

Министерство образования Республики Беларусь  
Учреждение образования «Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет компьютерного проектирования  
Кафедра проектирования информационно-компьютерных систем  
Дисциплина «Микроконтроллерные устройства»

«К ЗАЩИТЕ ДОПУСТИТЬ»  
Руководитель курсового проекта  
Ассистент кафедры ПИКС  
\_\_\_\_\_.\_\_\_\_\_.2024 К.С. Крез

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**  
к курсовому проекту  
на тему:  
**«Веб-сервис для проверки оригинальности различных работ на основе  
репозитория БГУИР»**

БГУИР КП 1-39 02 01 038 ПЗ

Выполнил студент группы 113802  
РАЗУМОВ Дмитрий Александрович

\_\_\_\_\_  
(подпись студента)  
Курсовой проект представлен на  
проверку \_\_\_\_\_.\_\_\_\_\_.2024

\_\_\_\_\_  
(подпись студента)

Минск 2024

## РЕФЕРАТ

БГУИР КП 1-39 02 01 038 ПЗ

**Разумов, Д.А.** Веб-сервис для проверки оригинальности различных работ на основе репозитория БГУИР: пояснительная записка к курсовому проекту/ Д.А. Разумов. – Минск: БГУИР, 2024. – 71 с.

Пояснительная записка 71 с., 41 рис., 3 табл., 33 источника, 3 приложения.

### ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ПРОВЕРКИ ОРИГИНАЛЬНОСТИ РАБОТ СТУДЕНТОВ НА ОСНОВЕ РЕПОЗИТОРИЯ БГУИР

Цель проектирования: разработка веб-сервиса, предназначенного для автоматизации процесса проверки оригинальности работ студентов и преподавателей на основе репозитория БГУИР.

Методология проведения работы: в ходе выполнения поставленных задач использовались принципы системного подхода, проведен анализ требований к сервису, реализован необходимый функционал.

Результаты работы: проведен анализ требований пользователей к веб-сервису, разработано общетехническое обоснование проекта, реализован механизм проверки оригинальности текстов, уделено внимание обработке ошибок и обеспечению корректной работы сервиса, разработан пользовательский интерфейс.

Веб-сервис позволяет загружать работы для проверки, получать детальные отчеты о проценте уникальности текста и о возможных источниках совпадений в репозитории БГУИР.

Область применения результатов: разработанное программное средство может быть использовано для создания аналогичных сервисов в образовательных учреждениях с целью проверки уникальности научных и учебных работ студентов и преподавателей.

## СОДЕРЖАНИЕ

По содержанию отдельный файл без нумерации, т.к. ссылки слитают и не ставятся страницы

## **ВВЕДЕНИЕ**

В рамках данного курсового проекта был разработан веб-сервис, направленный на помощь в проверке оригинальности различных работ на основе репозитория БГУИР. Данный веб-сервис обеспечивает работоспособность на различных операционных системах, таких как Windows, Linux и Mac OS X.

Основной целью данного проекта является обеспечение возможности автоматизации процесса проверки оригинальности работ студентов и преподавателей БГУИР. Важными задачами, поставленными перед разработчиками, было создание функционала, позволяющего пользователям загружать работы для проверки, получать детальные отчеты о проценте уникальности текста и о возможных источниках совпадений.

При выполнении курсового проекта были учтены государственные стандарты и нормативные документы, а также использованы рекомендации СТП БГУИР 01–2017 [1]. В пояснительной записке к работе будут представлены три основных раздела: анализ исходных данных и постановка задач; проектирование, разработка и тестирование программного средства; эксплуатация программного средства.

Для наглядного представления информации в рамках курсового проекта будут использованы графические материалы, включая ER-диаграмму базы данных, UML диаграмму состояний, UML диаграмму вариантов использования, графический интерфейс программного средства.

Проект был выполнен самостоятельно и протестирован на уникальность с помощью системы «Антиплагиат». Процент оригинальности составил 94.92 %. Все цитирования и использованные материалы будут четко указаны в «Списке использованных источников».

# **1 АНАЛИЗ ИСХОДНЫХ ДАННЫХ И ПОСТАНОВКА ЗАДАЧ**

## **1.1 Анализ исходных данных к курсовому проекту**

Современное образование и научная деятельность требуют эффективных инструментов для проверки оригинальности работ. Проверка оригинальности работ становится необходимостью для поддержания интеллектуальной честности и качества образования. Плагиат становится все более распространенной проблемой, поэтому разработка веб-сервиса для проверки оригинальности работ на базе репозитория БГУИР является актуальной задачей.

Студенты и преподаватели часто сталкиваются с необходимостью проверки работ на плагиат. Однако отсутствие удобного и доступного инструмента для проведения такой проверки может затруднять и замедлять процесс обучения и научной деятельности.

В рамках данного курсового проекта предлагается разработать веб-сервис, который будет предоставлять возможность проверки оригинальности различных работ на основе репозитория Белорусского государственного университета информатики и радиоэлектроники (БГУИР).

Веб-сервис для проверки оригинальности различных работ будет предоставлять удобный интерфейс для загрузки работ и проведения проверки на оригинальность. Основные функциональности сервиса включают:

- авторизация: пользователи смогут войти в систему для доступа к сервису используя свой аккаунт в системе ИИС БГУИР;
- загрузка работ: пользователи смогут загружать свои работы в систему для проверки на плагиат;
- проверка оригинальности: система будет проводить анализ загруженных работ и сравнивать их с содержимым репозитория БГУИР для выявления возможных совпадений;
- отчеты и результаты: после завершения анализа пользователи получат подробный отчет о результатах проверки, который можно скачать в формате pdf;
- управление работами: пользователи смогут просматривать и управлять своими загруженными работами, просматривать историю проверок и делать необходимые действия на основе результатов.

Разработка данного веб-сервиса для проверки оригинальности работ на основе репозитория БГУИР позволит обеспечить эффективную и доступную проверку работ для студентов и преподавателей. Веб-сервис «Антиплагиат» поможет бороться с плагиатом и поддерживать академическую честность, а также хранить все работы в едином структурированном виде.

Разработка подобного веб-сервиса будет способствовать улучшению качества образования и научной деятельности за счет предоставления эффек-

тивного инструмента для проверки оригинальности работ, способствовать развитию инструментов для борьбы с плагиатом и поддержанию академической честности в образовании и науке. Веб-сервис будет полезен как студентам, так и преподавателям, для обеспечения интеллектуальной честности и качества научных и учебных материалов.

## **1.2 Обзор существующих программных средств по теме курсового проекта**

В сети Интернет существует множество веб-сервисов по антиплагиату. Разрабатываются все более и более функциональные программные средства. Однако большое количество данных программных средств не говорит о бессмысленности разрабатываемого приложения. Несмотря на то, что существующие программы предоставляют различные функции, многие пользователи могут позволить только бесплатный интерфейс, не имея доступа к полному функционалу.

Было принято решение рассмотреть наиболее популярные и схожие с разрабатываемым приложением программные средства. Так для сравнения были выбраны: «Антиплагиат», «PlagScan», «Text.ru», «Antiplagius» и «Be1».

Антиплагиат – платформа, которая широко используется для выявления плагиата. Основана на сравнении текста работы с обширной базой данных научных публикаций, учебных материалов, интернет-ресурсов и других источников. Пользователи могут загружать свои работы на платформу и получать детальные отчеты о совпадениях с другими источниками. Помогает преподавателям, студентам и исследователям предотвращать академическое списывание и поощрять оригинальность мысли. Антиплагиат обладает высокой степенью точности в обнаружении плагиата и широко используется в образовательных учреждениях и научных организациях [2].

PlagScan – мощный инструмент для анализа оригинальности текста, который позволяет загружать и анализировать документы в различных форматах. PlagScan предоставляет гибкие настройки поиска плагиата и создает подробные отчеты о совпадениях, помогая пользователям эффективно бороться с копированием и поддерживать академическую честность [3].

Text.ru – многофункциональный инструмент для проверки текста, который помимо анализа грамматики и стиля также предлагает возможность проверки оригинальности контента. Пользователи могут использовать Text.ru для обнаружения плагиата и избежания нежелательных совпадений с другими источниками. Инструмент предоставляет детальные отчеты о совпадениях, что помогает поддерживать высокую уникальность и качество текстов [4].

Antiplagius – специализированное программное обеспечение, разработанное для обнаружения плагиата в онлайн-текстах. Antiplagius проводит сканирование веб-ресурсов с целью выявления совпадений и предоставляет пользователям подробные отчеты о найденных схожих фрагментах. Этот ин-

струмент помогает обеспечить уникальность контента и защитить авторские права [5].

Be1 – инновационное решение для борьбы с плагиатом, предназначенное как для образовательных, так и для коммерческих целей. Be1 предлагает широкий спектр функций для анализа текста и генерирует подробные отчеты о совпадениях с другими источниками. Инструмент помогает пользователям эффективно бороться с плагиатом и поддерживать высокий уровень академической и профессиональной честности [6].

При рассмотрении системы аутентификации важно обратить внимание на функциональные возможности. Обычно экран авторизации содержит поля для ввода адреса электронной почты и пароля, а также предлагает возможность аутентификации через социальные сети. Кроме того, присутствуют кнопки «Войти», «Регистрация» и «Забыли пароль». Однако, в данном случае, ни один из представленных веб-сервисов не соответствует требуемому функционалу, поскольку в системе не предусмотрена регистрация пользователей. Вместо этого необходимо обеспечить только процесс авторизации, который должен быть связан и взаимодействовать с информационной системой БГУИР.

На рисунке 1.1 представлены экраны сравниваемых веб-сервисов «BE1.ru», «Antiplagius», «Text.ru».

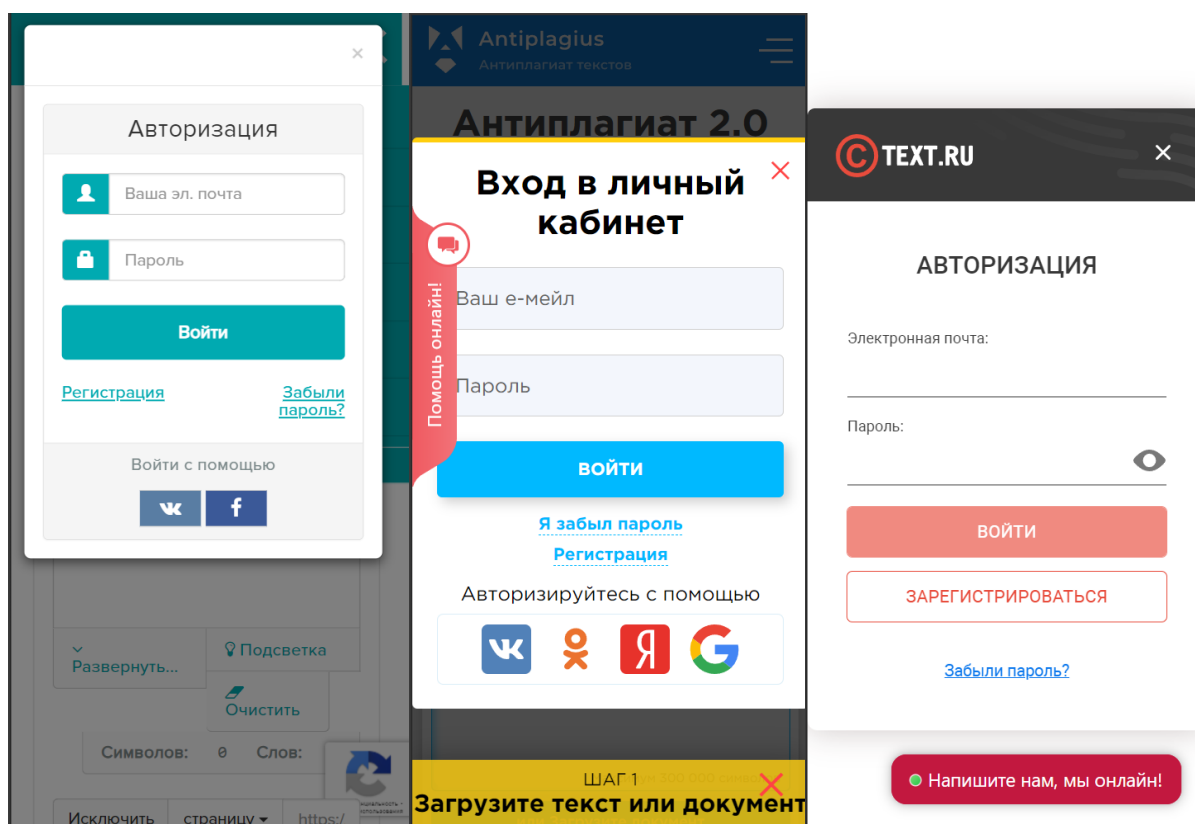


Рисунок 1.1 Экраны авторизации сравниваемых веб-сервисов «BE1.ru», «Antiplagius», «Text.ru»

На рисунке 1.2 представлены экраны сравниваемых веб-сервисов «PlagScan», «Антиплагиат», и разрабатываемого приложения «Антиплагиат».

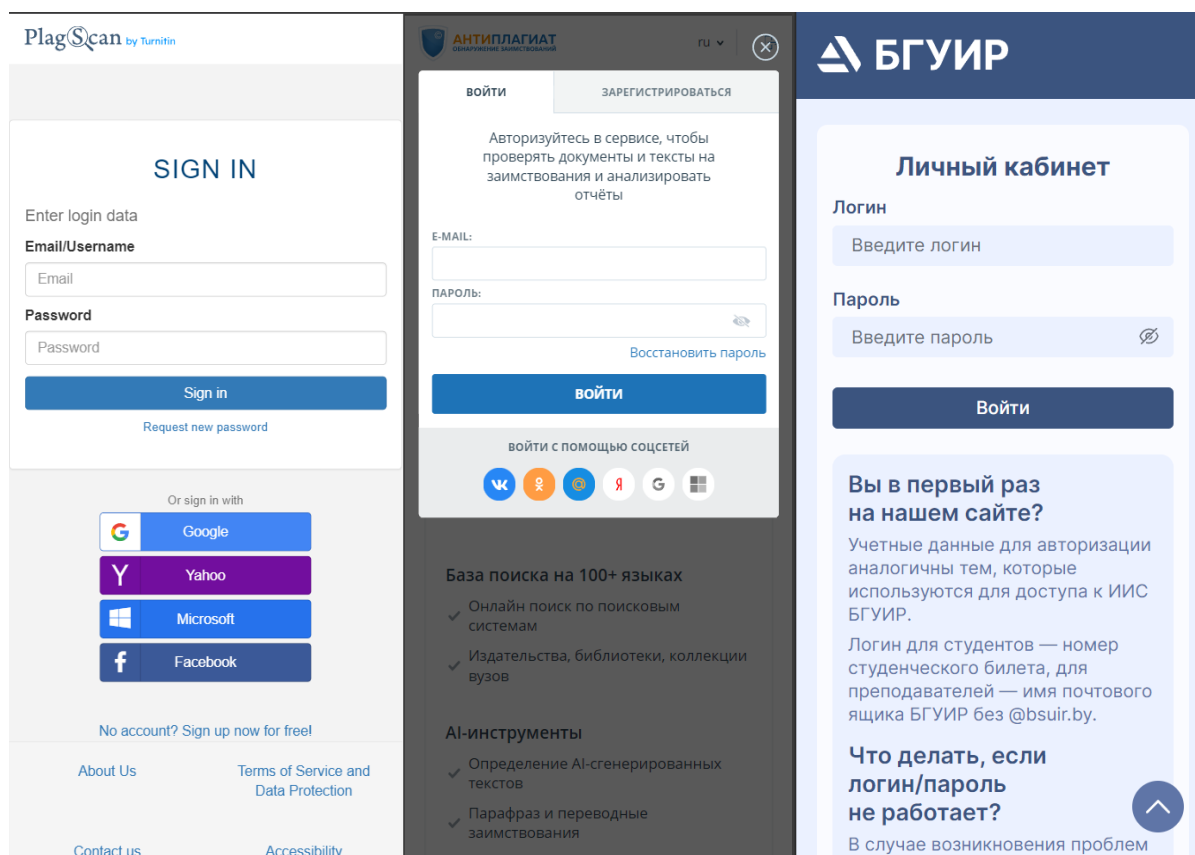


Рисунок 1.2 Экраны авторизации сравниваемых веб-«PlagScan», «Антиплагиат» и разрабатываемого веб-сервиса «Антиплагиат»

Рассмотрим непосредственно экран для работы с антиплагиатом. На рисунке 1.3 представлены экраны работы антиплагиата «BE1.ru», «PlagScan», «Text.ru». На первом скриншоте, относящемся к веб-сервису «Антиплагиат», заметно, что интерфейс достаточно информативен и интуитивно понятен для пользователя. Несмотря на то, что на экране содержится много информации, организована компактно и четко. Второй скриншот представляет экран веб-сервиса «PlagScan», где крупные кнопки и информация о балансе занимают значительное пространство, которое может вызывать необходимость прокрутки страницы для доступа ко всем функциям, что может быть неудобно для пользователей. На третьем скриншоте, принадлежащем веб-сервису «Text.ru», интерфейс выглядит минималистично и лишен лишних элементов, что позволяет пользователю сосредоточиться на основных задачах. Однако, отсутствие мобильной версии может быть значительным недостатком для пользователей, которые предпочитают использовать сервис с мобильных устройств.



