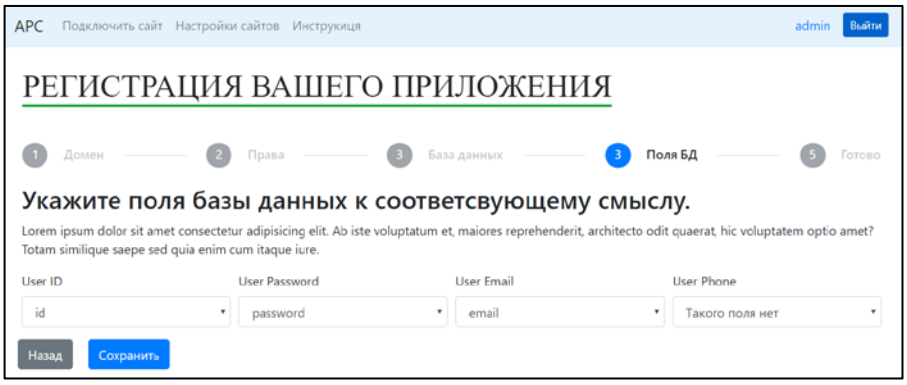

Рисунок 14 — Шаг 4

На финальном шаге 5 пользователь должен проверить все введённые им данные. Если все данные верны, то нужно нажать кнопку «Сохранить», после чего начнется процесс записи данных сайта в базу данных веб-модуля безопасности. Если пользователь обнаружил ошибку, то он имеет возможность вернуться на тот шаг, где была допущена ошибка и исправить ее (рисунок 15).

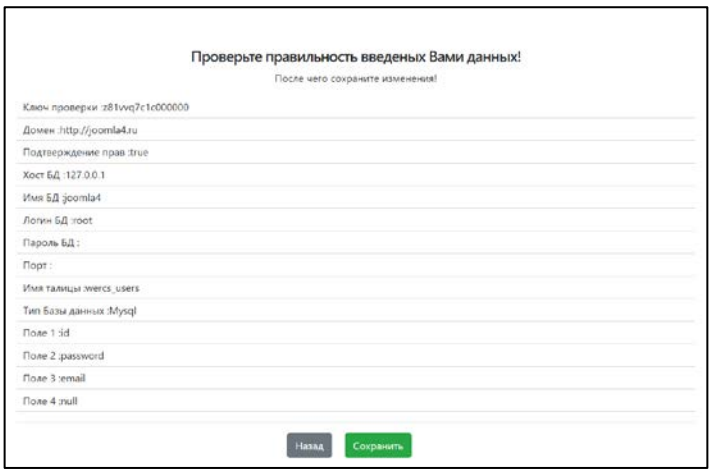

Рисунок 15 — Шаг 5

После того, как процесс сохранения данных завершиться, пользователь получит уведомление о статусе записи. Когда приложение будет успешно сохранено, можно будет перейти в настройки данного приложения из раздела «Мои сайты» на странице «Аккаунт».

2.5.4 Страница управление сайтами

Управление зарегистрированными сайтами производится на странице редактирования сайта. Перейти на данную страницу можно из раздела «Мои сайты» страницы «Аккаунт», кликнув по кнопке «Действия», а затем «Редактировать» (рисунок 16).

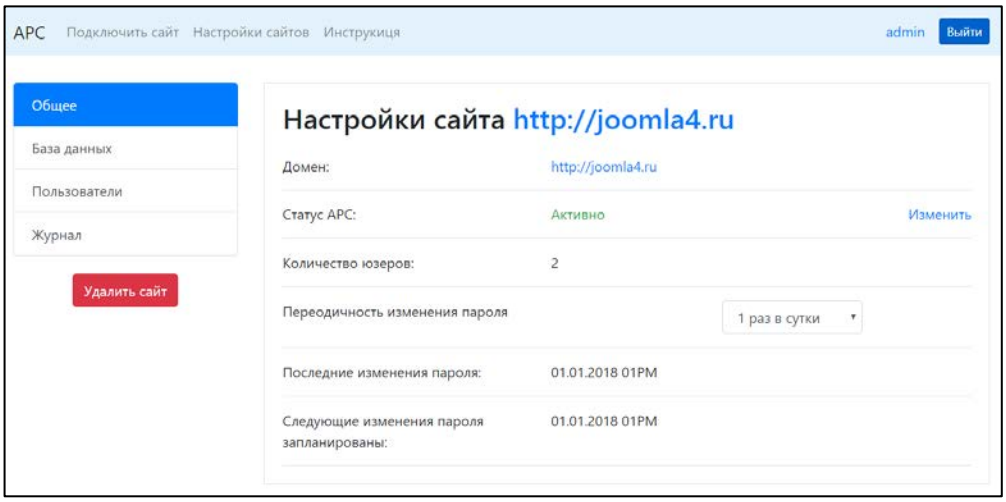

Рисунок 16 — Страница «Редактирование сайта»

Структура страницы такая же, как и на странице «Аккаунт». Слева находится область навигации. Под областью навигации располагается кнопка «Удалить сайт». На данной странице присутствуют такие разделы как:

- общее;
- база данных;
- пользователи;
- журнал;

В разделе «общее» представлена общая информация о зарегистрированном сайте:

- адрес сайта;
- статус работы веб-модуля;
- количество пользователей сайта;
- выбор периодичности смены паролей;
- дата последнего изменение пароля;

- дата следующего изменения пароля.

В разделе «База данных» представлена вся информация о базе данных сайта, с которой взаимодействует веб-модуль (рисунок 17).

