МИНИСТЕРСТВО ОБРАЗОВАНИЯ ОРЕНБУРГСКОЙ ОБЛАСТИ

**ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«ОРЕНБУРГСКИЙ КОЛЛЕДЖ ЭКОНОМИКИ И ИНФОРМАТИКИ»**

**(ГАПОУ «ОКЭИ»)**

**ОТЧЁТ**

ОКЭИ 09.02.07.9 0 25. 34 П

(код документа)

*09.02.07 «Информационные системы и программирование»*

По профилю специальности\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Квалификация: «Разработчик веб и мультимедийных приложений»*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*ПМ.09 Проектирование, разработка и оптимизация веб-приложений*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*МОАУ «СОШ №71»*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(место прохождения практики)

*55*

Количество листов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*17.05.2025*

Дата готовности \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Морозова Маргарита Вячеславовна*

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*студент группы 4вб2, Васильева Валерия Павловна*

Разработал \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Дорохова Елена Ильинична, учитель информатики*

Согласовано \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Соответствие отчета с заданием на практику \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Защищен\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Содержание**

[Введение 3](#_Toc11184)

[1 Анализ предприятия 6](#_Toc18776)

[1.1 Характеристика предприятия 6](#_Toc693)

[1.2 Анализ предметной области 8](#_Toc13846)

[2 Техническое задание 10](#_Toc31292)

[2.1 Требования к серверной части 10](#_Toc23772)

[2.2 Требование к разработке 11](#_Toc2547)

[2.3 Тестирование и документация 15](#_Toc15022)

[3 Разработка проектной документации 17](#_Toc11143)

[3.1 Анализ информационных потоков веб-приложения 17](#_Toc30831)

[3.2 Разработка документации функционального назначения 19](#_Toc8904)

[3.3 Разработка макетов и эскизов веб-приложения 22](#_Toc25147)

[3.4 Подбор графических материалов 31](#_Toc21338)

[4 Разработка веб-приложения 36](#_Toc28875)

[4.1. Создание шаблонов страниц веб-приложения 36](#_Toc2173)

[4.2 Реализация веб-приложения 40](#_Toc27141)

[4.3 Реализация динамических элементов сайта 45](#_Toc8583)

[5 Отладка и тестирование веб-приложения 48](#_Toc9442)

[5.1 Ревьюирование программного кода 48](#_Toc8857)

[5.2 Тестирование веб-приложения 48](#_Toc16844)

[5.3 Публикация веб-приложения в сети 57](#_Toc19485)

[6 Размещение и техническое сопровождение веб-приложения 59](#_Toc8862)

[6.1 Требования к аппаратным ресурсам 59](#_Toc30512)

[6.2 Руководство администратора 60](#_Toc2024)

[6.3 Руководство пользователя 62](#_Toc4008)

[7 Расчет экономической эффективности 64](#_Toc20662)

[Заключение 69](#_Toc25142)

[Список использованных источников 71](#_Toc17303)

[Приложение А(обязательное) 73](#_Toc9988)

[Приложение Б(обязательное) 74](#_Toc19074)

[Приложение В(обязательное) 75](#_Toc2409)

[Приложение Г(обязательное) 76](#_Toc22656)

[Приложение Д(обязательное) 77](#_Toc28678)

**Введение**

С каждым днем бурное развитие информатизации общества приводит к появлению новых технологий, делая быстрый доступ к информации неотъемлемой частью нашей жизни. Веб-сайты стали одним из наиболее удобных способов получения информации, и создание сайтов сегодня является одной из самых востребованных услуг. В современном мире веб-приложения становятся незаменимыми инструментами для организаций, предоставляя возможность оптимизировать процессы, расширить доступ к информации и повысить эффективность работы. Это программные решения, доступные через интернет-браузеры, которые позволяют выполнять различные функции, такие как обработка данных, управление информацией и взаимодействие с пользователями.

Создание информационных систем (ИС) в современном мире становится неотъемлемой частью развития любых сфер деятельности – от бизнеса и производства до сферы образования и здравоохранения. Особенно актуальными являются веб-приложения, которые позволяют автоматизировать внутренние процессы, повысить эффективность взаимодействия пользователей и обеспечить доступность данных в режиме реального времени. В условиях быстрого роста цифровых технологий и увеличения требований к удобству и скорости обслуживания, веб-приложения приобретают особую значимость, так как они обеспечивают централизованный доступ к функциям системы через интернет-браузер и способствуют оптимизации деятельности организаций.

Адаптивность веб-приложений становится важным аспектом. Высокая конкуренция требует от разработчиков создания решений, которые могут переключаться между различными устройствами и платформами, что обеспечивает пользователям возможность доступа к информации в любом месте и в любое время. Интерфейсы должны быть интуитивно понятными и отзывчивыми, что повышает общую удовлетворенность пользователей и способствует удержанию клиентов.

В основе разработки сайта лежит создание веб-приложения с использованием современных технологий «front-end» и «back-end» разработки. Для реализации проекта планируется использование различных инструментов и технологий, которые обеспечат его функциональность и производительность.

На этапе «front-end» будут задействованы «HTML», «CSS» и «JavaScript», а также библиотеки и фреймворки, что позволит обеспечить адаптивный дизайн и поддерживать отображение сайта на различных устройствах, включая мобильные телефоны и планшеты.

В «back-end» разрабатываемого приложения будут использованы языки программирования, такие как «JavaScript», что обеспечит гибкость и возможность создания сложного функционала. Для хранения и обработки данных планируется использование баз данных, таких как «SQlite», что позволит эффективно управлять информацией, необходимой для работы веб-приложения. Серверные технологии, помогут создать надежный серверный участок приложения, обеспечивая быстрое и эффективное взаимодействие между клиентом и сервером.

Для образовательных организаций веб-приложения могут стать незаменимым инструментом, способствуя повышению эффективности обучения, оптимизации административных процессов и улучшению взаимодействия с родителями и учениками. Например, веб-приложение может использоваться для создания электронных дневников, организации онлайн-тестирования, управления учебным процессом, а также для проведения онлайн-консультаций и создания виртуальных учебных площадок [1].

Кроме того, работа будет охватывать тестирование различных аспектов функциональности, совместимости, безопасности и производительности сайта.

Данная работа посвящена разработке веб-приложения для образовательной платформы «УчительPRO». Образовательная сфера постоянно требует современных инструментов, которые позволяют автоматизировать процессы обмена ресурсами, взаимодействия преподавателей и учеников, а также управления учебным контентом. На текущий момент многие образовательные организации используют устаревшие средства, такие как электронные таблицы или локальные базы данных, что не обеспечивает необходимого уровня автоматизации, безопасности и доступности данных.

Разработка веб-приложения «УчительPRO» является актуальной задачей, направленной на создание онлайн-ресурса, призванного оказывать поддержку и помощь учителям в их профессиональной деятельности, а также возможность для самообразования и расширения уровня знаний.

Цель данной работы – разработать веб-приложение, которое повысит эффективность обмена образовательными ресурсами, упростит управление информацией и обеспечит безопасный доступ к материалам для учителей и учеников.

Объектом является организация Муниципальное общеобразовательное автономное учреждение средняя общеобразовательная школа № 71 (МОАУ «СОШ №71»), профилем деятельности которой является предоставление качественного образовательного процесса, направленного на всестороннее развитие учащихся, а также публикация и обмен образовательным контентом, созданным как педагогами, так и самими учениками, с целью улучшения учебного процесса и повышения уровня подготовки.

Предметом является веб-сайт для обмена, разработанных пользователями, образовательными ресурсами.

Для достижения поставленной цели были определены следующие задачи:

* произвести анализ предметной области и требований к системе;
* обосновать необходимость разработки или модернизации нового веб-приложения;
* разработать техническое задание, регламентирующее требования;
* разработать проектную документацию;
* реализовать веб-приложение в соответствии с техническим заданием;
* разработать базу данных для хранения информационных ресурсов и пользовательских данных;
* произвести оптимизацию и ревьюирование программного кода;
* произвести публикацию веб-приложения в сети;
* произвести тестирование и отладку системы для обеспечения стабильной работы;
* подготовить документацию для пользователей и системных администраторов;
* засчитать экономическую эффективность, разработанного веб-приложения.

Изучение такой организации как МОАУ «СОШ №71» позволит глубже понять потребности образовательных учреждений и разработать веб-приложение, соответствующее актуальным запросам учителей и учащихся. Основными целями школы являются формирование общей культуры личности, ее адаптация к жизни в обществе, освоения профессиональных образовательных программ и повышение качества образования через внедрение инновационных технологий. Важно учитывать не только академические достижения, но и развитие социальных навыков учащихся, таких как сотрудничество, критическое мышление и креативность. Разработка цифровой платформы позволит интегрировать современные методы обучения и даст возможность учителям легко адаптировать учебные материалы под нужды класса.

Важно учитывать мнение всех участников образовательного процесса, чтобы веб-приложение стало удобным и полезным инструментом для реализации их потребностей и амбиций.

В рамках дальнейшего развития проекта планируется регулярное обновление функционала приложения на основе отзывов пользователей и новых образовательных тенденций. Это поможет не только адаптировать платформу к меняющимся условиям, но и предлагать пользователям наиболее актуальные и востребованные сервисы.

Результатом выполнения данной работы станет современное веб-приложение, которое значительно улучшит качество управления образовательным контентом, повысит скорость и удобство взаимодействия участников образовательной среды, а также обеспечит устойчивую работу системы в условиях увеличения нагрузки. Реализуя данные задачи, проект будет способствовать цифровой трансформации образовательных учреждений и укрепит их позиции на современном рынке образовательных услуг.

**1 Анализ предприятия**

**1.1 Характеристика предприятия**

Наименование организации: Муниципальное общеобразовательное автономное учреждение средняя общеобразовательная школа № 71 (МОАУ «СОШ № 71»).

Предприятие находится по юридическому адресу: г. Оренбург, ул. Джангильдина, 6.

Директор: Бодина Татьяна Михайловна.

Основная деятельность предприятия:

* реализация образовательных программ;
* воспитательная работа;
* обеспечение условий для осуществления образовательной деятельности.

МБОУ «СОШ №71» является одним из общеобразовательных учреждений города Оренбурга. Школа была основана в 1986 году и с тех пор активно развивается, внедряя современные образовательные технологии и методы обучения.

Учреждение обеспечивает начальное, основное и среднее общее образование для учащихся, а также предлагает широкий спектр дополнительных образовательных услуг.

В предоставление начального общего образования входит:

* организация и проведение уроков для учащихся 1-4 классов;
* реализация программ, направленных на развитие базовых навыков чтения, письма и математики;
* внедрение инновационных методов обучения и оценивания.

В предоставление основного общего образования входит:

* обучение учащихся 5-9 классов по различным предметам;
* подготовка к государственной итоговой аттестации (ГИА);
* проведение дополнительных занятий и консультаций для углубленного изучения предметов.

В предоставление среднего общего образования входит:

* обучение учащихся 10-11 классов;
* подготовка к ЕГЭ и другим формам итоговой аттестации;
* организация профильного обучения и индивидуальных траекторий развития.

Дополнительная образовательная деятельность включает в себя проведение кружков, секций и клубов по различным направлениям, что позволяет учащимся развивать свои интересы и таланты. В рамках этой деятельности организуются кружки по искусству, спорту, науке и технике, а также проводятся мастер-классы и тематические мероприятия, направленные на развитие творческих способностей учащихся.

Также, важным аспектом является организация летнего отдыха детей и подростков. В этом направлении проводятся летние лагеря с образовательными и развлекательными программами, а также осуществляется участие в межшкольных и городских мероприятиях, которые способствуют активному отдыху и социализации детей.

Вспомогательная деятельность включает в себя обеспечение питания учащихся, что осуществляется через организацию горячего питания в школьной столовой с контролем за качеством и разнообразием меню. Также важным элементом является предоставление медицинской помощи: работает школьная медсестра, которая обеспечивает первичную медицинскую помощь и проводит профилактические осмотры и вакцинации.

Кроме того, поддержание чистоты и порядка в учебных классах, коридорах и других помещениях школы является неотъемлемой частью работы. Регулярно проводятся санитарные мероприятия для обеспечения комфортной образовательной среды. Наконец, безопасность и охрана территории школы также имеют первостепенное значение. Для этого организована охрана территории, а также проводятся инструктажи по безопасности для учащихся и сотрудников.

За 38 лет школа выпустила более 2000 учащихся с аттестатом об окончании средней школы, из них 78 медалистов. Более 4 000 учащихся с аттестатом об окончании основной школы, из них 73 с отличием. С 21.09.2012 года школа получила статус автономной. Результативность образовательного процесса обеспечивается наличием квалифицированных педагогических кадров. Средний возраст педагогического коллектива составляет 40 лет, что является самым благоприятным возрастным показателем для осуществления инновационной деятельности. 3 педагога – почетные работники образования. Также школа имеет 18 педагогов, которые обладают высоким уровнем профессиональной квалификации.

Основными целями школы являются формирование общей культуры личности, ее адаптация к жизни в обществе, создание основы для осознанного выбора и последующего освоения профессиональных образовательных программ, воспитание трудолюбия, уважения к правам и свободам человека, любви к окружающей природе, родине, семье, формирование здорового образа жизни[17].

Цель программы развития – это создать на базе школы, в рамках образовательного процесса, условия для получения качественного, конкурентоспособного образования, способствующего социальной адаптации в обществе.

В современном образовании учителя сталкиваются с рядом трудностей при поиске и использовании качественных учебных материалов, методических разработок, игр и тестов. Существующие ресурсы часто разрознены, требуют значительных усилий для поиска необходимой информации, а также не предоставляют возможности для эффективного обмена разработками и опытом между педагогами.

Часто учителя вынуждены тратить больше времени на поиск и подготовку материалов, а не на непосредственное взаимодействие с учениками [18].

Поэтому, разрабатываемый сайт «УчительPRO» направлен на облегчение жизни учителям с помощью удобной платформы для обмена готовыми ресурсами, с удобным поиском, что освободит ценное время учителей и даст возможность скооперироваться для повышения качества образования.

**1.2 Анализ предметной области**

Образование и школа представляют собой важнейшую и многогранную предметную область, охватывающую все аспекты обучения, преподавания, управления и оценки в формальных образовательных учреждениях.

Основная цель образования заключается в том, чтобы обеспечить учащихся необходимыми знаниями, навыками и ценностями, которые помогут им успешно жить и активно участвовать в жизни общества. При анализе данной предметной области можно выделить несколько ключевых компонентов.

Во-первых, учащиеся являются центральными фигурами образовательного процесса, получая знания и навыки, которые формируют их будущее. Преподаватели, в свою очередь, выступают в роли специалистов, ответственных за обучение и поддержку своих учеников. А учебный план представляет собой структурированную последовательность тем и навыков, которые должны быть освоены учащимися.

Во-вторых, инфраструктура включает в себя здания, оборудование и ресурсы, необходимые для поддержки учебного процесса, а управление представляет собой структуры и процессы, обеспечивающие эффективное функционирование образовательных учреждений.

Проблема данной предметной области – это отсутствие единой платформы для обмена ресурсами среди учителей и всех, кто включен в образовательный процесс.

Организация выдвигает требование разработать и внедрить веб-приложение, которое будет служить платформой для публикации и обмена образовательными ресурсами среди преподавателей и всех участников образовательного процесса. Это веб-приложение будет сосредотачиваться на улучшении доступа к качественным учебным материалам и цифровым инструментам для взаимодействия между учителями и учениками.

Оно предоставит следующие функции пользователям:

* авторизация и регистрация;
* персонализированный профиль пользователя;
* поиск и фильтрация по ключевым словам, предмету, тегу;
* просмотр каталога с ресурсами;
* скачивание интересующих ресурсов;
* публикация своих ресурсов и разработок;
* загрузка различных форматов файлов (pdf, docx, pptx, jpg.);
* форма публикации с возможностью добавления описания, к разработке;
* просмотр скаченных или загруженных ресурсов;
* подписка на рассылку информации о новых ресурсах.

Веб-приложение повлияет на следующие бизнес-процессы:

* упрощение обмена ресурсами;
* повышение качества образования;
* оптимизация рабочих процессов.

Упрощение обмена ресурсами будет достигнуто за счет устранения необходимости в поиске материалов в отдельных источниках, что позволит педагогам тратить больше времени на обучение. Повышение качества образования произойдет за счет систематизации материалов и использования лучших практик, что приведет к улучшению качества преподавания и усвоения знаний учащимися. Оптимизация рабочих процессов случится благодаря автоматизации процессов, связанных с созданием и публикацией материалов, что позволит учителям быть более продуктивными [2].

Таким образом, для реализации веб-приложения необходимо будет определить основные требования к его разработке.

В рамках практики, целью которой является создание сайта по предметной области, предусмотрено выполнение ряда задач, направленных на глубокое погружение в учебный процесс и его организацию. Первым шагом станет проведение наблюдений за учебным процессом, что позволит получить представление о реальных условиях обучения и взаимодействии между учениками и преподавателями. Далее, важной задачей будет анализ учебных планов, программ и материалов, чтобы понять структуру и содержание образовательного процесса, а также выявить сильные и слабые стороны существующих ресурсов. Изучение методик преподавания и оценки знаний поможет определить наиболее эффективные подходы к обучению, которые могут быть внедрены на создаваемом сайте. Беседы с учителями, администрацией и родителями станут важным источником информации о потребностях и ожиданиях всех участников образовательного процесса, что поможет лучше адаптировать контент сайта под их запросы.

Современные тенденции в образовании включают интеграцию технологий, что подразумевает использование цифровых инструментов для улучшения учебного процесса. Глобальный охват подразумевает сотрудничество между школами и образовательными учреждениями по всему миру [5].

Важный аспект современного образования это что использование технологий онлайн-платформ учителями стало его неотъемлемой частью. Эти платформы предоставляют широкий доступ к разнообразным ресурсам, включая видеоуроки и учебные материалы, что позволяет учителям разнообразить свои занятия и сделать их более увлекательными для учеников.

Анализ предметной области показывает, что создание платформы «УчительPRO» может значительно улучшить доступ к образовательным ресурсам и взаимодействие между участниками обучающего процесса. Определение четких требований к приложению станет важным шагом к его успешной реализации и внедрению.

**2 Техническое задание**

Полное наименование веб-приложения: «УчительPRO: платформа для обмена образовательными ресурсами».

Условное обозначение веб-приложения: «УчительPRO».

Документы, на основании которых ведется разработка веб-приложения: «Договор о практической подготовке обучающихся».

Заказчик: МОАУ «СОШ №71».

Исполнитель: Васильева В. П., студент ГАПОУ «ОКЭИ».

Временные рамки для разработки веб-приложения: с 21.04.2025 по 17.05.2025.

Цели создания веб-приложения:

* оптимизация процесса обмена образовательными ресурсами;
* упрощение доступа к образовательным ресурсам;
* повышение качества образовательного процесса;
* адаптация к современным образовательным трендам;
* предоставление возможности обучения и самосовершенствования;
* создание персонализированного пользовательского опыта.

Функции, выполняемые веб-приложением:

* авторизация и регистрация;
* персонализированный профиль пользователя;
* поиск и фильтрация, по ключевым словам, предмету, тегу;
* просмотр каталога с ресурсами;
* скачивание интересующих ресурсов;
* публикация своих ресурсов и разработок;
* загрузка различных форматов файлов (.pdf, .docx, .pptx, .jpg.);
* форма публикации с возможностью добавления описания, к разработке;
* просмотр скаченных или загруженных ресурсов.

Функциональная модель сайта, представлена в приложении А в виде контекстной диаграммы, которая описывает взаимодействие системы с внешними сущностями: пользователями и администратором.

Для решения проблемы предметной области необходимо определить ключевые требования, которые обеспечат его успешное функционирование и защиту данных [15].

**2.1 Требования к серверной части**

Сервер должен быть оснащен многопроцессорной архитектурой, желательно с процессорами семейства «Intel Xeon» или «AMD EPYC», с тактовой частотой не менее 2,5 ГГц. Высокая производительность процессоров позволит эффективно обрабатывать одновременно несколько запросов и выполнять вычислительно интенсивные задачи.

Рекомендуемый объем оперативной памяти составляет не менее 16 Гб для обеспечения стабильной работы приложения при одновременной загрузке множества пользователей и обработки данных. В случае высокой нагрузки объем оперативной памяти можно увеличить до 32 Гб или более.

Выбор дистрибутива должен учитывать специфику используемых технологий и поддерживаемое программное обеспечение.

Рекомендуемый минимум для объема жесткого диска составляет 500 Гб, что позволит хранить базу данных, пользовательские данные и образовательные материалы. Рекомендуется использовать «SSD-диски» для повышения скорости чтения и записи данных, что, в свою очередь, увеличит быстродействие веб-приложения.

Сетевое соединение должно обеспечивать стабильную скорость передачи данных не менее 100 Мбит/с, желательно с резервированием канала связи. Это позволит обеспечить быстрое взаимодействие пользователей с сервером, особенно при загрузке и скачивании больших файлов [5].

К дополнительным требованиям относятся возможность использования виртуализации для оптимизации ресурсов сервера и обеспечения гибкости в управлении окружением. Также рекомендуется внедрение систем мониторинга для отслеживания состояния сервера, использования ресурсов и предотвращения потерь данных. Регулярное обновление программного обеспечения и систем безопасности является обязательным условием для поддержания безопасной и эффективной работы сервера.

**2.2 Требование к разработке**

2.2.1 Среда разработки

Для разработки веб-приложения «УчительPRO» должны быть выбраны современные инструменты, которые обеспечивают гибкость, функциональность и удобство в работе.

«Visual Studio Code» («VSCode») – это универсальная и мощная среда разработки, предназначенная для работы с различными языками программирования, такими как «HTML», «CSS», «JS» и многими другими.

Преимущества:

* интуитивно понятный интерфейс, что упрощает работу разработчиков;
* доступны мощные инструменты для написания кода;
* через плагины можно настроить среду под свои потребности.

«Figma» – это программа для коллаборативного дизайна и прототипирования, позволяющая командам разработчиков совместно работать над дизайном и визуализацией проектов.

Преимущества:

* возможность редактирования проектов несколькими участниками команды;
* можно создавать прототипы интерфейсов и визуализировать структуру;
* облегчает обратную связь между участниками команды разработки.

Использование «Visual Studio Code» и «Figma» в процессе разработки интернет-магазина обеспечивает комфортное рабочее окружение для программирования, дизайна и прототипирования, а также способствует эффективной командной работе, обеспечивая удобство, функциональность и качество процесса разработки [7].

2.2.2 Технологии

Для разработки веб-сайта должны быть использованы различные технологии, инструменты и языки программирования для обеспечения удобной работы.

«HyperText Markup Language» («HTML») выступает в качестве языка разметки, формирующего структуру и содержимое веб-страниц. Его простота и широкая распространенность позволяют создавать интуитивно понятный интерфейс пользователя. Преимуществом «HTML» является его кросс-браузерная совместимость и поддержка всеми современными браузерами.

«Cascading Style Sheets» («CSS») используется для стилизации веб-страниц, определяя внешний вид элементов, такие как цвета, шрифты, расположение и другие визуальные аспекты. «CSS» обеспечивает разделение стилей от структуры «HTML», что способствует улучшению читаемости кода, упрощает его поддержку и позволяет легко изменять внешний вид приложения. Возможность использования «CSS-фреймворков» значительно ускоряет процесс разработки и обеспечивает единообразие стиля.

«JavaScript» является языком программирования, обеспечивающим интерактивность веб-приложения. С помощью «JavaScript» будет реализована динамическая обработка данных на стороне клиента, валидация форм, обработка пользовательских событий и другие функции, обеспечивающие удобство использования приложения. «JavaScript» позволяет создавать динамические и отзывчивые интерфейсы, улучшая пользовательский опыт. Использование «JavaScript-фреймворков» или библиотек позволило бы повысить эффективность разработки и обеспечить более высокое качество кода.

«Node.js» – серверная платформа, которая позволяет использовать «JavaScript» на серверной стороне, что обеспечивает высокую производительность благодаря асинхронной обработке запросов[14].

«SQLite» – это встраиваемая база данных «SQL», которая не требует отдельного сервера. Это упрощает развертывание приложения, особенно для небольших проектов, не требующих высокой масштабируемости и производительности. «SQLite» обеспечивает простоту использования и интеграции с «JavaScript-кодом» через соответствующие библиотеки. Встраиваемый характер «SQLite» позволяет сделать приложение более автономным и портативным [15].

Для реализации некоторых функций могут быть использованы библиотеки и фреймворки:

* «Express.js» – фреймворк который предлагает мощные инструменты для маршрутизации, работы с запросами и настройками «middleware»;
* «Sequelize» – «ORM» для работы с реляционными базами данных с помощью JavaScript-объектов и методов;
* «Jest» – фреймворк для тестирования «JavaScript», который поддерживает тестирование приложений на «Node.js».

В целом, выбранный технологический стек позволит создать веб-приложение с понятным интерфейсом, динамическим поведением и относительно простой реализацией на стороне сервера.