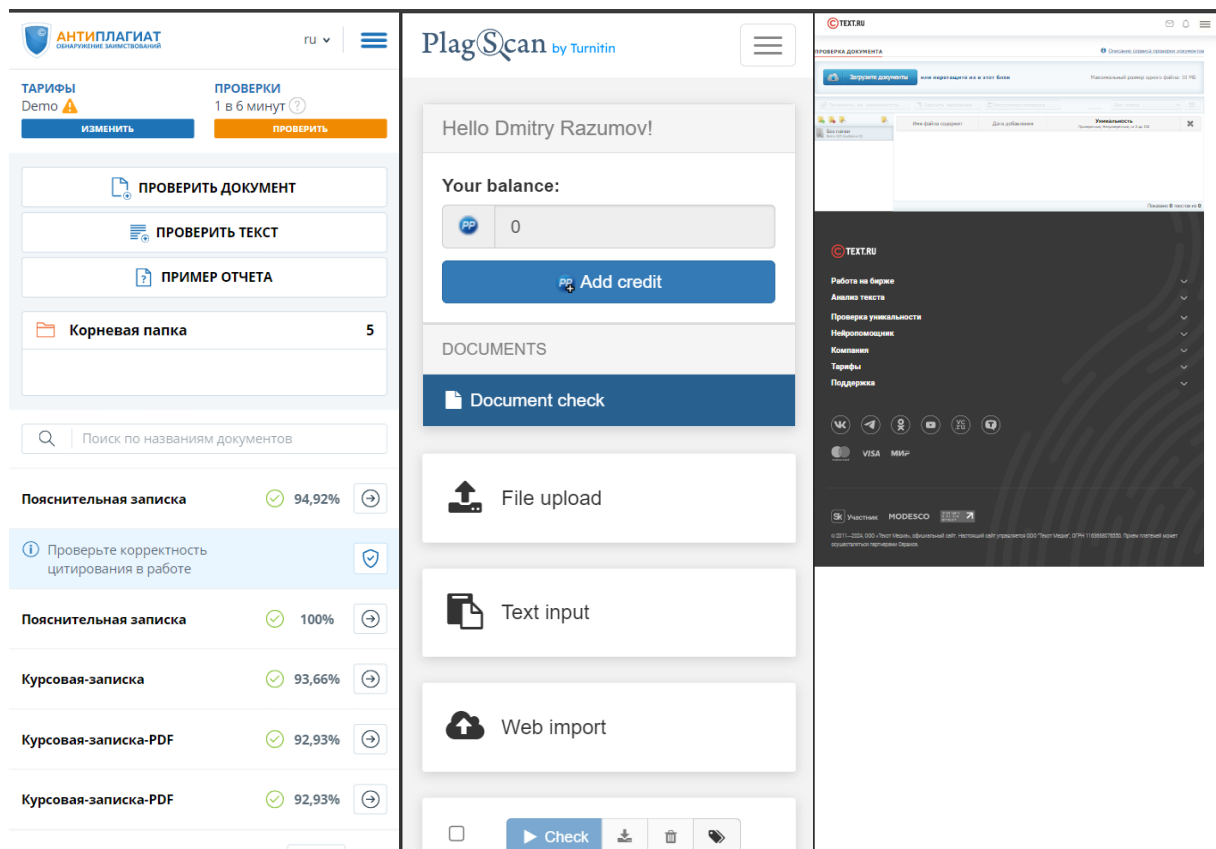


Рисунок 1.3 Экраны работы антиплагиата сравниваемых веб-сервисов «BE1.ru», «PlagScan», «Text.ru»

На рисунке 1.4 представлены экраны работы двух антиплагиатных приложений: «Antiplagius» и «Be1.ru». Первый скриншот отображает интерфейс «Antiplagius», который, несмотря на освобождение от избыточных элементов, выглядит достаточно устаревшим из-за маленького размера шрифта и излишней загруженности верхней части экрана. Кроме того, основной функционал антиплагиата кажется зажатым в узком пространстве. Рекламный баннер внизу экрана также привлекает слишком много внимания, что может отвлечь пользователя. На втором скриншоте представлен интерфейс приложения «BE1.ru», который отличается ясным, привлекательным и легким дизайном. В нем отсутствуют лишние элементы, такие как рекламные баннеры, что позволяет пользователю сосредоточиться исключительно на функционале. Это создает приятный опыт использования и повышает удовлетворенность пользователя. Третий скриншот представляет собой экран разрабатываемого веб-сервиса. Здесь система загрузки документа разработана таким образом, чтобы пользователь мог легко задать название файла, выбрать тип работ и загрузить файл для проверки. Отсутствие рекламы и других отвлекающих элементов позволяет пользователю эффективно выполнять основные задачи без лишних помех и создает благоприятное впечатление о сервисе.

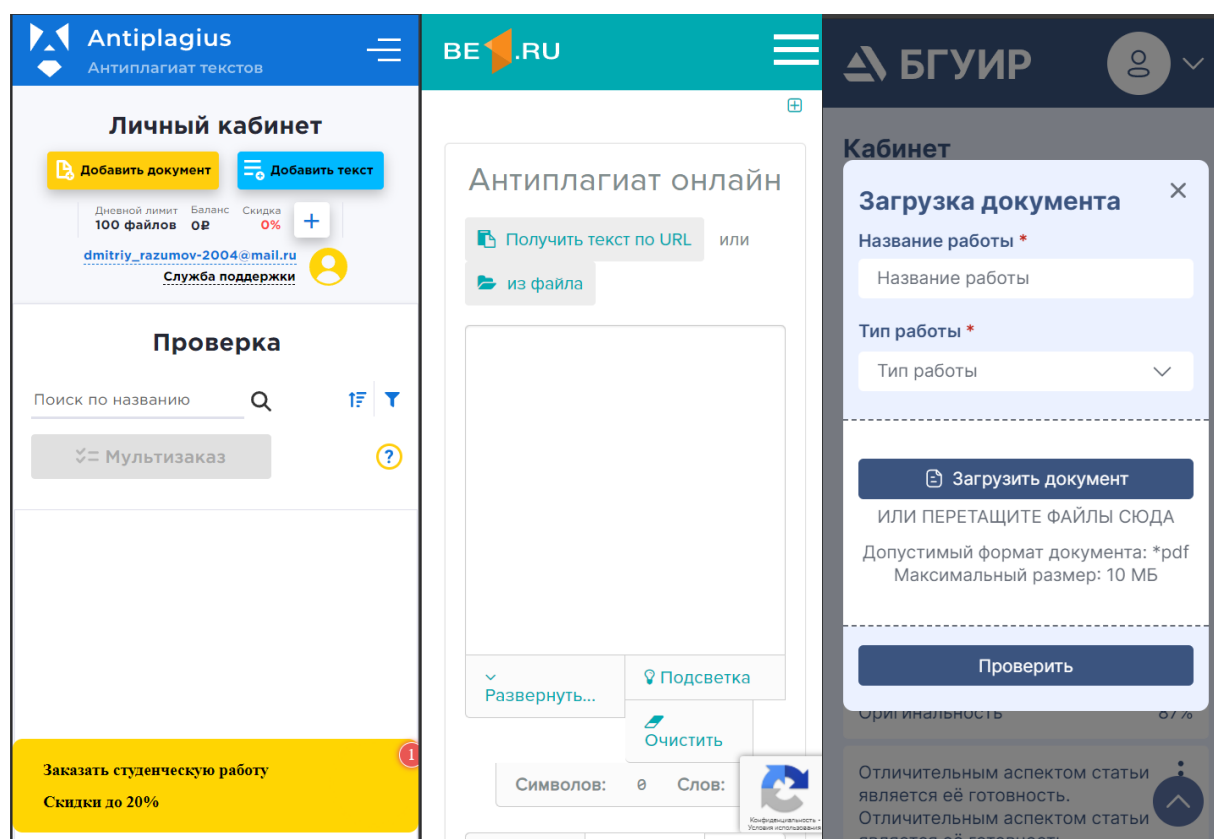


Рисунок 1.4 Экраны работы антиплагиата сравниваемых веб-сервисов «Antiplagius», «BE1.ru» и разрабатываемого веб-сервиса «Антиплагиат»

Возможность перейти в личный кабинет в самом веб-сервисе является значимым преимуществом для пользователей. Среди сравниваемых веб-сервисов личный кабинет представлен в приложениях «Антиплагиат», «Text.ru» и «Be1.ru». Однако, у каждого веб-сервиса есть свое собственное видение того, что должно включать в себя личное пространство пользователя. На экране веб-сервиса «Антиплагиат» представлены такие опции как удаление аккаунта, изменение пароля, адреса электронной почты и других учетных данных. В то время как экран веб-сервиса «Text.ru» выводит информацию о последнем входе в систему, дате регистрации, адресе электронной почты и других аспектах, связанных с профилем пользователя. Экран веб-сервиса «Be1.ru» обладает функционалом вывода и редактирования персональных данных, адреса электронной почты, а также социальных сетей и фотографий пользователя. Это дает пользователям возможность настройки и персонализации своего профиля в соответствии с их предпочтениями. В разрабатываемом веб-сервисе в личном кабинете представлена информация о ФИО и адресе электронной почты, но без возможности редактирования или удаления этих данных. Это может быть полезным в случае, если важно сохранить неподвижность некоторых личных данных для обеспечения безопасности и устойчивости аккаунта.

Экраны веб-сервисов «Антиплагиат», «Text.ru» представлены на рисунке 1.5.

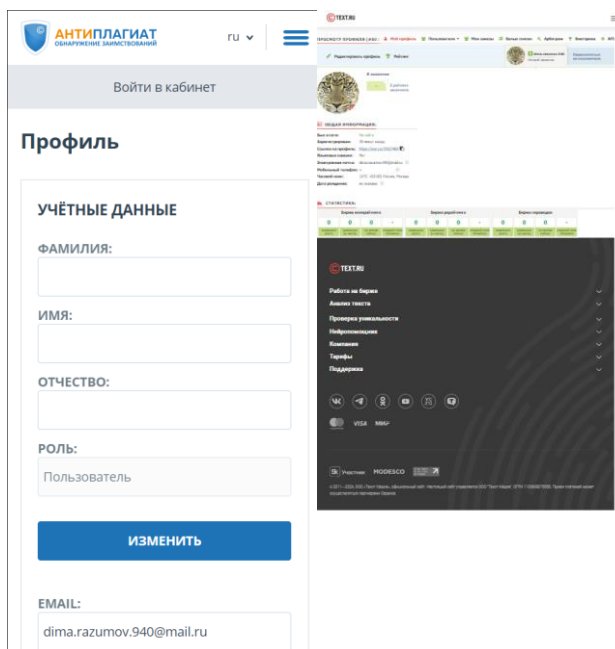


Рисунок 1.5 Экраны личного кабинета сравниваемых веб-сервисов  
«Антиплагиат», «Text.ru»

Экраны веб-сервисов «BE1.ru» и разрабатываемого веб-сервиса «Антиплагиат» представлены на рисунке 1.5.

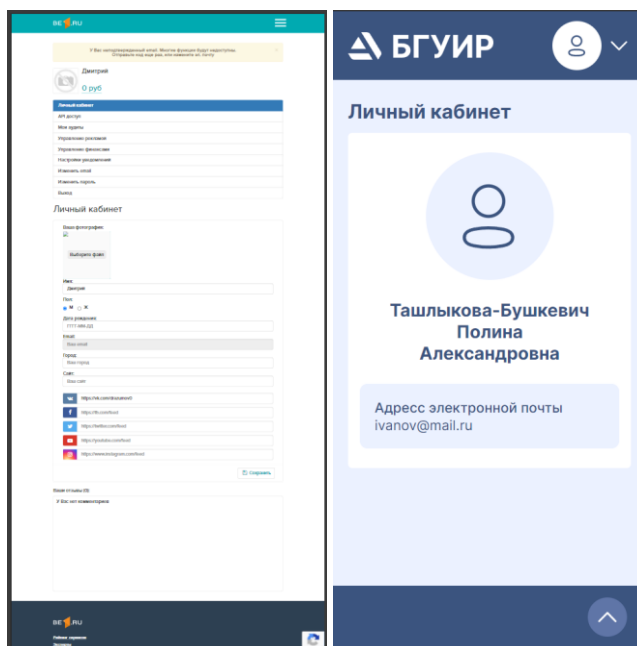


Рисунок 1.6 Экраны личного кабинета сравниваемых веб-сервисов «BE1.ru» и разрабатываемого веб-сервиса «Антиплагиат»

Каждая программа предлагает как платные, так и бесплатные функции. В бесплатном режиме могут быть ограничения на количество отправляемых файлов в день или в определенное время, на размер проверяемого файла, количество слов для проверки, и доступность формата PDF для получения отчета и так далее. Чтобы избежать таких ограничений, необходимо иметь премиум-аккаунт.

Также не все веб-сервисы имеют планшетную и мобильную реализацию, из-за этого появляются неудобства при работе на маленьких экранах.

Можно сделать вывод о том, что в сфере антиплагиата присутствует большое количество веб-сервисов, которые имеют многомиллионную аудиторию, то есть данная сфера очень востребована. Результат сравнения отображен в таблице 1.

Таблица 1 – Результат сравнения веб-сервисов

Критерий сравнения	BE1.RU	Antiplagius	Text.ru	PlagScan	Антиплагиат	Антиплагиат БГУИР
1	2	3	4	5	6	7
1 Адаптивность	-	+	-	+	+	+
2 Регистрация	+	+	+	+	+	-
3 Авторизация через сторонний ресурс	-	-	-	-	-	+
4 Личный кабинет	+	-	+	-	+	+
5 Бесплатный функционал антиплагиата	-	-	+	-	+	+

Сравнение существующих программных средств выделяет определенные характеристики, которым должен соответствовать разрабатываемый веб-сервис. Нет идеального решения, удовлетворяющего всем требованиям, но есть возможность создать веб-сервис, соответствующий большинству из них.

Это начинается от системы аутентификации и заканчивается функционалом антиплагиата.

Для удовлетворения разнообразных потребностей пользователей необходимо, чтобы веб-сервис был интуитивно понятным и легко доступным для использования. Основной акцент следует сделать на точности анализа и скорости обработки информации, чтобы пользователи могли эффективно использовать антиплагиат в повседневной деятельности.

### **1.3 Обоснование и описание выбора языка программирования, средств разработки, используемых технологий и сторонних библиотек**

В данном курсовом проекте использованы:

- язык гипертекстовой разметки текста HTML;
- каскадные таблицы стилей CSS;
- препроцессор SCSS;
- язык программирования JavaScript;
- интегрированная среда разработки VSCode;
- система контроля версий Git и онлайн-сервис GitHub;
- сборщик GULP для стилей, скриптов и изображений;
- для работы с дизайном Figma.

HTML – язык гипертекстовой разметки текста [7-9]. Это набор команд, следуя которым браузеры выводят на монитор различные документы и страницы сайтов. HTML является основным языком разметки для создания структуры веб-страниц. Является стандартом для создания веб-страниц, поддерживаемым всеми современными браузерами. С помощью HTML можно создавать различные типы веб-страниц, начиная от простых статических страниц и заканчивая сложными интерактивными веб-приложениями. HTML используется для создания структуры веб-страницы, определяя содержимое и семантику. Состоит из набора элементов (тегов), каждый из которых определяет определенный тип содержимого. HTML используется для создания основной структуры веб-страниц, которая затем стилизуется с помощью CSS и делается интерактивной с использованием JavaScript.

CSS – формальный язык, предназначенный для описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки [10]. Играет неотъемлемую роль в веб-разработке, предоставляя средства для стилизации и оформления веб-страниц. С помощью CSS можно задавать цвета, шрифты, стили, расположение отдельных блоков и другие аспекты представления внешнего вида веб-страниц. Одним из ключевых преимуществ CSS является возможность отделения логической структуры HTML-документа от визуального оформления, способствуя созданию более гибких, модульных и легко поддерживаемых веб-сайтов. CSS является стандартом в

веб-разработке и поддерживается всеми современными браузерами, что обеспечивает широкую совместимость и надежность. Предоставляет разработчикам широкий спектр инструментов для создания разнообразных дизайнерских решений, включая возможность задания цветов, шрифтов, размеров, отступов, анимации и многого другого. Основной целью разработки CSS является ограждение и отделение описания логической структуры веб-страницы от описания внешнего вида этой веб-страницы. Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления представлением, а также уменьшить сложность и повторяемость в структурном содержимом.

В данной работе в режиме разработки будет использоваться препроцессор SCSS для более гибкого и быстрого написания стилей, который в итоге соберется в один итоговый CSS файл с помощью сборщика. SCSS обладает расширенным и более читабельным синтаксисом по сравнению с обычным CSS, что позволяет разработчику писать код стилей быстрее и эффективнее, а также облегчает поддержку в будущем. Большим преимуществом SCSS перед CSS является возможность использования операторов и функций, переменных и миксинов, а также вложенных стилей. Данные возможности значительно упрощают процесс написания и поддержки стилей веб-приложений, делая более эффективным и удобным для разработчиков [11].

JavaScript (JS) является одним из наиболее распространенных и востребованных инструментов [12][13]. JavaScript обладает универсальностью, позволяя разрабатывать как клиентскую, так и серверную части веб-приложений. Благодаря кроссплатформенной поддержке JavaScript широко применяется на различных устройствах и операционных системах. Активное сообщество разработчиков постоянно развивает новые технологии и библиотеки для улучшения процесса разработки и повышения производительности. JavaScript поддерживает асинхронное программирование и обработку событий, что позволяет создавать отзывчивые и интерактивные веб-приложения. Большой выбор инструментов разработки, включая среды разработки, системы управления версиями и отладчики, делает процесс разработки более эффективным и удобным. В результате JavaScript остается привлекательным выбором для веб-разработки благодаря своей универсальности, широкой поддержке, активному сообществу и большому выбору инструментов.

Так же существует множество интегрированных сред разработки (IDE) для веб-сайтов с помощью JS, HTML, CSS, таких как VSCode, IntelliJ IDEA, WebStorm и другие [14]. Предоставляют разработчикам удобные инструменты для создания, отладки и тестирования приложений. При разработке программного средства для данного курсового проекта выбран VSCode.

VSCode – интегрированное средство разработки (IDE), обладающее обширным набором инструментов для улучшения процесса написания кода [15]. Одним из преимуществ является обширный набор операций рефакторинга, таких как переименование переменных и извлечение методов, которые

значительно упрощают изменение структуры кода и повышают читаемость. Кроме того, VSCode предоставляет контекстно-зависимые подсказки и автодополнение кода, что способствует увеличению производительности и снижению вероятности возникновения ошибок [16]. IDEA, с другой стороны, также предоставляет удобный интерфейс для создания новых проектов и импортирования существующих. Интегрируется с системами контроля версий, такими как Git, Subversion и Mercurial, обеспечивая удобное отслеживание изменений в коде. Более того, IDEA поддерживает различные языки программирования, включая PHP, Kotlin, JavaScript, Python и многие другие, что делает VSCode универсальным инструментом для разработки в рамках одной среды. Дополнительно, IDEA включает инструменты для работы с базами данных и SQL-запросами [17], позволяя разработчикам управлять и редактировать таблицы, выполнять запросы и многое другое, не выходя из среды разработки. Кроме того, с помощью плагинов и расширений разработчики могут расширять функциональность IDE и добавлять дополнительные инструменты по мере необходимости.

Выбор системы контроля версий Git и онлайн-сервиса GitHub обоснован исходя из ряда ключевых преимуществ и возможностей, которые предоставляют для эффективного управления кодом и совместной работы над проектами [18]. Git является одной из самых популярных и широко используемых систем контроля версий в мире разработки программного обеспечения. Распространенность обусловлена простотой в использовании, распределенной структурой, высокой скоростью работы и мощными функциями ветвления и слияния [19]. Git позволяет эффективно отслеживать изменения в коде, возвращаться к предыдущим версиям проекта и управлять конфликтами в изменениях. GitHub, с другой стороны, является онлайн-платформой для хостинга Git-репозитория, которая предоставляет дополнительные инструменты для совместной работы над проектами [20]. Git обладает широким набором функций, таких как задачи и проблемы (issues), запросы на объединение (pull requests), вики, управление правами доступа и многое другое. Использование Git и GitHub позволяет разработчикам эффективно организовывать рабочий процесс, отслеживать изменения в коде, реагировать на обратную связь со стороны других участников проекта, а также обеспечивать надежное хранение и резервное копирование кодовой базы. Благодаря распространенности и популярности, Git и GitHub являются стандартом в индустрии разработки программного обеспечения и обеспечивают надежную основу для успешного ведения проектов.

Выбор сборщика Gulp для обработки стилей, скриптов и изображений обусловлен несколькими факторами, которые способствуют эффективной и удобной разработке веб-приложений [21]. Прежде всего, Gulp является мощным инструментом автоматизации задач, который позволяет разработчикам оптимизировать процесс сборки и развертывания проекта. Обладает простым и интуитивно понятным синтаксисом, поэтому является легко доступным для

новичков и удобным для опытных разработчиков. С помощью Gulp можно легко настроить и выполнять различные задачи, такие как компиляция препроцессоров CSS в обычный CSS, минификация и оптимизация стилей и скриптов, сжатие изображений, автоматическое обновление браузера при изменении файлов и многое другое. Это позволяет существенно ускорить процесс разработки, сделать код более оптимизированным и улучшить производительность веб-приложения. Gulp также обладает богатой экосистемой плагинов, которые позволяют расширять функциональность и адаптировать под конкретные потребности проекта. Использование сборщика Gulp для обработки стилей, скриптов и изображений обеспечивает более эффективную и удобную разработку веб-приложений, улучшает процесс сборки проекта и помогает создать более оптимизированный и производительный код.