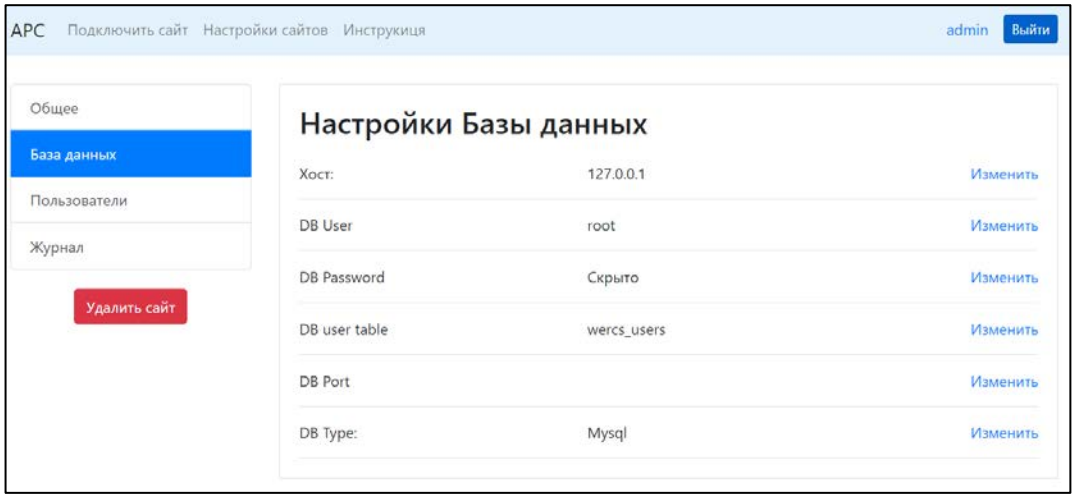

Рисунок 17 — Раздел «База данных»

В разделе «Пользователи» выводится список пользователей зарегистрированного сайта (рисунок 18). По нажатии на кнопку «Обновить список пользователей» будет произведено обновление списка пользователей и отображено на странице.

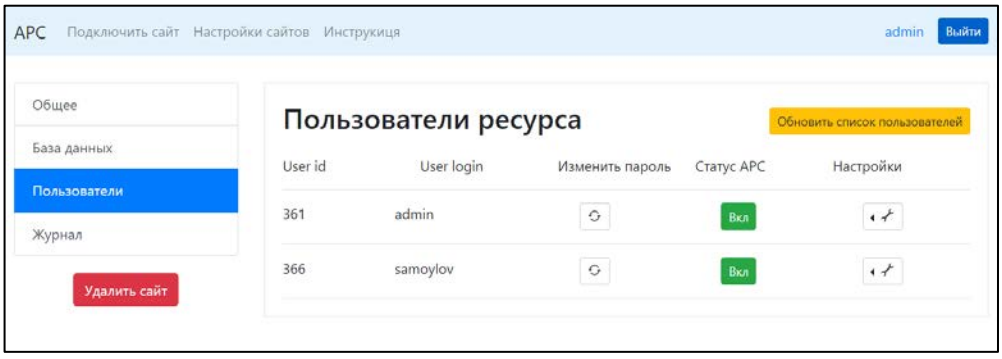

Рисунок 18 — Раздел «Пользователи»

Шапка таблицы разделена на 5 столбцов:

- ID пользователя в базе данных сайта;
- логин пользователя;
- кнопка, по нажатии которой будет единовременно изменен пароль конкретного пользователя;

- статус активности функции регулярной смены пароля для конкретного пользователя;
- индивидуальные настройки пароля пользователя.

По нажатии на кнопку настроек откроется блок, в котором регулируются настройки генерации пароля для пользователя (рисунок 19).

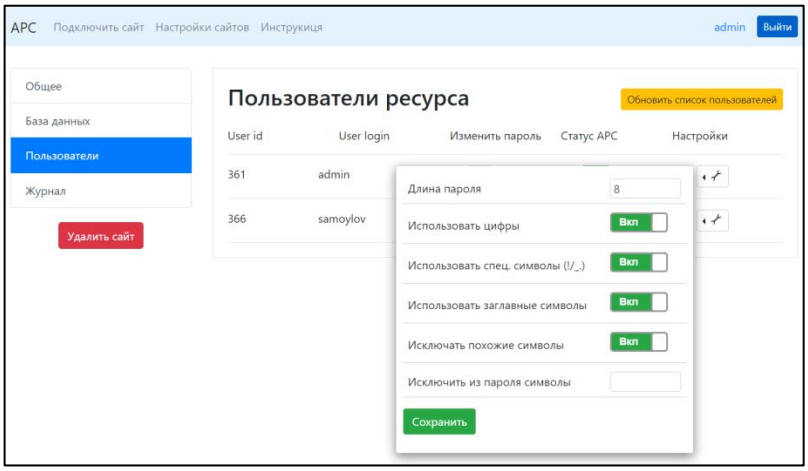

Рисунок 19 — Настройка генерации пароля

При регистрации сайта, у всех пользователей по умолчанию активны все пункты настроек.

В разделе «Журнал» представлена информация о всех действиях, которые производил веб-модуль с сайтом владельца (рисунок 20).