2.2.3 Требования к интерфейсу

Требования к адаптивности и юзабилити:

* интерфейс должен иметь адаптивную верстку (на ПК, планшетах, смартфонах);
* минимизировать количество действий пользователя для достижения целей;
* интерфейс должен поддерживать интерактивные элементы, такие как кнопки, выпадающие списки и формы;
* обеспечить быструю загрузку страниц, минимизируя время ожидания;
* анализировать поведение пользователей на сайте с помощью аналитических инструментов;
* графические элементы должны быть оптимизированы по размеру и формату для быстрого загрузки.

Требования к дизайну и стилистике:

* избегать перегруженности информацией на страницах, давая приоритет простоте и легкости в использовании;
* использовать простые формы и элементы интерфейса, которые помогут пользователям сосредоточиться на контенте.
* цвета должны быть спокойными, но с яркими акцентами для привлечения внимания;
* стилистика должна отражать сферу деятельности и направленность сайта;
* использовать читаемые шрифты с хорошим контрастом для лёгкости восприятия информации;
* разные стили шрифтов могут помочь в организации контента на странице.

Требование к иллюстрация и изображениям:

* использовать качественные изображения и иллюстрации, которые соответствуют тематике и добавляют визуальный интерес;
* иллюстрации могут быть использованы для визуализации понятий из сферы образования.

В качестве среды разработки макетов и логотипа будет использован онлайн-сервис для разработки интерфейсов и прототипирования – Figma.

В качестве источников бесплатных изображений должны быть использованы:

* Flaticon;
* Freepik;
* Storyset.

2.2.4 Требования к разработке базы данных

В качестве СУБД для разработки базы данных будет использоваться SQLite.

Этот выбор обусловлен следующими преимуществами:

* не требует настройки и эксплуатации отдельного сервера, что упрощает развертывание и внедрение системы;
* обеспечивает быструю работу с данными даже при больших объемах информации;
* не нуждается в выделенных ресурсах сервера;
* возможность использования на различных операционных системах.

Для создания и управления базой данных будет использоваться декларативный язык SQL, что обеспечивает:

* четкое определение структуры таблиц, связей и ограничений;
* простоту выполнения операций по вставке, обновлению, удалению и выборке данных;
* возможность автоматического поддержания целостности данных и соблюдения правил бизнес-логики.

База данных должна полностью соответствовать разработанной информационной модели системы, которая включает:

* все необходимые сущности;
* определение связей между сущностями;
* атрибуты каждой сущности.

Доступ к базе данных должен быть разграниченным и контролируемым.

Для каждого типа пользователя должны быть установлены свои роли и права, позволяющие выполнять только разрешенные операции.

Необходимо реализовать механизмы авторизации и аутентификации при работе с базой данных, чтобы обеспечить защиту данных от несанкционированного доступа.

2.2.5 Требования к разграничению прав

В веб-приложении будут предусмотрены различные уровни доступа, которые будут соответствовать статусу пользователя.

Если пользователь не будет авторизован, то он сможет просматривать общедоступные разделы сайта, такие как поиск, просмотр ресурсов и ознакомление с информацией. Однако для редактирования и управления функционалом ему потребуется пройти авторизацию.

После авторизации пользователь получит доступ к дополнительным возможностям, таким как редактирование контента, добавление материалов и управление личными данными. Однако доступ к определённым функциям и разделам будет ограничен уровнем доступа, соответствующим роли пользователя.

Администратор будет пользователем с полными правами, который сможет управлять всеми пользователями, изменять системные настройки, редактировать любой контент и выполнять административные операции, связанные с безопасностью и настройками системы.

Вход в систему будет осуществляться через процедуру авторизации, которая будет основана на логине и пароле. Авторизация будет защищена от несанкционированного доступа.

В случае необходимости пользователь сможет восстановить пароль через электронную почту или другие средства идентификации. Для защиты данных будут применяться меры по предотвращению подбора паролей и других видов атак, такие как ограничение количества неудачных попыток входа.

Все ключевые действия пользователей, такие как вход и выход из системы, создание, редактирование и удаление контента, будут регистрироваться в журнале. Эти логи будут использоваться для аудита, мониторинга активности и повышения общей безопасности системы.

Логика разграничения прав будет реализована на уровне системы, поэтому пользователи с разными ролями будут видеть только разрешённые разделы и функции. Запрет доступа к административным разделам и функциям редактирования будет осуществляться через механизм авторизации, который будет проверять роли и права перед выполнением операций. Это исключит возможность обхода прав доступа и обеспечит высокий уровень безопасности и целостность данных.

**2.3 Тестирование и документация**

Для корректной работы веб-приложения «УчительPRO» необходимо выполнить ключевые действия, касающиеся тестирования и подготовки документации. Так как тестирование является неотъемлемой частью процесса разработки, обеспечивающей высокое качество и стабильность приложения. Необходимо реализовать юнит-тестирование, целью которого является проверка отдельных компонентов приложения на корректность выполнения заданных функций. Для этого следует использовать фреймворки, такие как «Jest» или «Mocha» для «JavaScript». Каждая функция и метод должны быть покрыты юнит-тестами, чтобы гарантировать их корректную работу.

Интеграционное тестирование также должно быть проведено для проверки взаимодействия между модулями и компонентами приложения. Для этого необходимо создать тесты, которые проверяют, как различные модули работают вместе, используя инструменты, такие как «Cypress» или «Postman» для «API-тестирования». Функциональное тестирование должно оценить функциональность приложения с точки зрения конечного пользователя. Это включает тестирование сценариев использования, где проверяются все функции приложения, такие как регистрация, авторизация, загрузка файлов и доступ к материалам.

Нагрузочное тестирование необходимо для оценки производительности приложения под высокой нагрузкой. Для этого следует использовать инструменты, такие как «Apache JMeter» или «Gatling», для моделирования большого количества одновременных пользователей и измерения реакции приложения. Ожидается, что это позволит определить пределы производительности приложения и выявить узкие места в системе.

Функциональная модель сайта, представлена в приложении Б в виде контекстной диаграммы, которая описывает взаимодействие системы с внешними сущностями: пользователями и администратором [16].

Тестирование безопасности должно быть проведено для выявления уязвимостей приложения. Это включает тестирование на проникновение и использование инструментов для анализа безопасности. Ожидается, что выявленные уязвимости будут устранены, что повысит уровень безопасности приложения.

И так же неотъемлемой частью процесса разработки является создание технической документации является важным этапом, обеспечивающим поддержку и дальнейшее развитие приложения. Основные элементы документации включают архитектурную документацию, которая представляет собой подробное описание архитектуры приложения, включая диаграммы классов, диаграммы последовательностей и схему базы данных. В документации должно быть общее видение системы, описание архитектурного стиля, компоненты системы и их взаимодействие, а также технические детали реализации.

Инструкции по развертыванию должны содержать пошаговые рекомендации по развертыванию приложения на сервере, включая требования к окружению и необходимые конфигурации. Включение установки необходимых зависимостей и сред, примеров конфигурационных файлов, шагов по миграции базы данных и настроек окружения также является обязательным.

Инструкции по обслуживанию должны описывать процесс поддержки приложения, включая рекомендации по регулярному обновлению зависимостей, процедуры резервного копирования и восстановления базы данных, регулярные тесты и мониторинг состояния приложения.

Пользовательская документация должна содержать инструкции и руководство для пользователей по работе с приложением, включая шаги по регистрации и созданию учетной записи, объяснения функций и возможностей платформы, а также раздел часто задаваемых вопросов и решения распространенных проблем.

Использование системы контроля версий для управления исходным кодом проекта является необходимостью. «Git» должен быть использован для отслеживания изменений в коде и обеспечения совместной работы команды. Рекомендуется создавать отдельные ветки для разработки новых функций и исправления ошибок, регулярно коммитить изменения с содержательными сообщениями и осуществлять процесс слияния изменений с основной веткой через «Pull Requests», что обеспечит код-ревью и обсуждение изменений [12].

Таким образом, качественное тестирование и документация обеспечивают безопасность и удобство использования веб-приложения «УчительPRO», что, в свою очередь, позитивно сказывается на его развитии и поддержке.

**3 Разработка проектной документации**

**3.1 Анализ информационных потоков веб-приложения**

Работа в системе организована таким образом, что в ходе анализа в качестве основной входной информации могут выступать:

* техническое задание на разработку веб-приложения;
* список образовательных ресурсов и их метаданных;
* требования к дизайну интерфейса;
* данные о пользователях и ролях.

Основные сценарии использования системы позволяют реализовать следующие функции:

* поиск и просмотр образовательных материалов;
* добавление и редактирование ресурсов авторизованными пользователями;
* управление аккаунтами и настройками системы администраторами;
* автоматическую генерацию отчетов о деятельности пользователей и активности сайта.

Основные процессы:

* обработка запросов пользователя — главный процесс, который принимает запросы и отображает результаты.
* работа с ресурсами — хранение, добавление, редактирование ресурсов (в основном педагогом и админом).
* работа с пользовательскими профилями — управление профилями пользователей (педагог/учащийся).
* генерация отчетов и логов — для администратора.
* отображение каталога — отображение ресурсов пользователю.

Роли пользователей системы и описание как они взаимодействуют с системой представлены в таблице 1.

|  |  |  |  |
| --- | --- | --- | --- |
| Пользователи | С чем взаимодействуют | Что делает | Результат  (что получает) |
| Гость | Обработка запросов | Просматривать открытые ресурсы, искать | Каталог, результаты поиска |
| Педагог | Обработка запросов и управление контентом | Добавлять, редактировать ресурсы, обновлять профиль | Обновлённый каталог, профиль |
| Учащийся | Обработка запросов | Просматривать каталог, профиль | Контроль системы, отчёты, логи |
| Админ | Обработка запросов и управление системой | Просматривать каталог, профиль | Контроль системы, отчёты, логи |

Описание потоков данных:

* пользователи вводят запросы на поиск, а также данные для входа в систему (логин, пароль).
* в системе происходит проверка учетных данных (авторизация), после чего осуществляется разграничение доступа по ролям;
* в качестве выхода – отображение результатов поиска, подтверждения операций, отчётов о деятельности.
* в системе сохраняются и обновляются сведения о ресурсах, логах действий пользователей и настройках системы.

Таким образом, на рисунке 1 представлена диаграмма потоков данных в нотации DFD.

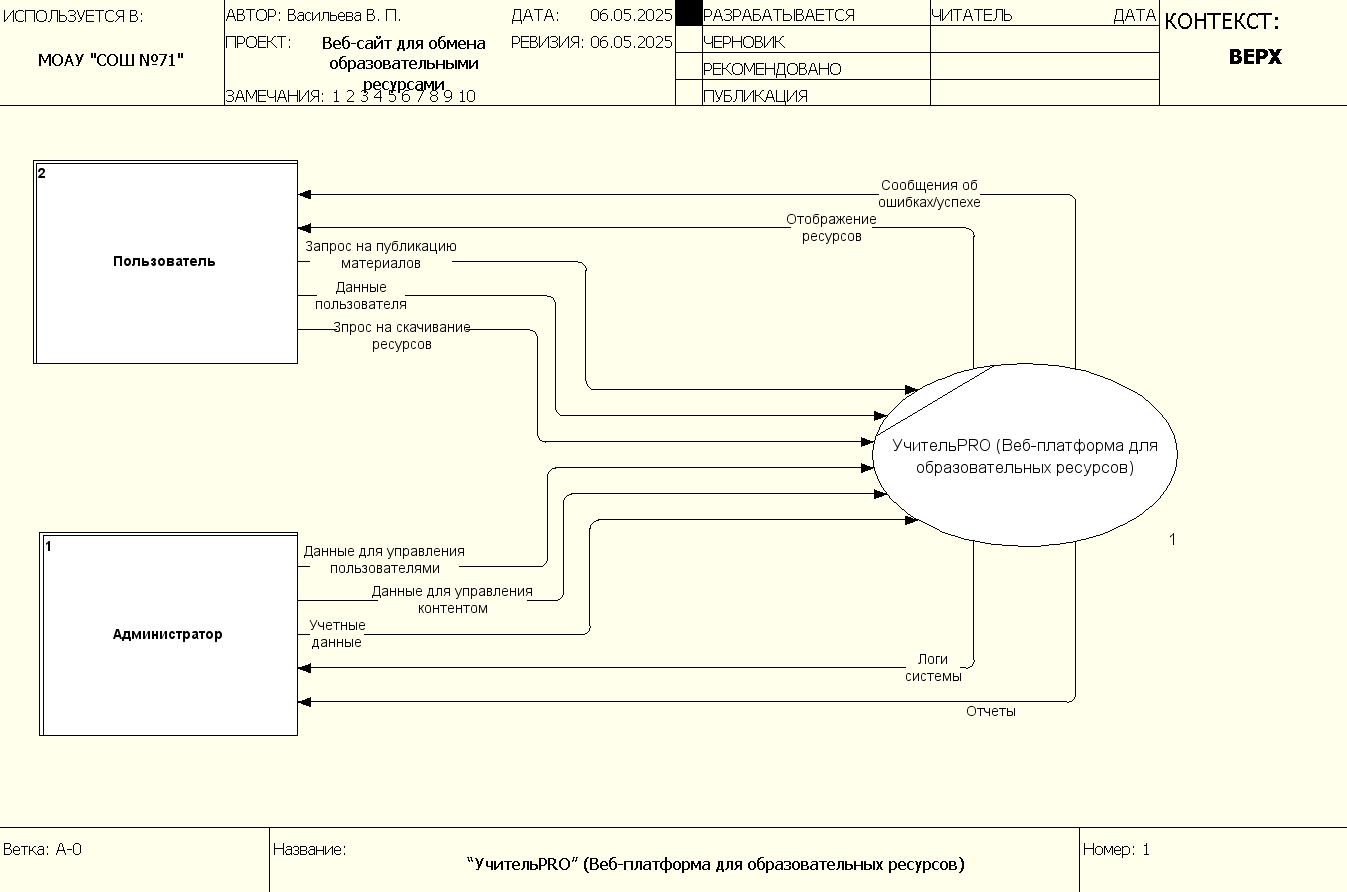


Рисунок 1 – Пример диаграммы в нотации DFD.

Диаграмма прецедентов для веб-приложения «УчительPRO» иллюстрирует взаимодействие различных типов пользователей и администратора с функционалом разрабатываемого сайта. На диаграмме выделяются три основных актора, каждый из которых имеет свои особенности и возможности.

Первый актор – пользователь, который представляет незарегистрированного или неавторизованного пользователя с ограниченным доступом к функционалу сайта (гость). Он может зарегистрироваться, что позволит создать аккаунт на сайте, просматривать главную страницу, включая доступ к каталогу разработок, а также искать и скачивать образовательные материалы.

Второй актор – авторизованный пользователь с ролью «педагог», который успешно прошел регистрацию и авторизацию, открывая доступ к расширенному набору возможностей. Он может авторизоваться для входа в свой аккаунт, настраивать параметры для публикуемых разработок, публиковать собственные образовательные ресурсы, включая возможность загрузки файлов, а также просматривать личный профиль, что всегда включает доступ к списку скаченных и разработанных ресурсов с возможностью расширенного просмотра и загрузки файлов.

Третий актор – авторизованный пользователь с ролью «учащийся», который успешно прошел регистрацию и авторизацию, открывая доступ к расширенному набору возможностей. Он может авторизоваться для входа в свой аккаунт, просматривать личный профиль, что всегда включает доступ к списку скаченных ресурсов с возможностью расширенного просмотра и загрузки файлов.

Четвёрный актор – администратор, который обладает полным контролем над сайтом, включая управление пользователями и контентом. Администратор может модерацией контента контролировать и одобрять ресурсы, публикуемые пользователями, управлять пользователями, добавляя, удаляя и редактируя информацию о них, а также просматривать статистику, что предоставляет доступ к статистическим данным о использовании сайта [20].

Связь «include» означает, что один прецедент всегда включает в себя другой, например, просмотр главной страницы всегда включает просмотр каталога разработок.

Связь «extend» означает, что один прецедент может расширить другой, но это не является обязательным, например, публикация разработки может быть расширена загрузкой файлов.

Диаграмма прецедентов для веб-приложения «УчительPRO» представлена на рисунке 2.

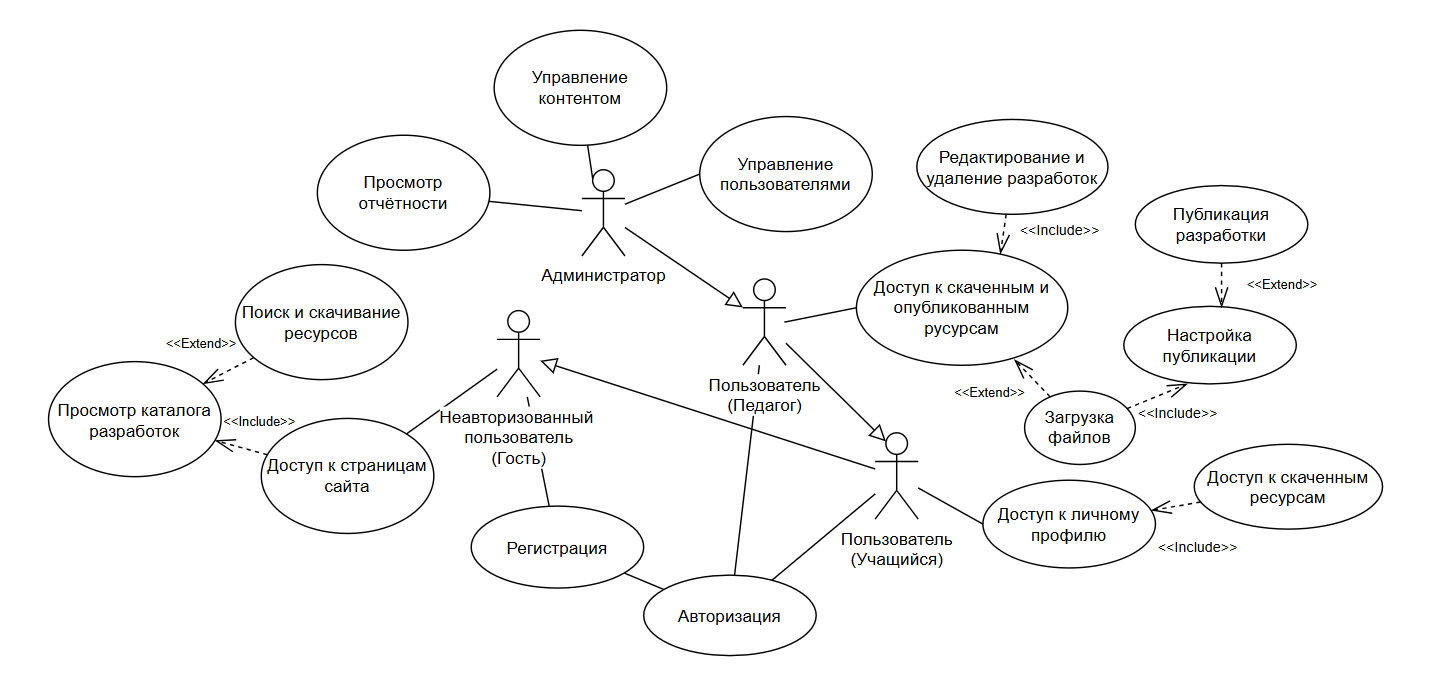


Рисунок 2 – Диаграмма прецедентов

В результате анализа информационных потоков было выявлено, что основными источниками данных являются запросы пользователей, файлы ресурсов, метаданные и параметры поиска. Внутри системы данные проходят обработку через центральный процесс, который обеспечивает отображение каталога, формирование отчетов и управление профилями. Такой анализ позволил определить ключевые точки взаимодействия и обеспечить корректную организацию потоков информации для эффективной работы системы.

**3.2 Разработка документации функционального назначения**

В ходе проектирования системы «УчительPRO» особое внимание будет уделяться созданию надежной и расширяемой базы данных, которая обеспечит хранение и управление всей информацией, необходимой для функционирования веб-приложения. Основной этап будет заключаться в формировании системы таблиц, включающей описание сущностей, их атрибутов и связей. В данной главе будет представлена структура базы данных, в том числе ER-диаграмма, информационная и физическая модели.

Также важно писание таблиц, обеспечивающих хранение данных о пользователях, ролях, историях скачиваний, разработках, профилях, категориях и тегах. Для некоторых таблицы будут приведены технические характеристики, такие как ключи, типы данных, ограничения и образцы данных.

Следующий этап проектирования будет предполагать формирование физической модели базы данных на базе выбранной системы управления – SQLite, с учетом требований к обеспечению эффективности, целостности и безопасности данных. В таблицах 2-4 приведен пример технического описания реляционных таблиц.

Таблица 2 – Техническое описание таблицы «users»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Ключ | Тип, длина | Обязательность значения | Логическое ограничение | Примеры данных |
| id | PK | int | not null | Check (Номер >0) | 1, 2, 3 |
| name |  | varchar  (100) | not null |  | Иван Иванов |
| email |  | varchar  (100) | not null |  | ivan@mail.ru |
| role\_id | FK | int | not null |  | 1, 2, 3 |
| password |  | varchar  (225) | not null |  | qwer1234 |

Таблица 3 – Техническое описание таблицы «developments»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Ключ | Тип, длина | Обязательность значения | Логическое ограничение | Примеры данных |
| id | PK | int | not null | Check (Номер >0) | 1, 2, 3 |

Продолжение таблицы 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| name |  | varchar  (100) | not null |  | «Учебное пособие» |
| title\_description |  | text | not null |  | «Книга по математике» |
| file\_path |  | varchar  (225) | not null |  | /files/урок.pdf |
| category\_id | FK | varchar  (225) | not null |  | 1, 2, 3 |
| preview |  | int | not null |  | математика.jpg |
| profession |  | varchar  (225) | null |  | «Учитель информатики» |

Таблица 4 – Техническое описание таблицы «roles»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Ключ | Тип, длина | Обязательность значения | Логическое ограничение | Примеры данных |
| id | PK | int | not null | Check (Номер >0) | 1, 2, 3 |
| role\_name |  | varchar  (50) | not null |  | user, admin |

На ER-диаграмме, представленной на рисунке 3, изображено 10 классов объектов. Основной вид связи между классами объектов - 1:М с обязательной опциональностью на стороне «много» .

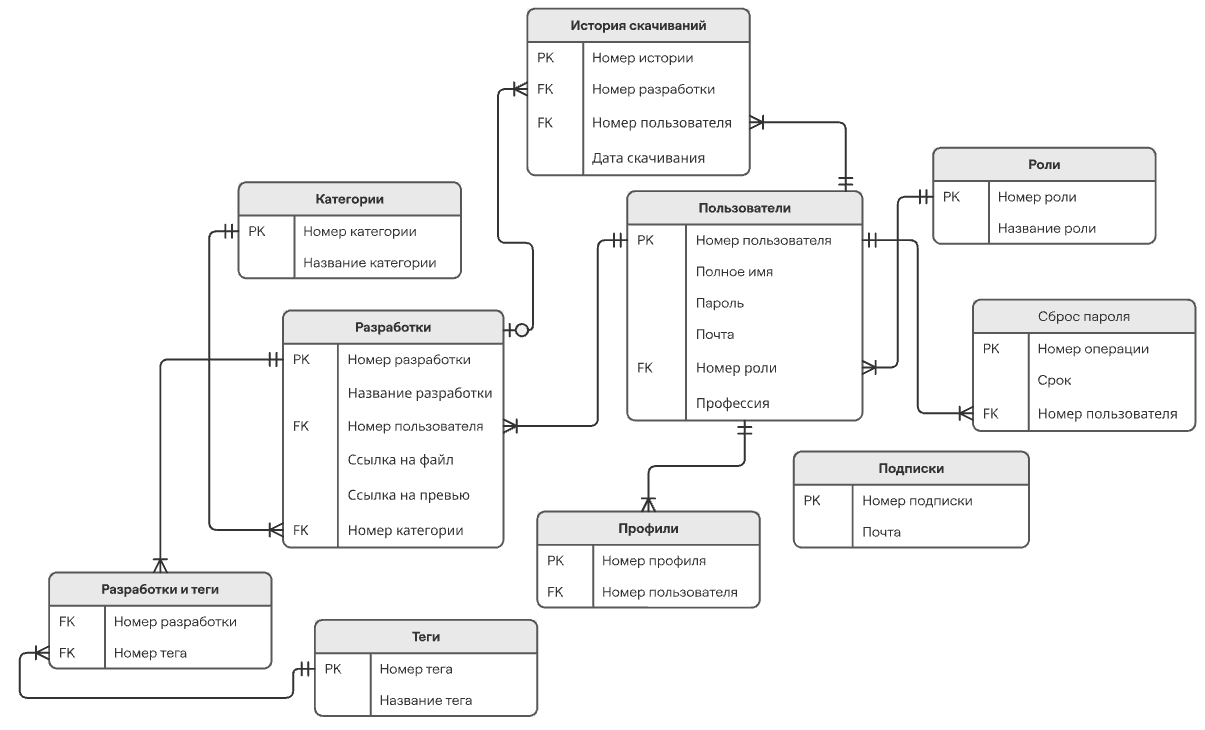


Рисунок 3 – Пример ER-диаграммы

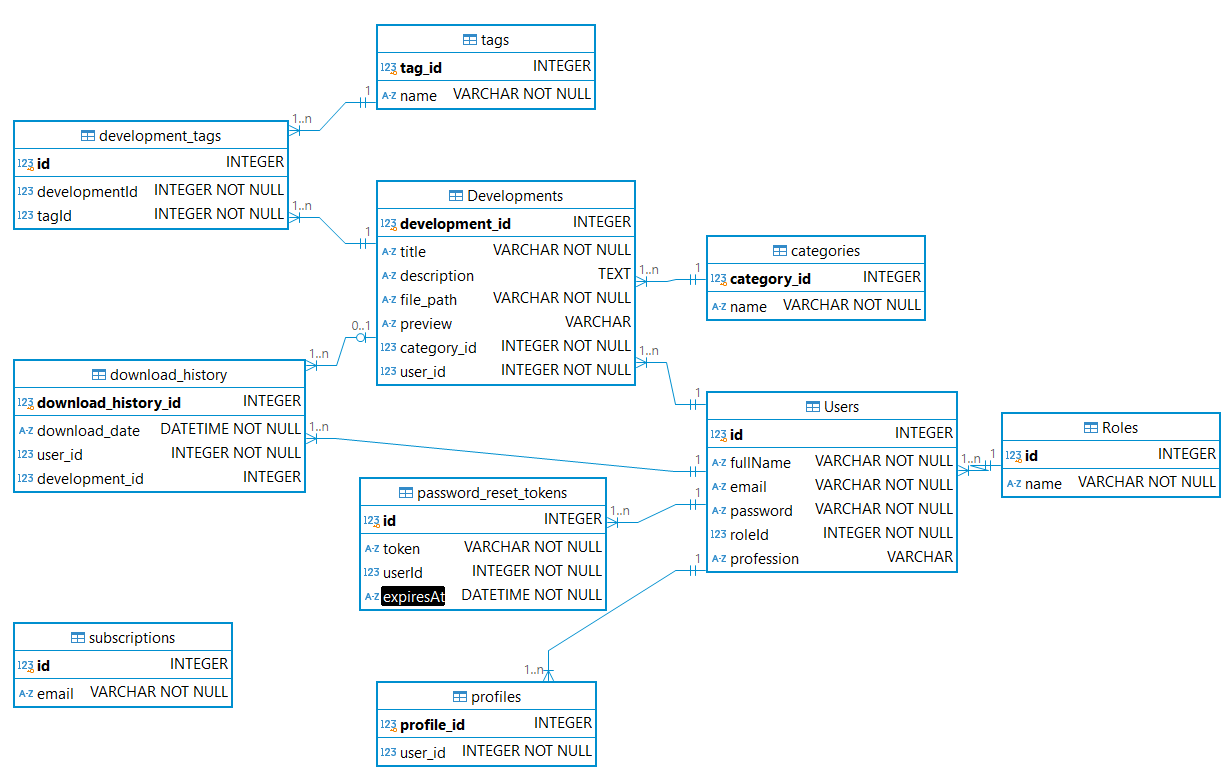


Рисунок 4 – Реализация ER-диаграммы

В результате выполненной разработки документации определены основные функции и задачи веб-приложения, а также описание ролей пользователей и их взаимодействий с системой.