Figma – инструмент для дизайна, который позволяет создавать интерфейсы и макеты веб-приложений с учетом современных требований и тенденций [22]. Figma предоставляет возможность совместной работы над проектом, позволяя всем участникам команды, включая дизайнеров и разработчиков, работать над одним и тем же макетом одновременно, снижая время на коммуникацию и ускоряя процесс разработки. Использование Figma для разработки предоставляет разработчикам мощный инструмент для создания качественного дизайна, улучшает командную работу, ускоряет процесс разработки и помогает создать более удобные и привлекательные веб-сервисы.

Наличие ранее описанных мощных и многофункциональных библиотек, взаимодействие с HTML, CSS, JS, а также крайне удобной среды разработки с множеством дополнительных функций, ускоряющих процесс разработки за счет своей производительности, предоставляет огромное количество возможностей для реализации массивного функционала программного средства. Также данное программное средство имеет огромный плюс в виде своей адаптивности под разные операционные системы, что позволит не испытывать проблем с поддержкой в будущем.

#### **1.4 Постановка задач на курсовое проектирование**

Необходимо спроектировать и разработать графический интерфейс веб-сервиса, предназначенного для проверки оригинальности различных работ (курсовые, дипломные, научные, статьи и т. д.) на основе репозитория БГУ-ИР. Цель системы заключается в обеспечении проверки оригинальности работ на основе репозитория БГУИР.

Требования к языкам программного обеспечения являются крайне важными и ориентированы на использование русского языка, что обеспечивает удобство и понятность для конечных пользователей.

Графический интерфейс системы состоит из нескольких страниц:

- главная страница;
- страница авторизации;



- страница личного кабинета;
- страница кабинета пользователя;
- страница роль пользователя.

Главная страница предназначена для общего ознакомления с сервисом и предоставляет основную информацию о его возможностях. Страница авторизации обеспечивает безопасный доступ к функциям системы. Личный кабинет предоставляет возможность пользователям управлять своими данными и настройками. Кабинет пользователя позволяет загружать работы на проверку оригинальности. Роль пользователя определяет его права и доступ к функционалу в соответствии с уровнем его привилегий.

Этапы разработки веб-сервиса включают в себя несколько ключевых шагов. В первом этапе производится обзор предметной области проекта и анализ аналогичных программных средств. Это позволяет определить преимущества и недостатки существующих решений и обосновать выбор языка программирования и средств разработки.

Второй этап включает в себя:

- создание архитектуры программного средства;
- разработка базы данных;
- алгоритмы функционирования приложения;
- проектирование пользовательского графического интерфейса.

Результатом этого этапа является полностью функционирующее программное средство, готовое к тестированию и дальнейшей эксплуатации.

Третий этап предполагает составление инструкции по развертыванию и эксплуатации приложения. Это важный шаг, который обеспечивает понятность и простоту процесса установки и использования системы для конечных пользователей.

Последний этап включает оформление пояснительной записки и графического материала. Сюда входят плакаты и чертежи, такие как ER-диаграмма базы данных, UML диаграмма состояний и вариантов использования, а также графический интерфейс программного средства. Этот этап важен для документирования разработки и предоставления полной информации о системе.

Таким образом, сформулированы этапы разработки, которые необходимо выполнить в ходе курсовой работы для успешной реализации веб-сервиса проверки оригинальности работ на основе репозитория БГУИР.

## **2 ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА И ТЕСТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА**

### **2.1 Проектирование и реализация способа хранения данных программного средства**

База данных – совокупность структурированных взаимосвязанных данных, относящихся к определённой предметной области и организованных для решения определенных задач разными пользователями. База данных предназначена для хранения, изменения и обработки информации.

В ходе разработки используется PostgreSQL – мощная реляционная база данных с открытым исходным кодом, которая широко применяется для веб-сервисов и мобильных приложений [24]. PostgreSQL предоставляет разработчикам надежное хранилище данных с расширенными возможностями работы с данными и поддержкой многих языков программирования для разработки приложений.

PostgreSQL обладает рядом преимуществ, которые делают привлекательный выбор для разработчиков:

1 Надежность: PostgreSQL предлагает высокую степень надежности и целостности данных благодаря своей транзакционной архитектуре и механизмам обеспечения целостности данных.

2 Гибкость и масштабируемость: PostgreSQL легко масштабируется вместе с ростом вашего веб-приложения, обеспечивая высокую производительность даже при больших объемах данных.

3 Расширяемость: PostgreSQL предоставляет широкий спектр расширений и возможностей для настройки, что позволяет разработчикам создавать уникальные и эффективные решения под свои потребности.

4 Поддержка стандартов: PostgreSQL полностью совместим с ANSI SQL и предлагает расширенные возможности работы с географическими данными, JSON и другими современными типами данных.

5 Безопасность: PostgreSQL обеспечивает защиту данных с помощью шифрования и других мер безопасности.

6 Безопасность Существует активное сообщество разработчиков и экспертов, которые поддерживают и развивают PostgreSQL, что обеспечивает непрерывное развитие и поддержку базы данных.

Django предоставляет встроенные механизмы для взаимодействия с различными системами управления базами данных (СУБД), включая PostgreSQL. В этом разделе рассматривается, как Django связывается с PostgreSQL, как управляет базой данных и как обеспечивается работа всех компонентов системы.

Для подключения Django к PostgreSQL необходимо настроить параметры подключения в файле настроек проекта settings.py. Основные параметры включают указание типа базы данных, имя базы данных, пользователя, па-

роль и хост. Пример подключения базы данных в settings.py представлен на рисунке 2.1.

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql',  
        'NAME': 'test',  
        'USER' : 'shemerei',  
        'PASSWORD' : 'qwsdxc09',  
        'HOST' : 'localhost',  
        'PORT' : '5432',  
        'DISABLE_SERVER_SIDE_CURSORS': True,  
    }  
}
```

Рисунок 2.1 – Пример подключения базы данных в settings.py

Эти настройки позволяют Django подключаться к базе данных PostgreSQL и использовать её для хранения и извлечения данных.

Django использует механизм миграций для создания и изменения схемы базы данных на основе моделей, определённых в коде. Миграции позволяют синхронизировать состояние базы данных с определениями моделей.

Для создания миграций используется команда makemigrations, которая анализирует изменения в моделях и создаёт соответствующие файлы миграций.

Для применения миграций и обновления базы данных используется команда migrate.

Эти команды генерируют SQL-запросы, которые выполняются в базе данных для создания или изменения таблиц и других объектов.

Django ORM позволяет взаимодействовать с базой данных, используя Python-классы и методы вместо написания прямых SQL-запросов. Модели Django представляют собой классы, которые автоматически связываются с таблицами в базе данных.

Модели в Django определяются как подклассы django.db.models.Model. Каждое поле модели соответствует колонке в таблице базы данных.

Основой архитектуры хранения данных являются модели, описывающие структуры данных и связи между ними. В данном разделе описаны модели Document, Status, Type и User, их поля, а также используемые методы и сигналы.

Модель Status используется для хранения информации о статусах документов. Программная реализация данной модели показана на рисунке 2.2.

```

class Status(models.Model):
    name = models.CharField(max_length=100, unique=True, verbose_name='Название')
    html_clase = models.CharField(max_length=100, unique=True, verbose_name='html клас', default='queue')

    class Meta:
        db_table: str = 'status'
        verbose_name: str = 'Статус'
        verbose_name_plural: str = 'Статусы'

    def __str__(self):
        return self.name

```

Рисунок 2.2 – Программная реализация модели Status

Поля модели Status:

- name: название статуса;
- html\_clase: CSS-класс для отображения статуса.

Модель Type предназначена для хранения типов работ, связанных с документами. Программная реализация модели Type представлена на рисунке 2.3.

```

class Type(models.Model):
    name = models.CharField(max_length=100, unique=True, verbose_name='Название')

    class Meta:
        db_table: str = 'type'
        verbose_name: str = 'Тип работы'
        verbose_name_plural: str = 'Типы работ'

    def __str__(self):
        return self.name

```

Рисунок 2.3 – Программная реализация модели Type

Поля модели Type:

- name: название типа работы.

Модель Document является основной моделью, описывающей документы, загружаемые пользователями. Программная реализация модели Document представлена на рисунке 2.4.

```

class Document(models.Model):
    user = models.ForeignKey(to=User,
                             on_delete=models.CASCADE,
                             blank=True, null=True
                             # Здесь устанавливаем значение по умолчанию как None
    )
    name = models.CharField(max_length=100, blank=True, null=True, unique=True, verbose_name='Название')
    result = models.DecimalField(max_digits=4, decimal_places=2, blank=True, null=True, verbose_name='Оригинальность')
    status = models.ForeignKey(to=Status, default=Status.objects.first, on_delete=models.CASCADE)
    type = models.ForeignKey(to=Type, on_delete=models.CASCADE, default=Type.objects.first)
    time_created = models.DateTimeField(auto_now_add=True, verbose_name='Дата и время загрузки документа')
    data = models.FileField(upload_to="pdf_files/", verbose_name='документ')
    txt_file = models.FileField(upload_to='txt_files/', blank=True, null=True)

    def save(self, *args, **kwargs):
        if not self.user_id:
            self.user = kwargs.pop('user', None) # Используем переданный пользователь, если есть
            super().save(*args, **kwargs)

    class Meta:
        db_table: str = 'Document'
        verbose_name: str = 'Документ'
        verbose_name_plural: str = 'Документы'
        ordering: ("time_created")

    def __str__(self):
        return self.name

```

Рисунок 2.4 – Программная реализация модели Document

Поля модели Status:

- user: ссылка на модель User, указывающая владельца документа;
- name: название документа;
- result: процент оригинальности документа;
- type: тип работы, связанной с документом;
- time\_created: дата и время загрузки документа;
- data: файл PDF документа;
- txt\_file: файл с текстовым содержимым документа;
- status: текущий статус документа.

В Django стандартная модель пользователя предоставляет базовый функционал для аутентификации и управления пользователями. Включает в себя необходимые поля и методы для работы с пользователями, такие как логин, пароли и профили. Django предлагает две базовые модели для работы с пользователями: AbstractUser и AbstractBaseUser. В большинстве случаев используется AbstractUser, которая включает в себя все стандартные поля и методы, необходимые для пользователя.

Модель AbstractUser является частью пакета django.contrib.auth.models и наследует от AbstractBaseUser и PermissionsMixin. Предоставляет удобный способ для расширения модели пользователя, добавляя дополнительные поля и методы.

Модель AbstractUser включает следующие поля:

- username: Имя пользователя. Уникальное поле, используемое для идентификации пользователя;
- first\_name: Имя пользователя;

- last\_name: Фамилия пользователя;
- email: Электронная почта пользователя;
- password: Хэшированный пароль пользователя;
- is\_staff: Флаг, указывающий, является ли пользователь членом персонала сайта;
- is\_active: Флаг, указывающий, активен ли пользователь;
- date\_joined: Дата и время регистрации пользователя;
- last\_login: Дата и время последнего входа пользователя.

Для добавления новых полей к модели User, создаём пользовательскую модель, наследующуюся от AbstractUser, добавим отчество (sur\_name) и аватар пользователя (image). Программная реализация расширения модели User представлена на рисунке 2.5.

```
class User(AbstractUser):
    sur_name = models.CharField(max_length=100, blank=True, null=True, verbose_name='отчество')
    image = models.ImageField(upload_to='user_image', blank=True, null=True, verbose_name='Аватар')
    class Meta:
        db_table: str = 'User'
        verbose_name: str = 'Пользователь'
        verbose_name_plural: str = 'Пользователи'