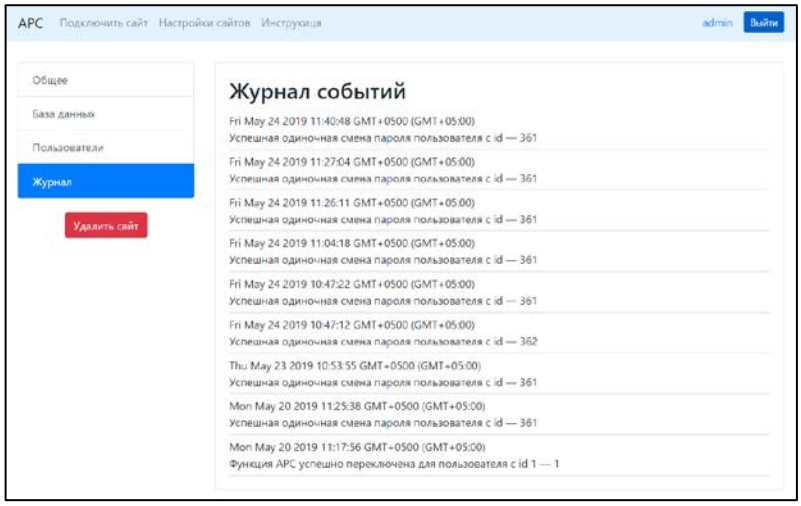

Рисунок 20 — Раздел «Журнал»

Каждое действие регистрируется и записывается в журнал. Таким образом администратор сайта в любой момент времени в курсе того, что происходит с паролями пользователей его сайта.

2.5.5 Обеспечение безопасности веб-модуля

Работа с базой данных требует особенного внимания к вопросу обеспечения безопасности. Для веб-модуля безопасности был предпринят комплекс мер обеспечивающий безопасность конфиденциальных данных.

Соединение. Веб-модуль работает на протоколе https. HTTPS — расширение протокола HTTP для поддержки шифрования в целях повышения безопасности [24]. Данные в протоколе HTTPS передаются поверх криптографических протоколов SSL или TLS. При установке соединения по протоколу HTTPS компьютер пользователя и сервер сначала генерируют некий секретный ключ, а затем уже обмениваются информацией, при этом шифруя ее с помощью данного ключа. Ключ создается заново при каждом сеансе связи, так что перехватить и подобрать его практически невозможно — он представляет собой число с количеством знаков более ста. Для полной надежности используется еще и цифровой сертификат для идентификации сервера. Первое, что делает браузер при установке соединения по HTTPS, — проверяет подлинность сертификата. Только после ее подтверждения начинается обмен данными.

База данных. Все конфиденциальные данных хранятся в базе данных в зашифрованном виде. Данные шифруются криптографическим алгоритмом RSA ключом равным 512 бит (рисунок 21). RSA — криптографический алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших целых чисел [30].


```
{
    "_id" : ObjectId("5ce245bf3a85e83080ae8fbc"),
    "isActive" : false,
    "owner" : ObjectId("5ca6f92bc6adea2f30cc49c4"),
    "domain" : "km8K06ocRGCcisKQrlpyI+QxsywCgK9pElztspSB/Z2rhz1csXqOFUP4FljxIuXjM+DpPERsZxHzMnORvCAUEw==",
    "dbHost" : "ix5KkBl5Op/rUtzoaauubn2WelhXnHtzAlQT9FnzV+HnMBhHpNCpTP1DsLRCr8lMEPBRZ3mwh3+kooLU14ipYA==",
    "dbName" : "BXFU3ai5FMgkxvPBpVpaXEqweD+Ive/FgHSE4GxATee4VF2oZKf1dJEzwSPSXvTWncSf+8Pf4IS0oIX7Yb5PzA==",
    "dbUser" : "IjcdghTiMJK73LqwtlvCCtp2kCAXGfz9A6zlB+dvpIw36B4EhfS/d3hQQ9VPZeixCvl2ybYLoR3SD7sEMRn5fg==",
    "dbPassword" : "U+b2NSAAEoLRGmEAwUURwlVSLwRT7tx/JMQ4/EVTpUeyZMsz7dHonIVCtjjcXOxRqOF5HXPKmsqtQ4eDr7uycg==",
    "dbTable" : "KE1q8ya1N6f8r2XctxiLCMJnGVCMK7vhkawtmbxmoPAZmgIqLw0g1zYhu5xN78lM3hBp+UMXvLKOxAD+H19qFQ==",
    "dbPort" : null,
    "dbType" : "Mysql",
    "colUserId" : "id",
    "colUserPassword" : "password",
    "colUserEmail" : "email",
    "colUserPhone" : "null",
    "createAt" : ISODate("2019-05-20T06:14:23.173Z"),
    "createdAt" : ISODate("2019-05-20T06:14:23.183Z"),
    "updatedAt" : ISODate("2019-05-20T06:14:23.183Z"),
    "__v" : 0
}
```


Рисунок 21 — Зашифрованные данные в базе данных веб-модуля

При запросе к БД данные расшифровывается ключом, который находится в переменных окружения, таким образом, ключ достать невозможно.

2.5.6 Описание демонстрационного сайта

Тестирование работы веб-модуля безопасности было произведено на локальном сайте (рисунок 22). На сайте установлена CMS Joomla 3.9.6 с версией PHP 5.3.13. Версия PHP 5.3.13 поддерживает технологию хеширования паролей bcrypt. Целю создания данного сайта стала необходимость проверки работоспособности веб-модуля безопасности, а именно изменения паролей данных аккаунтов и отправка новых паролей на почту.