**3.3 Разработка макетов и эскизов веб-приложения**

Эскиз или прототип дизайна веб-сайта представляет собой предварительное визуальное представление будущего веб-приложения или сайта. Он содержит основные элементы интерфейса, расположение разделов и функционал приложения.

Главная цель эскиза – позволить разработчикам, дизайнерам и заказчикам получить общее представление о будущем продукте, определить его функционал, взаимосвязи элементов и общий внешний вид.[4].

На рисунке 5, представлены «Шапка» и «Подвал» сайта, они имеют функцию навигации и контактной информации соответственно, они дублируются на каждой странице.



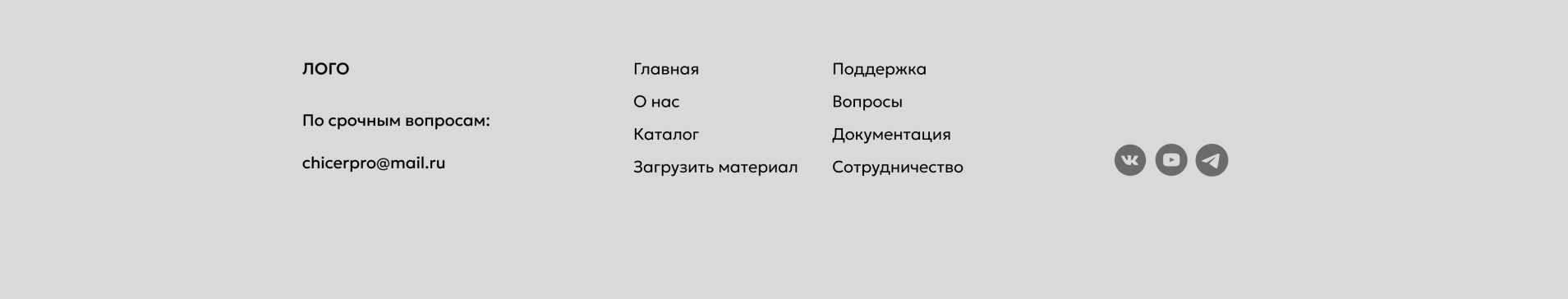


Рисунок 5 – Эскиз «Шапки» и «Подвала» сайта

На рисунке 6 представлен эскиз контента главной страницы, визуализирующий ее ключевые структурные элементы. Здесь отображены такие блоки, как слайдер с баннерами, информация об услугах и перечень преимуществ платформы. Слайдер с баннерами призван привлечь внимание пользователя и кратко представить основные возможности ресурса. Раздел с информацией об услугах, форма подписки на рассылку, целевая аудитория пользователей и примеры разработок.

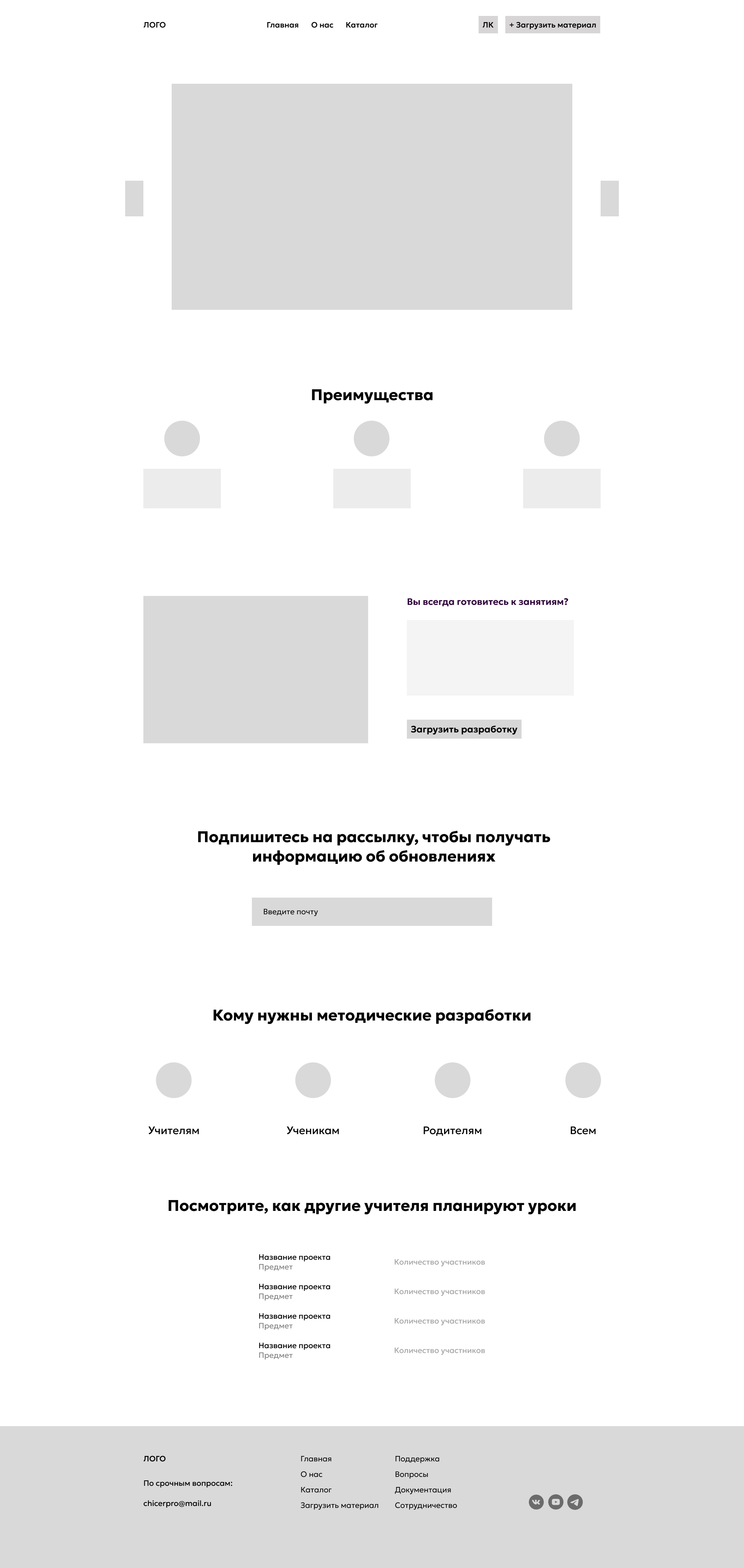


Рисунок 6 – Эскиз страницы «Главная»

На рисунке 7 представлен эскиз контента страницы о компании, где находится рекламный баннер, история с миссией компании и главной целью сайта, а также ее ценности и контактная информация для связи.

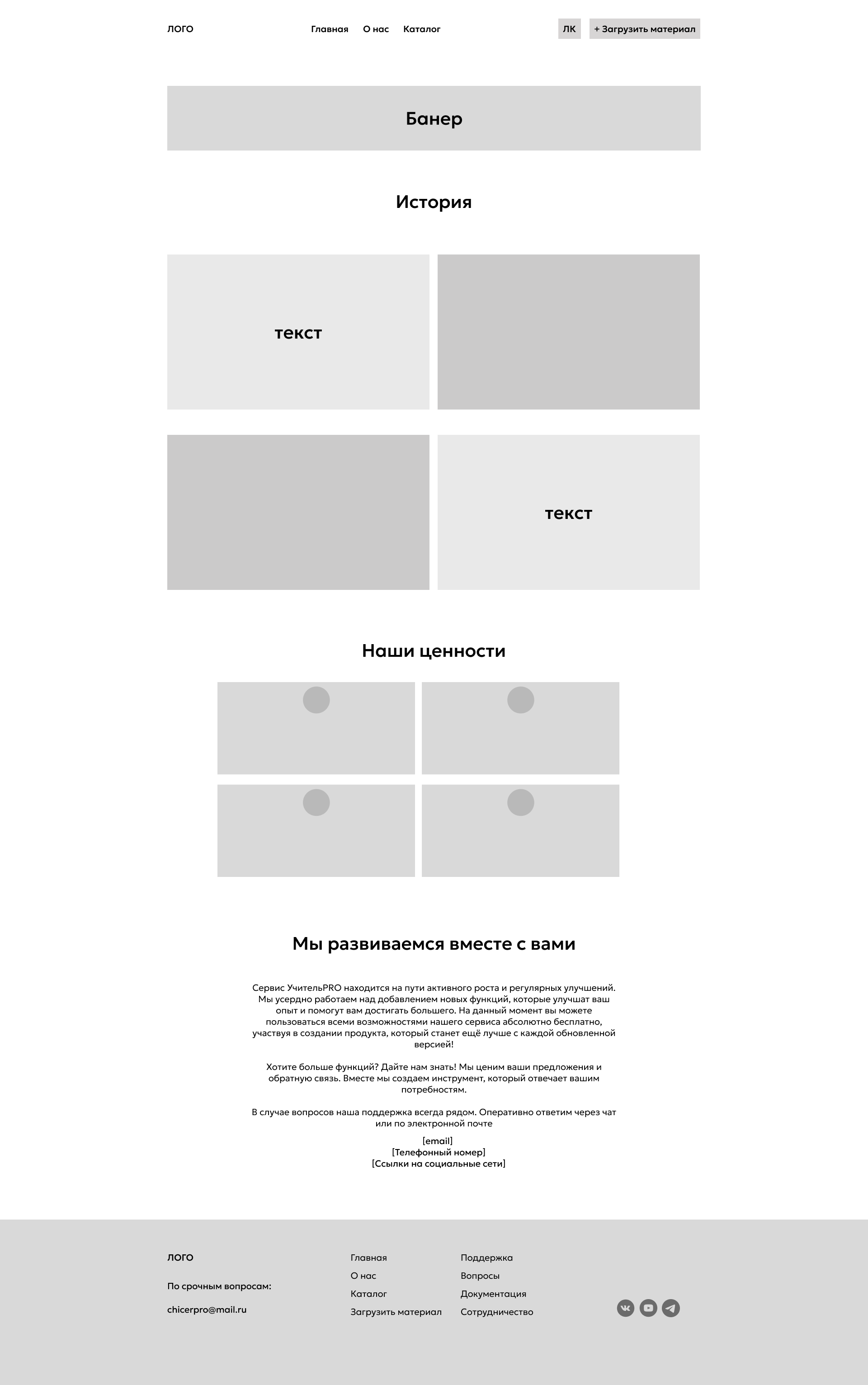


Рисунок 7 – Эскиз страницы «О нас»

На рисунке 8 представлен эскиз контента страницы с каталогом, где находится рекламный баннер, поиск и фильтрация, сам каталог с разработанными материалами других пользователей.

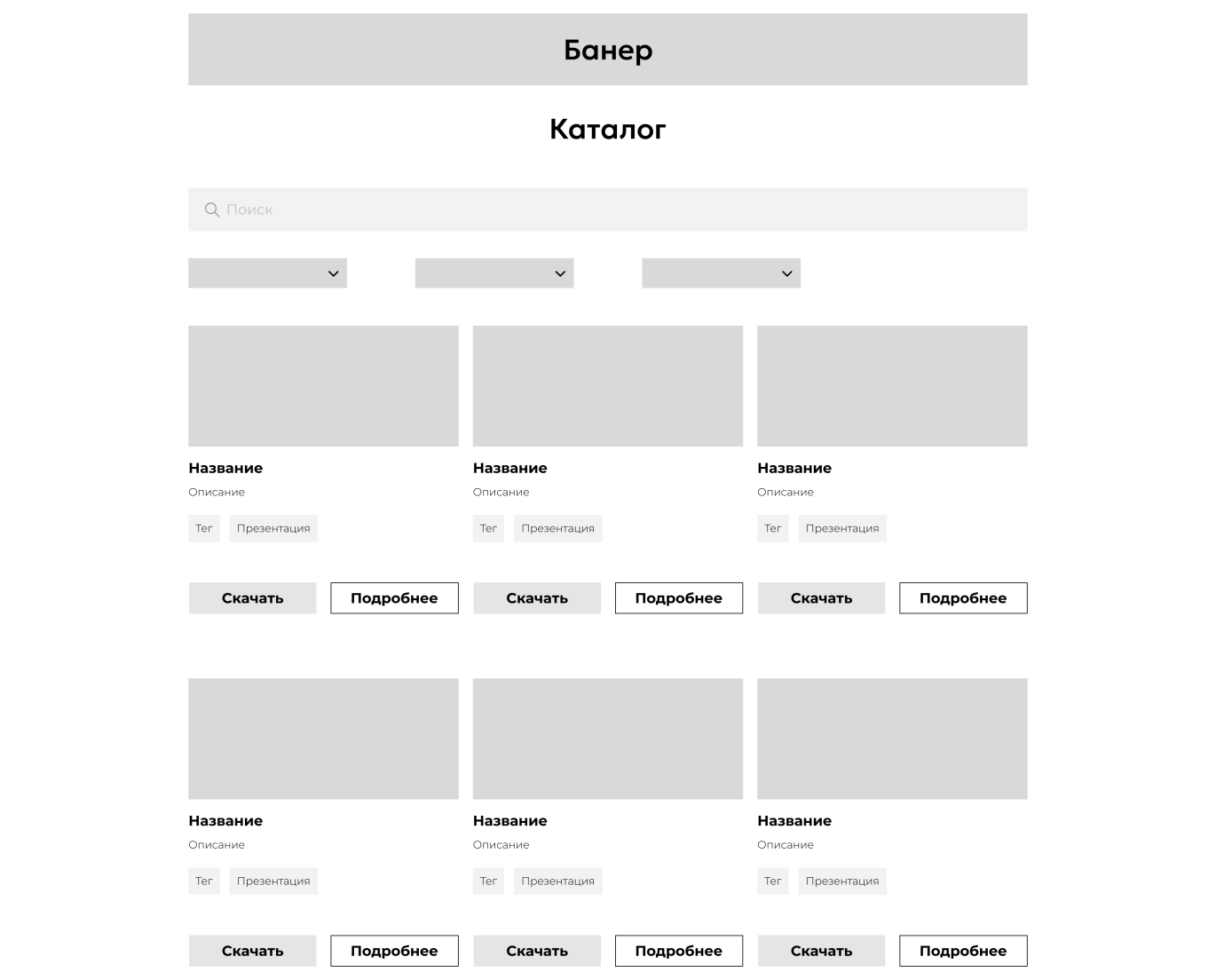


Рисунок 8 – Эскиз страницы «Каталог»

А на рисунке 9 изображен эскиз страницы, где хранятся загруженные личные разработки пользователя, так же можно будет увидеть все скаченные пользователем ресурсы.



Рисунок 9 – Эскиз страницы «Ваши разработки»

На рисунке 10 изображен эскиз контента страницы готового ресурса, где представлена карточка разработки, ее описание, написанное самим пользователем, категории и теги для поиска и так же кнопка для скачивания файла.

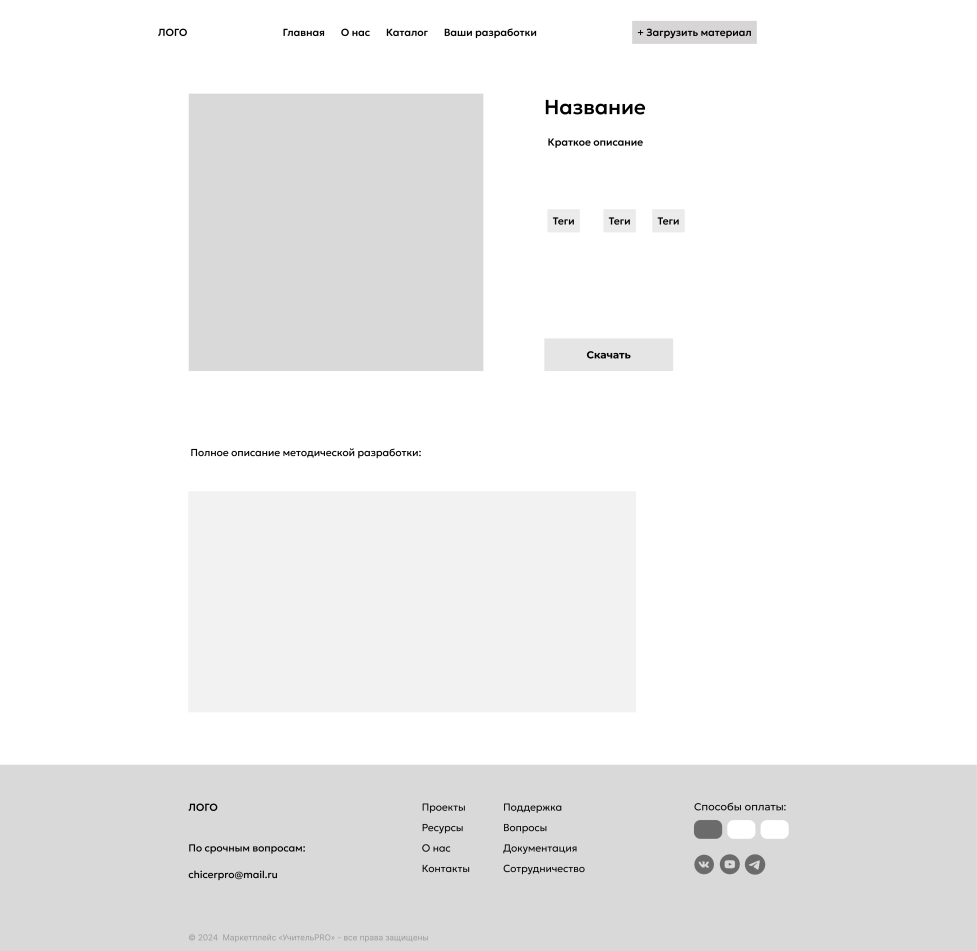


Рисунок 10 – Эскиз страницы «Страница продукта»

Таким образом, эскиз дизайна веб-сайта является важным инструментом в процессе разработки, который помогает определить концепцию, функционал и внешний вид будущего веб-приложения.

При работе с системой выделены 2 основных вида пользователей: обычные пользователи (авторизованные и неавторизованные) и администраторы.

Обычные пользователи (учителя и ученики):

* доступ к основному контенту, включая каталог, загрузку материалов и личный кабинет;
* возможность просмотра и скачивания материалов.

Администраторы:

* полный доступ к функционалу управления платформой, включая администрирование пользователей и контента;
* возможность модерации загруженных материалов и обработки отзывов.

Так же доступные функции зависят от того, авторизован пользователь или нет, у неавторизованных пользователей есть возможность просматривать каталог и даже скачивать ресурсы, но не будет возможности публиковать личные разработки.

Для каждого вида пользователя будет создана своя конкретная роль в системе: «администратор» и «пользователь» [2].

Основными формами взаимодействия пользователя с системой на первом этапе будут формы авторизации и регистрации. Их эскизы представлены на рисунках 11 и 12.

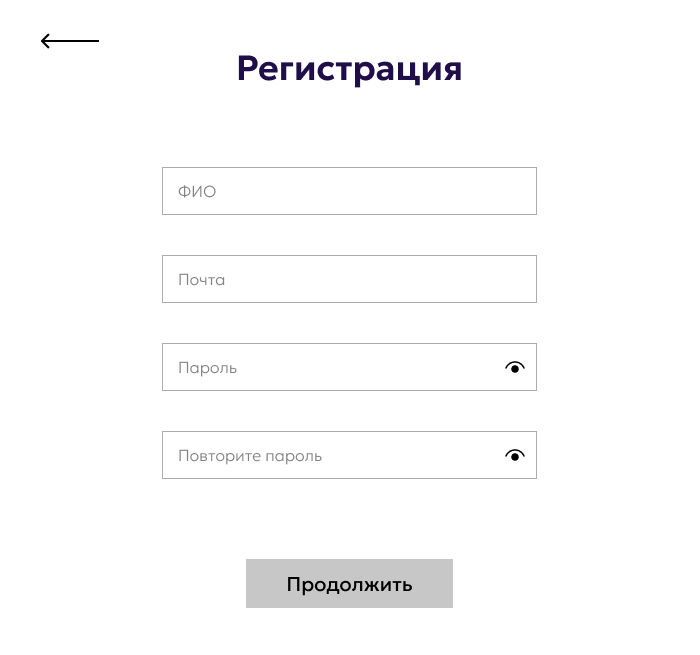


Рисунок 11 – Эскиз формы регистрации

Эти формы, предоставляют доступ к персонализированным функциям и защищенному контенту платформы.



Рисунок 12 – Эскиз формы авторизации

На втором этапе основная форма взаимодействия будет связана с личный кабинетом и будет отличаться в зависимости от роли.

На рисунке 13 представлен эскиз формы, где пользователь может загрузить свои разработки, оформить ресурс для удобного поиска, выбрать категорию и тег, выбрать привлекательную обложку и информативное описание.