    def __str__(self):
        return self.username
```

Рисунок 2.5 – Программная реализация модели расширения User

Поля модели User добавленные в результате расширения:

- sur\_name: Отчество пользователя. Поле CharField с максимальной длиной 100 символов;
- image: Аватар пользователя. Поле ImageField, позволяющее загружать и хранить изображения.

После создания пользовательской модели необходимо указать Django использовать её вместо стандартной модели. Для этого в файле настроек settings.py добавляем следующую строку AUTH\_USER\_MODEL = 'your\_app\_name.User'.

Использование ORM Django позволяет легко управлять связями между моделями и автоматизировать выполнение рутинных задач, что значительно упрощает разработку и поддержку приложения.

PostgreSQL представляет собой надежные инструменты для разработчиков в области веб-разработки, обеспечивая высокую производительность, надежность и защиту данных.

## 2.2 Проектирование и разработка графического интерфейса

Проектирование и разработка графического интерфейса – один из самых важных этапов создания веб-сервиса, так как именно интерфейс является основным способом взаимодействия пользователя с приложением.

Перед началом проектирования необходимо определить целевую аудиторию и ее потребности. Это поможет создать удобный и понятный интерфейс, который будет максимально соответствовать требованиям пользователей.

Для построения грамотного интерфейса нужно соблюдать принципы UI/UX [25].

UI (User Interface) и UX (User Experience) – процесс создания удобного и привлекательного интерфейса для пользователей. Этот процесс включает в себя разработку дизайна, который будет удобен для использования и будет привлекать пользователей к продукту.

UI (User Interface) – интерфейс, который пользователи видят и используют для взаимодействия с продуктом. UI включает в себя все элементы, такие как кнопки, меню, тексты и изображения, которые помогают пользователям выполнять задачи в приложении.

UX (User Experience) – опыт, который пользователь получает при использовании программного продукта. UX включает в себя такие аспекты, как удобство использования, скорость работы, интуитивность интерфейса и т. д.

Есть несколько основных правил, которым должно соответствовать приложение:

1 Интуитивность и простота: приложение должно быть простым в использовании, чтобы пользователь мог быстро разобраться в нем и понять, как выполнять нужные действия.

2 Визуальная согласованность: все элементы приложения должны быть оформлены в едином стиле, чтобы пользователь легко мог ориентироваться в приложении.

3 Удобство навигации: приложение должно иметь понятную и логичную систему навигации, чтобы пользователь мог найти нужную информацию.

4 Скорость работы: приложение должно быстро загружаться и работать без задержек, чтобы пользователь не терял терпение и не переключался на другое приложение.

Таким образом, необходимо создавать простой и интуитивно понятный интерфейс, который будет легко осваиваться пользователями. Отсутствие лишних элементов и явных инструкций также может улучшить юзабилити интерфейса.

Пользовательский интерфейс веб-сервиса «Антиплагиат» состоит из следующих экранов:

– страница авторизации;

- страница кабинета;
- страница с результатами;
- страница личного кабинета;
- страница с ролями пользователей;
- модальное окно «Загрузка документа»;
- модальное окно «Отправить документ на защиту»;
- модальное окно «Оценить документ».

Рассмотрим графический дизайн для мобильных устройств.

В первую очередь, при проектировании пользовательского интерфейса нужно создать грамотный экран «Авторизации». Дизайн экрана «Авторизации» представлен на рисунке 2.6.

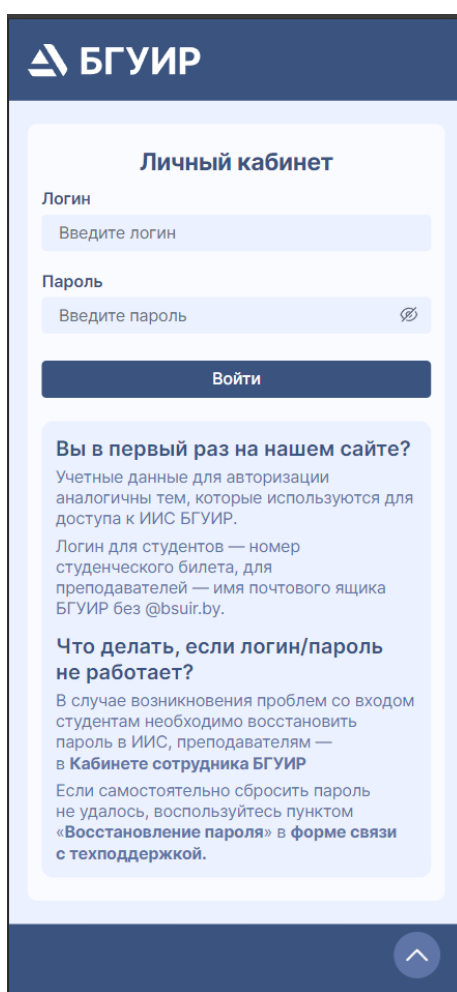


Рисунок 2.6 – Дизайн страницы «Авторизации»

Страница «Авторизации» является лицом приложения, это тот экран, с которым пользователь будет взаимодействовать в первую очередь.

Данный экран имеет минимальное количество элементов дизайна. Имеет поля логина и пароля, а также для пароля кнопка показать/скрыть па-



роль, кнопку «Войти», а также информационный блок со ссылками на сторонние ресурсы для помощи пользователю в случае если пользователь находится первый раз на веб-сервисе или если логин/пароль не работает.

После успешной авторизации автоматически запускается экран «Кабинета». На странице «Кабинета» размещены некоторые документы пользователя, если они имеются. Также страница «Кабинета» содержит поиск по названию документа, пагинацию и кнопку «Проверить документ». Для каждого документа есть кнопка в виде иконки «Подробнее», нажимая появляется панель по работе с документом с двумя кнопками. Дизайн страницы «Кабинет» со сразу открытым выпадающим списком для одного из документов представлен на рисунке 2.7.

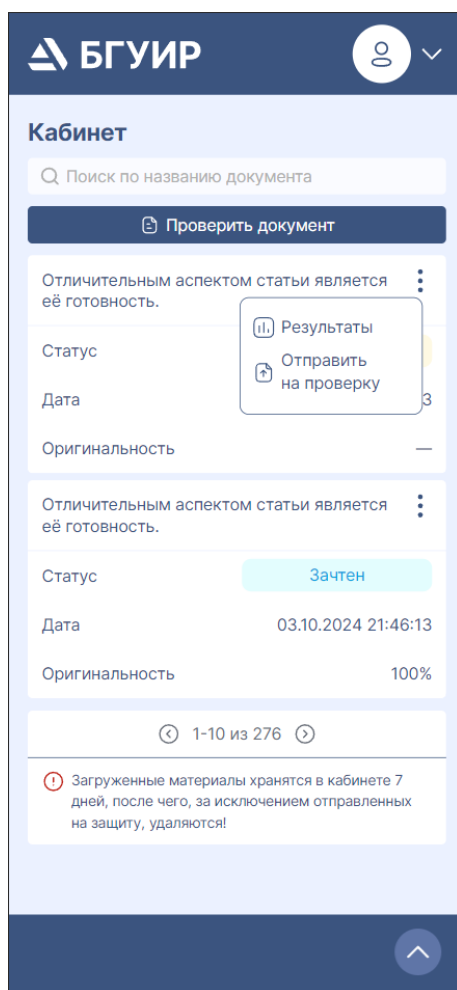


Рисунок 2.7 – Дизайн страницы «Кабинет» со сразу открытым выпадающим списком для документа

Нажав на кнопку «Отправить на проверку», в выпадающем меню документа, открывается модальное окно «Отправить документ на защиту» с подробной информацией о документе и кнопкой «Отправить». Дизайн модального окна «Отправить документ на защиту» показан на рисунке 2.8.

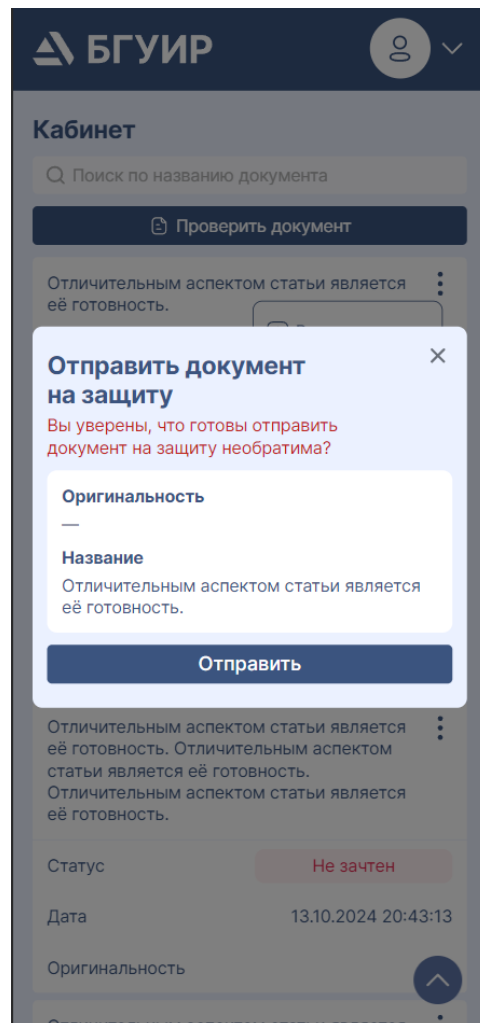


Рисунок 2.8 – Дизайн модального окна «Отправить документ на защиту»

Кнопка «Проверить документ» на странице «Кабинета» является ключевым элементом в процессе загрузки новых материалов для проверки оригинальности. При нажатии на эту кнопку открывается модальное окно, которое предоставляет удобный интерфейс для загрузки документа. Модальное окно не только содержит поля для ввода названия работы и выбора типа работы из списка, но и область для загрузки самого документа в формате \*.pdf. Это позволяет пользователю удобно и быстро загружать необходимые материалы для проверки оригинальности. Особенно удобной функцией является возможность предварительного просмотра загруженного документа прямо в модальном окне. Это обеспечивает пользователю дополнительный контроль над загруженными данными и упрощает процесс проверки перед отправкой на анализ. Кроме того, наличие функции удаления загруженного документа позволяет пользователю аккуратно управлять своими данными, удаляя лишнее или ошибочно загруженное содержимое.

Пример дизайна модального окна «Загрузка документа» представлен на рисунке 2.9.

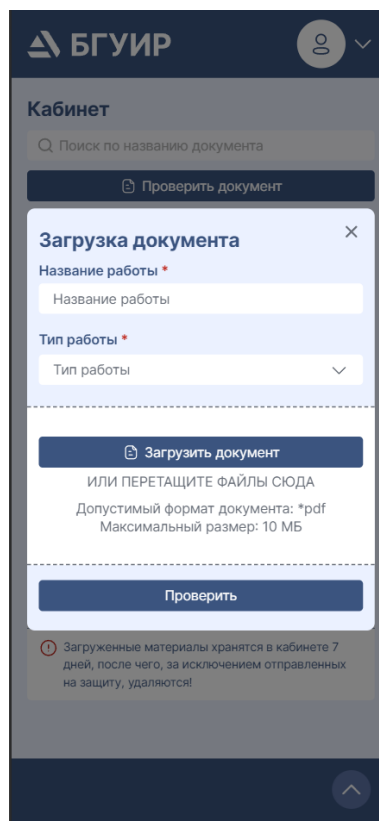


Рисунок 2.9 – Дизайн модального окна «Загрузка документа»

Когда пользователь авторизован, на каждой странице есть в шапке веб-сервиса иконка, нажимая на которую появляется выпадающее меню с возможностью перехода по страницам и кнопкой «Выхода» для выхода из системы. Дизайн выпадающего меню показан на рисунке 2.10.

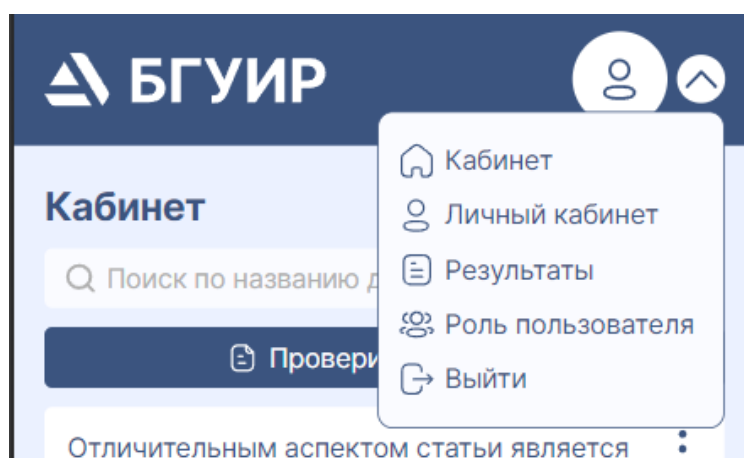


Рисунок 2.10 – Дизайн выпадающего списка меню

Нажимая на кнопку «Личный кабинет» в выпадающем меню, происходит переход на экран «Личный кабинет». Дизайн страницы «Личный кабинет» представлен на рисунке 2.11.

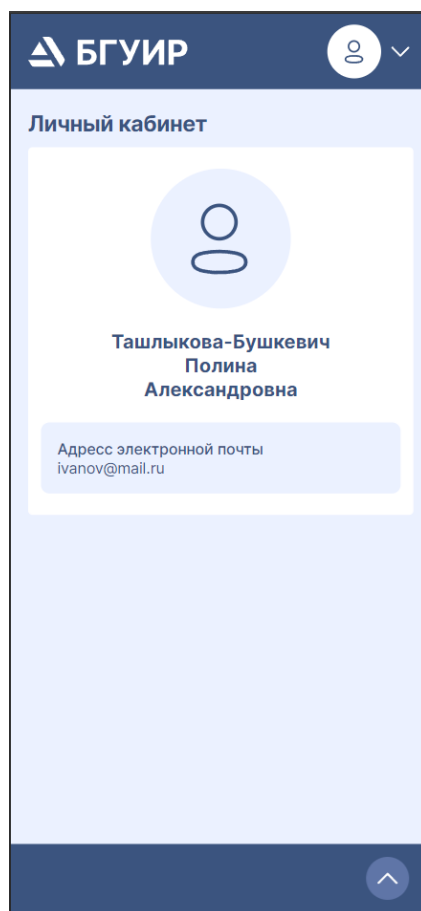


Рисунок 2.11 – Дизайн страницы «Личный кабинет»

Кнопка «Роль пользователя» в выпадающем меню предоставляет доступ к странице «Роль пользователя», однако эта страница доступна только для администраторов. Для обычных пользователей не отображается в выпадающем меню, что способствует упрощению интерфейса и предотвращает лишнее путаницу.

На странице «Роль пользователя» пользователи могут выполнить поиск по ФИО и управлять пагинацией, что делает процесс поиска и навигации более удобным. Это особенно важно при работе с большим объемом данных, таких как информация о пользователях.

Таким образом, кнопка «Роль пользователя» обеспечивает доступ к соответствующей странице, а дизайн этой страницы способствует удобству и эффективности пользовательского опыта.

Дизайн страницы «Роль пользователя», изображенный на рисунке 2.12, разработан с учетом удобства использования.

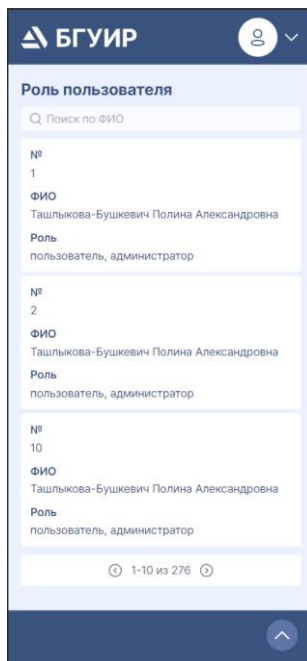


Рисунок 2.12 – Дизайн страницы «Роль пользователя»

Нажимая на кнопку «Результаты» в выпадающем меню осуществляется переход на страницу «Результаты». На странице отображается поиск по ФИО, пагинация и результаты. Дизайн страницы «Результаты» с открытым выпадающим списком для данного результата представлен на рисунке 2.13.

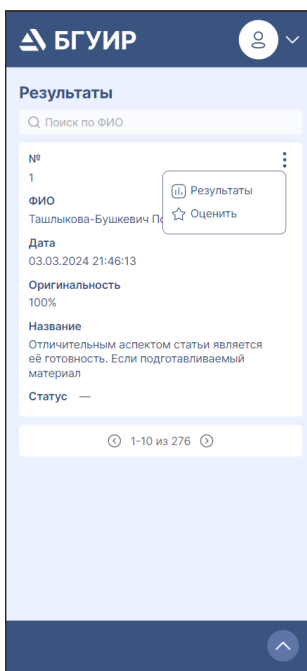


Рисунок 2.13 – Дизайн страницы «Результаты» со сразу выпадающим списком для результата

При нажатии на кнопку «Оценить» открывается модальное окно с названием «Оценить документ». Это окно предоставляет подробную информацию о результате проверки документа и содержит две кнопки действий: «Отправить на доработку» и «Зачтен».

Дизайн модального окна «Оценить документ» представлен на рисунке 2.14.

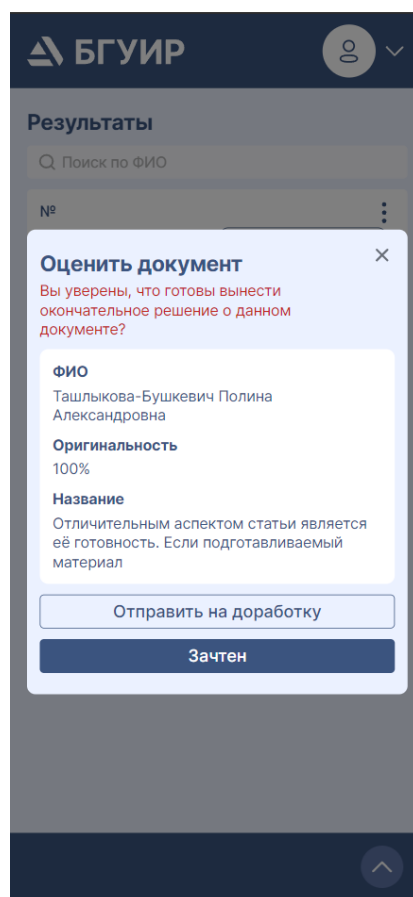


Рисунок 2.14 – Дизайн модального окна «Оценить документ»

Проектирование и разработка графического интерфейса приложения процесс, требующий не только технической компетентности, но и творческого подхода. Правильно спроектированный интерфейс может сделать веб-сервис более привлекательным для пользователей и удобным в использовании. Привлекательный и интуитивно понятный интерфейс может стать ключевым фактором успешного взаимодействия с сервисом, привлекая новых пользователей и удерживая уже существующих.

В рамках данного раздела осуществлялась разработка веб-сервиса «Антиплагиат», и графический интерфейс играл важную роль в создании. Интерфейс был спроектирован с учетом требований пользователей и особенностей функционала сервиса, чтобы обеспечить максимальное удобство использования и приятный внешний вид.

## 2.3 Тестирование программного средства

Тестирование программного обеспечения имеет ключевое значение в процессе разработки, внедрения и сопровождения программного продукта. Основная цель – обнаружение ошибок, дефектов и недостатков в программе, а также проверка соответствия программного продукта требованиям и ожиданиям пользователей [26].

По степени автоматизации тестирование разделяется на автоматизированное и ручное. Автоматизированное тестирование программного обеспечения основано на написании кода. Методика данного тестирования позволяет описать ожидаемый сценарий, сравнить с реальным поведением и указать расхождения. Такое тестирование позволяет получить быстрые и точные результаты. Однако важно понимать, что инструменты автоматизированного тестирования могут оказаться затратными. Поэтому автоматизация наиболее применима в сложных приложениях с большой функциональной частью. Значительным минусом является то, что с помощью автоматизированного тестирования нельзя оценить цветовые решения программного средства. Это не является первостепенным, но может повлиять на качество программного средства. Ручное тестирование – более долгий и трудоемкий процесс. Но позволяющий получить первый пользовательский опыт и отзыв реального человека. Такое тестирование также позволяет протестировать дизайн приложения. Внедрение ручного тестирования всегда менее затратно, чем автоматизированного. Для небольших проектов более рационально использовать именно ручное тестирование.

В данном дипломном проекте используется ручное тестирование. Результаты тестирования представлены в таблице 3.

Таблица 3 – Тестирование программного продукта

Сценарий	Выполняемое действие	Ожидаемый результат
1	2	3
1 Авторизация пользователя	Открыть веб-сервис	На экране отображена авторизация в веб-сервис
	Вести логин и пароль в соответствующих полях	На экране отображены заполненные поля
	Нажать на кнопку «Войти»	Совершен переход на «Кабинет»
2 Просмотр «Кабинета»	Нажать иконку выпадающего меню в шапке веб-сервиса	Открыто меню

Продолжение таблицы 3

	Нажать на ссылку «Кабинет»	Совершен переход на страницу «Кабинет
3 Загрузка нового документа	Нажать на кнопку «Проверить документ»	Открывается модальное окно «Загрузка документа»
	Заполнить поля, загрузить документ и нажать кнопку «Проверить»	Документ загружен и отображается в «Кабинете»
4 Отправить документ на проверку	Нажать на троеточие у документа	Открывается выпадающее меню для данного документа