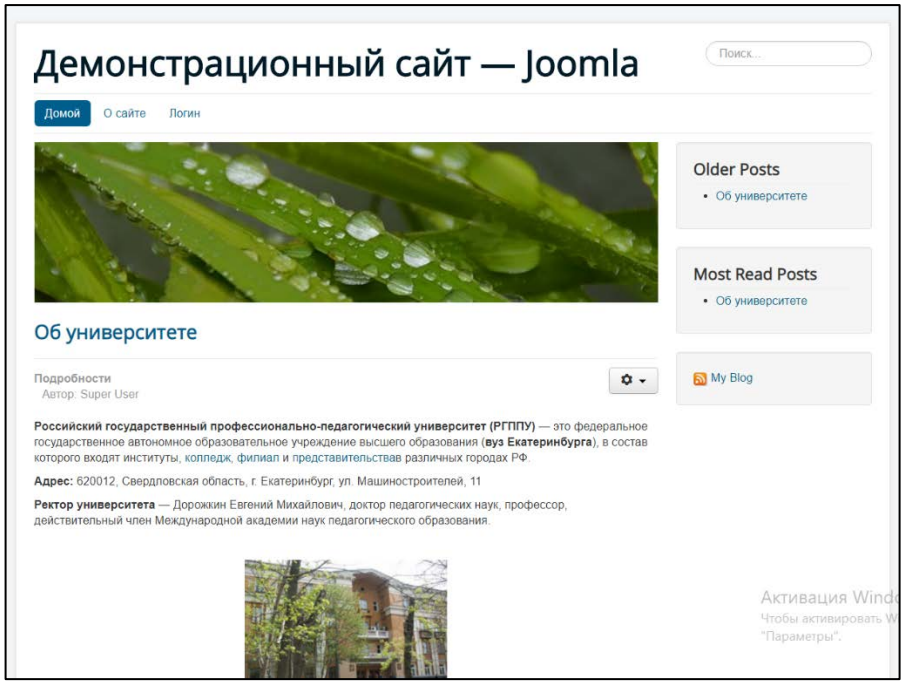

Рисунок 22 — Демонстрационный сайт

Структура сайта разделена на следующие блоки:

- шапка;
- область контента;
- панель навигации;
- подвал.

На сайте была зарегистрировано 5 аккаунтов пользователей и назначены различные группы пользователей. Через административную панель веб-

модуля безопасности были произведены успешные манипуляции над паролями данных аккаунтов (рисунок 23).

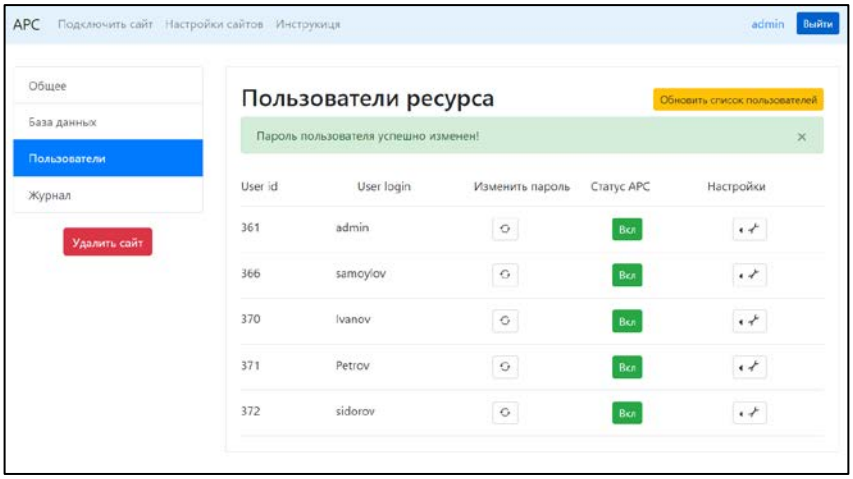

Рисунок 23 — Административная панель

В результате пароли были изменены и отправлены на электронный адрес получателя (рисунок 24).

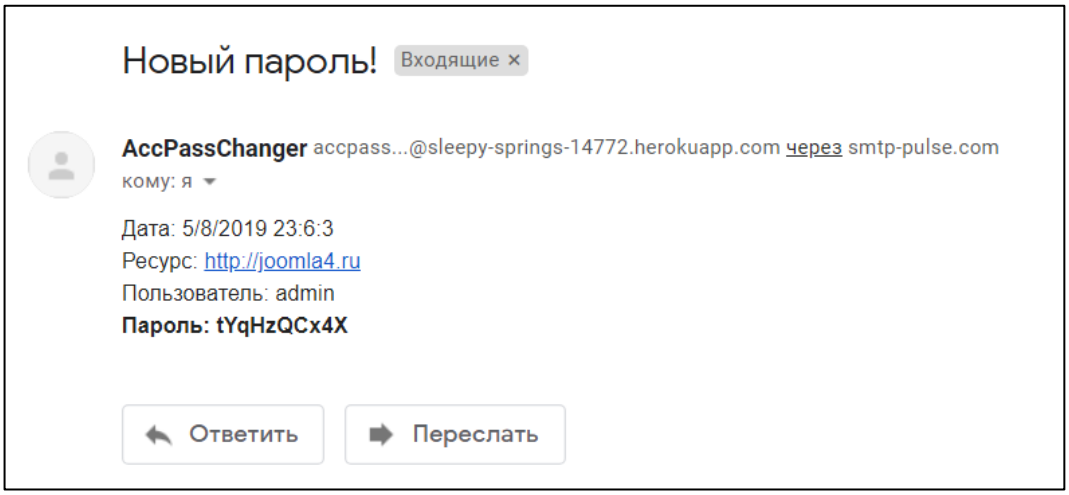

Рисунок 24 — Полученный пароль в электронном письме

Таким образом, было проведено тестирование работы веб-модуля безопасности.

2.6 Описание обучающей инструкции и теоритического материала

Для веб-модуля безопасности была разработана электронная инструкция. В электронной инструкции произведено разделение пошаговых руко-

водств по категориям веб-страниц, с которыми пользователь будет взаимодействовать в процессе использования веб-модуля безопасности «АРС».

Электронная инструкция предназначена для администраторов веб-ресурсов (веб-сайтов) использующих в своей работе политику аутентификации и авторизации аккаунтов пользователей с помощью логина и пароля. Инструкция обеспечивает объяснение по использованию ресурса и пошаговые руководства по правильной настройке и подключению веб-сайта администратора к базе данных веб-модуля безопасности «АРС» (рисунок 25).