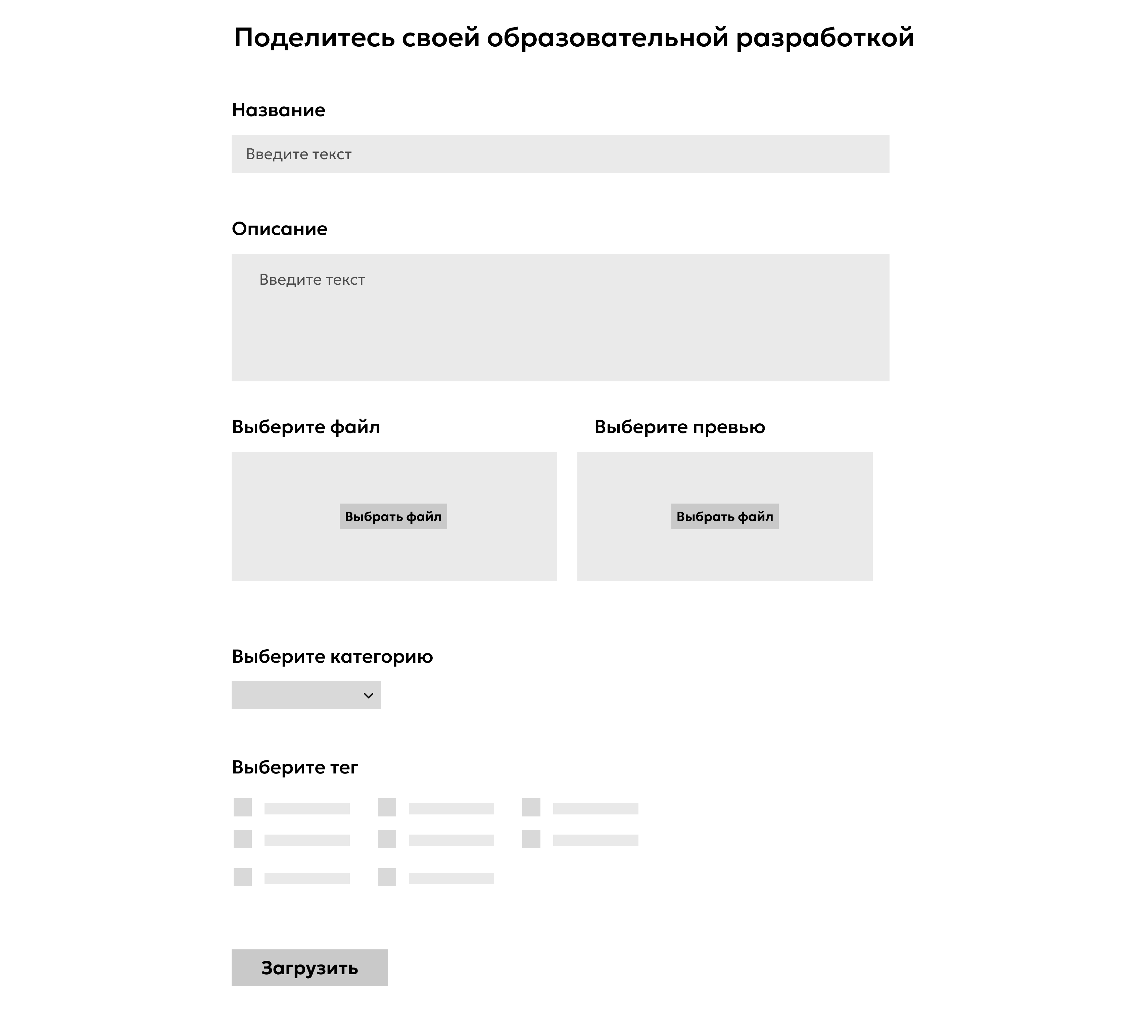


Рисунок 13 – Эскиз формы загрузки материала

Рисунок 14 иллюстрирует панель администратора, которая предоставляет функционал для управления контентом и пользователями веб-платформы. В этой панели можно добавлять, менять и удалять теги и категории.

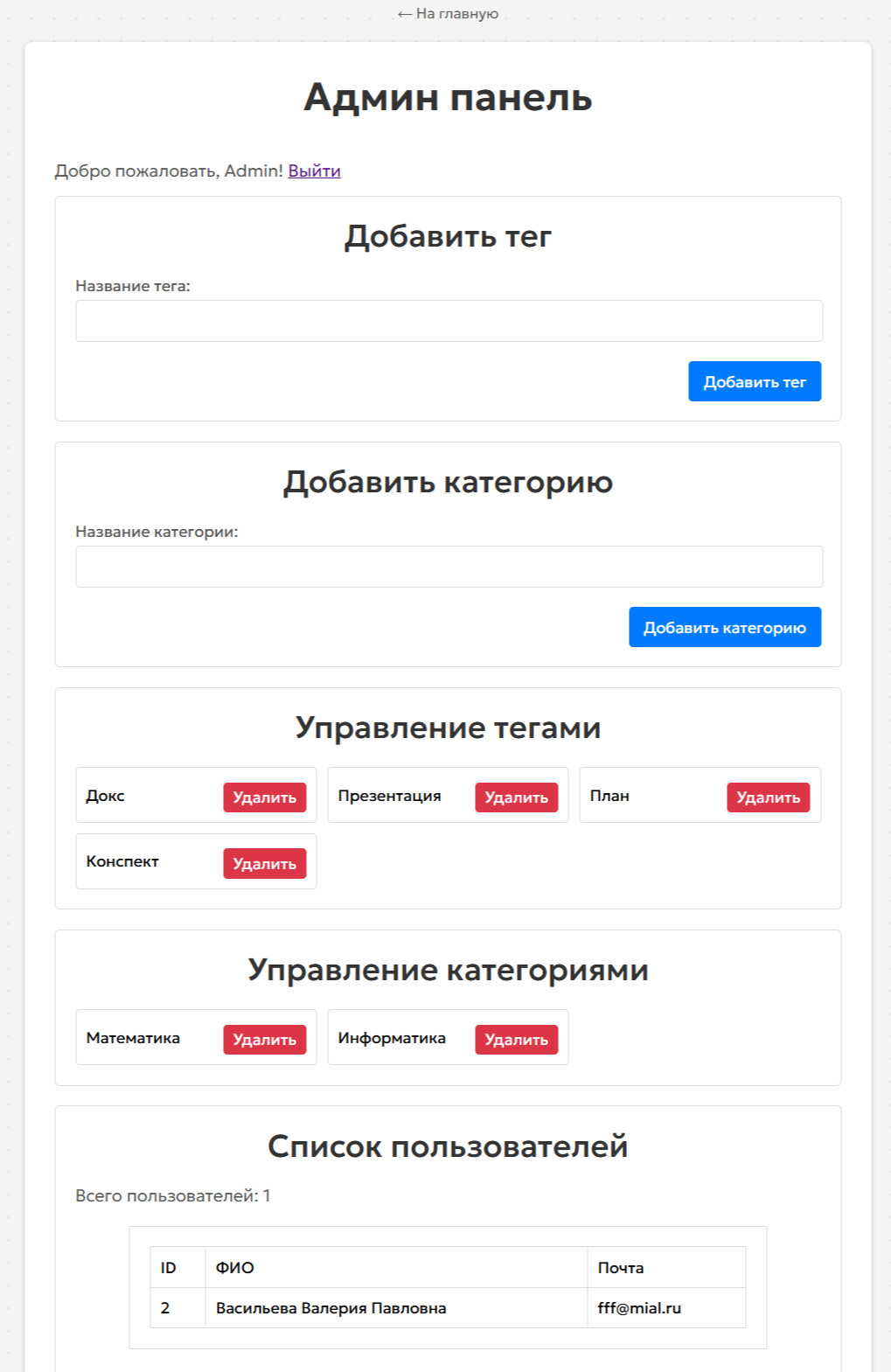


Рисунок 14 – Формы на странице администратора

На рисунке 15 представлены формы в панели администратора, которые отвечают за удаление категорий и тегов. Также ниже отображается все зарегистрированные пользователи.

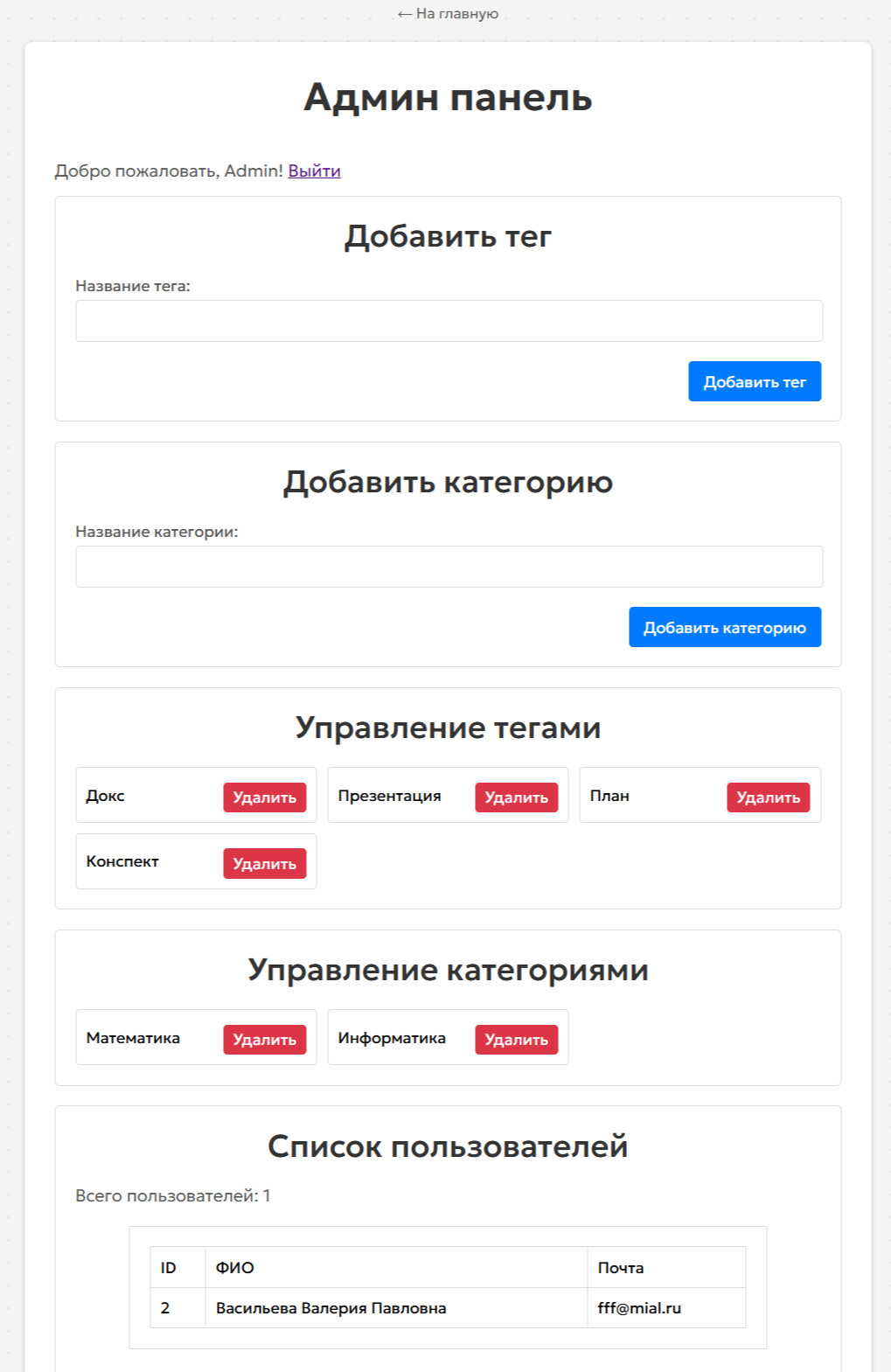


Рисунок 15 – Продолжение страницы администратора

Модели пользовательского интерфейса для веб-приложения «УчительPRO» разрабатываются на основе ролей пользователей и их функций. Интерфейс должен быть удобным и эффективным для образовательного процесса.

**3.4 Подбор графических материалов**

Подбор графических материалов – это процесс отбора, выбора и использования различных изображений, иллюстраций, фотографий или других графических элементов для создания дизайна, рекламы, веб-сайтов или других видов визуальных материалов.

Логотип сайта «УчительPRO», показанный на рисунке 16 создавался в программе для дизайна Figma и представляет собой стилизованную звёздочку, вписанную в круг, что символизирует уверенность и лёгкость в решении даже самых сложных педагогических задач. Звезда в этом контексте символизирует знания, достижения, стремление к совершенству и сложность, ассоциируясь с «заданиями повышенной сложности», знакомыми учителям. Она становится символом ценности ресурса «УчительPRO», который помогает справиться с любой, даже самой сложной задачей, делая процесс обучения проще и эффективнее.

Круг, в свою очередь, символизирует безопасность, защиту, единство, бесконечность и завершённость, подчёркивая цикличность процесса обучения. Он обнимает звезду, что символизирует поддержку сайта в преодолении трудностей и повышении эффективности работы учителей, чтобы обучение было гармоничным и полным.

Название «УчительPRO» размещено рядом с логотипом, создавая целостную композицию и подчёркивая взаимосвязь визуального образа и названия. Цветовая гамма логотипа состоит из сине-фиолетового сочетания, которое создаёт ощущение спокойствия, доверия, профессионализма, интеллекта и креативности. Фиолетовый цвет символизирует творчество, но стабильностью и достоверностью – качествами, присущими профессиональному учителю, а также интуицию и мистику, намекая на то, что на сайте «УчительPRO» можно найти необычные решения и инновационные подходы к обучению.

Для тёмного фона предусмотрен белый вариант логотипа, который создаёт контраст и делает его более заметным и читабельным.

Рисунок 16 – Логотип сайта «УчительPRO»

В целом, логотип выполнен в лаконичном и содержательном стиле, отражая ценности сайта и его направленность. Он явно указывает на то, что «УчительPRO» – это не просто ресурс, а настоящий помощник для учителей, который помогает сделать обучение более эффективным и увлекательным.

Для сайта были выбраны следующие иконки, представленные на рисунке 17:

* иконка лупы для поиска, что обеспечивает наглядность и играет роль кнопки;
* иконки-стрелочки, обозначающие направление движения, например, «Развернуть список», они помогают пользователям понять, какие действия они могут выполнить при нажатии на эти кнопки;
* меню-бургер для мобильной версии;
* иконки популярных социальных сетей и мессенджеров, таких как YouTube, Вконтакте, Телеграм они помогут пользователям быстро перейти на соответствующие страницы сайта в социальных сетях.

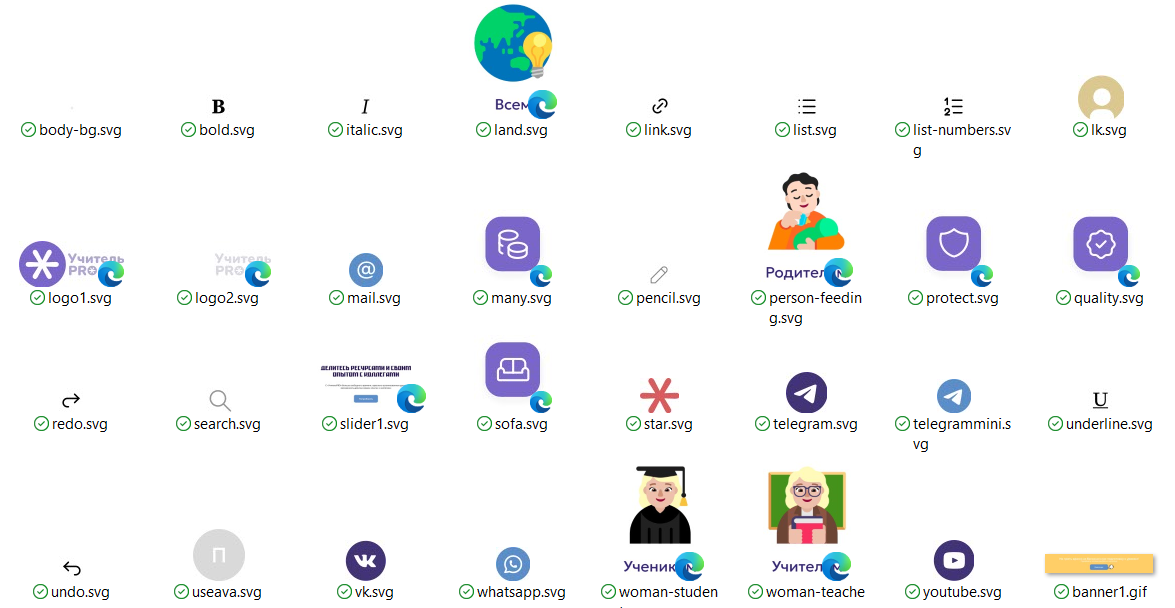


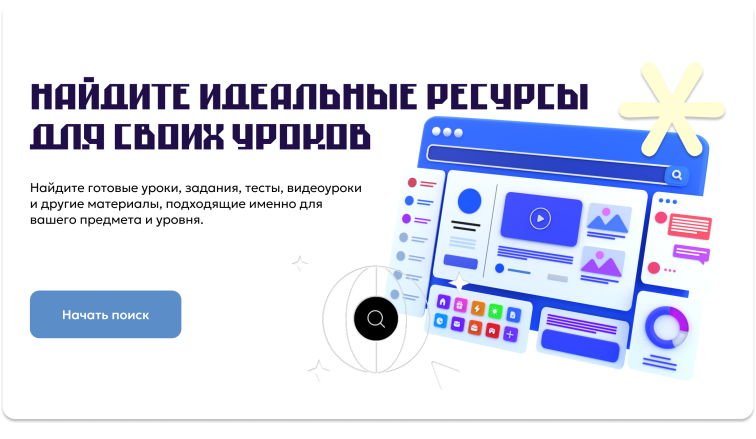
Рисунок 17 – Иконки

Эти иконки не только облегчают взаимодействие пользователя с сайтом, но и делают его более понятным и удобным для использования.

Для создания баннеров в слайдере на главной странице сайта, показанных на рисунке 18, использовался подход, основанный на комбинации готовых 3D-изображений и GIF-анимации.

Первым шагом стал поиск готовых 3D-изображений, соответствующих тематике сайта «УчительPRO». Были выбраны изображения образовательных объектов, а также предметов обучения, включая книги, учебники, глобусы и лабораторное оборудование. Важным аспектом стали 3D-модели учителей, изображённые в ситуациях обучения.

На втором этапе была проведена работа по комбинированию 3D-изображений. В некоторых баннерах использовались GIF-анимации для добавления динамики и оживления композиций. Завершающим этапом стало добавление GIF-изображений.



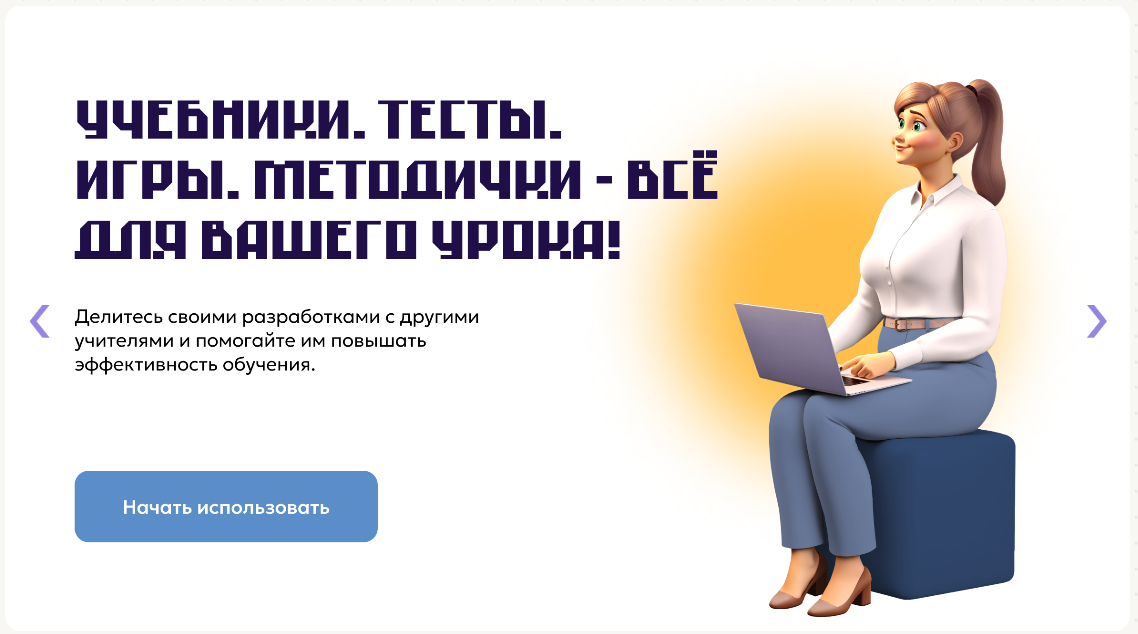


Рисунок 18 – Изображения для баннеров в слайдере

Из найденных материалов была создана единая композиция, что позволило сформировать динамичный и привлекательный визуальный ряд для баннеров.

Дальше на рисунке 19 на главной странице изображена композиция из стокового изображения учителя и вручную отрисованной, с помощью фигур, папки для документов. Такие изображения делают сайт визуально привлекательным.

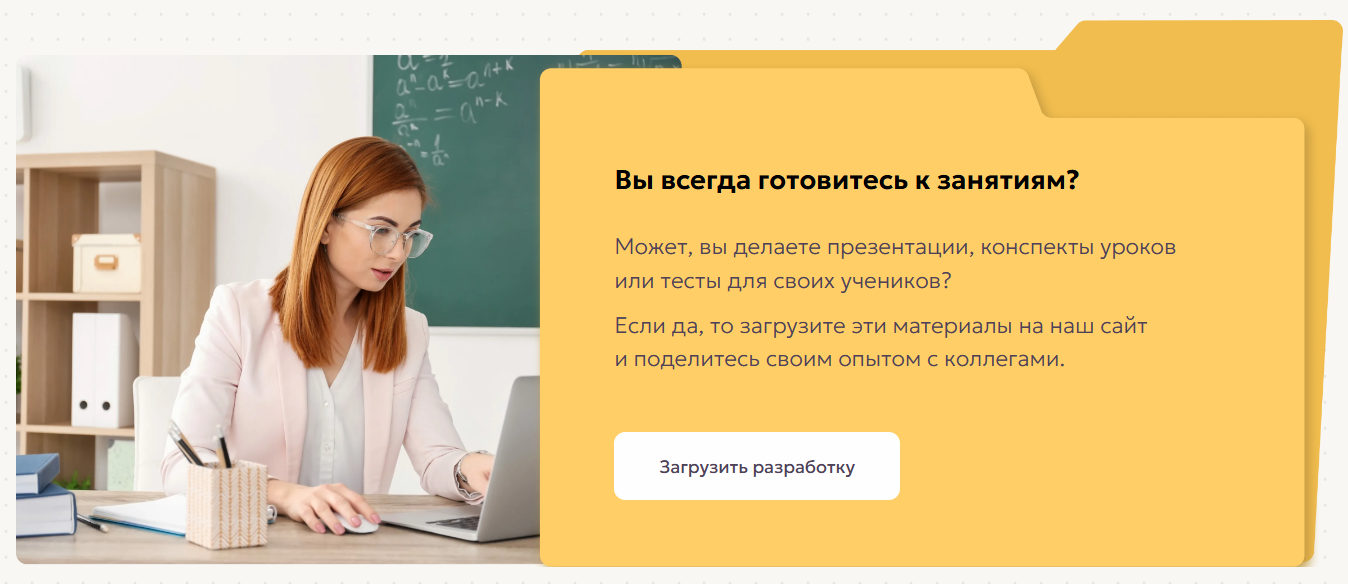


Рисунок 19 – Изображение для главной страницы

На рисунке 20 показаны сгенерированные изображения, которые дополнительно обработаны для улучшения качества, изменения размеров или добавления элементов, соответствующих стилю сайта. Готовые изображения были интегрированы для страницы «О нас» для разбавления текста.



Рисунок 19 – Изображения истории «О нас»

Использование нейросетей для создания изображений позволило сэкономить время и ресурсы, а также обеспечить высокое качество визуального контента, который соответствует тематике ресурсов на сайте.

В баннере для каталога, представленного на рисунке 16, использована анимация в Figma, реализованная как бегущая строка с продающей фразой. Анимация экспортирована в GIF-формат с помощью плагина.

Его цель – привлечь внимание к каталогу и побудить пользователей пробрести разработки. GIF-анимация привлекает внимание и выделяется на фоне статических элементов страницы, а бегущая строка позволяет разместить краткий, но ёмкий продающий текст.

В баннер для страницы «О нас», который также изображён на рисунке 20, внедрено готовое GIF-изображение с призывом к действию. Его цель побудить пользователей перейти в каталог после прочтения страницы «О нас». На баннере присутствует кнопка «В каталог», которая является частью GIF-изображения, а весь баннер является кликабельным элементом, ведущим на страницу каталога.

GIF-изображение с призывом к действию и кнопкой «В каталог» максимально понятно и эффективно побуждает пользователей к переходу в каталог.