	Нажать на кнопку «Отправить на проверку»	Открывается модальное окно «Отправить документ на защиту»
	Нажать на кнопку «Отправить»	Документ отправляется на проверку
5 Скачать результаты	Нажать на троеточие у документа	Открывается выпадающее меню для данного документа
	Нажать на кнопку «Результаты»	Скачиваются результаты данного документа в формате *.pdf
6 Просмотр «Личного кабинета»	Нажать иконку выпадающего меню в шапке веб-сервиса	Открыто меню
	Нажать на ссылку «Личный кабинет»	Совершен переход на страницу «Личный кабинет»
7 Просмотр страницы «Роль пользователей» (режим админа)	Нажать иконку выпадающего меню в шапке веб-сервиса	Открыто меню
	Нажать на ссылку «Роль пользователя»	Совершен переход на страницу «Роль пользователя»
8 Просмотр страницы «Результаты» (режим админа)	Нажать иконку выпадающего меню в шапке веб-сервиса	Открыто меню
	Нажать на ссылку «Результаты»	Совершен переход на страницу «Результаты»



Продолжение таблицы 3

9 Скачать результаты (режим админка)	Нажать на троеточие у документа	Открывается выпадающее меню для документа
	Нажать на кнопку «Результаты»	Скачиваются результаты в формате *.pdf
10 Оценить документ	Нажать на троеточие у документа	Открывается выпадающее меню для данного документа
	Нажать на ссылку «Оценить»	Открывается модальное «Оценить документ»
	Нажать на кнопку «Отправить на доработку»	Документ отправляется на доработку
	Нажать на кнопку «Зачтен»	Документ зачтен
11 Поиск по названию документа или по ФИО	Ввести поисковые данные в строку поиска	Вывод искомой информации
12 Выход из веб сервиса	Нажать иконку выпадающего меню в шапке веб-сервиса	Открыто меню
	Нажать на кнопку «Выйти»	Выход из программы и переход на страницу «Авторизации»
13 Скроллинг страницы в самый вверх	Нажать в правом низу экрана кнопку со скроллингом	Скроллинг страницы в самый вверх
14 Пагинация	Нажимая на кнопку вправо	Отображаются следующие 10 записей
	Нажимая на кнопку влево	Отображаются предыдущие 10 записей

После тщательного тестирования программного продукта не было обнаружено никаких ошибок, что свидетельствует об успешном прохождении всех этапов проверки. Это означает, что все компоненты программы функционируют корректно и соответствуют установленным требованиям. Результаты тестирования подтверждают высокое качество разработанного программного средства. Отсутствие ошибок после тестирования обеспечивает надежность и стабильность работы программы, что повышает уверенность в ее эффективности и способности удовлетворить потребности пользователей. Таким образом, успешное завершение этапа тестирования подтверждает, что программное средство готово к предоставлению конечным пользователям.

## **3 ЭКСПЛУАТАЦИЯ ПРОГРАММНОГО СРЕДСТВА**

### **3.1 Ввод в эксплуатацию и обоснование минимальных технических требований к оборудованию**

Для размещения веб-сервиса в интернете необходимо выбрать платформу, на которой веб-сервис будет доступен для пользователей.

Хостинг – сервис, который предоставляет серверное пространство и другие ресурсы для того, чтобы ваш веб-сайт был доступен онлайн [27]. Существует множество хостинг-провайдеров, каждый из которых предлагает свои услуги и особенности.

Существуют различные хостинг-провайдеры, такие как: «Beget», «Holsinger», «SiteGround», «Bluehost».

Hostinger предлагает доступные цены и широкий спектр функций, известные своими быстрыми серверами и простым в использовании интерфейсом. Однако, может быть ограничение на количество баз данных и доменов, что может быть недостатком для некоторых проектов [28].

SiteGround предлагает высокую производительность и превосходную поддержку, инструменты безопасности и управления сайтом также находятся на высоком уровне [29].

Bluehost предлагает надежный хостинг с хорошей скоростью и производительностью. Предоставляют бесплатный SSL-сертификат и доменное имя в течение первого года [30].

Beget обеспечивает удобное и относительно недорогое размещение веб-проектов. Кроме того, у Beget есть функция бесплатного использования для одного сайта, что может быть очень выгодно для студентов или начинающих разработчиков [31][32].

В данном случае используется услуга хостинга от Beget, предоставляющая возможность разместить бесплатно один сайт, одну базу данных и имеющая ограничения по объему памяти, так как Beget предлагает хорошее сочетание надежности, простоты использования и доступности. Кроме того, бесплатное использование для одного сайта делает хостинг привлекательным вариантом для тех, кто хочет сэкономить на размещении своих проектов в интернете.

Процесс загрузки веб-сервиса на хостинг «Beget» начинается с прохождения регистрации, где пользователю необходимо указать свой номер телефона. После успешной регистрации на указанный номер приходят данные для авторизации: логин и пароль. Этот процесс обеспечивает безопасный доступ к панели управления хостингом и предоставляет пользователю возможность управлять своим веб-сервисом. Далее пользователь переходит на страницу авторизации, где вводит полученные учетные данные. При корректном вводе данных пользователь попадает в панель управления, представленную на рисунке 3.1.

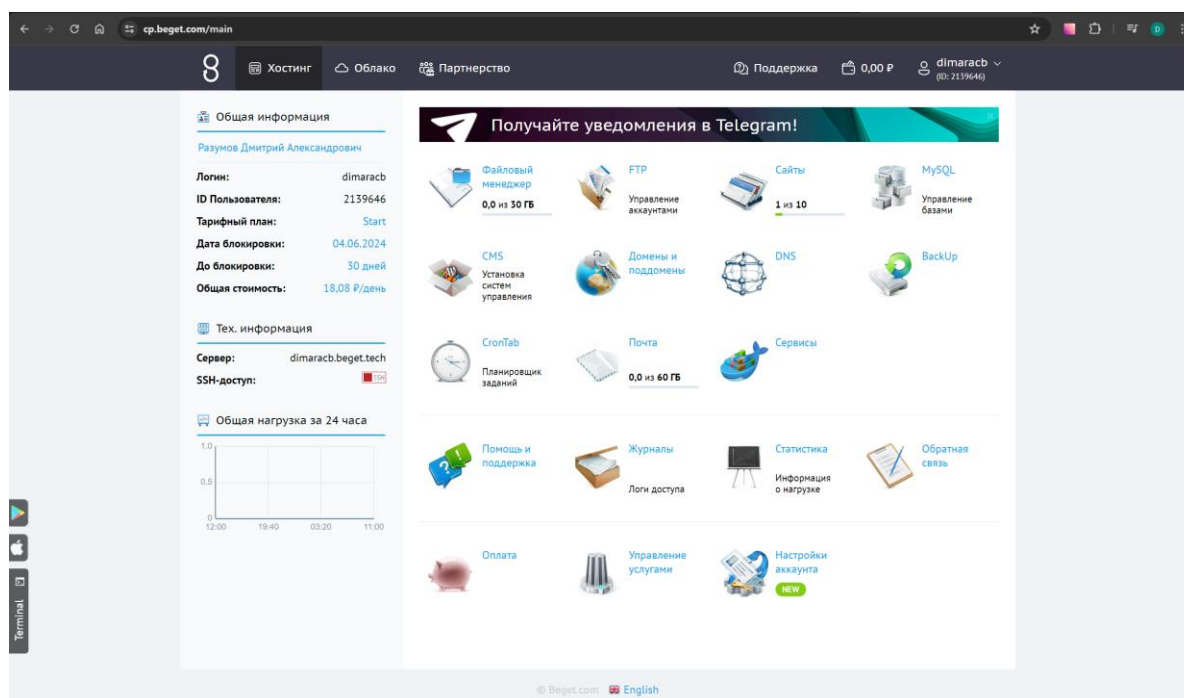


Рисунок 3.1 – Панель управления Beget

На локальном устройстве архивируем все файлы курсового проекта и при помощи Файлового менеджера загружаем на сервер, предварительно удалив присутствующие файлы по умолчанию папке public. Пример загрузки файлов через файловый менеджер представлен на рисунке 3.2.

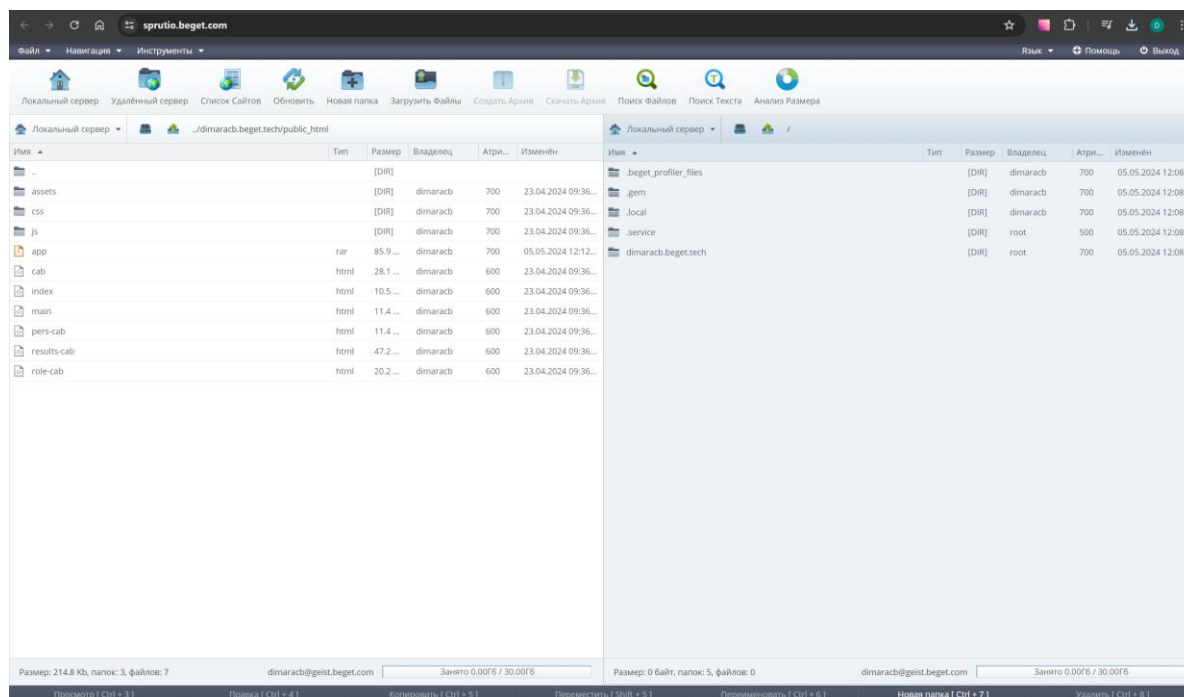


Рисунок 3.2 – Загрузка файлов через файловый менеджер на хостинг

Далее нужно указать версии используемых языков программирования на хостинге, такие же, которые использовались в разработке, для совмещения версий.

В случае успешной загрузки всех данных на хостинг веб-сервис откроется по выданной ссылке в левом меню блока.

Для работы и внесения правок на сервере нужно зайти в раздел FTP и создать FTP-аккаунт, к которому можно подключиться через FileZilla или другое подобное программное средство для внесения изменений сразу на сервере. Пример создания FTP-аккаунта представлен на рисунке 3.3.

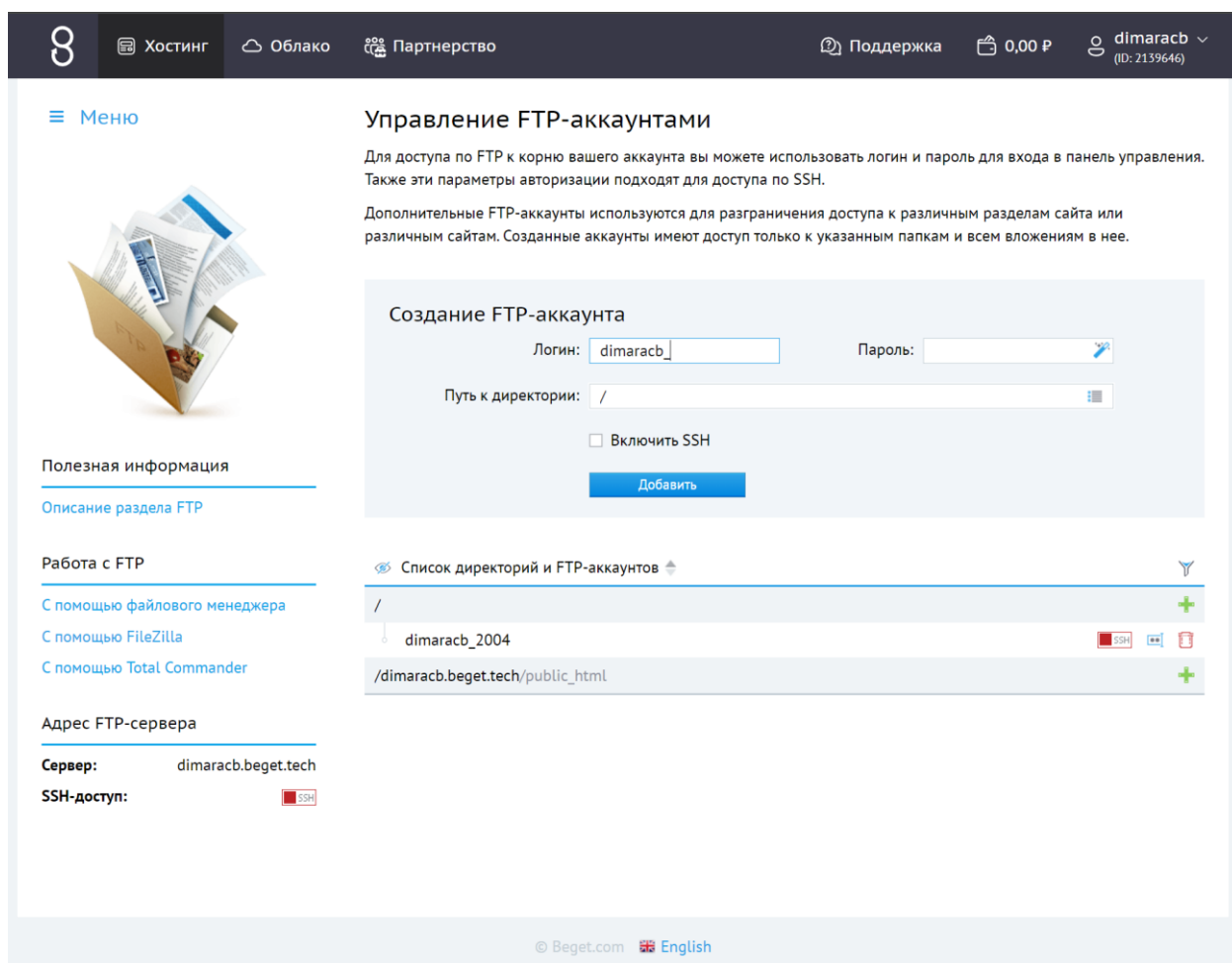


Рисунок 3.3 – Создание FTP-аккаунта

После установления соединения с FTP-сервером получаем доступ ко всем файлам, что позволяет вносить необходимые изменения в веб-сервис. Этот процесс обеспечивает удобный способ управления файлами на хостинге и позволяет осуществлять оперативное внесение изменений в веб-сервис.

Пример подключения через FileZilla представлен на рисунке 3.4.

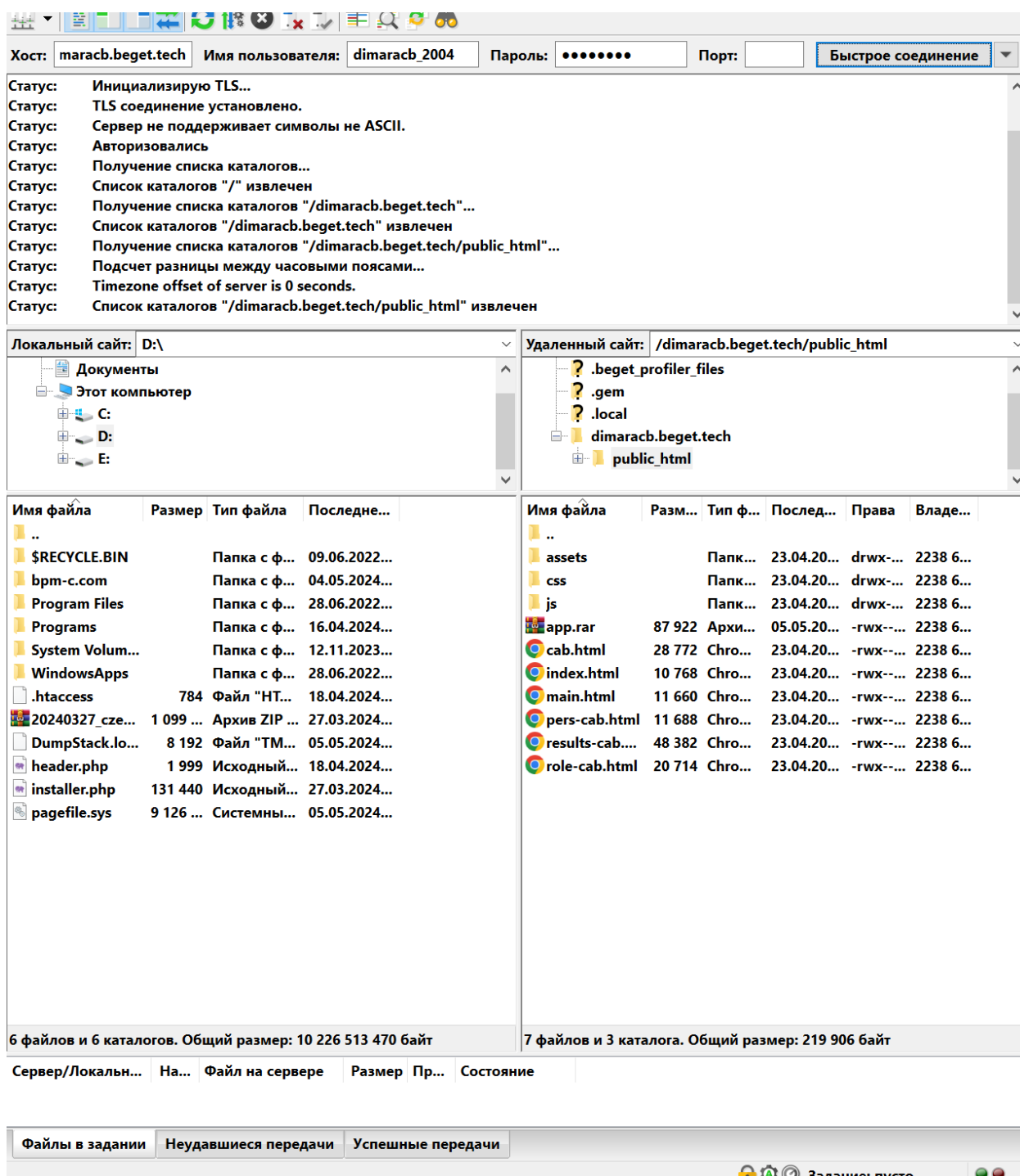


Рисунок 3.4 – Подключение к веб-сайту для изменений через FileZilla

Обоснование минимальных технических требований к оборудованию для открытия верстки сайта основывается на нескольких ключевых аспектах:

1 Производительность: для открытия требуется оборудование, способное обрабатывать HTML, CSS и другие веб-ресурсы без задержек. Даже на

начальном этапе проекта, пользователи ожидают быстрого и плавного отображения страниц сайта.

2 Ресурсы браузера: браузеры должны иметь достаточное количество ресурсов для загрузки и отображения верстки. Это включает в себя оперативную память, процессорное время и место для кэширования данных.

3 Совместимость и доступность: оборудование должно соответствовать требованиям современных браузеров и учитывать различные устройства и разрешения экранов, чтобы обеспечить правильное отображение верстки на всех платформах.

4 Быстродействие сети: для эффективной загрузки сайта необходима стабильная и быстрая сетевая связь, особенно в случае использования медиа-контента или внешних ресурсов.

5 Безопасность: оборудование должно быть защищено от возможных угроз, таких как вредоносные программы или атаки злоумышленников, чтобы обеспечить безопасность.

Исходя из вышеприведенных аспектов, минимальные технические требования к оборудованию для открытия данного веб-сервиса могут включать в себя следующее: персональный компьютер или ноутбук современного поколения с процессором не ниже Intel Core i3 или аналогичным, оперативной памятью не менее 4 ГБ и достаточным объемом свободного места на диске для временного хранения данных.

Также важна поддержка 3G, 4G, Wi-Fi для подключения к сети Интернет. Использование современного веб-браузера, такого как Google Chrome, Mozilla Firefox, Safari, Microsoft Edge, обеспечивающего оптимальное отображение и исполнение веб-страниц, также является важным аспектом.

Стабильное интернет-соединение с минимальной скоростью загрузки и выгрузки данных обеспечит быструю загрузку верстки сайта и минимальные задержки при использовании веб-сервиса.

Эти минимальные требования обеспечат возможность открытия веб-сервиса на устройствах пользователя с минимальными задержками и с учетом основных аспектов производительности и безопасности.

### **3.2 Руководство по эксплуатации программным средством**

Для открытия необходимо иметь устройство, которое имеет возможность выхода в сеть Интернет. Веб-сервис загружен на хостинг и открывается по общедоступной ссылке в сети интернет [33].

Рассмотрим руководство по эксплуатации программным средством в десктопной версии.

В первую очередь, при проектировании пользовательского интерфейса, нужно создать грамотный экран «Авторизации». Экран «Авторизации» является лицом приложения, тот экран, с которым пользователь будет взаимодействовать в первую очередь. Должен сочетать в себя приятный интерфейс, ко-

торый сможет понравиться пользователю, а также быть интуитивно понятен. Пример страницы «Авторизации» представлен на рисунке 3.5.

**БГУИР**

### Личный кабинет

Логин

123

Пароль

\*\*\*\*\*

Войти

**Вы в первый раз на нашем сайте?**  
Учетные данные для авторизации аналогичны тем, которые используются для доступа к ИИС БГУИР.  
Логин для студентов — номер студенческого билета, для преподавателей — имя почтового ящика БГУИР без @bsuir.by.

**Что делать, если логин/пароль не работает?**  
В случае возникновения проблем со входом студентам необходимо восстановить пароль в ИИС, преподавателям — в Кабинете сотрудника БГУИР.  
Если самостоятельно сбросить пароль не удалось, воспользуйтесь пунктом «Восстановление пароля» в форме связи с техподдержкой.

Рисунок 3.5 – Страница «Авторизации»

При посещении веб-сервиса впервые или при возникновении проблем с входом, пользователь обнаружит информационный блок с подсказками о действиях в данной ситуации. В этом блоке будут представлены ссылки на сторонние ресурсы, где пользователь сможет получить дополнительную информацию или помощь по восстановлению логина или пароля.

Кроме того, для удобства пользователей, предусмотрена функция отображения пароля в открытом виде. Нажав на специальную иконку в виде глаза, пользователь сможет сделать ввод пароля видимым, что облегчит процесс ввода и предотвратит возможные ошибки.

После заполнения всех необходимых полей и нажатия кнопки «Войти», пользователь будет перенаправлен на страницу «Кабинета». Если в процессе входа возникнет какая-либо ошибка, пользователю немедленно будет показано сообщение о неверно заполненных полях. Кроме того, рядом с каждым полем будет расположена иконка, при наведении на которую пользователь сможет получить более подробную информацию об ошибке. Это поможет пользователям исправить ошибки быстро и без лишних усилий.

Пример исключительной ситуации с подробным описанием ошибки и функционалом показа пароля представлен на рисунке 3.6.

**Личный кабинет**

Логин

123

Некорректный формат логина!

Менее 8 символов

Пароль

12312

Войти

Рисунок 3.6 – Исключительная ситуация с подробным описанием ошибки и функционалом показа пароля при авторизации

Каждый пользователь, зайдя в систему получает доступ к навигационному меню в шапке веб-сервиса при наведении на иконку, для перехода между страницами или выхода из аккаунта нажав кнопку «Выйти». Также есть кликабельный логотип, который ведет на страницу «Кабинет». Пример навигационного меню показан на рисунке 3.7.

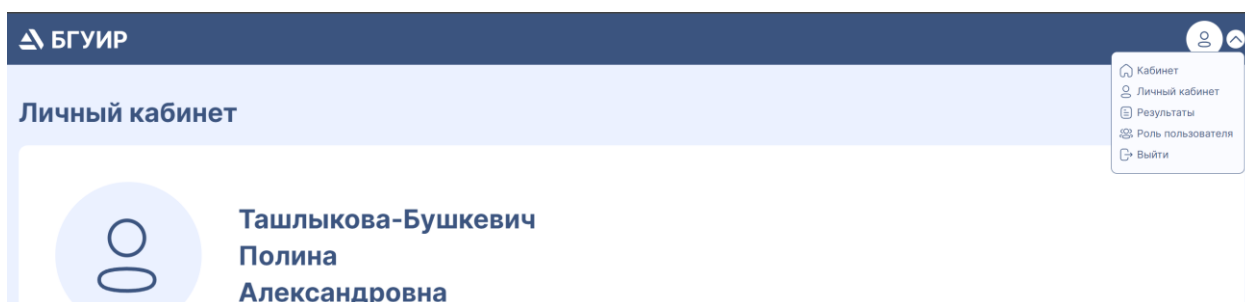