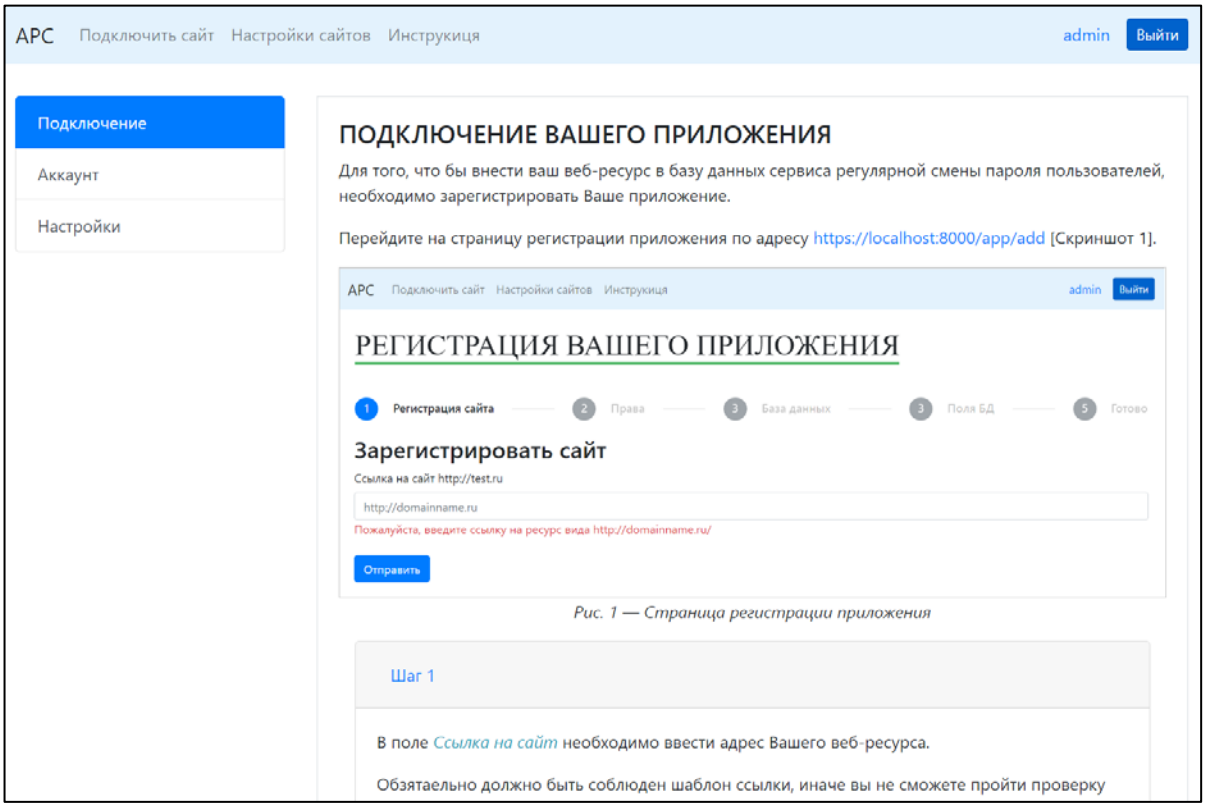

Рисунок 25 — Электронная инструкция

Объектом исследования является процесс обучения администраторов веб-ресурсов по настройке веб-модуля безопасности «АРС».

Предметом исследования документационные материалы по аутентификации и авторизации пользователей.

Цель настоящего исследования — разработать электронную инструкцию по настройке сервиса «Регулярной смены пароля пользователей веб-сайта».

В соответствии с поставленной целью в работе определены следующие задачи:

1. Проанализировать литературу и интернет-источники по теме исследования.
2. Спроектировать структуру электронной инструкции.
3. Спроектировать интерфейс электронной инструкции.
4. Наполнить содержанием электронную инструкцию.

Электронная инструкция располагается на сайте веб-модуля безопасности «АРС» в разделе «Инструкция» по адресу https://localhost:8000/instruction. В электронной инструкции произведено разделение пошаговых руководств по категориям веб-страниц, с которыми пользователь будет взаимодействовать в процессе пользования сервисом «регулярной смены пароля пользователей веб-сайта». В электронной инструкции подробно и пошагово описаны действия по настройке и подключению приложения администратора к базе данных веб-модуля безопасности «АРС».

В разделе «Аккаунт» описаны действия по управлению настройками аккаунта и его информационной безопасности.

В разделе «Настройки» описаны действия по управлению настройками зарегистрированных приложений администратором.

Электронная инструкция разделена на следующие категории:

- подключение приложения;
- аккаунт пользователя;
- настройки аккаунта пользователя и его веб-ресурсов.

В свою очередь данные категории разделены на следующие подкатегории:

1) подключение приложения:
    - шаг 1 — зарегистрировать сайт;
    - шаг 2 — подтверждение прав;
    - шаг 3 — база данных;
    - шаг 4 — поля БД;

2) аккаунт пользователя:
   - раздел «общее»;
   - раздел «безопасность»;
3) настройки веб-ресурсов пользователя:
   - раздел «мои сайты».

2.7 Размещение на веб-сервере и апробация

Разработанный веб-модуль «АРС» прошел апробацию в центре веб-технологий и программирования Федерального государственного автономного образовательного учреждения высшего образования «Российский государственный профессионально-педагогический университет» (РГППУ) в г. Екатеринбурге.

Был размещен на локальном веб-сервере отдела, далее были произведены пробные смены паролей аккаунтов сайта monument.rsvpu.ru на базе CMS Wordpress (рисунок 26).