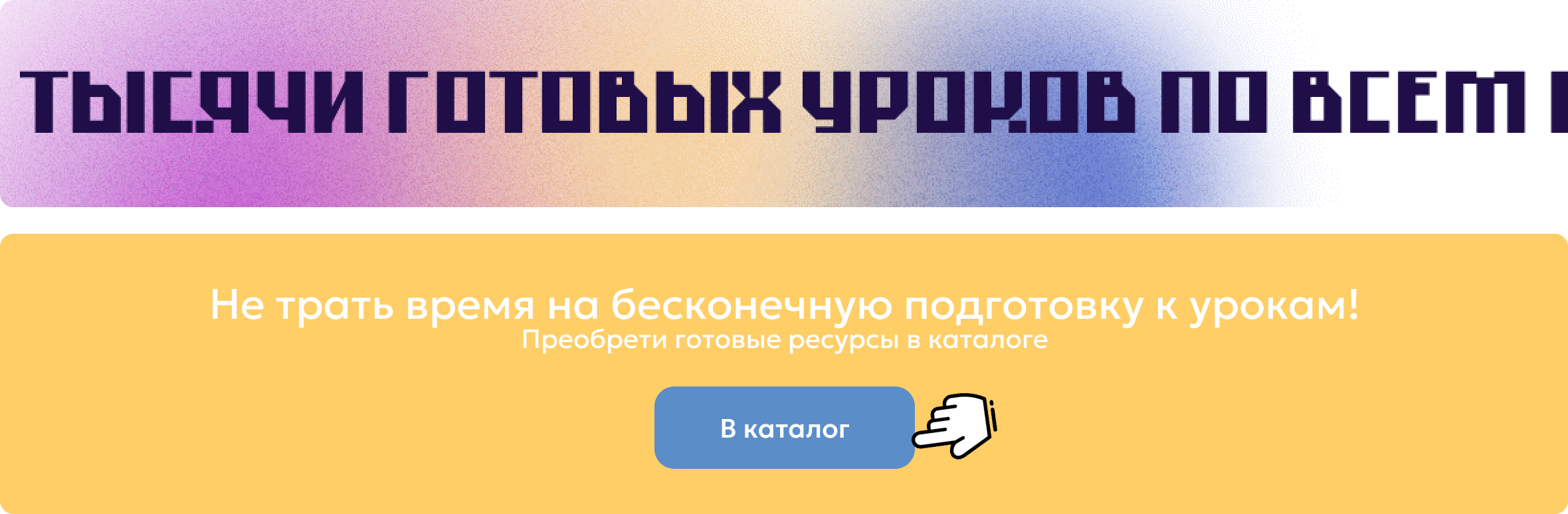


Рисунок 20 – Анимированные баннеры

Оба баннера созданы с целью привлечения внимания пользователей к каталогу. Баннер для каталога привлекает внимание своей динамичностью и информирует пользователей о содержании каталога.

Для создания визуальной составляющей сайта были использованы современные технологии и инструменты, такие как 3D-изображения, GIF-анимация и Figma, что позволило создать привлекательные и динамичные визуальные элементы, отражающие концепцию сайта и привлекающие внимание пользователей.

В целом, выбор изображений основан на стремлении привлечь внимание зрителей, передать эмоциональное содержание и создать единое визуальное впечатление.

**4 Разработка веб-приложения**

**4.1. Создание шаблонов страниц веб-приложения**

Создание шаблонов веб-страниц – это процесс разработки универсального дизайна и структуры страницы, который можно применять к нескольким страницам схожего содержания. Шаблон включает в себя разметку, стили, элементы интерфейса и другие компоненты, которые остаются неизменными на всех страницах, созданных на основе данного шаблона.[9]

В начале определяем основные разделы страницы: шапка, навигационное меню, секции с контентом и подвал.

В шапке обычно размещают логотип сайта, контактную информацию, ссылки на социальные сети и другие важные элементы.

Навигационное меню должно быть легко доступным и помогать пользователям перемещаться по сайту.

Контент должен быть разделен на логические блоки или секции, например, каталог, о нас, разаработки и т.д. Каждая секция должна иметь чёткую структуру и визуальное оформление.

Подвал обычно содержит дополнительные ссылки, контактную информацию, копирайт.

Дополнительные элементы, такие как формы обратной связи, карта сайта, поиск и другие виджеты, также могут быть добавлены в структуру страницы в зависимости от потребностей сайта.

В коде шаблонов представлены основные элементы, которые будут использоваться на протяжении всей вёрстки сайта.

**«section» – э**то HTML-элемент, который используется для группировки контента на странице. В данном случае, каждый «section» представляет собой отдельный блок на главной странице, а все секции сгруппированны в тег «main».

**«script src=script.js» э**тот код подключает внешний JavaScript файл script.js. Он будет использоваться для реализации функционала слайдера.

HTML-элемент «**footer»**, который используется для добавления информации в нижней части страницы.

Блок «**div class=logo»** будет содержать логотип сайта, который находится в блоке «**div class=nav»** содержащем навигационное меню.

Для обеспечения единообразия и удобства поддержки интерфейса в проекте реализуются общие шаблоны элементов страниц – header (шапка) и footer (подвал), представленных на рисунках 21 и 23, которые вынесены в отдельные файлы-части (partials). Это позволяет централизованно управлять внешним видом и содержимым данных элементов.

Шаблоны подключаются на каждой странице с помощью специальных команд в шаблонизаторе (например, EJS):

* <%- include('partials/header', {user: user}) %>;
* <%- include('partials/footer') %>.

Стоит обратить внимание, что при подключении header в него передается объект user, что позволяет динамически изменять содержимое шапки в зависимости от состояния авторизации и роли пользователя. Например, для неавторизованных гостей отображаются одни элементы, а для авторизованных пользователей – дополнительные ссылки, в зависимости от роли (педагог, студент, администратор).

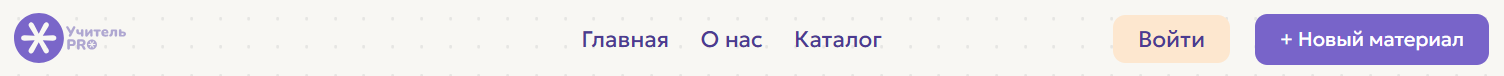


Рисунок 21 – Шаблон «шапки» сайта

Данный шаблон хедера реализован фрагментом кода, изображённом на рисунке 22.



Рисунок 21 – Код реализации шаблона «шапки» сайта

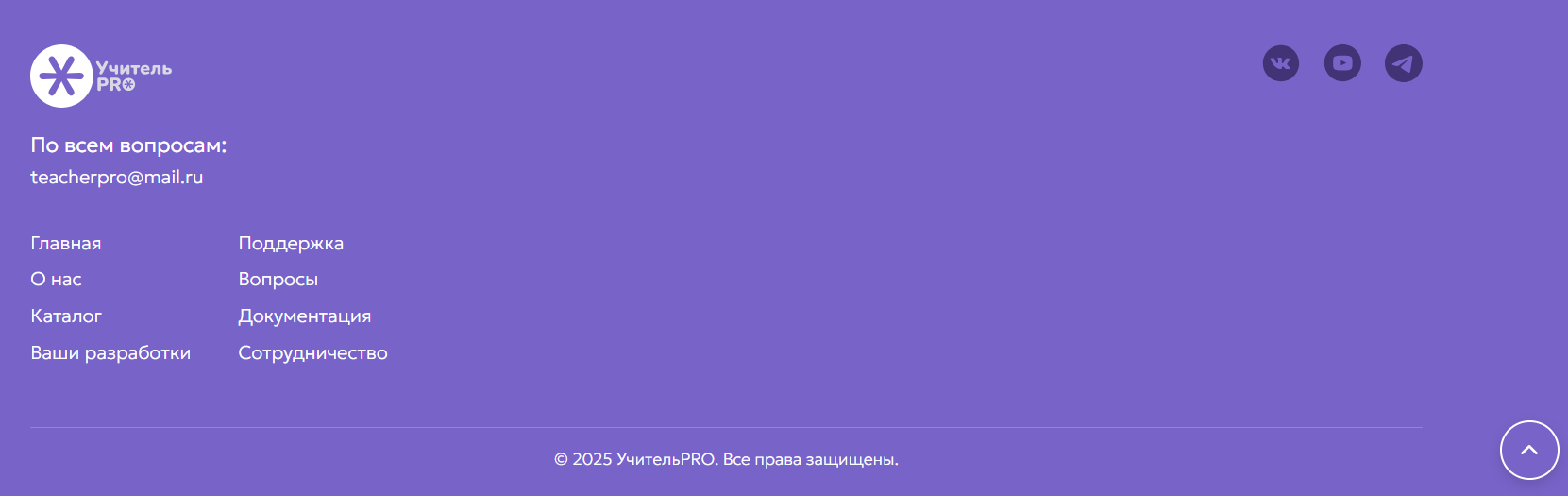


Рисунок 23 – Шаблон «подвала» сайта

Данный шаблон футера реализован кодом, изображённом на рисунке 24.



Рисунок 24 – Код реализации шаблона «подвала» сайта

В каталоге ресурсов для повышения удобства и визуальной однородности используются повторяющиеся компоненты, которые отображают краткую информацию о каждой разработке, пример карточки разработки представлен на рисунке 25.

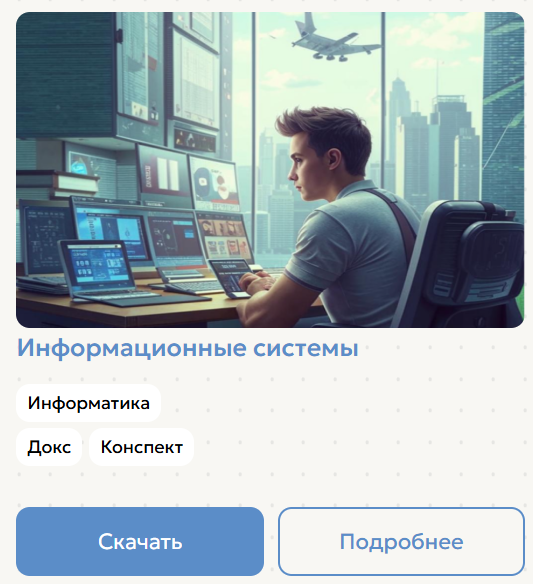


Рисунок 25 – Шаблон карточки разработки в каталоге

Данный шаблон карточки реализован фрагментом кода, изображённом на рисунке 26.



Рисунок 26 – Код реализации шаблона карточки

Детальная страница разработки, изображённая на рисунке 27, строится по похожей логике: она использует отдельный шаблон, в который передаются полные данные конкретного ресурса, что позволяет отображать расширенную информацию, описание, файлы другую связанную информацию.



Рисунок 27 – Шаблон детальной карточки

Данный шаблон реализован фрагментом кода, изображённом на рисунке 28.



Рисунок 28 – Код реализации шаблона подробной карточки

**4.2 Реализация веб-приложения**

Первоначально, для пользователей важна безопасность данных: пароли пользователей должны передаваться по защищенному каналу связи («HTTPS») и храниться в базе данных в зашифрованном виде. Для этой цели используется криптографические хеш-функции с солью, такие как «bcrypt», которые делают невозможным восстановление исходного пароля даже в случае утечки базы данных, пример отображения в базе данных представлен на рисунке 29.

Помимо защиты паролей, стоит обратить внимание на безопасность самого приложения. Регулярное тестирование на уязвимости и обновление используемых библиотек помогают минимизировать риски.

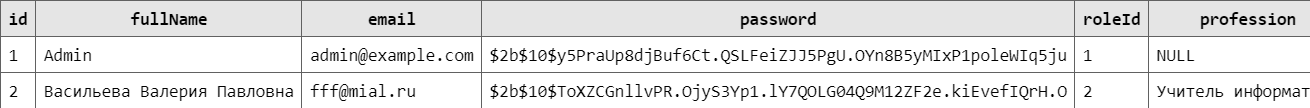


Рисунок 29 – Демонстрация хеширования пароля

Разделение ролей позволяет назначать пользователям различные уровни доступа к функциональности приложения. Реализация разделения ролей начинается с определения набора ролей и соответствующих им прав доступа. Затем в процессе регистрации пользователю может быть назначена определенная роль.

При авторизации информация о роли пользователя сохраняется, и дальнейшая работа приложения строится с учетом этой роли, определяя доступные пользователю функции и данные. Код реализации проверки разделения ролей представлен на рисунке 30.



Рисунок 30 – Код реализации проверки разделения ролей

При разработке форм регистрации и авторизации ключевым аспектом является валидация данных, подразумевающая проверку вводимой пользователем информации на соответствие заданным критериям. Это включает в себя проверку формата адреса электронной почты на наличие символа «@» и соответствия повторно введенного пароля. Шаблон формы регистрации представлен на рисунке 31.

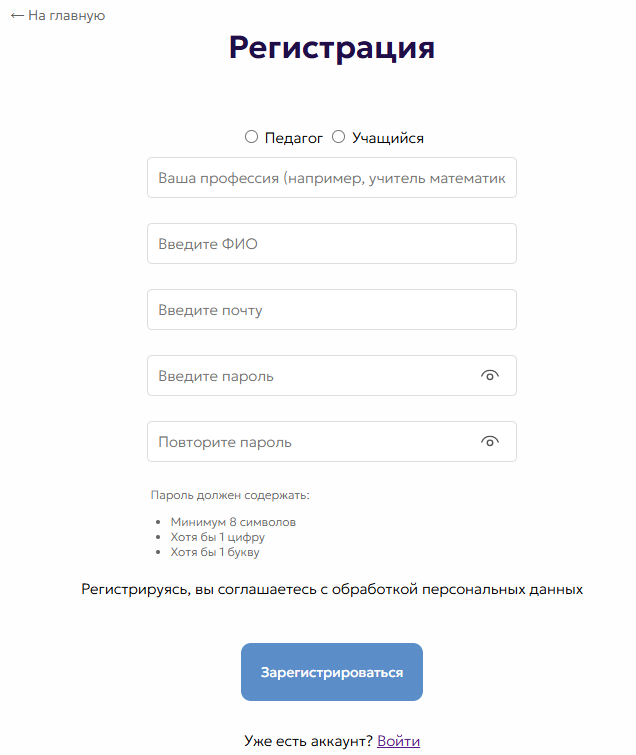


Рисунок 31 – Форма регистрации

На рисунке 32 представлен код реализации отображения разработок в личном кабинете пользователя.

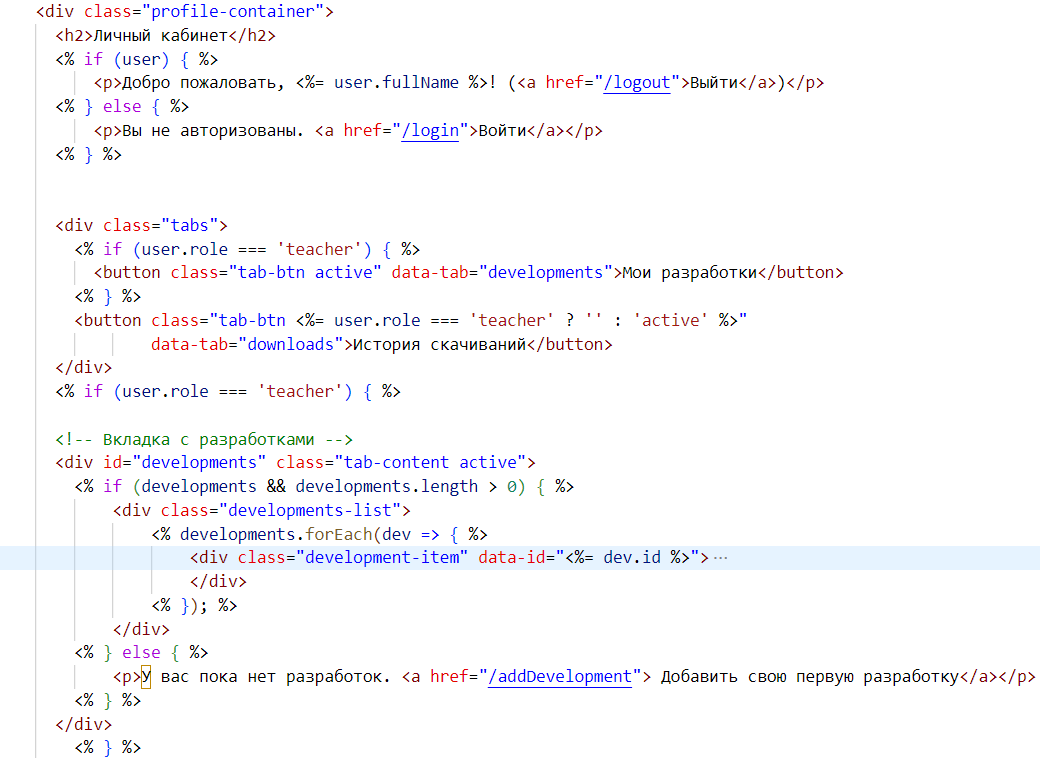


Рисунок 32 – Код реализации отображения разработок в личном кабинет

е

Рисунок 33 демонстрирует реализацию макета личного кабинета, где представлены основные разделы, позволяющие пользователям просматривать свои ресурсы.

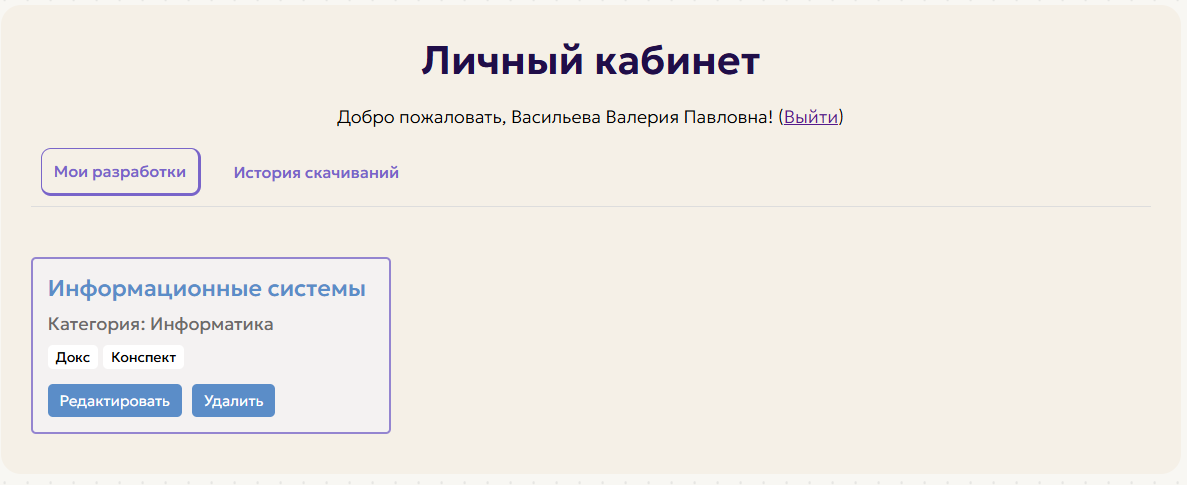


Рисунок 33 – Реализация макета личного кабинета

На рисунке 34 представлен код реализации отображения контента в каталоге, где пользователи могут видеть доступные образовательные ресурсы и информацию о них. Этот рисунок акцентирует внимание на удобстве навигации и возможности быстрого поиска.



Рисунок 34 – Код реализации отображения каталога

На рисунке 35 представлена структура и дизайн пользовательского интерфейса, который обеспечивает доступ к образовательным ресурсам. Он включает в себя: функцию поиска, фильтры и сортировку, список ресурсов.

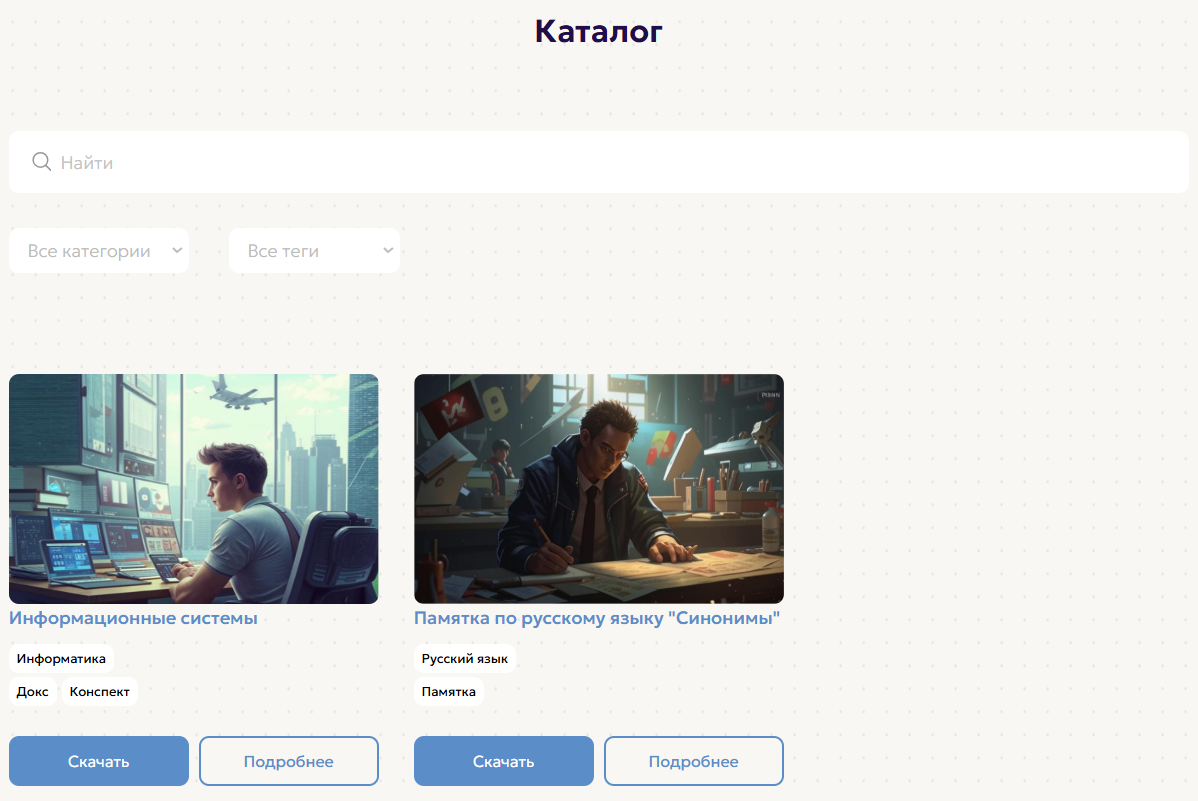


Рисунок 35 – Реализация макета каталога

Учитывая, что веб-приложение может открываться через мобильное устройство, все страницы сайта должны быть адаптированы и корректно отображаться на телефонах. На рисунке 36 представлен каталог в мобильном варианте.

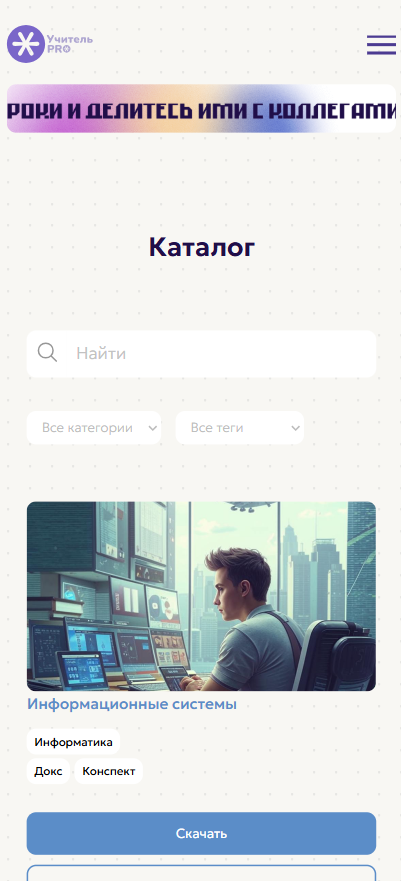


Рисунок 36 – Реализация мобильной версии каталога

На рисунке 37 представлен фрагмент кода добавления ресурса через форму, он структурирован для обработки и валидации данных.



Рисунок 37 – Код реализации добавления ресурса через форму

Рисунок 38 демонстрирует форму для загрузки ресурсов с текстовыми полями, кнопками и инструкциями.

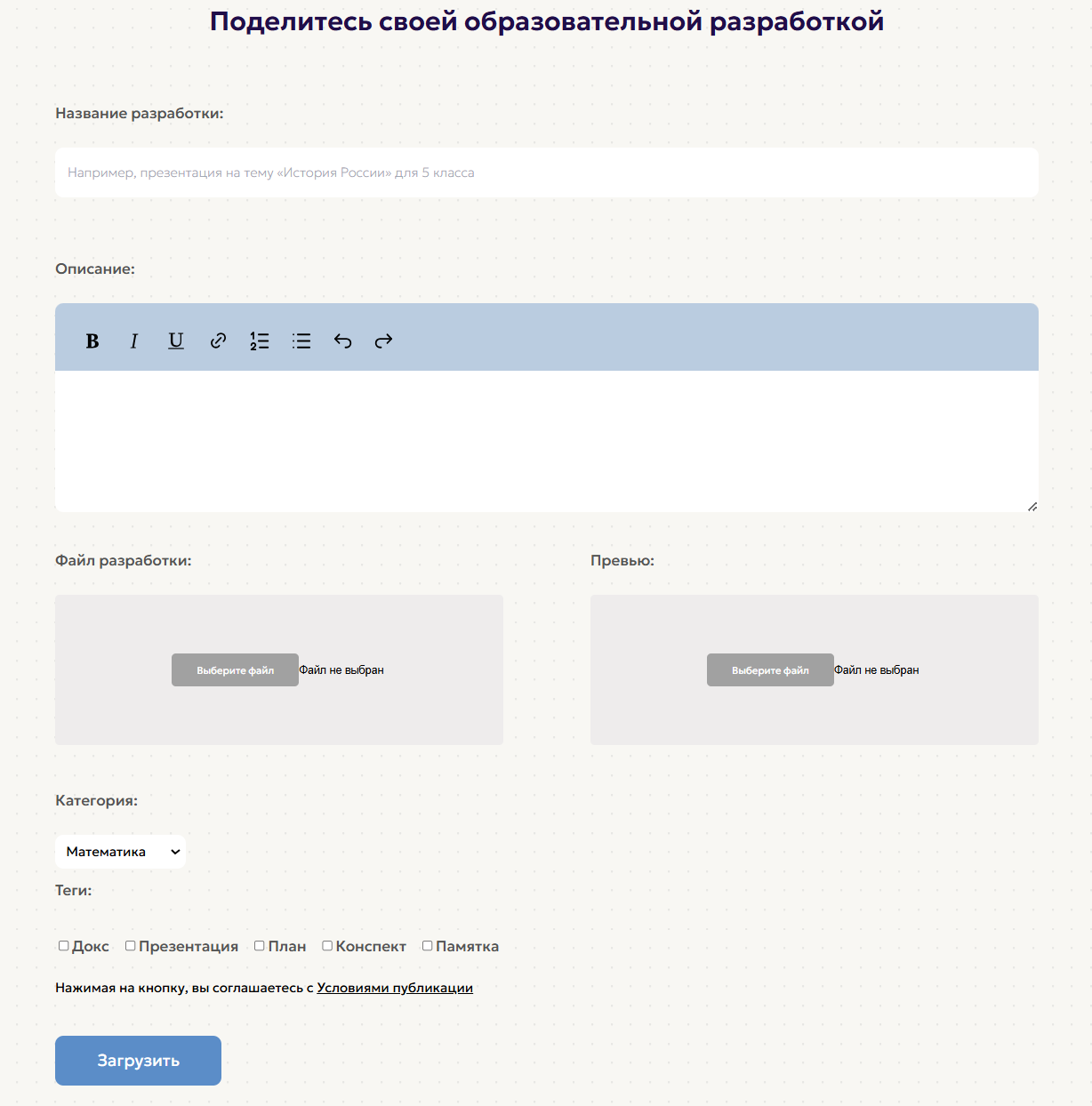


Рисунок 38 – Реализация макета формы для загрузки ресурсов

**4.3 Реализация динамических элементов сайта**

Динамические элементы сайта – это те, которые изменяются в зависимости от действий пользователя или внешних факторов. Они делают сайт более интерактивным и привлекательным.