Рисунок 3.7 – Навигационное меню

В разделе «Личный кабинет» пользователь может увидеть свое ФИО и адрес электронной почты. Однако, изменение личной информации невозможно, поскольку вся информация берется из репозитория БГУИР и автоматически загружается в систему. Это гарантирует актуальность и достоверность данных, а также предотвращает возможные ошибки или несоответствия в личной информации пользователя.



Пример страницы «Личного кабинета» показан на рисунке 3.8.

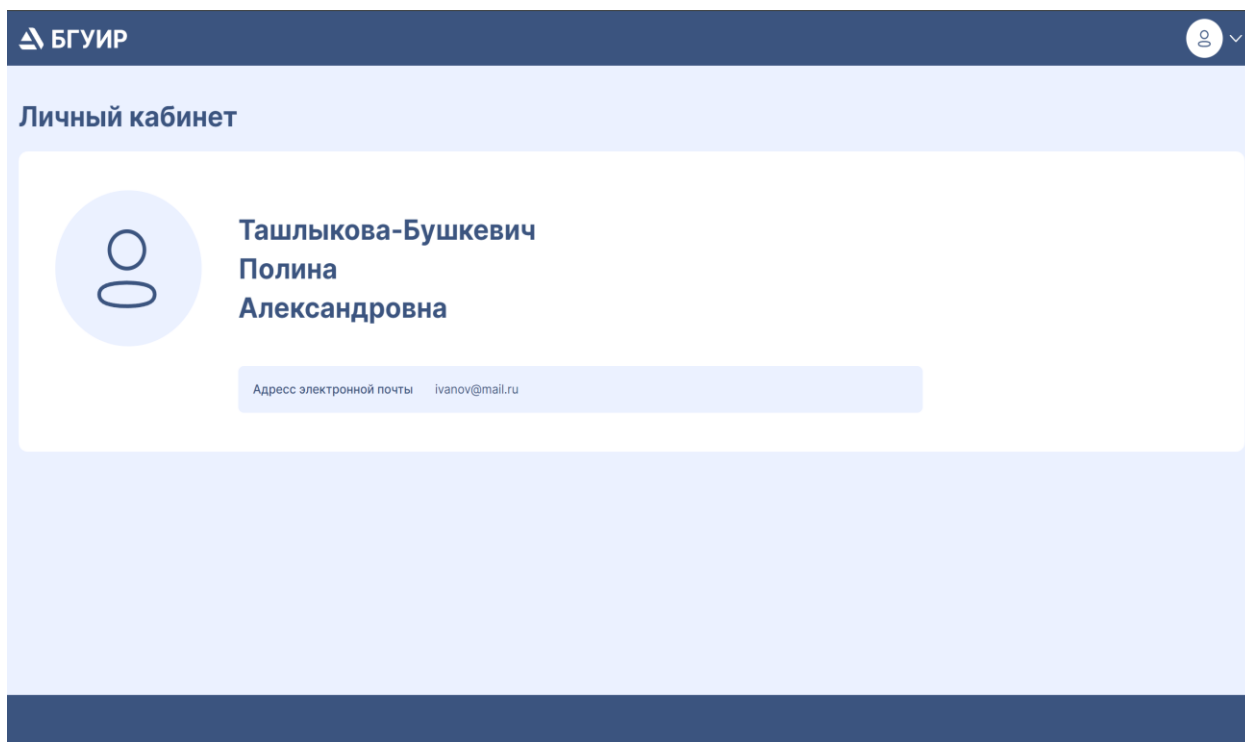


Рисунок 3.8 – Страница «Личного кабинета»

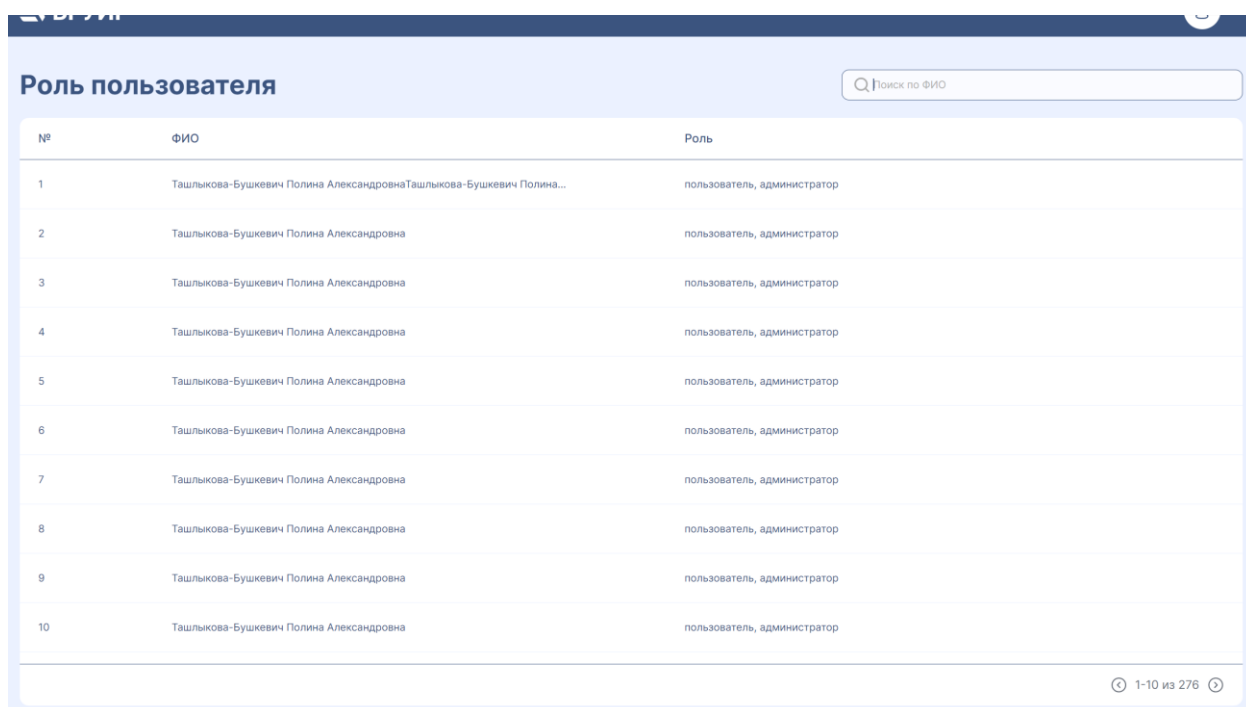
После перехода на страницу «Роль пользователя» пользователь обнаружит, что данный функционал доступен только в режиме администратора, что гарантирует безопасность и контроль доступа к данным. Это важно для обеспечения конфиденциальности информации и предотвращения несанкционированного доступа к данным пользователей.

Для удобства взаимодействия с этой страницей предусмотрено поле поиска по ФИО, которое облегчает быстрое нахождение нужной информации. Это особенно удобно в случае, когда необходимо быстро найти конкретного пользователя среди большого количества записей.

Дополнительно, внизу страницы представлена пагинация, которая позволяет переходить между записями с информацией о пользователях, если на экране отображается более 10 пользователей одновременно. Это позволяет упорядочить вывод информации и обеспечить легкость навигации по большому объему данных. Пагинация делает процесс просмотра информации более удобным и эффективным.

Основной контент страницы состоит из вывода информации о пользователях, их номера, ФИО и роли. Такое представление данных упрощает восприятие информации и обеспечивает пользователям быстрый доступ к нужной информации, необходимой для управления ролями пользователей.

Пример страницы «Роль пользователя» показан на рисунке 3.9.



№	ФИО	Роль
1	Ташлыкова-Бушкевич Полина АлександровнаТашлыкова-Бушкевич Полина...	пользователь, администратор
2	Ташлыкова-Бушкевич Полина Александровна	пользователь, администратор
3	Ташлыкова-Бушкевич Полина Александровна	пользователь, администратор
4	Ташлыкова-Бушкевич Полина Александровна	пользователь, администратор
5	Ташлыкова-Бушкевич Полина Александровна	пользователь, администратор
6	Ташлыкова-Бушкевич Полина Александровна	пользователь, администратор
7	Ташлыкова-Бушкевич Полина Александровна	пользователь, администратор
8	Ташлыкова-Бушкевич Полина Александровна	пользователь, администратор
9	Ташлыкова-Бушкевич Полина Александровна	пользователь, администратор
10	Ташлыкова-Бушкевич Полина Александровна	пользователь, администратор

Рисунок 3.9 – Страница «Роль пользователя»

При переходе на страницу «Кабинет» пользователь обнаружит удобное поле поиска по названию документа, что значительно облегчает процесс поиска конкретного документа среди имеющихся записей. Это особенно полезно в случаях, если большое количество документов усложняет поиск.

Дополнительно, внизу страницы присутствует пагинация, которая позволяет переходить между записями с информацией о документах в случае, если на экране отображается более 10 документов одновременно. Такой подход упорядочивает информацию и обеспечивает удобство навигации по большому объему данных.

На странице также присутствует информационный блок с подсказкой, который помогает пользователям быстро разобраться в функционале страницы и использовать весь функционал наиболее эффективно.

Основной контент страницы представляет собой вывод информации о документах, включая их название, статус, дату загрузки на проверку, оригинальность и соответствующую кнопку действия. Статус документа может быть различным, включая «в очереди», «проверен», «не зачтен», «в защите» и «зачтен». Этот вывод информации помогает пользователям отслеживать статусы своих документов и оперативно реагировать на изменения.

Дата, отображаемая рядом со статусом, равна дате загрузки документа на проверку, что позволяет пользователям ориентироваться в хронологии загрузки и обработки документов.

Пример страницы «Кабинета» показан на рисунке 3.10.

Кабинет			
Поиск по названию документа		Проверить документ	
Название	Статус	Время	Оригинальность
Отличительным аспектом статьи является её готовность.	206111 в очереди	03.03.2024 21:46:13	—
Пояснительная записка	Проверен	02.06.2023 12:24:56	87%
Отличительным аспектом статьи является её готовность. Отличительным аспектом статьи...	Не зачтен	13.10.2024 20:43:13	91%
Отличительным аспектом статьи является её готовность является её готовность	В защите	03.10.2024 21:46:13	83%
Отличительным аспектом статьи является её готовность.	Зачтен	03.10.2024 21:46:13	100%

Загруженные материалы хранятся в кабинете 7 дней, после чего, за исключением отправленных на защиту, удаляются!

1-10 из 276

Рисунок 3.10 – Страница «Кабинета»

Если же название документа слишком большое и не помещается в одну строку, обрезается и оканчивается троеточием, но для того чтобы увидеть полное название документа, нужно навести на название и на экране покажется блок с полным названием. Пример полного названия документа представлен на рисунке 3.11.

Название	Статус	Время	Оригинальность
Отличительным аспектом статьи является её готовность.	206111 в очереди	03.03.2024 21:46:13	—
Пояснительная записка	Проверен	02.06.2023 12:24:56	87%
Отличительным аспектом статьи является её готовность. Отличительным аспектом статьи...	Не зачтен	13.10.2024 20:43:13	91%
Отличительным аспектом статьи является её готовность. Отличительным аспектом статьи является её готовность. Отличительным аспектом статьи является её готовность.	В защите	03.10.2024 21:46:13	83%
Отличительным аспектом статьи является её готовность.	Зачтен	03.10.2024 21:46:13	100%

Рисунок 3.11 – Пример вывода полного названия документа

Нажимая на иконку троеточия у каждого документа, появляется выпадающее меню с выбором «Результаты» нажимая происходит скачивание результатов документа и кнопка «Отправить на проверку» для отправки документа на проверку.

Пример клика на иконку троеточия показан на рисунке 3.12.

Название	Статус	Время	Оригинальность
Отличительным аспектом статьи является её готовность.	206111 в очереди	03.03.2024 21:46:13	—
Пояснительная записка	Проверен	02.06.2023 12:24:56	87%
Отличительным аспектом статьи является её готовность. Отличительным аспектом статьи...	Не зачтен	13.10.2024 20:43:13	91%
Отличительным аспектом статьи является её готовность является её готовность	В защите	03.10.2024 21:46:13	83%
Отличительным аспектом статьи является её готовность.	Зачтен	03.10.2024 21:46:13	100%

Рисунок 3.12 – Пример клика на иконку троеточия

Нажимая на кнопку «Отправить на проверку» открывается модальное окно для подтверждения отправки документа на защиту. В данном модальном окне выводиться вопрос подтверждения действия и информация о документе, а также кнопка подтверждения отправки на проверку, нажимая на которую данный документ отправиться на проверку. Для того чтобы закрыть данное модальное окно, необходимо нажать на иконку крестика или же на область вне модального окна. Пример модального окна для отправки на проверку представлен на рисунке 3.13.

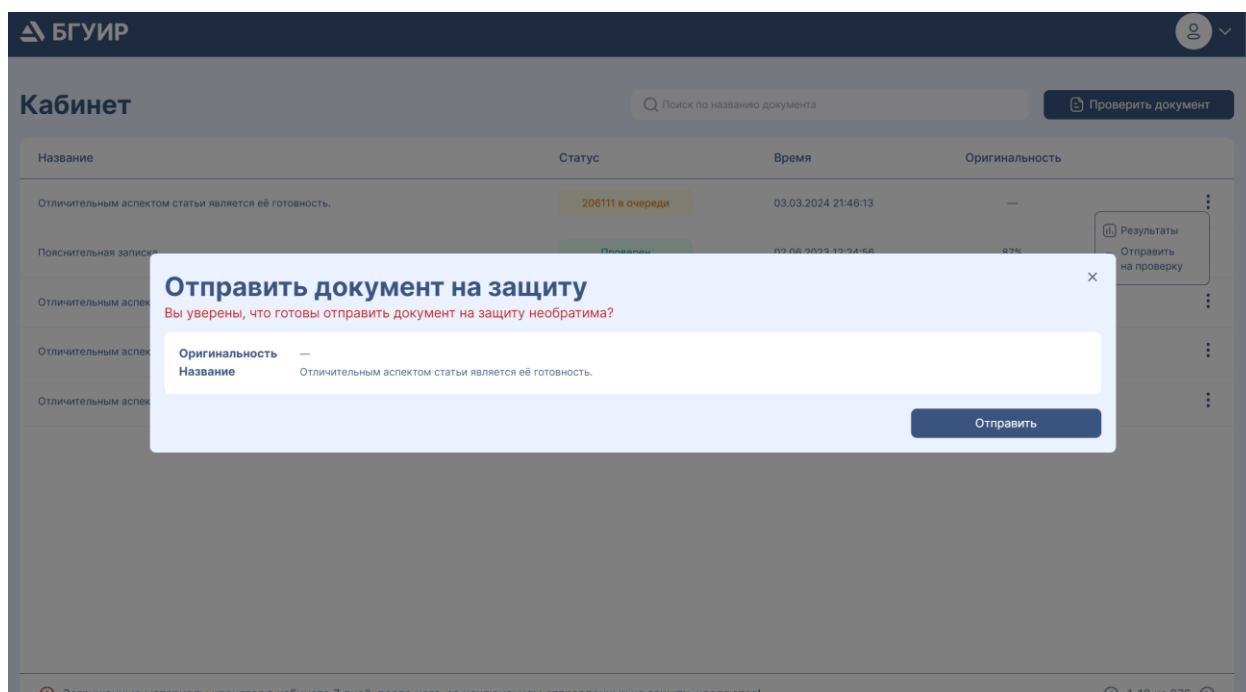


Рисунок 3.13 – Модальное окно при клике на «Отправить на проверку»

Для отправки нового документа на проверку в системе, пользователю достаточно выполнить простое действие: нажать на кнопку «Проверить документ», которая расположена рядом с полем поиска. После этого автоматически открывается модальное окно, предлагающее пользователю загрузить новый документ для последующей проверки. Пример модального окна для загрузки нового документа представлен на рисунке 3.14.

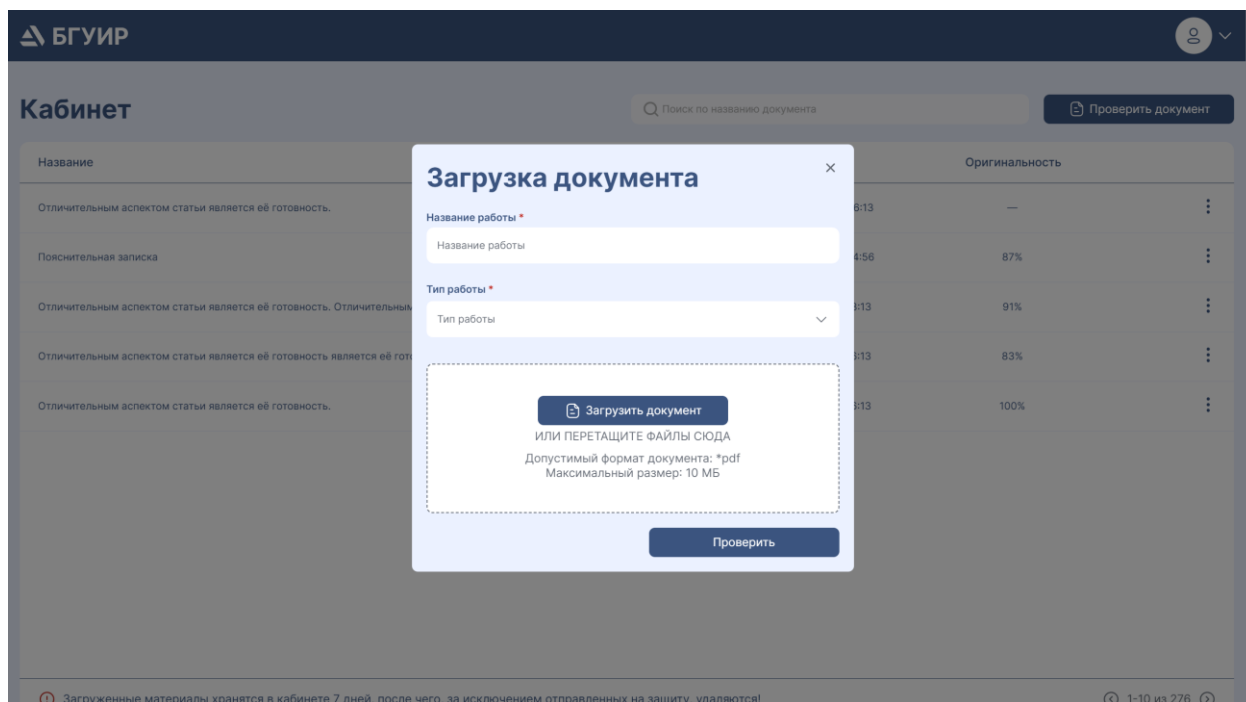


Рисунок 3.14 – Модальное окно для загрузки нового документа

После нажатия кнопки «Проверить документ» и открытия модального окна, пользователю предоставляется возможность выбрать тип работы из выпадающего списка и указать название работы. Затем пользователь может загрузить документ, воспользовавшись кнопкой «Загрузить документ» или просто перетащив выбранный файл в область загрузки документа. Этот метод позволяет пользователям выбирать наиболее удобный способ загрузки и обеспечивает гибкость в использовании сервиса. Если выбранный документ оказывается не нужным, пользователь может удалить его из формы, нажав на кнопку с крестиком рядом с ним. Это предотвращает возможные ошибки и обеспечивает пользователю контроль над загружаемыми файлами. После заполнения всех необходимых полей пользователь может нажать кнопку «Проверить», чтобы отправить документ на проверку. При этом модальное окно закроется автоматически. Кроме того, для закрытия модального окна пользователь может также нажать на кнопку с крестиком или на область вне модального окна. Это обеспечивает дополнительный способ управления процессом и повышает удобство использования, предоставляя пользователям свободу действий без лишних препятствий.

Пример заполненной формы представлен на рисунке 3.15.

The screenshot displays a web application interface for document management. A modal window titled "Загрузка документа" is open, allowing a user to upload a new document. The form includes a text input for the document name (filled with "123"), a dropdown menu for the document type (set to "Item-1"), and a file selection area showing a selected PDF file named "Пояснительная записка.pdf". A "Проверить" (Check) button is located at the bottom of the modal. In the background, the "Кабинет" (Cabinet) interface is visible, featuring a search bar, a table of documents with columns for name, originality, and status, and a "Проверить документ" button. The table lists several documents, each with a unique ID, a name, and a percentage indicating originality (e.g., 87%, 91%, 83%, 100%).

Рисунок 3.15 – Заполненная форма загрузки документа

После перехода на страницу «Результаты» пользователь обнаружит удобное поле поиска по ФИО, что значительно облегчает поиск нужной информации среди имеющихся записей. Это поле поиска позволяет быстро фильтровать результаты и находить конкретные документы по заданному имени.

Дополнительно, внизу страницы представлена пагинация, которая позволяет переходить между страницами с информацией о результатах документов. Особенно полезно в случае, если на экране отображается большое количество записей, и пользователю требуется легко навигировать по множеству данных. Пагинация способствует упорядочиванию вывода информации и обеспечивает удобство использования интерфейса.

Основной контент страницы состоит из вывода информации о результатах документов. Каждая запись включает в себя название документа, его текущий статус, дату загрузки на проверку, информацию об оригинальности и соответствующую кнопку действия. Статус документа может варьироваться от «зачтен» до «отправлен на доработку», что помогает пользователям отслеживать текущее состояние своих работ и оперативно реагировать на изменения.

Дата, отображаемая рядом со статусом, соответствует дате загрузки документа на проверку. Это позволяет пользователям ориентироваться в хронологии загрузки и обработки документов, что особенно важно при работе с большим объемом информации.

Пример страницы «Результаты» показан на рисунке 3.16.

Результаты						Поиск по ФИО	
№	ФИО	Дата	Оригинальность	Название	Статус		
1	Ташлыкова-Бушкевич Полина Александровна	03.03.2024 21:46:13	100%	Отличительным аспектом статьи является её готовность. Если...	—	⋮	
2	Разумов Дмитрий Александрович	02.06.2024 05:46:13	100%	Пояснительная записка курсовой работы	—	⋮	
3	Разумов Дмитрий Александрович	02.06.2024 05:46:13	100%	Пояснительная записка курсовой работы	—	⋮	
4	Ташлыкова-Бушкевич Полина Александровна	03.03.2024 21:46:13	100%	Отличительным аспектом статьи является её готовность.	—	⋮	
5	Разумов Дмитрий Александрович	02.06.2024 05:46:13	100%	Пояснительная записка курсовой работы	—	⋮	
6	Ташлыкова-Бушкевич Полина Александровна	03.03.2024 21:46:13	100%	Отличительным аспектом статьи	—	⋮	