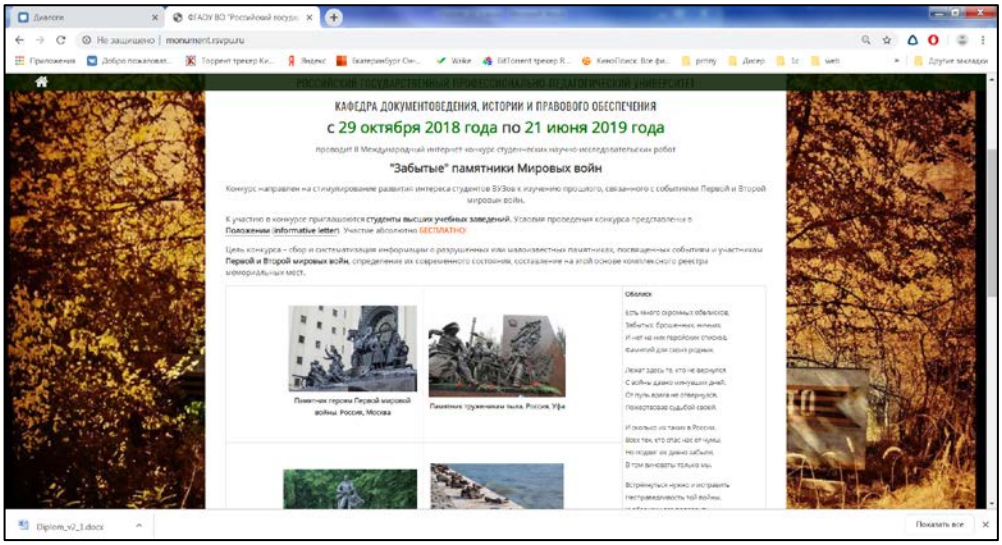

Рисунок 26 — Сайт «monument.rsvpu.ru»

Данные от аккаунта были введены в приложение настроена смена пароля оператора — преподавателя кафедры документоведения, истории и правового обеспечения раз в месяц пароль будет приходить на указанный электронный ящик (рисунок 27).

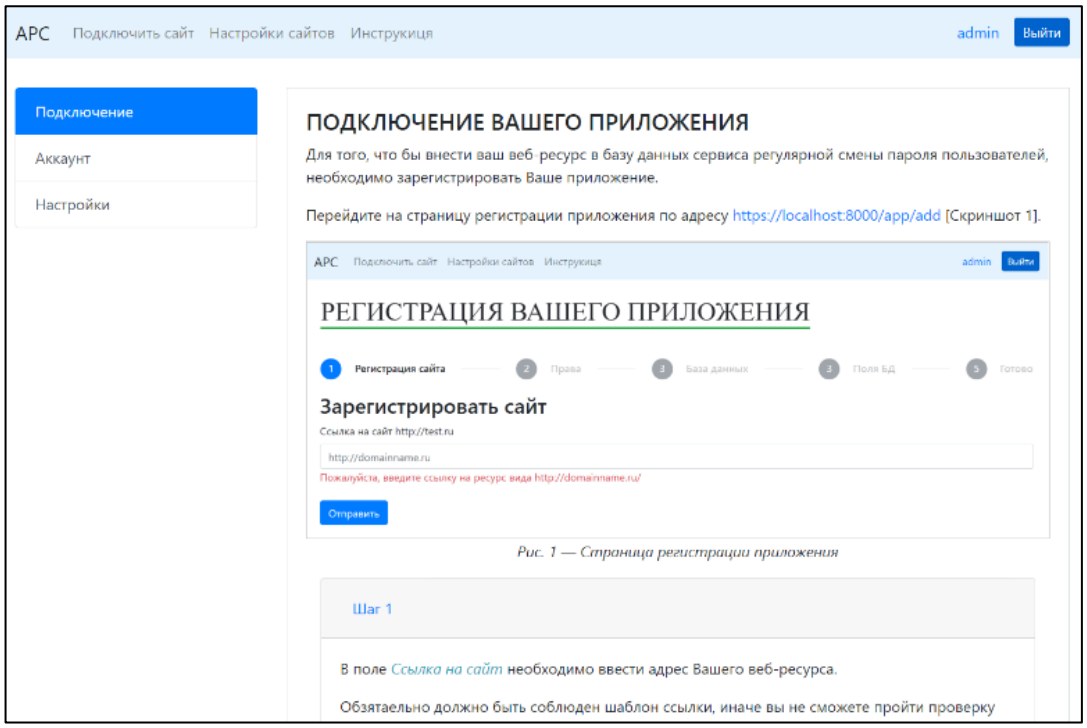

Рис. 1 — Страница регистрации приложения

Рисунок 27 — Параметры подключения

По результатам апробации необходимо отметить следующие:

- была добавлена возможность сконфигурировать индивидуальные настройки генерации пароля для каждого пользователя;
- была добавлена возможность изменения пароля администратором для одного пользователя;
- была добавлена возможность выбора частоты изменения пароля.

ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломной работы был разработан веб-модуль «АРС», а так же разработана электронная инструкция для администраторов веб-сайтов использующих в своей работе политику аутентификации и авторизации аккаунтов пользователей с помощью логина и пароля.

Сопоставление результатов работы с поставленными задачами позволяет заключить следующее:

Проведенный анализ литературы, интернет-источников выявил, основные направления использования аутентификации пользователя, что определило необходимость и важность разработки веб-модуля.

Также были определенны параметры генерации нового пароля, а именно:

- длина пароля;
- использование цифр;
- использование специальных символов;
- использование заглавных букв;
- использование похожих символов;
- исключение из пароля символов.

Разработанный веб-модуль позволит администраторам сайтов обеспечить дополнительную защиту аккаунтов и конфиденциальных данных пользователей своих веб-ресурсов.