Для создания динамических анимаций и эффектов на странице HTML был использован JavaScript или простейшие анимации в CSS. [6]

Слайдер на главной странице представляет собой динамический элемент интерфейса, который демонстрирует набор изображений с gif-изображениями в циклическом режиме, он представлен на рисунке 39. Он обеспечивает пользователям возможность навигации между слайдами с помощью кнопок или стрелок. Каждый слайд содержит не только визуальный контент, но и текстовые элементы, такие как подписи и заголовки, что позволяет сделать информацию более доступной и привлекательной.

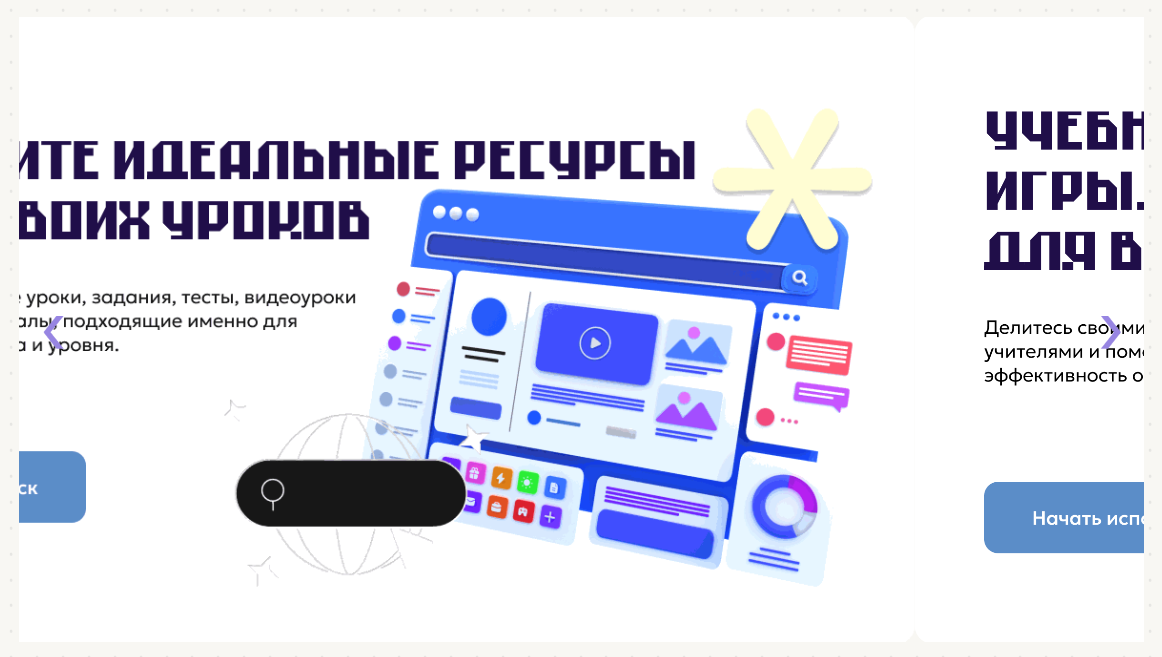


Рисунок 39 – Реализация слайдера на сайте

Для реализации слайдера была выбрана одна из популярных библиотек, таких как Slick, что обеспечивает высокую производительность и гибкость, код представлен на рисунке 40. В HTML-коде был создан контейнер для слайдера с определенным классом, который отвечает за стилизацию. Внутри этого контейнера размещены отдельные слайды, каждый из которых включает в себя изображения, а также текстовые элементы.



Рисунок 40 – JS для реализации слайдера

На стороне JavaScript происходит инициализация выбранной библиотеки слайдера, настройка параметров, таких как количество отображаемых слайдов, тип анимации и режим автовоспроизведения. Также добавлены обработчики событий для кнопок навигации и свайпа.

Одним из важных элементов адаптивного интерфейса является динамическое мобильное меню-бургер, которое обеспечивает удобную навигацию на мобильных устройствах, оно представлено на рисунке 41.



Рисунок 41 – Реализация меню на мобильном устройстве

Динамическое меню реализовано с помощью JavaScript, что позволяет автоматически показывать или скрывать его при взаимодействии пользователя. В основном, оно встроено в шаблон хедера и управляется с помощью добавления или удаления CSS-классов, обеспечивающих его отображение. На рисунке 42, представлен скрипт для работы меню в мобильной версии.



Рисунок 42 – Скрипт реализации меню на мобильном устройстве

**5 Отладка и тестирование веб-приложения**

**5.1 Ревьюирование программного кода**

**5.2 Тестирование веб-приложения**

Тестирование веб-приложения – это процесс проверки функциональности, производительности, безопасности и других аспектов веб-приложения для обеспечения его качества и корректной работы. Цель тестирования веб-приложения заключается в выявлении ошибок, недочетов и потенциальных проблем, которые могут повлиять на пользовательский опыт или безопасность приложения.

Для тестирования юзабилити веб-приложения «УчительPRO» была выбрана методика тест-кейсов. Этот выбор обусловлен следующими факторами: структурированность, так как методика позволяет систематизировать процесс тестирования, четко определяя цели, шаги и ожидаемые результаты; повторяемость, поскольку тест-кейсы можно легко воспроизвести, что позволяет сравнивать результаты тестирования, проводимого разными группами пользователей.

Процесс тестирования веб-приложения включает в себя создание тест-кейсов (наборов шагов для проверки определенной функциональности), выполнение тестов, регистрацию обнаруженных проблем и их отслеживание до устранения. Тестирование веб-приложения также может включать в себя проверку совместимости с различными браузерами и устройствами, а также оценку производительности приложения при различных нагрузках.

В таблице 1 представлена информация о процессе проверки корректности регистрации нового пользователя. Данная таблица служит инструментом для анализа и оценки всех необходимых параметров, обеспечивающих успешную регистрацию пользователя в системе, что критически важно для обеспечения безопасности и удобства работы с веб-приложением.

Таблица 1 – Проверка корректной регистрации нового пользователя

|  |  |  |  |
| --- | --- | --- | --- |
| №1 | Проверка корректной регистрации нового пользователя | | |
| Ссылка на ТЗ | Васильева В. П. | Высокий | Регистрация |
| 1. Ввести в поля ФИО и почту корректные данные (например, «Анна Петрова», «anna.petrova@mail.ru»). 2. Ввести пароль (например, «Пароль123»). 3. Ввести пароль повторно для проверки. 4. Нажать кнопку «Зарегистрироваться». | | Ожидаемый результат:  Добавление данных нового пользователя в базу данных и появление окна авторизации для входа в личный профиль. | Фактический результат:   * данные добавлены в базу данных; * появилась форма авторизации; * пользователь может авторизироваться. |

На рисунке 27 представлена работа формы регистрации пользователя, на которой видно, как пользователь вводит свои личные данные. В форме предусмотрены поля для заполнения, включая ФИО, адрес электронной почты и пароль. Пользователь также должен повторно ввести пароль для проверки на совпадение, что обеспечивает дополнительный уровень безопасности.

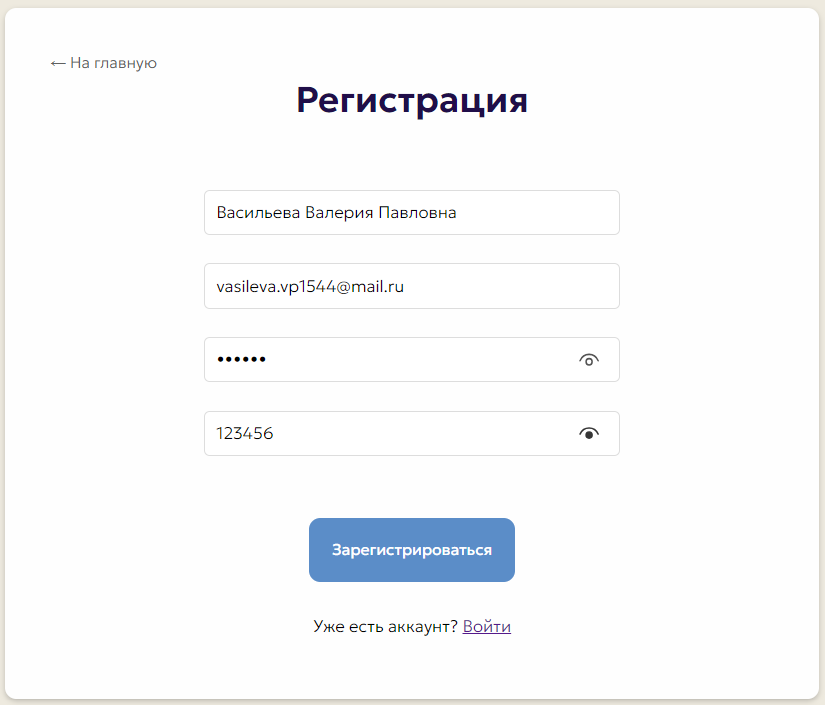


Рисунок 27 ‒ Работа формы регистрации пользователя

На рисунке 28 представлена форма авторизации, которая появляется после первоначального обращения на сайт. В этой форме пользователь имеет возможность ввести ранее зарегистрированные данные, включая логин и пароль, что позволяет получить доступ к личному кабинету.

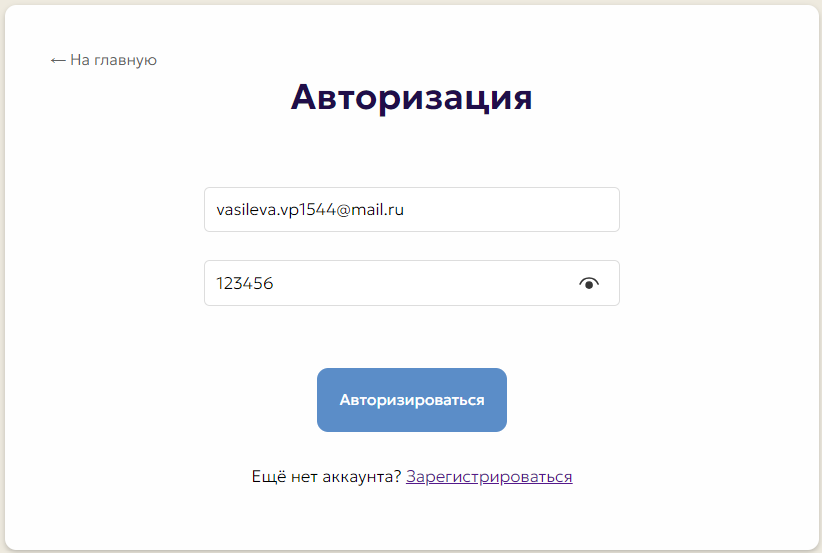


Рисунок 28 ‒ Форма авторизации

Рисунок 29 демонстрирует раздел отчетности о пользователях в административной панели. После регистрации, как упоминалось ранее, данные пользователя сохраняются в базе данных и используются для авторизации. На этом рисунке видно, как эти данные представлены администратору, предоставляя информацию о зарегистрированных пользователях.

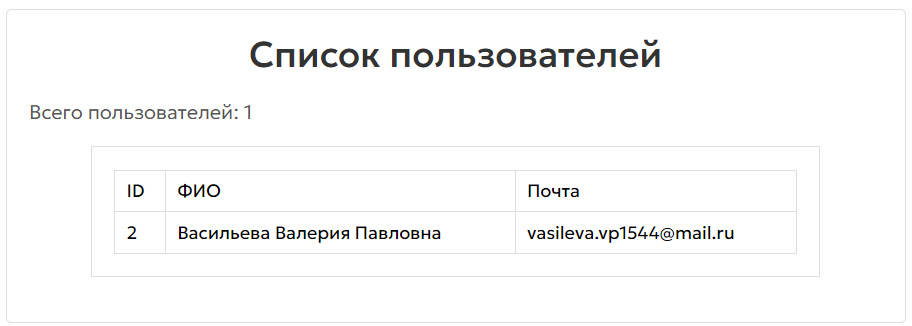


Рисунок 29 ‒ Отчетность о пользователях в админ-панели

В таблице 2 представлена информация о процессе проверки формы добавления разработки через администратора.

Таблица 2 – Проверка формы добавления разработки через администратора

|  |  |  |  |
| --- | --- | --- | --- |
| №2 | Проверка формы добавления разработки через администратора | | |
| Ссылка на ТЗ | Васильева В. П. | Высокий | Добавление разработок |
| 1. Подготовить тестовые данные: файл в поддерживаемом формате (например, PDF) и изображение для обложки разработки, и заполнить форму публикации (название, описание, категории и теги). 2. Загрузить файлы и нажать кнопку «Загрузить». 3. Выбрать теги. 4. Подтвердить загрузку. | | Ожидаемый результат:  Материал успешно загружается и отображается в каталоге опубликованных разработок, загруженный файл можно скачать, а данные корректно отображаются в базе данных и используются для формирования каталога. | Фактический результат:   * данные добавлены в базу данных; * теги не добавились в базу данных и не взаимодействуют с каталогом; * карточка разработки корректно отображается в каталоге (превью, заголовок, категория и кнопки); * загруженный файл успешно скачивается. |

Рисунок 30 демонстрирует работу формы для публикации образовательной разработки. На изображении представлены все ключевые элементы интерфейса, включая поля ввода и кнопки для подтверждения действия, а также визуальные подсказки, которые помогают пользователю правильно заполнить форму. Эта функциональность упрощает процесс добавления материалов и способствует улучшению общего квалификационного уровня контента на платформе.

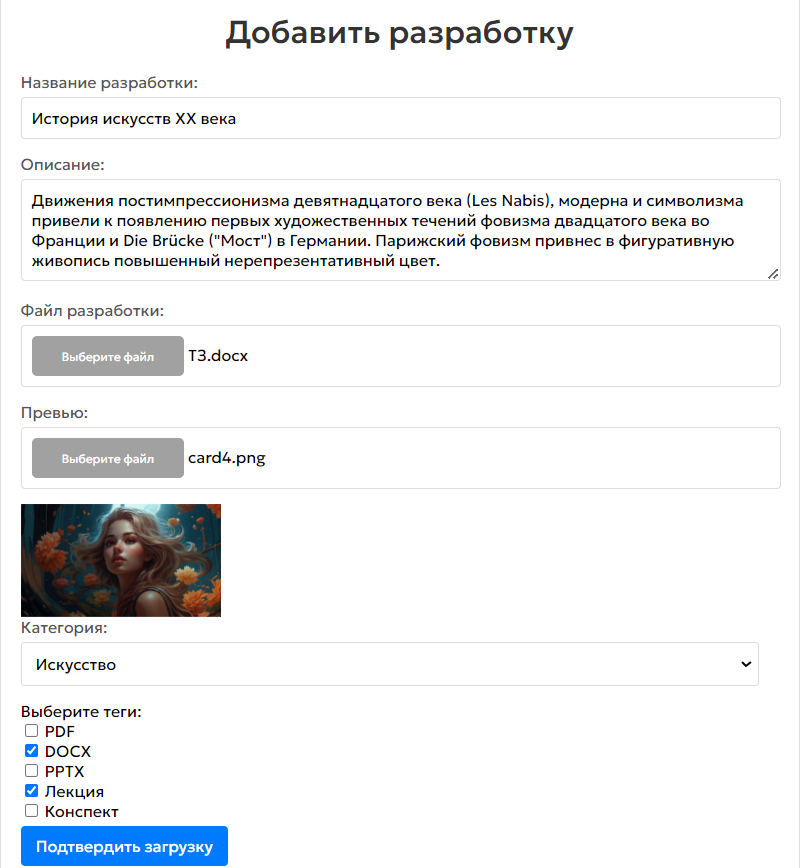


Рисунок 30 ‒ Работа формы для публикации образовательной разработки

Данные о разработке добавляются в базу данных, для отображения в каталоге, который представлен на рисунке 31.

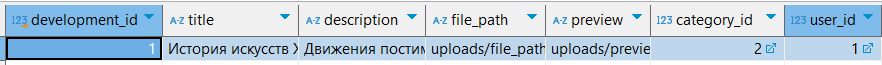


Рисунок 31 ‒ Отображение добавленной разработки в базе данных

На рисунке 32 представлен фрагмент каталога с отображением недавно добавленной разработки. Здесь можно увидеть, как новая разработка интегрируется в общую структуру каталога.

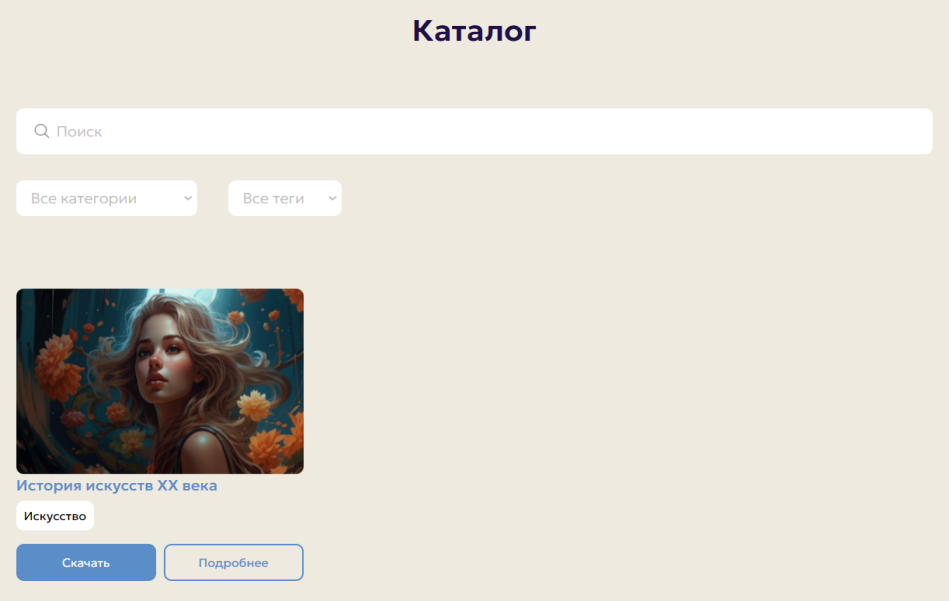


Рисунок 32 ‒ Отображение добавленной разработки в каталоге

В таблице 3 представлены результаты проверки функциональности удаления опубликованных разработок.

Таблица 3 – Проверка удаления опубликованной разработки

|  |  |  |  |
| --- | --- | --- | --- |
| №3 | Проверка удаления опубликованной разработки | | |
| Ссылка на ТЗ | Васильева В. П. | Высокий | Удаление разработок |
| 1. Перейти в панель администратора (только для администратора) в раздел «Удаление и редактирование разработок». 2. Выбрать опубликованную разработку для удаления. 3. Нажать кнопку «Удалить». 4. Подтвердить удаление. | | Ожидаемый результат: Разработка успешно удалена, данные отсутствуют в базе данных и не отображаются в каталоге. | Фактический результат:   * данные удалены из базы данных; * разработка не отображается в каталоге. |

Данные, которые ранее были добавлены в базу данных, после удаления разработки из каталога, удалились и из базы данных, представлено на рисунке 33.

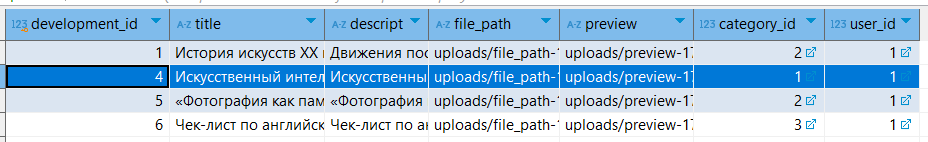


Рисунок 33 ‒ Отображение добавленных разработок в базе данных

После попытки удалить разработку в веб-приложении появляется окно с подтверждением этого действия, представлено на рисунке 34. Это окно служит для информирования пользователя о том, что он собирается выполнить необратимую операцию, и дает возможность подтвердить или отменить данное действие.

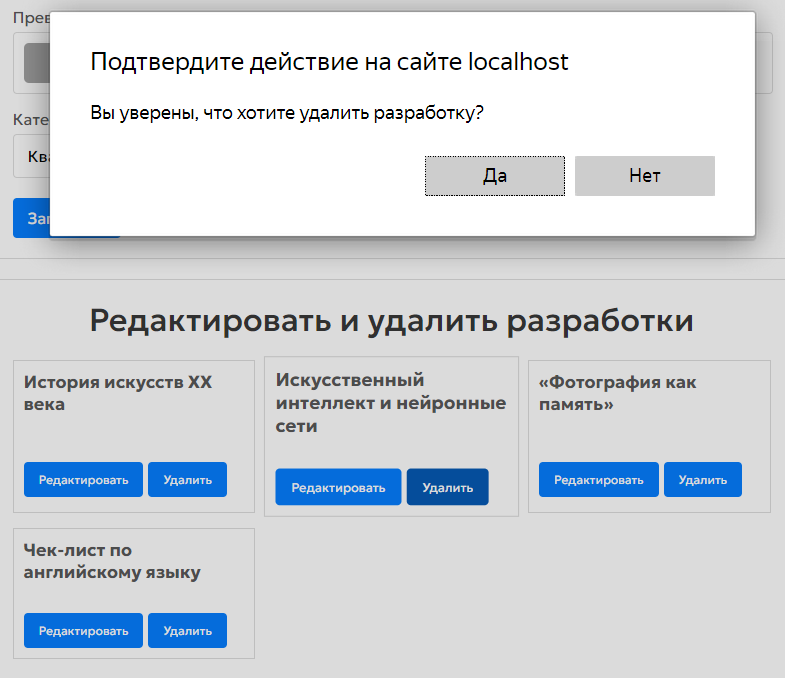


Рисунок 34 ‒ Форма и сообщение удаления разработки

После удаления данных о разработке из базы данных происходит удаление самой разработки из каталога приложения, представлено на рисунке 35. Этот процесс важен для поддержания чистоты и актуальности данных, обеспечивая, что пользователи видят только те элементы, которые доступны и активны в системе.

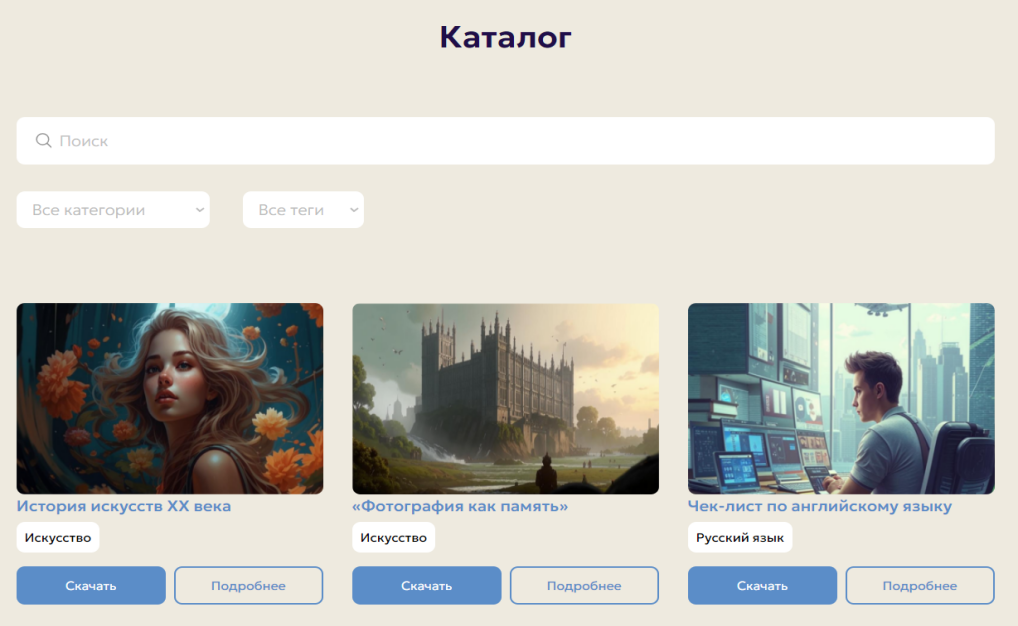


Рисунок 35 ‒ Отсутствие удаленной разработки в каталоге

Рисунок 36 иллюстрирует отсутствие данных об удаленной разработке в системе. На данном изображении видно, что запись, ранее присутствовавшая как в каталоге, так и в базе данных, полностью исчезла. Это подтверждает успешность и полноту операции удаления, а также демонстрирует, что пользователи теперь видят только актуальную информацию.