7	Разумов Дмитрий Александрович	02.06.2024 05:46:13	100%	Пояснительная записка курсовой работы	—	⋮	
8	Разумов Дмитрий Александрович	02.06.2024 05:46:13	100%	Пояснительная записка курсовой работы	—	⋮	
9	Ташлыкова-Бушкевич Полина Александровна	03.03.2024 21:46:13	100%	Отличительным аспектом статьи является её готовность. Если...	—	⋮	
10	Разумов Дмитрий Александрович	02.06.2024 05:46:13	100%	Пояснительная записка курсовой работы	—	⋮	
						1-10 из 276	

Рисунок 3.16 – Пример страницы «Результаты»

При нажатии на иконку троеточия, которая представлена у каждого документа, пользователю открывается выпадающее меню с несколькими опциями. Одной из этих опций является «Результаты», при выборе которой происходит скачивание результатов документа в формате PDF. Дополнительно, в выпадающем меню присутствует кнопка «Оценить». Нажатие открывает модальное окно, где пользователь может оставить оценку данной работы.

Пример клика на иконку троеточия показан на рисунке 3.17.

Результаты						Поиск по ФИО	
№	ФИО	Дата	Оригинальность	Название	Статус		
1	Ташлыкова-Бушкевич Полина Александровна	03.03.2024 21:46:13	100%	Отличительным аспектом статьи является её готовность. Если...	—	⋮	
2	Разумов Дмитрий Александрович	02.06.2024 05:46:13	100%	Пояснительная записка курсовой работы	—	⋮	
3	Разумов Дмитрий Александрович	02.06.2024 05:46:13	100%	Пояснительная записка курсовой работы	—	⋮	
4	Ташлыкова-Бушкевич Полина Александровна	03.03.2024 21:46:13	100%	Отличительным аспектом статьи является её готовность.	—	⋮	
5	Разумов Дмитрий Александрович	02.06.2024 05:46:13	100%	Пояснительная записка курсовой работы	—	⋮	
6	Ташлыкова-Бушкевич Полина Александровна	03.03.2024 21:46:13	100%	Отличительным аспектом статьи	—	⋮	

Рисунок 3.17 – Пример клика на иконку троеточия

При нажатии на кнопку «Оценить» пользователю открывается модальное окно, предназначенное для оценки документа. В модальном окне пользователю предоставляется подтверждение для внесения окончательного решения о данном документе. Также в модальном окне отображается информация о выбранном документе, включая ФИО автора, название документа и оригинальность. Данный функционал помогает пользователям в принятии решений по оценке документов и обеспечивает дополнительную информацию для принятия обоснованных решений. Предоставление подтверждения перед окончательной оценкой помогает избежать случайных или нежелательных действий, что способствует точности и надежности оценки работ.

Пример вывода модального окна показан на рисунке 3.18.

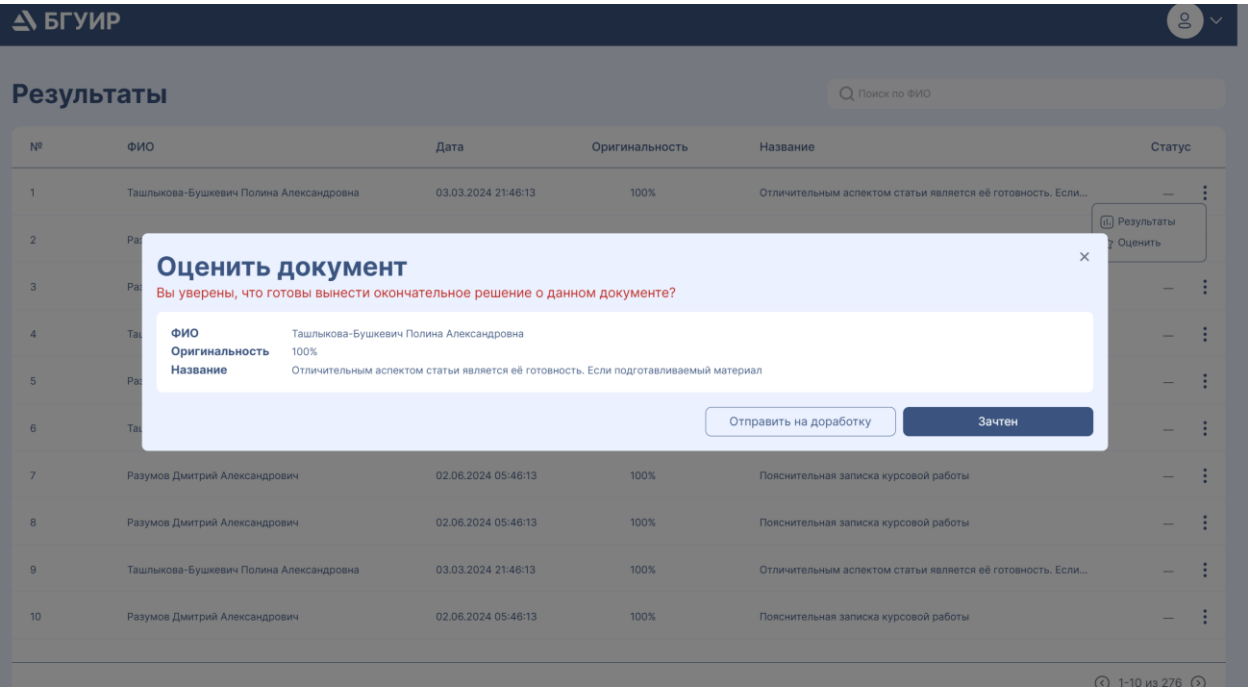


Рисунок 3.18 – Модальное окно «Оценки документа»

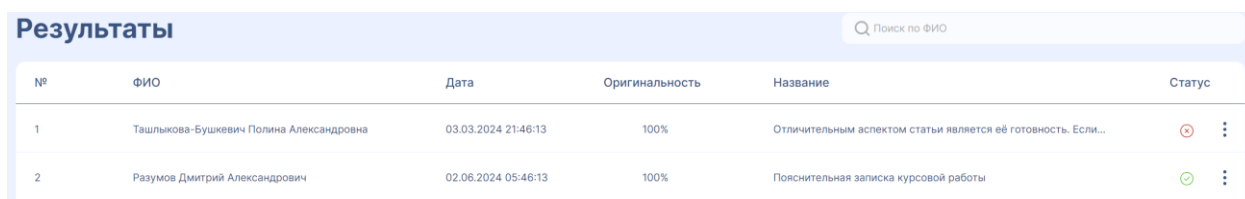
Также есть две кнопки «Отправить на доработку» и «Зачтен», при клике на кнопку «Отправить на доработку» документ отправляется на доработку и статус документа меняется на отрицательный. Статус документа после клика на кнопку «Отправить на доработку» показан на рисунке 3.19.

№	ФИО	Дата	Оригинальность	Название	Статус
1	Ташлыкова-Бушкевич Полина Александровна	03.03.2024 21:46:13	100%	Отличительным аспектом статьи является её готовность. Если...	❌
2	Разумов Дмитрий Александрович	02.06.2024 05:46:13	100%	Пояснительная записка курсовой работы	—

Рисунок 3.19 – Статус при клике на кнопку «Отправить на доработку»



При клике на кнопку «Зачтен» статус документа меняется на положительный. Статус документа после клика на кнопку «Зачтен» показан на рисунке 3.20.



№	ФИО	Дата	Оригинальность	Название	Статус
1	Ташлыкова-Бушкевич Полина Александровна	03.03.2024 21:46:13	100%	Отличительным аспектом статьи является её готовность. Если...	Зачтен
2	Разумов Дмитрий Александрович	02.06.2024 05:46:13	100%	Пояснительная записка курсовой работы	Зачтен

Рисунок 3.20 – Статус при клике на кнопку «Зачтен»

После клика на кнопку «Зачтен» или же на кнопку «Отправить на доработку» модальное окно закрывается, также для того, чтобы закрыть модальное окно достаточно нажать на кнопку крестика или же на область вне модального окна.

Также для мобильной версии предусмотрена кнопка скроллинга наверх страницы, для быстрого перехода в вверх страницы и любого положения. Пример кнопки скроллинга наверх показан на рисунке 3.21.

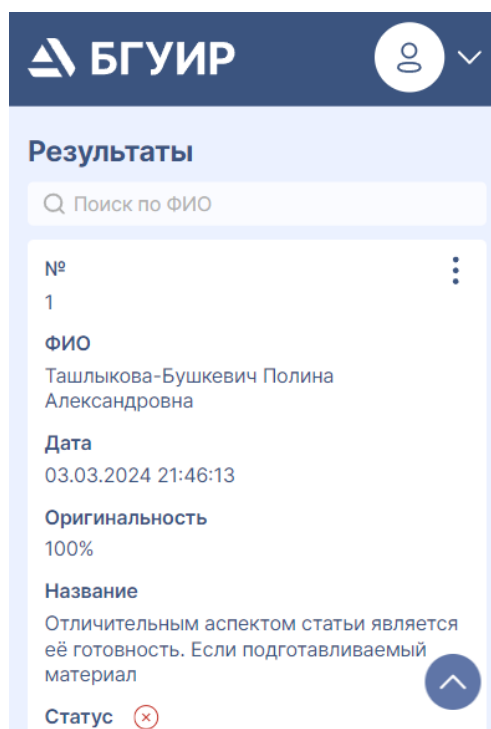


Рисунок 3.21 – Пример кнопки скроллинга

Таким образом, в данном разделе было представлено руководство по вводу в эксплуатацию и эксплуатации разработанного программного продукта.

## ЗАКЛЮЧЕНИЕ

В рамках данного проекта был разработан веб-сервис, предназначенный для автоматизации процесса проверки оригинальности различных научных и учебных работ студентов и преподавателей на основе репозитория БГУИР. Основной целью проекта было обеспечение возможности автоматизации проверки оригинальности работ с целью повышения эффективности и удобства процесса для пользователей.

В процессе выполнения проекта были использованы принципы системного подхода, что позволило эффективно анализировать требования пользователей и реализовывать необходимый функционал. Был проведен анализ требований пользователей к веб-сервису, разработано общетехническое обоснование проекта, а также реализован механизм проверки оригинальности текстов. Особое внимание было уделено обработке ошибок и обеспечению корректной работы сервиса.

Результатом работы стал разработанный веб-сервис, который позволяет загружать работы для проверки и получать детальные отчеты о проценте уникальности текста и о возможных источниках совпадений в репозитории БГУИР. Программное средство может быть использовано для создания аналогичных сервисов в образовательных учреждениях, что расширяет область применения.

В процессе работы над проектом были учтены государственные стандарты и нормативные документы, а также использованы рекомендации СТП БГУИР 01–2017. Вся используемая информация была четко указана в «Списке использованных источников».

Важным результатом проекта является создание удобного и эффективного инструмента для проверки оригинальности работ, что способствует повышению качества образования и научной работы. Однако, в дальнейшем планируется расширение функционала веб-сервиса. Планируется разработка дополнительных модулей для анализа содержания работ с использованием методов машинного обучения. Это поможет выявлять не только формальные признаки плагиата, но и структурные и семантические сходства, что повысит качество проверки оригинальности. Важным аспектом будет также улучшение интерфейса и пользовательского опыта. Планируется доработка дизайна и функциональности пользовательского интерфейса, с учетом обратной связи от пользователей и требований к удобству использования. Помимо этого, будет проведено тестирование и оптимизация производительности сервиса, чтобы обеспечить его стабильную и эффективную работу при большой нагрузке и объеме данных.

Таким образом, в будущем планируется значительное расширение функциональности и возможностей веб-сервиса для проверки оригинальности работ.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] БГУИР ГОСТ СТП 01–2017 [Электронный ресурс]. – Режим доступа: [https://www.bsuir.by/m/12100229\\_1\\_122976.pdf](https://www.bsuir.by/m/12100229_1_122976.pdf) – Дата доступа: 10.05.2024.
- [2] Антиплагиат [Электронный ресурс]. – Режим доступа: <https://antiplagiat.ru/> – Дата доступа: 12.04.2024.
- [3] PlagScan [Электронный ресурс]. – Режим доступа: <https://www.turnitin.com/> – Дата доступа: 12.04.2024.
- [4] Text.ru [Электронный ресурс]. – Режим доступа: <https://text.ru/doccheck> – Дата доступа: 12.04.2024.
- [5] Antiplagius [Электронный ресурс]. – Режим доступа: <https://antiplagiatus.ru/> – Дата доступа: 12.04.2024.
- [6] BE1.ru [Электронный ресурс]. – Режим доступа: <https://be1.ru/antiplagiat-online/> – Дата доступа: 12.04.2024.
- [7] MDN Web Docs [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/ru/docs/Web/HTML> – Дата доступа: 10.03.2024.
- [8] W3Schools [Электронный ресурс]. – Режим доступа: <https://www.w3schools.com/html/> – Дата доступа: 05.03.2024.
- [9] Основы HTML – Изучение веб-разработки [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/ru/docs/Web/HTML> – Дата доступа: 08.03.2024.
- [10] Руководство по CSS: каскадные таблицы [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/ru/docs/Web/CSS/Reference> – Дата доступа: 10.03.2024.
- [11] Основы SCSS [Электронный ресурс]. – Режим доступа: <https://sass-scss.ru/guide/> – Дата доступа: 12.03.2024.
- [12] Современный учебник JavaScript [Электронный ресурс]. – Режим доступа: <https://learn.javascript.ru/> – Дата доступа: 14.03.2024.
- [13] UI Дока JavaScript [Электронный ресурс]. – Режим доступа: <https://doka.guide/js/> – Дата доступа: 14.03.2024.
- [14] Интегрированная среда разработки [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Интегрированная\\_среда\\_разработки](https://ru.wikipedia.org/wiki/Интегрированная_среда_разработки) – Дата доступа: 06.03.2024.
- [15] VSCode [Электронный ресурс]. – Режим доступа: <https://code.visualstudio.com/> – Дата доступа: 06.03.2024.
- [16] VS Code настройка установка плагины [Электронный ресурс]. – Режим доступа: <https://www.youtube.com/watch?v=nxCLXMBI4e4> – Дата доступа: 06.03.2024.
- [17] MySQL [Электронный ресурс]. – Режим доступа: <https://www.mysql.com/> – Дата доступа: 05.04.2024.
- [18] Git, GitHub для новичков [Электронный ресурс]. – Режим доступа: [https://www.youtube.com/watch?v=zZBiln\\_2FhM](https://www.youtube.com/watch?v=zZBiln_2FhM) – Дата доступа: 17.03.2024.

- [19] GIT [Электронный ресурс]. – Режим доступа: <https://git-scm.com/> – Дата доступа: 20.04.2024.
- [20] GitHub [Электронный ресурс]. – Режим доступа: <https://blog.hubspot.com/website/what-is-github-used-for> – Дата доступа: 12.04.2024.
- [21] Gulp установка настройка плагина [Электронный ресурс]. – Режим доступа: <https://www.youtube.com/watch?v=jU88mLuLWlk&t=2055s> – Дата доступа: 12.03.2024.
- [22] Гид по Figma для разработчиков [Электронный ресурс]. – Режим доступа: <https://www.figma.com/community/file/886207890389884742> – Дата доступа: 12.03.2024.
- [23] Алексеев, В. Ф. Структуры и базы данных. Пособие для курсового проектирования: пособие / В. Ф. Алексеев, И. Н. Богатко, Г. А. Пискун. – Минск: БГУИР, 2017. – 84 с.: ил.
- [24] PostgreSQL: Документация – Режим доступа: <https://postgrespro.ru/docs/postgresql> – Дата доступа: 04.04.2024.
- [25] Принципы UI/UX. – Режим доступа: [https://habr.com/ru/companies/SECL\\_GROUP/articles/182208/](https://habr.com/ru/companies/SECL_GROUP/articles/182208/) – Дата доступа: 10.04.2024.
- [26] Зачем нужно тестирование? Когда оно начинается и когда заканчивается, какие задачи стоят перед тестированием программного обеспечения? [Электронный ресурс]. – Режим доступа: <https://dzen.ru/a/ZGEgrWptlGbZJKJi> – Дата доступа: 05.01.2024.
- [27] Что такое хостинг [Электронный ресурс]. – Режим доступа: <https://hostiq.ua/info/what-is-hosting/> – Дата доступа: 04.05.2024.
- [28] Хостинг провайдер Hostinger [Электронный ресурс]. – Режим доступа <https://ishosting.com/ru/> – Дата доступа: 04.05.2024.
- [29] Хостинг провайдер SiteGround [Электронный ресурс]. – Режим доступа: <https://eu.siteground.com/> – Дата доступа: 04.05.2024.
- [30] Хостинг провайдер Bluehost [Электронный ресурс]. – Режим доступа <https://www.bluehost.com/> – Дата доступа: 04.05.2024.
- [31] Руководство по панели управления Beget [Электронный ресурс]. – Режим доступа: <https://beget.com/ru/kb/manual> – Дата доступа: 04.05.2024.
- [32] Выгрузка проекта на хостинг [Электронный ресурс]. – Режим доступа: <https://www.youtube.com/watch?v=UPnVcTZUwXE&t=854s> – Дата доступа: 04.05.2024.
- [33] Разработанный веб-сервис [Электронный ресурс]. Режим доступа: <http://dimaracb.beget.tech/> – Дата доступа: 07.05.2024.

# ПРИЛОЖЕНИЕ А

## (Обязательное)

### Отчёт о проверке на заимствования в системе «Антиплагиат»

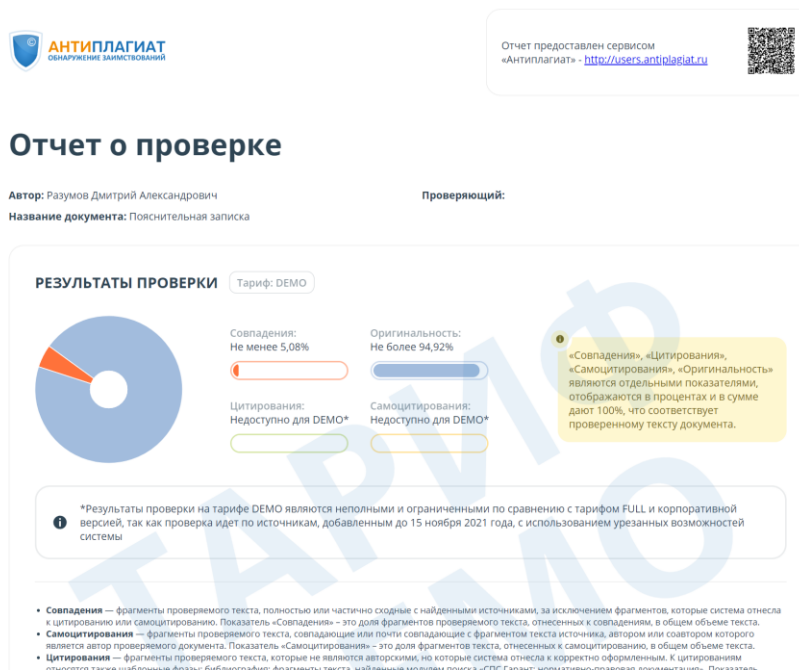


Рисунок А.1 – Отчет о проверке на заимствования в система «Антиплагиат»

## ПРИЛОЖЕНИЕ Б

(обязательное)

### Листинги программного кода

#### Файл main.html