Таким образом, следует считать, что задачи дипломной работы полностью выполнены и цель исследования достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Адаптивная криптографическая хеш функция формирования ключа [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/Bcrypt (дата обращения: 11.05.2019).

2. Аутентификация [Электронный ресурс]. — Режим доступа: https://habr.com/ru/company/dataart/blog/311376/ (дата обращения: 08.04.2019).

3. Аутентификация в HTTP. Проверка подлинности пользователей в HTTP [Электронный ресурс]. — Режим доступа: https://zametkinapolyah.ru /servera-i-protokoly/tema-11-autentifikaciya-v-http-proverka-podlinnosti-polzova te lej-v-http.html (дата обращения: 11.05.2019).

4. Аутентификация пользователя. Веб [Электронный ресурс]. — Режим доступа: https://life-prog.ru/view_programmer.php?id=158&page=16 (дата обращения: 11.05.2019).

5. Аутентификация. Теория и практика обеспечения безопасного доступа к информационным ресурсам [Электронный ресурс]. — Режим доступа: http://it-ebooks.ru/publ/it_secutity/authentification/15-1-0-644 (дата обращения: 10.05.2019).

6. Аутентификация: от паролей до открытых ключей [Электронный ресурс]. — Режим доступа: http://www.williamspublishing.com/Books/5-8459-0341-6.html (дата обращения: 10.05.2019).

7. ГОСТ 2. 601-2013. Единая система конструкторской документации (ЕСКД). Эксплуатационные документы (с Поправкой). — Введ. 01.01.2002. — Москва: Изд-во стандартов, 2005. — 17 с.

8. ГОСТ Р 1.0-2004 Стандартизация в Российской Федерации. Основные положения. — Введ. 05.02.2003.— Москва: Изд-во стандартов, 2005. — 26 с.

9. ГОСТ Р 1.5-2004 Стандартизация в Российской Федерации. Стандарты национальные Российской Федерации. Правила построения, изложения, оформления и обозначения. — Введ. 03.03.2003.— Москва: Изд-во стандартов, 2005. — 18 с.

10. Данные по компрометации сайтов [Электронный ресурс]. — Режим доступа: https://www.securitylab.ru/news/482195.php (дата обращения: 10.05.2019).

11. Документно-ориентированная система управления базами данных MongoDB. [Электронный ресурс]. — Режим доступа: https://www.mongodb.com/ (дата обращения: 11.05.2019).

12. Идентификация и аутентификация [Электронный ресурс]. — Режим доступа: http://www.osp.ru/os/1996/04/178931 (дата обращения: 08.04.2019).

13. Как написать инструкцию по эксплуатации товара [Электронный ресурс]. — Режим доступа: https://kaplunoff.com/blog/kak-napisat/kak-napisat-instrukciyu-k-tovaru (дата обращения: 07.04.2019).

14. Как написать инструкцию так, чтобы тебя поняли [Электронный ресурс]. — Режим доступа: https://habr.com/ru/company/icl_services/blog/421503 (дата обращения: 07.04.2019).

15. Как написать понятное руководство — советы для копирайтеров [Электронный ресурс]. — Режим доступа: http://molyanov.ru/kak-napisat-ponyatnoe-rukovodstvo-sovety-dlya-kopirajterov/ (дата обращения: 07.04.2019).

16. Менеджер привилегированного доступа — Thycotic [Электронный ресурс]. — Режим доступа: https://thycotic.com/ (дата обращения: 15.05.2019).

17. Метод аутентификации с использованием динамических ключей [Электронный ресурс]. — Режим доступа: https://samizdatt.net/?do=pdf&idf=27396701 (дата обращения: 10.05.2019).

18. Написание инструкций пользовательского интерфейса [Электронный ресурс]. — Режим доступа: https://habr.com/ru/post/95948/ (дата обращения: 07.04.2019).

19. Нативная реализация JS BCrypt для Node. [Электронный ресурс]. — Режим доступа: https://www.npmjs.com/package/bcrypt-nodejs (дата обращения: 11.05.2019).

20. Обзор способов аутентификации [Электронный ресурс]. — Режим доступа: https://habr.com/ru/company/dataart/blog/262817/ (дата обращения: 11.05.2019).

21. Организационно-распорядительные документы [Электронный ресурс]. — Режим доступа: http://textarchive.ru/c-2895353-p3.html (дата обращения: 07.04.2019).

22. Основы информационной безопасности автоматизированных систем [Электронный ресурс]. — Режим доступа: http://baumanpress.ru/books/558/558.pdf (дата обращения: 11.05.2019).

23. Примеры и рекомендации удобных инструкций [Электронный ресурс]. — Режим доступа: https://habr.com/ru/post/153973/ (дата обращения: 07.04.2019).

24. Протокол HTTPS и уязвимости сайта: безопасные связи [Электронный ресурс]. — Режим доступа: https://www.kp.ru/guide/bezopasnost-saita.html (дата обращения: 11.05.2019).

25. Прототипно-ориентированный сценарный язык программирования Javascript [Электронный ресурс]. — Режим доступа: https://www.javascript.com/ (дата обращения: 11.05.2019).

26. Электронный документ. Правила его составления и оформления. Электронная цифровая подпись [Электронный ресурс]. — Режим доступа: https://studwood.ru/594859/menedzhment/elektronnyy_dokument_pravila_sostavleniya_oformleniya_elektronnaya_tsifrovaya_podpis (дата обращения: 07.04.2019).

27. Blowfish (cipher) [Электронный ресурс]. — Режим доступа: https://en.wikipedia.org/wiki/Blowfish_(cipher) (дата обращения: 19.05.2019).