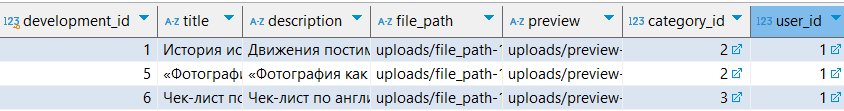


Рисунок 36 ‒ Отсутствие данных об удаленной разработке

В таблице 4 представлены результаты проверки работы функции поиска в каталоге по ключевым словам. Данная проверка была проведена для оценки эффективности и удобства использования этой ключевой функции в веб-приложении.

Таблица 4 – Проверка работы поиска в каталоге по ключевым словам

|  |  |  |  |
| --- | --- | --- | --- |
| №4 | Проверка работы поиска в каталоге по ключевым словам | | |
| Ссылка на ТЗ | Васильева В. П. | Высокий | Поиск |
| 1. Перейти в каталог разработок. 2. Ввести ключевое слово в строку поиска (например, «математика»). 3. Нажать кнопку «Поиск». | | Ожидаемый результат: Список разработок фильтруется по введенному ключевому слову и отображает только релевантные результаты. | Фактический результат:   * список отображает только релевантные разработки. |

При вводе в поиск слов, которые фигурируют в разработке, каталог автоматически фильтруется и выдает релевантный результат, представлено на рисунке 37.

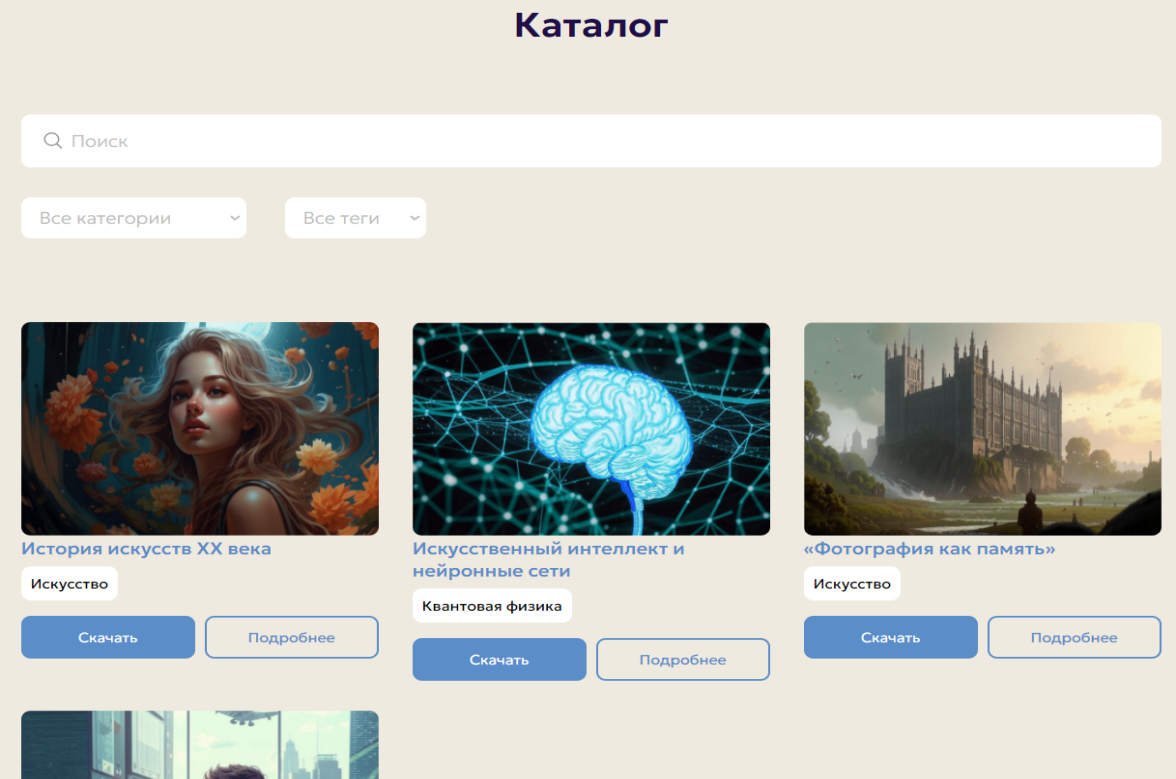


Рисунок 37 ‒ Отображение каталога до ввода слов в поиск

После ввода ключевых слов, каталог начинает автоматически перестраиваться и выдает результат поиска, представлено на рисунке 38.

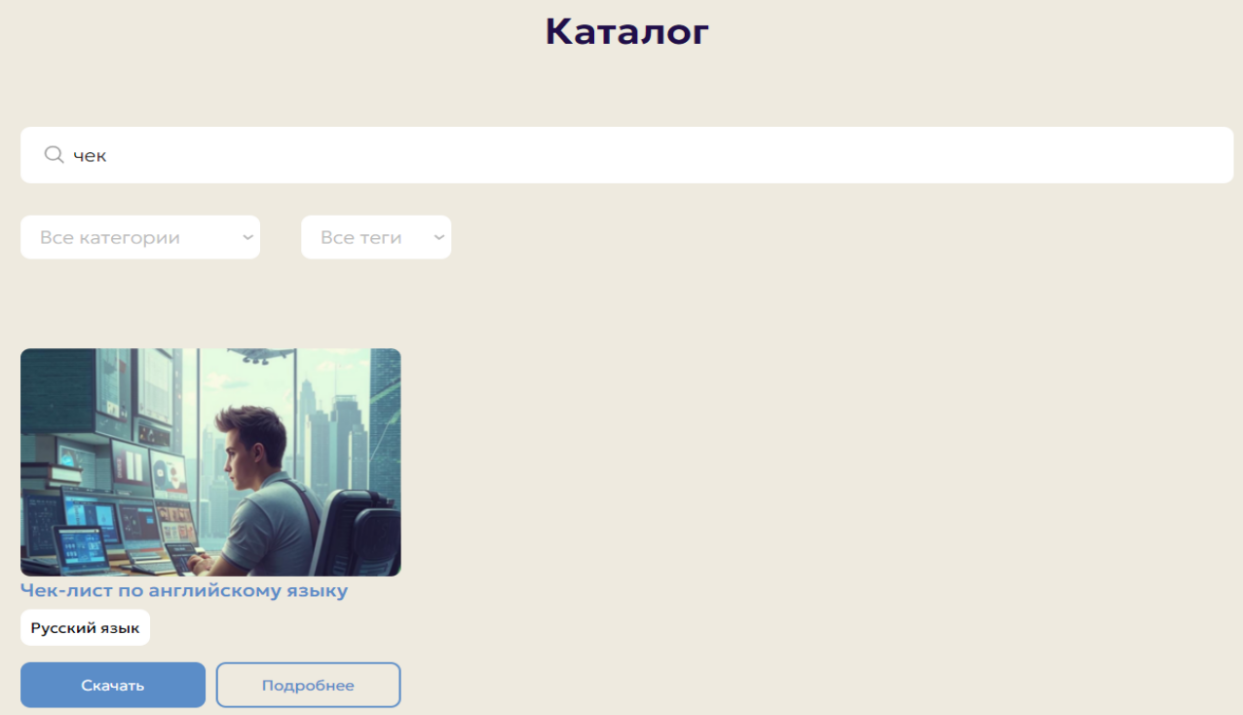


Рисунок 38 ‒ Отображение разработки, найденной по ключевому слову

В случае, если в каталоге нет разработок с вводимым словом, каталог выдаст, что ничего не найдено, представлено на рисунке 39.

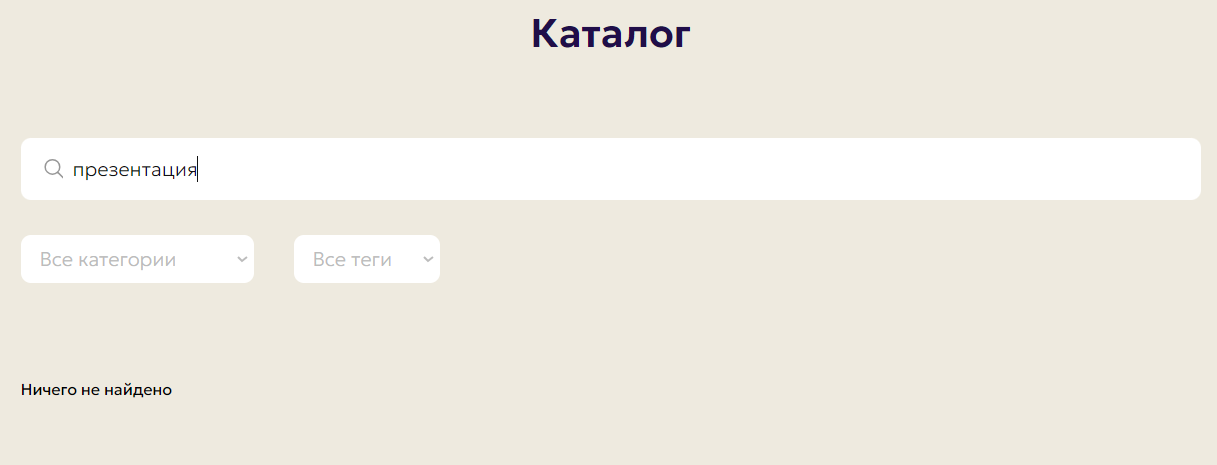


Рисунок 39 ‒ Случай, когда по ключевым словам ничего не найдено

Выявленные ошибки и недостатки будут устранены в соответствии с рекомендациями, представленными в отчете о тестировании.

После устранения ошибок и недостатков будет проведено повторное тестирование для проверки работоспособности продукта.

После проведения тестирования и устранения ошибок и недостатков продукт будет готов к представлению заказчику для оценки и утверждения.

Эффективное тестирование веб-приложения позволяет обеспечить высокое качество приложения, удовлетворение пользователей и защиту от потенциальных угроз безопасности.

**5.3 Публикация веб-приложения в сети**

Размещение веб-приложения «УчительPRO» включает предоставление исходного кода, структурированного в соответствии с общепринятыми стандартами. Проект организован в следующие директории:

* «/frontend» для клиентской части («HTML», «CSS», «JavaScript»);
* «/backend» для серверной части (код на «Node.js»);
* «/database» для миграций и скриптов конфигурации базы данных;
* «README.md» с инструкциями по установке, настройке и запуску проекта;
* «package.json» для «Node.js», содержащий список зависимостей проекта.

Исходный код предоставляется через систему контроля версий, например, «Git», с доступом к репозиторию на платформах «GitHub», «GitLab» или «Bitbucket». Это позволяет отслеживать изменения и управлять историей проекта.

Для размещения веб-приложения рекомендуется использовать популярные хостинг-провайдеры, такие как «Heroku» для быстрого развертывания, «AWS» («Amazon Web Services») для высокой степени настройки и масштабируемости, «DigitalOcean» для виртуальных серверов («Droplets») или «VPS» (виртуальные частные серверы) для пользователей, желающих больше контроля над сервером, таких как «Linode» или «Vultr».

После размещения веб-приложения необходимо выполнить следующие шаги для проверки его функционирования:

* доступ к серверу через «SSH» для «VPS»;
* проверка установленных зависимостей в соответствии с требованиями в файле «requirements.txt» или «package.json»;
* установка необходимых сред выполнения на целевом сервере;
* настройка базы данных с использованием миграционных скриптов из директории «/database»;
* проверка корректности подключения к базе данных и создания всех необходимых таблиц;
* настройка переменных окружения, и параметры подключения к базе данных через интерфейс управления хостингом;
* запуск серверного приложения с использованием соответствующих команд;
* использование систем управления процессами для автоматического перезапуска приложения в случае сбоя;
* открытие веб-приложения в браузере по указанному «URL» для тестирования основных функциональных возможностей, таких как регистрация, авторизация, публикация материалов и доступ к ресурсам;
* проверка работы приложения на различных устройствах и браузерах для обеспечения кроссбраузерной совместимости.

После успешного развертывания веб-приложение будет доступно по публичному «URL», предоставленному хостингом. Если заказчик желает использовать собственный домен, необходимо настроить «DNS-записи», указывающие на «IP-адрес» сервера, путем добавления записей «A» или «CNAME» в панели управления доменом.

Размещение и техническое сопровождение веб-приложения «УчительPRO» требуют тщательной подготовки и соблюдения установленных процедур. Следование данным инструкциям позволит обеспечить успешное развертывание, работоспособность и доступность приложения для пользователей. Правильная структура исходного кода и выбор надежного хостинг-провайдера способствуют повышению эффективности и стабильности работы приложения.

**6 Размещение и техническое сопровождение веб-приложения**

**6.1** **Требования к аппаратным ресурсам**

Веб-приложение «УчительPRO» разработано с учетом современных стандартов и технологий, что обеспечивает его совместимость с различными операционными системами и устройствами посредством стандартных веб-браузеров. Однако для развертывания серверной части применяются следующие минимальные требования:

* процессор Intel Pentium 166 МГц или лучше;
* объём оперативного запоминающего устройства 1024 Мбайт или выше;
* объём на жёстком диске не менее 3 Гбайт;
* монитор с SVGA видеокартой;
* клавиатура;
* манипулятор «мышь».

Поскольку веб-приложение работает через браузер, требования к пользовательскому оборудованию минимальны:

* любые современные ОС (Windows, macOS, Linux, Android, iOS);
* любой современный браузер (Google Chrome, Mozilla Firefox, Яндекс);
* процессор от 1 ГГц и выше;
* оперативная память не менее 512 Мбайт;
* монитор с поддержкой разрешения не менее 1024 на 768 пикселей;
* клавиатура;
* манипулятор «мышь».

Требования к оборудованию для администраторов и разработчиков:

* многоядерные процессоры (например, Intel Core i5, i7 или выше);
* объем оперативной памяти не менее 8 Гбайт;
* монитор с поддержкой разрешения не менее 1024 на 768 пикселей;
* жесткий диск SSD для высокой скорости работы;
* средства мониторинга и диагностики оборудования;
* клавиатура, мышь или другое устройство ввода для взаимодействия.

Требования к электропитанию и охлаждению:

* источник бесперебойного питания (ИБП) для серверных устройств;
* системы вентиляции и кондиционирования.

Дополнительные требования:

* использование антивирусных программ, системы защиты сети;
* регулярное создание резервных копий базы данных и важных файлов.

Обеспечение минимальных аппаратных требований гарантирует доступность системы для большинства пользователей с использованием стандартных устройств и поддерживаемых операционных систем. В случае увеличения числа пользователей или расширения функциональности, рекомендуется планировать масштабируемую архитектуру, включающую более мощное серверное оборудование и оптимизированные базы данных.

**6.2 Руководство администратора**

6.2.1 Назначение программного средства

Веб-приложение «УчительPRO» предназначено для обеспечения взаимодействия между учебными заведениями, преподавателями, студентами и разработчиками образовательных ресурсов. Основное предназначение системы – организация централизованного хранения, управления и публикации образовательных материалов, а также предоставление инструментов для их поиска, скачивания и редактирования.

Административное программное средство обеспечивает централизованный контроль над содержимым платформы, управлением пользователями, ролями, а также контролем доступа к различным функциям системы. В его функциональный спектр входит:

* назначение ролей и прав доступа для разных категорий пользователей (администратор, преподаватель);
* добавление, редактирование, удаление учебных ресурсов, категорий и тегов;
* контроль истории скачиваний и активности пользователей;
* создание резервных копий базы данных и обеспечение безопасности системы;
* мониторинг работоспособности системы и логирование событий.

Этот инструмент предназначен для системных администраторов и ответственных за техническое сопровождение системы, обеспечивая их возможностью эффективно управлять системой в условиях эксплуатации и обновлений.

6.2.2 Характеристики программного средства

Веб-приложение «УчительPRO» разработано с применением передовых технологий и стандартов веб-разработки, что гарантирует его функционирование на широком спектре современных устройств и операционных систем через совместимые браузеры. Ниже представлены технические характеристики.

Технологии разработки:

* backend на базе языка Node.js;
* frontend HTML, CSS, JavaScript, современные библиотеки и фреймворки;
* база данных реляционная СУБД – SQLite.

Инфраструктура:

* веб-сервер Apache, Nginx или встроенный сервер приложений.
* cреда выполнения соответствующая выбранному языку программирования Node.js runtime.

Объем памяти:

* для серверной части – не менее 1 Гбайт оперативной памяти;
* для клиентской – не более минимальных требований, определённых для современных браузеров.

Требования к браузеру:

* современные браузеры, поддерживающие HTML5 и CSS3, без необходимости установки дополнительных плагинов или расширений.

Системные требования:

* для сервера: процессор с минимальной частотой 166 МГц или выше;
* объем диска не менее 3 Гбайт;
* подключение к сети с пропускной способностью не менее 10 Мбит/с.

6.2.3 Обращение к программе

Для запуска и использования веб-приложения «УчительPRO», разработанного на платформе Node.js, необходимо выполнить несколько шагов.

Сначала убедитесь, что на вашем компьютере установлены последние версии Node.js и npm. Рекомендуется использовать редактор Visual Studio Code для удобства разработки и запуска. Затем скачайте или склонируйте проект в выбранную папку на вашем компьютере. После этого откройте терминал в папке с проектом, что можно сделать в Visual Studio Code с помощью сочетания «Ctrl + \» или через меню. Выполните команду для установки необходимых модулей, введя npm install. Эта команда автоматически установит все библиотеки, указанные в файле package.json.

После успешной установки зависимостей запустите сервер, выполнив в терминале команду npm start. Если в файле package.json прописан скрипт start, то сервер запустится, и в консоли появится сообщение, подобное «Сервер запущен по порту http://localhost:3000». Далее откройте любой современный веб-браузер и введите указанный URL (http://localhost:3000 или другой, если порт был изменен). После этого полностью загрузится сайт «УчительPRO», и вы сможете использовать его функциональные возможности. То как выглядит запуск в терминале показано на рисунке 10.

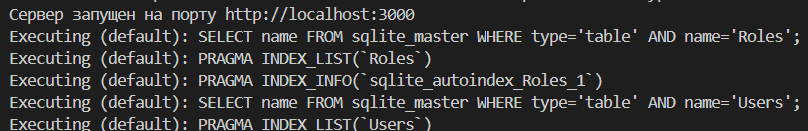


Рисунок 37 – Запуск в локальном режиме

Если необходимо запускать систему на удаленном сервере или в другом окружении, рекомендуется выполнить дополнительные настройки, такие как использование менеджеров процессов, например, pm2, и настройки доменных имен.

6.2.4 Входные и выходные данные

Веб-приложение принимает следующие категории входных данных:

* учётные данные пользователей;
* загружаемые файлы;
* метаданные разработок;
* запросы на поиск и фильтрацию.

В результате обработки входных данных система формирует следующие выходные данные:

* отображение каталога материалов;
* отчёты о скачиваниях;
* персональная статистика пользователей;
* уведомления о новых материалах.

6.2.5 Сообщения программисту

В рамках разработки системы «УчительPRO» специально не предусмотрены уникальные системные сообщения для пользователя, за исключением стандартных сообщений веб-браузера или ошибок сервера. В то же время, важную роль занимает корректное документирование и обработка сообщений, возникающих при работе системы.

Внутри исходного кода системы предусмотрены многочисленные комментарии, которые служат ориентиром для разработчиков и техподдержки. Они описывают структуру функций, логику отдельных блоков, особенности работы с базой данных, а также возможные места для расширения или доработки системы.

Описания ошибок и исключений включают стандартные сообщения.

«Ошибка при выполнении SQL-запроса» – возникающая при нарушении целостности или ошибке выполнения операции с базой данных.

«Некорректный формат входных данных» – при несоответствии формы ожидаемым параметрам.

«Не удалось подключиться к серверу» – при сбое соединения с сервером.

Для упрощения сопровождения системы в исходном коде предусмотрены комментарии типа:«//Проверка пользователя на права доступа».

Эти комментарии помогают разработчикам быстро понять структуру и логику программного кода, а также облегчают внесение изменений и развитие системы.

**6.3 Руководство пользователя**

Ежедневно создаются новые программные продукты, сервисы и часто пользователям бывает что-то непонятно при освоении какой-нибудь сложной программы, поэтому каждому новому продукту желательно собственное руководство.

Большинство людей не хочет разбираться с чем-то незнакомым без персонального, всегда доступного и понятного помощника. А именно им и является хорошее руководство пользователя.

6.3.1 Руководство оператора

6.3.1.1 Назначение программного средства

Веб-приложение «УчительPRO» предназначено для выполнения ниже представленных целей в рамках образовательной деятельности.

Для разработчика:

* создание, тестирование и сопровождение системы, внесение изменений и дополнений в функционал;
* обеспечение стабильной работы системы, исправление ошибок и развитие интерфейса;
* сопровождение базы данных и интеграционных процессов.

Для клиента (пользователя системы – учителя, сотрудника учреждения):

* выполнение задач по управлению учебным процессом, редактированию и обновлению учебных ресурсов;
* просмотр и создание отчетов о работе системы, мониторинг состояния и активности;
* взаимодействие с системой через удобный интерфейс для достижения образовательных целей.

Для руководителя:

* контроль деятельности пользователей и администраторов системы;
* аналитика и формирование отчетов для принятия управленческих решений;
* настройка прав доступа и управление ролью пользователей;
* обеспечение безопасности данных и контроль за выполнением нормативных требований.

Общая задача системы – повысить эффективность учебного процесса за счет автоматизации рутинных задач, обеспечения прозрачности и удобства управления информацией.

6.3.1.2 Условия выполнения программы

Поскольку веб-приложение работает через браузер, требования к пользовательскому оборудованию минимальны:

* любые современные ОС (Windows, macOS, Linux, Android, iOS);
* любой современный браузер (Google Chrome, Mozilla Firefox, Яндекс);
* процессор от 1 ГГц и выше;
* оперативная память не менее 512 Мбайт;
* монитор с поддержкой разрешения не менее 1024 на 768 пикселей;
* клавиатура;
* манипулятор «мышь».

6.3.1.3 Выполнение веб-приложения

**7 Расчет экономической эффективности**

При анализе экономической эффективности в первую очередь необходимо рассчитать стоимость разработки веб-приложения «УчительPRO».

Используя результаты этих расчетов, следует вычислить экономическую эффективность результатов деятельности МОАУ «COШ №71» по реализации программного обеспечения после опубликации сайта.

График выполнения работ представлен в таблице 35.

Таблица 35 – График выполнения работ по разработке «УчительPRO»

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование работ | Исполнитель | Длительность работы | |
| Дни | Часы |
| Разработка технического задания |  | 1 | 8 |
| Планирование |  | 2 | 16 |
| Рабочее проектирование |  | 14 | 112 |
| Отладка и тестирование |  | 5 | 40 |
| Обобщение и оценка результатов |  | 2 | 16 |

Таким образом, на разработку «УчительPRO» для МОАУ «COШ №71» было затрачено 23 дня или 184 часа (при восьмичасовом рабочем дне).

При расчете стоимости (составлении сметы затрат) разработки веб-приложения учитываются следующие виды расходов:

* стоимость материалов и покупных изделий;
* основная заработная плата;
* дополнительная заработная плата;
* страховые взносы;
* накладные расходы;
* затраты на машинное время (затраты на электроэнергию).

Перечень затрат на материалы и покупные изделия приведен в таблице 36.

Таблица 36 – Затраты на материалы и покупные изделия

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование | Ед. измерения | Количество | Цена за ед., руб | Стоимость, руб |
| Доступ в Интернет | мес. | 1 | 520 | 520 |
| Канцтовары | Шт. | 2 | 50 | 100 |
| Бумага формата А4 | упаковка | 1 | 550 | 550 |
| Итого |  |  |  | 1170 |

Таким образом, затраты на материалы и покупные изделия равны:

(1)

К этой статье относится основная и дополнительная заработная плата разработчика веб-приложения. Результаты расчета фонда заработной платы представлены в таблице 37.