```
<main class="auth-page">
  <section class="auth">
    <div class="container">
      <div class="auth__container">
        <h1 class="auth__title title">Личный кабинет</h1>
        <form class="auth__form" id="auth-form">
          <div class="auth__form-input input">
            <label class="input__label"
for="login">Логин</label>
            <input autocomplete="off" class="input__input"
type="text" placeholder="Введите логин" name="login" />
            <div class="input__btn-notify js-notify" data-
notify="Менее 8 символов">
              <svg>
                <use xlink:href="assets/img/sprite.svg#input-
error"></use>
              </svg>
            </div>
            <div class="input__error">Некорректный формат
логина!</div>
          </div>
          <div class="auth__form-input input input--pass">
            <label class="input__label"
for="login">Пароль</label>
            <input
              autocomplete="off"
              class="input__input"
              type="password"
              placeholder="Введите пароль"
              name="password"
            />
            <button type="button" class="input__btn">
              <svg>
                <use xlink:href="assets/img/sprite.svg#pass-
open"></use>
              </svg>
              <svg>
                <use xlink:href="assets/img/sprite.svg#pass-
close"></use>
              </svg>
            </button>
            <div class="input__btn-notify js-notify" data-
notify="Менее 8 символов">
              <svg>
```

```

        <use xlink:href="assets/img/sprite.svg#input-
error"></use>
        </svg>
    </div>
    <div class="input__error">Некорректный формат
пароля!</div>
    </div>
    <button class="auth__form-btn btn btn-def" form="auth-
form" type="submit">Войти</button>
    </form>
    <div class="auth__content">
        <div class="auth__content-block">
            <h2 class="auth__content-title">Вы в первый раз на
нашем сайте?</h2>
            <p>Учетные данные для авторизации аналогичны тем,
которые используются для доступа к ИИС БГУИР.</p>
            <p>
                Логин для студентов - номер студенческого билета,
для преподавателей - имя почтового ящика БГУИР без
                @bsuir.by.
            </p>
        </div>
        <div class="auth__content-block">
            <h2 class="auth__content-title">
                Что делать, если логин/пароль <br />
                не работает?
            </h2>
            <p>
                Если самостоятельно сбросить пароль не уда-
лось, воспользуйтесь пунктом
                <a
href="https://courses.bsuir.by/customforms/form1.html">«Восстано
вление пароля»</a> в
                <a
href="https://courses.bsuir.by/customforms/form1.html">форме
связи с техподдержкой.</a>
            </p>
        </div>
    </div>
</div>
</section>
@include('../templates/btnUp.html')

```

## Файл pers-cab.html

```

<main>
    <section class="cab">
        <div class="container">

```

```

        <div class="cab__container">
            <div class="cab__titles">
                <h1 class="cab__titles-title title">Личный
кабинет</h1>
            </div>
            <div class="cab__block cab-pers">
                <div class="cab-pers__content">
                    <div class="cab-pers__main">
                        <div class="cab-pers__icon">
                            <svg>
                                <use
xlink:href="assets/img/sprite.svg#profile"></use>
                            </svg>
                        </div>
                        <div class="cab-pers__name">
                            <p>Ташлыкова-Бушкевич</p>
                            <p>Полина</p>
                            <p>Александровна</p>
                        </div>
                        <div class="cab-pers__info">
                            <div class="cab-pers__info-block">
                                <b>Адресс электронной почты</b>
                                <a
href="mailto:ivanov@mail.ru">ivanov@mail.ru</a>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </section>
        @include('../templates/btnUp.html')
    </main>

```

## Файл header.html

```

<header class="header">
    <div class="container">
        <div class="header__container">
            <a href="./cab.html" class="header__logo logo">
                
            </a>
            <div class="header__menus js-menus">
                <div class="header__btn js-menu-btn">
                    <div class="header__btn-icon">
                        <svg>
                            <use
xlink:href="assets/img/sprite.svg#profile"></use>
                        </svg>
                    </div>
                </div>
            </div>
        </div>
    </div>

```



```

        <div class="header__btn-arrow">
            <svg>
                <use xlink:href="assets/img/sprite.svg#header-
btn"></use>
            </svg>
        </div>
    </div>
    <div class="header__menu js-menu">
        <a href="./cab.html" class="header__menu-item">
            <div class="header__menu-icon">
                <svg>
                    <use xlink:href="assets/img/sprite.svg#menu-
cab"></use>
                </svg>
                <svg>
                    <use xlink:href="assets/img/sprite.svg#menu-cab-
active"></use>
                </svg>
            </div>
            <span>Кабинет</span>
        </a>
        <a href="./pers-cab.html" class="header__menu-item">
            <div class="header__menu-icon">
                <svg>
                    <use xlink:href="assets/img/sprite.svg#menu-
user"></use>
                </svg>
                <svg>
                    <use xlink:href="assets/img/sprite.svg#menu-
user-active"></use>
                </svg>
            </div>
            <span>Личный кабинет</span>
        </a>
        <a href="./results-cab.html" class="header__menu-
item">
            <div class="header__menu-icon">
                <svg>
                    <use xlink:href="assets/img/sprite.svg#menu-
result"></use>
                </svg>
                <svg>
                    <use xlink:href="assets/img/sprite.svg#menu-
result-active"></use>
                </svg>
            </div>
            <span>Результаты</span>
        </a>
        <a href="./role-cab.html" class="header__menu-item">
            <div class="header__menu-icon">
                <svg>

```



```

&.--active {
  .header__btn-arrow {
    transform: rotate(180deg);

    svg {
      background-color: $white;
      fill: $blue400;
    }
  }
}

&__btn-icon {
  border-radius: 50%;
  @include sizes(50px, 50px);
  background-color: $white;
  @include flexCenter;

  svg {
    @include sizes(25px, 25px);
    stroke: $blue400;
    fill: none;
    stroke-width: 1.5;
  }
}

&__btn-arrow {
  @include transition(transform);
  @include sizes(25px, 25px);
}

&__btn-arrow svg {
  fill: $white;
  stroke: none;
  background: none;
  border-radius: 50%;
  transition:
    fill 0.2s ease,
    background 0.2s ease;
}

&__menu {
  border: 1px solid $blue400;
  padding: 15px 15px 15px 10px;
  border-radius: 8px;
  background-color: $blue50;
  @include flexColumn;
  row-gap: 8px;
  position: absolute;
  top: 42px;
  right: 0;
}

```

```

width: 183px;
@include hidden;
z-index: -2;

&.--active {
  @include visible;
  z-index: 2;
}

svg {
  fill: none;
  stroke: $blue400;
}
}

&__menu-item {
  display: flex;
  align-items: center;
  column-gap: 5px;

  span {
    border-bottom: 1px solid transparent;
    @include regular14;
    @include transition(border-color);
  }

  @include hover {
    svg {
      &:first-child {
        @include hidden;
      }
      &:last-child {
        @include visible;
      }
    }

    span {
      color: $blue400;
      border-color: $blue400;
    }
  }
}

&__menu-icon {
  position: relative;
  @include sizes(20px, 20px);

  svg {
    transition:
      opacity 0.2s ease,
      visibility 0.2s ease;
  }
}

```

```

        position: absolute;

        &:first-child {
            fill: none;
            stroke: $blue400;
            @include visible;
        }
    }
}
}

```

### **Файл main.scss**

```

.auth {
    margin: 0 auto;

    &__container {
        margin: 0 auto;
        background-color: $blue50;
        border-radius: 8px;
        padding: 30px 42px;
        width: 628px;
    }

    &__title {
        text-align: center;
        margin-bottom: 10px;
    }

    &__content {
        border-radius: 10px;
        background-color: $blue200;
        padding: 30px 42px 30px 30px;
    }

    &__content-title {
        @include semibold20;
        margin-bottom: 5px;
    }

    &__content-block {
        &:not(:last-child) {
            margin-bottom: 30px;
        }

        p {
            @include regular16;
            color: $blue300;

            &:not(:last-child) {

```

```

        margin-bottom: 5px;
    }
}

a {
    @include bold16;
    @include transition(color);
    @include hover {
        color: $blue400;
    }
}

}

&__form {
    margin-bottom: 30px;
    @include flexColumn;
    row-gap: 25px;
}

&__form-btn {
    @include medium16;
    @include sizes(100%, 48px);
    margin-top: 5px;
}

}

@media screen and (max-width: $mob) {
    .auth {
        &__container {
            padding: 20px 12px;
            width: 100%;
        }

        &__content {
            padding: 24px 16px;
        }

        &__form {
            row-gap: 16px;
            margin-bottom: 20px;
        }

        &__form-btn {
            height: 32px;
            margin-top: 7px;
        }

        &__content {
            padding: 12px;
        }
    }
}

```

```

    &__content-block {
      @include mb(12px);
    }

    &__content-title {
      text-wrap: balance;
    }
  }
}

```

## Файл tippy.js

```

document.addEventListener('DOMContentLoaded', () => {
  const files = document.querySelectorAll('.js-notify-tippy');

  if (files && window.innerWidth > 1100) {
    files.forEach((file) => {
      file.addEventListener('mouseover', (event) => {
        const text = event.currentTarget;
        const txt = text.textContent;
        const txtHeight = text.scrollHeight;
        const height = 24;

        if (txt.split('\n').length >= 2 && txtHeight > height) {
          console.log('заебок');
          tippy(file, {
            content: txt,
            placement: 'bottom',
            arrow: true,
            theme: 'light',
            maxWidth: '400px',
            followCursor: true,
          });
        }
      });
    });
  }
});

if (document.querySelector('.js-notify')) {
  tippy('.js-notify', {
    content(reference) {
      return reference.getAttribute('data-notify');
    },
    placement: 'bottom',
    animation: 'fade',
    maxWidth: '200px',
  });
}

```

## Файл more.js

```
const mores = document.querySelectorAll('.more-btns');

function handleMore() {
  if (mores) {
    mores.forEach((more) => {
      const moreBtn = more.querySelector('.more-btns__btn');
      const moreContent = more.querySelector('.more-
btns__content');

      moreBtn.addEventListener('click', () => {
        moreContent.classList.toggle('--active');
      });
    });
  }
}

document.addEventListener('click', (event) => {
  if (mores) {
    mores.forEach((more) => {
      const moreContent = more.querySelector('.more-
btns__content');
      const isClickInsideMore = more.contains(event.target);

      if (!isClickInsideMore) {
        moreContent.classList.remove('--active');
      }
    });
  }
});

document.addEventListener('DOMContentLoaded', () => {
  handleMore();
});
```

## Файл drag-area.js

```
/* eslint-disable func-names */
const dragArea = document.querySelector('.drag-area');
const button = dragArea.querySelector('button');
const input = dragArea.querySelector('input');
const fileDisplay = document.querySelector('.drag-area__files');
const dragAreaForm = document.querySelector('.drag-area__form');

function clickBtn() {
  button.addEventListener('click', () => input.click());
}

function handleDragOver() {
```



```

    dragArea.addEventListener('dragover', function (event) {
        event.preventDefault();
        this.classList.add('--active');
    });
}

function handleDragLeave() {
    dragArea.addEventListener('dragleave', function () {
        this.classList.remove('--active');
    });
}

function displayFile(file) {
    const fileReader = new FileReader();
    fileReader.onload = function () {
        const fileElement = document.createElement('div');
        fileElement.setAttribute('class', 'drag-area__file');

        const fileName = document.createElement('div');
        fileName.setAttribute('class', 'drag-area__filename');
        fileName.textContent = file.name;

        const deleteButton = document.createElement('button');
        deleteButton.setAttribute('class', 'drag-area__fileclose');
        deleteButton.innerHTML = `
            <svg>
                <use xlink:href="assets/img/sprite.svg#close-
file"></use>
            </svg>
        `;

        deleteButton.addEventListener('click', () => {
            fileDisplay.removeChild(fileElement);
            dragArea.classList.remove('--error');
            dragArea.classList.remove('--files');
            input.value = '';
        });

        fileElement.appendChild(fileName);
        fileElement.appendChild(deleteButton);
        fileDisplay.appendChild(fileElement);
    };
    fileReader.readAsDataURL(file);
}

const isValid = (file) => file.type === 'application/pdf' &&
file.size <= 10485760;

function isValidatation(file) {
    if (isValid(file)) {
        displayFile(file);
    }
}

```

```

        dragArea.classList.add('--files');
    } else {
        dragAreaForm.classList.add('--error');
    }
}

function handleDrop() {
    dragArea.addEventListener('drop', (event) => {
        event.preventDefault();
        dragAreaForm.classList.remove('--active');

        const file = event.dataTransfer.files[0];
        isValidatation(file);
    });
}

function handleChangeInput() {
    input.addEventListener('change', (event) => {
        const file = event.target.files[0];
        isValidatation(file);
    });
}

document.addEventListener('DOMContentLoaded', () => {
    clickBtn();
    handleDragOver();
    handleDragLeave();
    handleDrop();
    handleChangeInput();
});

```

### **Файл dropdown.js**

```

document.addEventListener('DOMContentLoaded', () => {
    const dropdowns = document.querySelectorAll('.js-dropdown');

    if (dropdowns) {
        dropdowns.forEach((dropdown) => {
            const inner = dropdown.querySelector('.js-dropdown-
inner');
            const title = dropdown.querySelector('.js-dropdown-
value');
            const labels = dropdown.querySelectorAll('.js-dropdown-
item label');

            inner.addEventListener('click', () => {
                if (dropdown.classList.contains('--active')) {
                    dropdown.classList.remove('--active');
                } else {
                    dropdown.classList.add('--active');
                }
            });
        });
    }
});

```

```

    }
  });

  for (let index = 0; index < labels.length; index += 1) {
    labels[index].addEventListener('click', (e) => {
      if (title) {
        title.textContent = e.target.textContent;
        dropdown.classList.add('--filled');
      }
    });
  }

  document.addEventListener('click', (event) => {
    if (!event.target.closest('.js-dropdown-inner')) {
      dropdown.classList.remove('--active');
    }
  });
});
}
});

```

### Файл inputs.js

```

const changeVisiblePassword = () => {
  const passInputs = document.querySelectorAll('.input--pass');

  if (passInputs) {
    passInputs.forEach((passInput) => {
      const input = passInput.querySelector('input');
      const btnPass = passInput.querySelector('.input__btn');

      btnPass.addEventListener('mousedown', (event) => {
        event.preventDefault();
        btnPass.classList.toggle('--open');
        if (input.getAttribute('type') === 'text') {
          input.setAttribute('type', 'password');
        } else {
          input.setAttribute('type', 'text');
        }
        input.focus();
      });
    });
  }
};

const changeInputWithError = () => {
  const inputErrors = document.querySelectorAll('.input.--error');

  if (inputErrors) {

```

```

        inputErrors.forEach((inputError) => {
            inputError.addEventListener('input', function () {
                this.classList.remove('--error');
            });
        });
    }
};

const changeVisibleResetSearch = () => {
    const search = document.querySelector('.search');

    if (search) {
        const btnRes = search.querySelector('.search__btn-reset');
        const input = search.querySelector('input');

        input.addEventListener('input', () => {
            if (input.value) {
                btnRes.classList.add('--active');
            } else {
                btnRes.classList.remove('--active');
            }
        });

        btnRes.addEventListener('click', () => {
            btnRes.classList.remove('--active');
            input.focus();
        });
    }
};

document.addEventListener('DOMContentLoaded', () => {
    changeVisiblePassword();
    changeInputWithError();
    changeVisibleResetSearch();
});

```

### Файл btn-up.js

```

document.addEventListener('DOMContentLoaded', () => {
    const btnUp = document.querySelector('.js-btn-up');

    btnUp.addEventListener('click', () => {
        window.scrollTo({
            top: 0,
            behavior: 'smooth',
        });
    });
});

```

**ПРИЛОЖЕНИЕ В**  
**(Обязательное)**  
**Ведомость курсового проекта**