28. CMS Joomla безопасные связи [Электронный ресурс]. — Режим доступа: https://www.joomla.org (дата обращения: 11.05.2019).

29. ExpressJS — фреймворк для создания веб-приложения [Электронный ресурс]. — Режим доступа: https://expressjs.com (дата обращения: 11.05.2019).

30. NodeJS RSA библиотека [Электронный ресурс]. — Режим доступа: https://www.npmjs.com/package/node-rsa (дата обращения: 11.05.2019).

31. SPDY — протокол прикладного уровня для передачи веб-контента [Электронный ресурс]. — Режим доступа: https://www.chromium.org/spdy/spdy-whitepaper (дата обращения: 13.05.2019).

ПРИЛОЖЕНИЕ А

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Российский государственный профессионально-педагогический университет»

Институт инженерно-педагогического образования
Кафедра информационных систем и технологий
Направление подготовки 44.03.04 Профессиональное обучение (по отраслям)
Профиль «Информатика и вычислительная техника»
Профилизация «Информационная безопасность»

УТВЕРЖДАЮ
Заведующий кафедрой

_____ И.А. Суслова
подпись и.о. фамилия

« ____ » _____ 2019 г.

ЗАДАНИЕ
на выполнение выпускной квалификационной работы бакалавра

студента (ки) *4* курса группы *ИБ-402*
Самойлова Александра Сергеевича
фамилия, имя, отчество полностью

1. Тема ***Веб-модуль автоматизации процесса генерации паролей интернет-сайтов «account password changer»***
утверждена распоряжением по институту от «____» _____ 20 г. № ___
2. Руководитель ***Ченушкина Светлана Владимировна***
фамилия, имя, отчество полностью

ученая степень	ученое звание	*Ст. преподаватель каф. ИС* (должность)	*РГППУ* (место работы)

3. Место преддипломной практики *Российский государственный профессионально-педагогический университет г. Екатеринбург*
4. Исходные данные к ВКР 1. ГОСТ 2. 601-2013. Единая система конструкторской документации (ЕСКД). Эксплуатационные документы
2. Основы информационной безопасности автоматизированных систем
5. Содержание текстовой части ВКР (перечень подлежащих разработке вопросов)
Проанализировать литературу и интернет-источники, посвященные аутентификации интернет-сайтов.

Подготовить проект веб-модуля с учетом специфики аутентификации различных системах управления контента (CMS) для создания интернет-сайтов.

Реализовать функционал веб-модуля в выбранных технологиях.

Подготовить интернет-сайт для демонстрации работы веб-модуля.

Разработать обучающий материал по организации процесса аутен-тификации с использованием веб-модуля.

6. Перечень демонстрационных материалов *презентация выполненная в MS Power Point, веб-модуль автоматизации процесса генерации паролей интернет-сайтов «account password changer»*

7. Календарный план выполнения выпускной квалификационной работы

№ п/п	Наименование этапа дипломной работы	Срок выполнения этапа	Процент выполнения ВКР	Отметка руководителя о выполнении
1	Сбор информации по выпускной квалификационной работе		10%	подпись
2	Выполнение работ по разрабатываемым вопросам и их изложение в пояснительной записке:		60%	подпись
2.1	Анализ проблем и тенденций		10%	подпись
2.2	Анализ нормативной документации		10%	подпись
2.3	Анализ литературы по аутентификации		10%	подпись
2.4	Разработка электронной инструкции		15%	подпись
2.5	Исправление недочетов и ошибок веб-модуля		15%	подпись
3	Оформление текстовой части ВКР		10%	подпись
4	Выполнение демонстрационных материалов к ВКР		10%	подпись
5	Нормоконтроль		5%	подпись
6	Подготовка доклада к защите в ГЭК		5%	подпись

8. Консультанты по разделам выпускной квалификационной работы

Наименование раздела	Консультант	Задание выдал		Задание принял	
		подпись	дата	подпись	дата

Руководитель _____ _____ *Задание получил* _____ _____
подпись дата подпись студента дата

9. Дипломная работа и все материалы проанализированы.
Считаю возможным допустить ***Самойлова А.С.*** к защите выпускной квалификационной работы в государственной экзаменационной комиссии.
Руководитель _____ _____
подпись дата

10. Допустить ***Самойлова А.С.*** к защите выпускной квалификационной работы
фамилия и. о. студента
в государственной экзаменационной комиссии (протокол заседания кафедры от «___» _____ 20____г., № _____)
Заведующий кафедрой _____ _____
подпись дата

ПРИЛОЖЕНИЕ Б

В функцию генерации пароля передаются входные параметры, которые учитываются при генерации нового пароля (рисунок 28).

```
const generator = require('generate-password');

module.exports = (length, numbers, symbols, uppercase, excludeSimilarCharacters, exclude, sctrict) => {
    return generator.generate({
        length,
        numbers,
        symbols,
        uppercase,
        excludeSimilarCharacters,
        exclude,
        sctrict
    });
}
```

Рисунок 28 — Функция генерации пароля

Ниже представлена функция отправки нового пароля в электронном письме на почтовый ящик (рисунок 29).

```
// send mail with defined transport object
  transporter.sendMail(mailOptions, async (error, info) => {
    console.log(process.env.EMAIL, process.env.PASSWORD)
    if (error) {
      console.log(error)
    } else {
      connection.query(`UPDATE ${app.dbTable} SET ${mainApp.colUserPassword} =
        if (err) {
          console.log(err)
          res.json({
            ok: false,
            msg: 'Ошибка при обновлении пароля!'
          })
        } else {
          res.json({
            ok: true,
            msg: 'Пароль пользователя успешно изменен!'
          })
        }
      })
    }
  })
```

Рисунок 29 — Функция отправки письма на почтовый ящик