Таблица 37 – Расчет фонда заработной платы

|  |  |  |  |
| --- | --- | --- | --- |
| Должность: программист | Количество проработанных дней | Размер ежедневной платы, руб. | Заработная плата в месяц, руб |
| Основная заработная плата | 23 | 480 | 11040 |
| Дополнительная заработная плата (80% от основной) |  |  | 8832 |
| Итого |  |  | 19872 |

В статью «Дополнительная заработная плата» входят выплаты, предусмотренные трудовым договором с руководством МОАУ «COШ №71» на разработку «УчительPRO». Размер дополнительной заработной платы разработчика веб-приложения определяется в размере 80 процентов от основной заработной платы.

832 руб. (9)

Следовательно, разработчику сайта всего начислено 19872 руб.

К отчислениям на социальные нужды относят страховые взносы в ПФР, ФСС, ФФОМС и взносы на страхование от несчастных случаев на производстве и профзаболеваний.

Страховые взносы рассчитываются в размере 32,2% от фонда заработной платы, что составит:

. (10)

Отчисления в пенсионный фонд ЗПФ составляют 24 процента от фонда заработной платы и равны:

. (11)

Отчисления в фонд обязательного медицинского страхования Змс равны:

. (12)

Отчисления на социальное страхование Зсс равны:

. (13)

Отчисления на обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний равны:

. (14)

Численные значения отчислений на социальные нужды (страховые взносы), представлены в таблице 38.

Таблица 38 – Расчет отчислений на социальные нужды (страховые взносы)

|  |  |  |
| --- | --- | --- |
| Отчисления на социальные нужды (страховые взносы) | Тарифы страховых взносов % | Суммы страховых взносов руб. |
| Отчисления в ПФР | 24 | 4 769,3 |
| Отчисления в ФОМС | 5,1 | 1 013,5 |
| Отчисления в ФСС | 2,9 | 576,29 |
| Отчисления на обязательное социальное страхование от несчастных случаев на производстве и профессиональных | 0,2 | 39,70 |
| Всего | 32,2 | 6 398,80 |

Размеры тарифов страховых взносов устанавливаются Федеральными законами. На момент разработки проекта необходимо руководствоваться действующим законодательством.

Накладные расходы, косвенные затраты – расходы, затраты, сопровождающие, сопутствующие основному производству, но не связанные с ним напрямую, не входящие в стоимость труда и материалов - дополнительные к основным затратам расходы для обеспечения процессов производства и обращения.

Накладные расходы Зн фирмы составляют 20 процентов (условно) от суммы основной и дополнительной заработной платы:

(15)

Как следует из данных таблицы 23, на разработку и последующую отладку веб-приложения «УчительPRO» для МОАУ «COШ №71» потребовалось 23 рабочих дня.

В среднем с учетом перерывов программист работает за компьютером 6 часов в день. Себестоимость одного кВт/ч электроэнергии (С1квт/ч) для организаций составляет 3 рубля 30 копеек.

При проведении расчетов в проекте необходимо в расчеты брать существующие на дату расчета тарифы. Суммарная мощность энергопотребителей для АРМ программиста складывается из мощности, потребляемой системным блоком персонального компьютера, монитором, принтером и другим периферийным оборудованием, которая составляет 1,5 кВт. Следовательно, за 8 часов работы программиста суммарное энергопотребление за день составит:

. (16)

Таким образом, стоимость машинного времени Змаш, необходимого для разработки веб-приложения, составит:

(17)

Затраты на машинное время учитываются как затраты на электроэнергию.

В результате выше произведенных расчетов мы получили итоговые затраты на разработку, представленные в таблице 39.

Таблица 39 – Итоговая смета затрат

|  |  |
| --- | --- |
| Наименование статей расходов | Сумма, руб |
| Стоимость материалов и покупных изделий | 1170 |
| Основная заработная плата | 11040 |
| Дополнительная заработная плата | 8832 |
| Отчисления на социальные нужды (32,2% от п.2.4) | 6398,8 |
| Накладные расходы (20% от п.2.5) | 3974,4 |
| Затраты на машинное время (затраты на электроэнергию) | 910,8 |
| Итого | 32326 |

Цена веб-приложения Ц определяется итоговыми затратами и прибылью, которая, в свою очередь, составляет 30% (условно) от фонда заработной платы:

(18)

Разработка веб-приложения преследовала цель экономии рабочего времени специалистов на анализирование вычислительных устройств для вынесения решения по дальнейшему действию с ними.

Одним из основных методов определения экономической эффективности служит метод сравнения результатов работы до и после проведения мероприятия.

Расчет экономической эффективности по этому методу проведен на основе данных о количестве принятых и обработанных заявок пользователей ВТ, влияющих на отправление в ремонт устройства, до и после размещения веб приложения.

Усредненные данные о реализации программного обеспечения за период с 10 сентября по 25 октября 2023 года (до размещения веб-приложения) по сравнению с аналогичными данными за период с 1 ноября по 29 ноября 2023 года (после размещения веб-приложения) приведены в таблице 40.

Таблица 40 – Усредненные данные о результатах реализации программного обеспечения «Техническая поддержка» до и после размещения веб-приложения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Затраты на диагностику устройств | | Затраты на ремонт устройств | | Итого |
| До | После | До | После |  |
|  |  | 67 800 | 1 200 | -98,33% |
| 250 | 2 500 |  |  | +100% |
| Общие затраты до | 68050 | Общие затраты после | 3700 | 100% - 5,4% |

Из приведенных данных в таблице 28 можно сделать вывод, что благодаря реализации программного продукта общие затраты на обслуживание устройств снизились на 79,5%:

(19)

Цена программного продукта составляет , в то время как расходы, которые несет компания, при внедрении «УчительPRO» несет на меньше.

(20)

Таким образом, экономический эффект Э в 1,7 раза превышает затраты на разработку вет-сайта.

. (21)

Срок окупаемости проекта равен 0,59 года.

Следовательно, разработка и размещение веб-сата «УчительPRO» было экономически оправдано.

**Заключение**

Целью описываемого веб-приложения «УчительPRO» является создание эффективной платформы обмена образовательными ресурсами среди учителей и учеников. Это приложение направлено на автоматизацию процессов, связанных с обменом информации, публикацией материалов и улучшением образовательного процесса. С его помощью пользователи смогут не только делиться своими разработками, но и находить необходимые ресурсы для обучения, что способствует повышению качества образовательного процесса.

Как показывает педагогическая практика, применение в процессе обучения интерактивных технологий существенно повышает как количественные, так качественные показатели усваиваемого обучающимися учебного материала. Интерактивные технологии обучения позволяют выстроить систему организованного взаимодействия преподавателя и обучающихся, которая гарантирует педагогически эффективное воздействие, позволяющее в результате создать условия для переживания обучаемыми происходящей ситуации.

В ходе производственной практики были выполнены работы по исследованию потребностей целевой аудитории, а также разработаны все необходимые документы для успешной реализации проекта. Для создания веб-приложения был проведен анализ предметной области, составлено техническое задание, в котором определялись аппаратные и функциональные требования к разработке. Кроме того, была выполнена разработка проектной документации, включающей создание макетов и эскизов веб-приложения, а также подбор графических материалов, что наглядно иллюстрирует дизайн и функциональность платформы [10].

Далее была осуществлена реализация веб-приложения, включающая разработку интерфейса с использованием современных технологий «front-end» и «back-end». В процессе разработки были реализованы шаблоны и динамические элементы интерфейса, что обеспечило интуитивно понятное взаимодействие пользователей с приложением. Важным этапом стало тестирование функциональности, выявление и устранение ошибок, что позволило гарантировать стабильную и безопасную работу веб-приложения.

Разработан привлекательный и современный дизайн, соответствующий ожиданиям целевой аудитории и принципам юзабилити. Наполнение платформы качественным контентом включало сбор и структурирование материалов, отвечающих образовательным потребностям пользователей. Обеспечен интуитивно понятный интерфейс, что способствует легкости навигации и повышает удовлетворенность пользователей. Создана база данных для загруженных образовательных разработок, обеспечивающая эффективное хранение и управление материалами [15].

Реализован функционал сайта с использованием современных технологий программирования, проведено тестирование на наличие ошибок и недочетов, что позволило выявить и устранить проблемы до запуска. Все найденные недочеты были исправлены, после чего веб-сайт был успешно запущен на хостинг.

Данное веб-приложение «УчительPRO» представляет собой мощный инструмент для образовательного процесса, который может использоваться как учителями, так и учениками. Оно не только облегчает обмен ресурсами, но и создает возможность для самообразования, позволяя пользователям находить и использовать разнообразные образовательные материалы, создаваемые их коллегами [10].

Данное приложение привнесло возможность:

* использовать широкий спектр учебных материалов, включая методические пособия, презентации, конспекты;
* делиться своими наработками, что способствует развитию профессионального сообщества;
* поиска и фильтрации позволили учителям быстро находить нужные материалы, экономя время и усилия;
* использовать новые методы и технологии в обучении, что способствует улучшению качества образования.

Одним из ключевых достоинств приложения является его удобный и простой в навигации интерфейс, который облегчает пользователям поиск и использование образовательных ресурсов.

В дальнейшем могут быть реализованы дополнительные функции, такие как интеграция с системами онлайн-тестирования, создание системы рейтинга материалов, а также возможности для настройки уведомлений о новых ресурсах, что будет способствовать расширению функциональности платформы.

Таким образом, веб-приложение «УчительPRO» имеет большой потенциал для повышения интерактивности образовательного процесса и соответствия потребностям пользователей.

В заключении, это веб-приложение может успешно конкурировать с аналогичными платформами на рынке и стать незаменимым помощником для педагогов и учащихся, предлагая удобные инструменты для обмена образовательным контентом, что, в конечном счете, улучшит процесс обучения и сделает его более доступным и интересным.

**Список использованных источников**

1. Байбородова, Л. В. Педагогические технологии в 3 ч. Часть 1. Образовательные технологии: Учебник и практикум для академического бакалавриата / Л. В. Байбородова [и др.]; под общей редакцией Л. В. Байбородовой, А. П. Чернявской. – 2-е, – М.: Издательство Юрайт, 2019. – 258 с.
2. Белоконова, С. С., Назарова, В. В. Web-технологии в профессиональной деятельности учителя: учебное пособие. – Москва; Берлин: Директ-Медиа, 2020. – 180 с.
3. Борисенко И.Г., Черных С.И. Виртуализация отечественного образовательного пространства: монография [Текст] / И.Г. Борисенко – 1-е. – Красноярск: Сиб.федер. ун-т, 2016. 172 с.
4. Босова, Л. Л. Современные тенденции развития школьной информатики в России и за рубежом [Текст] / Л. Л. Босова – 1-е. – Москва: Информатика и образование, 2019. – 32 с.
5. Вигерс, Карл. Разработка требований к программному обеспечению = Software Requirements: пер. с англ.; 3-е издание, дополненное / Карл Виггерс, Джой Битти – СПб.: Издательство «BHV», 2020. – 736 с.;
6. Воронин, Б. А. Системный анализ: методические указания по выполнению курсового проекта для студентов, обучающихся применением дистанционных образовательных технологий / Б. А. Воронин. – Томск: ФДО ТУСУР, 2021. – 82 с.
7. Гвоздева, В. А., Лаврентьева, И. Ю. Основы построения автоматизированных систем: Учебник. – М.: ИД «ФОРУМ»: ИНФРА – М, 2019. – 320 с.
8. Куликова, Н. Ю., Данильчук, Е. В. Использование мультимедийных интерактивных средств при обучении учащихся школ // Известия Волгоградского государственного педагогического университета. 2019. № 04(143). С. 72–80.
9. Соловова, Н.В., Суханкина, Н.В., Дмитриева, Д.С., Дмитриев, Д.С. Цифровая педагогика: технологии и методы: учебное пособие – Самара: Самарский национальный исследовательский университет имени академика С.П. Королева, 2020. – 128 с.
10. Тхакушинов, А.К., Бибалова, С.А., Сиюхова, А.М. Интерактивные технологии как средство повышения эффективности процесса обучения // Вестник Майкопского государственного технологического университета. 2019. Вып. 4
11. Чернявская А. П., Байбородова Л. В., [и др.]. Педагогические технологии: Учебный курс. – М.: Издательство Юрайт, 2019. – 258 с.
12. Алексеев, Р. К. Тестирование документации программного обеспечения Р. К. Алексеев. [Электронный ресурс]. – Режим доступа: https://www.performance-lab.ru/pressreleases/testirovanie-dokumentatsii-programmn – 02.04.2025г.
13. Данильчук, Е. В., Куликова, Н. Ю., Чернышова, М. В., Волков, Д. В. Обучение информатике в условиях виртуализации образовательного пространства Современные проблемы науки и образования. 2019. № 6. [Электронный ресурс]. – Режим доступа: http://www.science-education.ru/ru/article/view?id=29323 – 22.03.2025г.
14. Helpiks.org: Достоинства и недостатки языка [Электронный ресурс]. – Режим доступа: https://helpiks.org/6-21879.html – 10.04.2025г.
15. Логическая структура БД. [Электронный ресурс]. – Режим доступа: https://intuit.ru/studies/courses/5/5/lecture/124?page=4/ – 05.04.2025г.
16. Методология IDEF0. [Электронный ресурс]. – Режим доступа: https://www.sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6\_2 – 04.04.2025г.
17. МОАУ «СОШ № 71» город Оренбург – Текст : электронный // orenschool71 : [Электронный ресурс]. – [Электронный ресурс]. – https://orenschool71.gosuslugi.ru/ – 24.03.2025г.
18. Смирнова, А. А. Образовательные онлайн-платформы как явление современного мирового образования: к определению понятия // Искусственные общества. [Электронный ресурс]. – Режим доступа: https://artsoc.jes.su/s207751800005274-0 – 23.03.2025г.
19. Федотова, В. С. Цифровые инструменты и сервисы в работе учителя: учебное пособие /Ленинградский государственный университет имени А.С. Пушкина, – [Электронный ресурс]. – https://biblioclub.ru/i ndex.php?page= book&id=611279 – 29.03.2025г.
20. UML Use Case Diagrams [Электронный ресурс]. – Режим доступа: uml-diagrams.org – 25.03.2025г.

**Приложение А**

***(обязательное)***

**UML-диаграмма прецедентов**

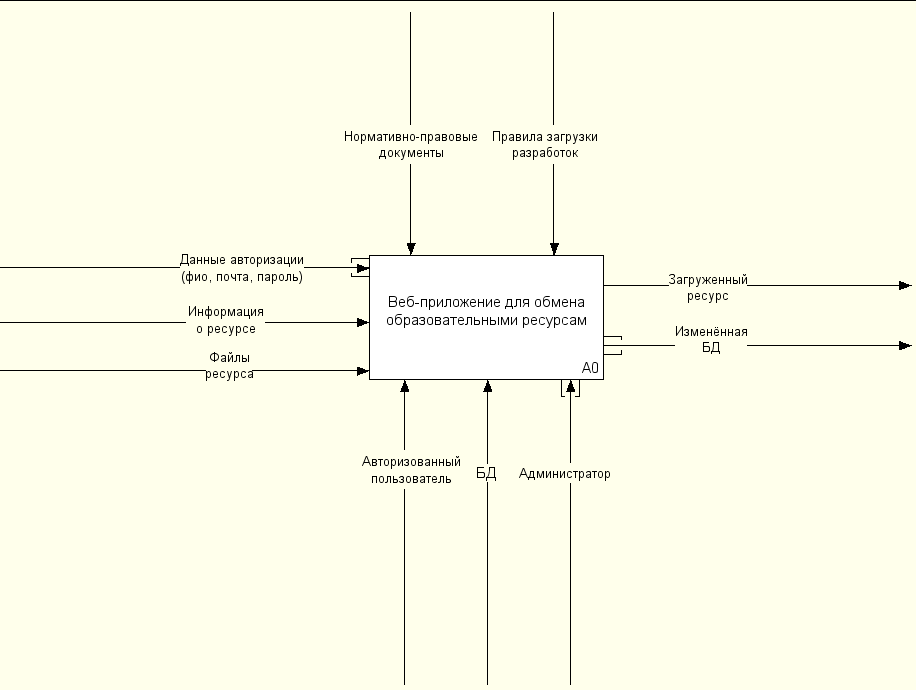


Рисунок А.1 – Функциональная модель

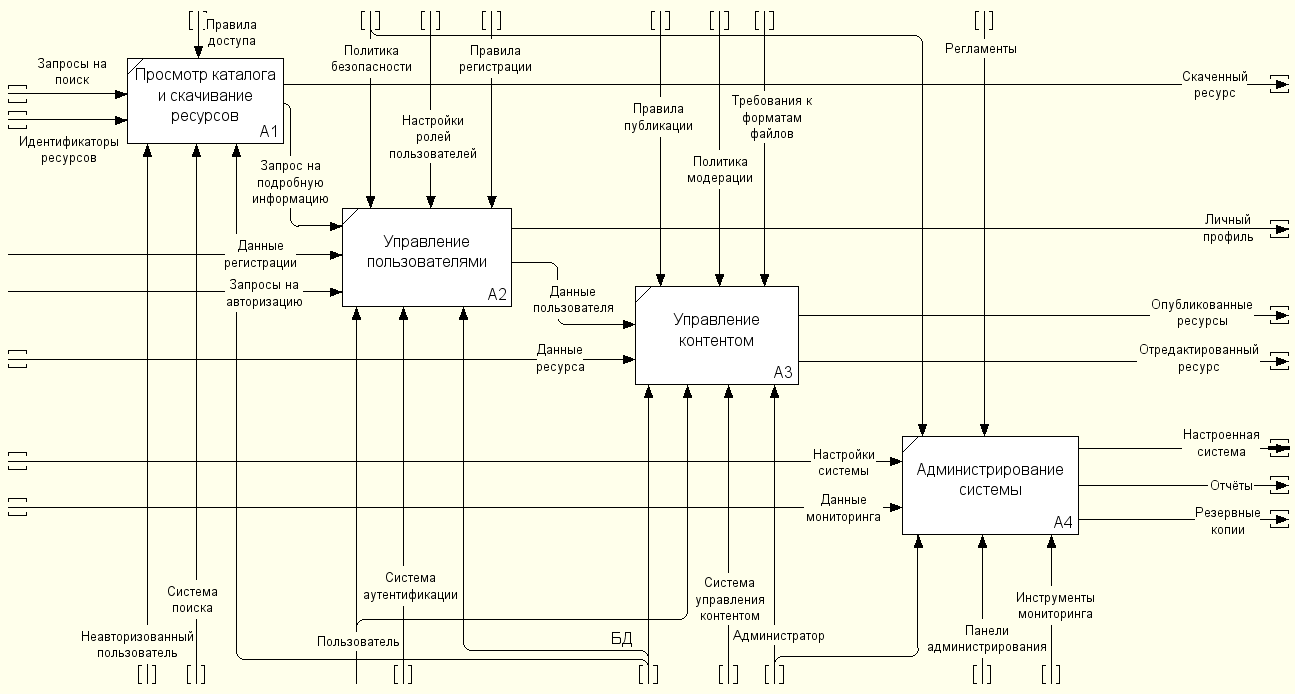


Рисунок А.2 – Функциональная модель (детализация)

**Приложение Б**

***(обязательное)***

**Информационная модель**

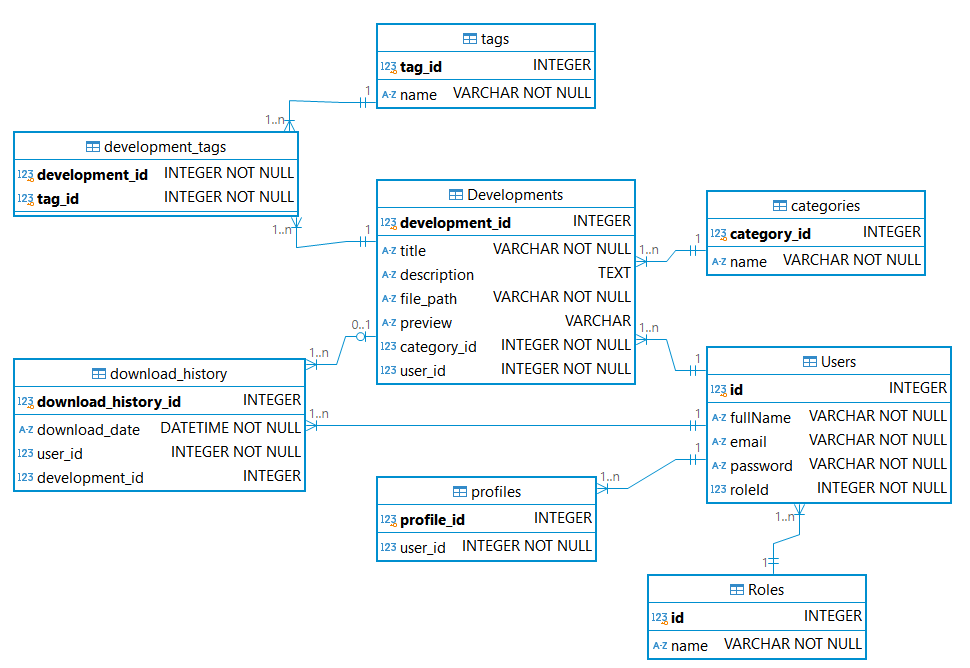


Рисунок Б.1 – Информационная модель

**Приложение В**

***(обязательное)***

**ER-диаграмма**

**Приложение Г**

***(обязательное)***

**SQL-скрипты**

**Приложение Д**

***(обязательное)***

**Код реализации веб-приложения**

Код реализации формы регистрации:

    <div class="form-container">

        <a href="/index"><span class="close" id="closeConfirm">← На главную</span></a>

    <h1>Регистрация</h1>

        <% if (error) { %>

        <p class="error"><%= error %></p>

        <% } %>

        <form method="post" action="/register">

            <div class="input-group">

               <input type="text" id="fullName" name="fullName" placeholder="Введите ФИО" required

                 <% if (error && error.includes('ФИО')) {%> class = "invalid" <%}%>>

            </div>

             <div class="input-group">

               <input type="email" id="email" name="email" placeholder="Введите почту" required

                 <% if (error && error.includes('email')) {%> class = "invalid" <%}%>>

            </div>

            <div class="input-group">

              <div class="password-container">

                <input type="password" id="password" name="password" placeholder="Введите пароль" required

                 <% if (error && error.includes('Пароль')) {%> class = "invalid" <%}%>>

                 <a href="#" class="password-control" onclick="return show\_hide\_password(this);"></a>

              </div></div>

            <div class="input-group">

              <div class="password-container">

                <input type="password" id="confirmPassword" name="confirmPassword" placeholder="Повторите пароль" required

                  <% if (error && error.includes('Пароли не совпадают')) {%> class = "invalid" <%}%>>

                  <a href="#" class="password-control" onclick="return show\_hide\_password2(this);"></a>

              </div>

             </div>

            <input type="submit" value="Зарегистрироваться">

        </form>

          <div class="auth-link">

             <p>Уже есть аккаунт? <a href="/login">Войти</a></p>

          </div></div>

        <script>

    function show\_hide\_password(target){

    const input = document.getElementById('password');

    if (input.getAttribute('type') == 'password') {

        target.classList.add('view');

        input.setAttribute('type', 'text');

    } else {

        target.classList.remove('view');

        input.setAttribute('type', 'password');

    }

    return false;

}

function show\_hide\_password2(target){

    const input = document.getElementById('confirmPassword');

    if (input.getAttribute('type') == 'password') {

        target.classList.add('view');

        input.setAttribute('type', 'text');

    } else {

        target.classList.remove('view');

        input.setAttribute('type', 'password');

    }

    return false;}

</script>

Код работы поисковой системы в каталоге:

    document.addEventListener('DOMContentLoaded', function() {

    const searchForm = document.getElementById('searchForm');

    const searchInput = document.getElementById('searchInput');

    const catalogContainer = document.getElementById('catalogContainer');

    const cards = Array.from(catalogContainer.getElementsByClassName('cardct'));

    searchForm.addEventListener('submit', function(event) {

        event.preventDefault(); // Предотвращаем перезагрузку страницы

        const searchTerm = searchInput.value.toLowerCase().trim();

        // Удаляем старые карточки

        catalogContainer.innerHTML = '';

        if (searchTerm) {

            // Сортируем карточки в зависимости от ключевых слов

            const filteredCards = cards.filter(card => {

                const title = card.querySelector('.title').textContent.toLowerCase();

                return title.includes(searchTerm);

            });

            // Вставляем отфильтрованные карточки в каталог

            filteredCards.forEach(card => {

                catalogContainer.appendChild(card);

            });

            // Если нет совпадений, показываем сообщение

            if (filteredCards.length === 0) {

                const noResults = document.createElement('p');

                noResults.textContent = 'Ничего не найдено';

                catalogContainer.appendChild(noResults);

            }

        } else {

            // Если строка поиска пустая, отображаем все карточки в исходном порядке

            cards.forEach(card => {

                catalogContainer.appendChild(card);

            });

        }

    });

    // Дополнительная функция для немедленного отображения поиска по вводу

    searchInput.addEventListener('input', function() {

        // Обработка для фильтрации при вводе без нажимания кнопки

        const searchTerm = searchInput.value.toLowerCase().trim();

        catalogContainer.innerHTML = '';

        if (searchTerm) {

            const filteredCards = cards.filter(card => {

                const title = card.querySelector('.title').textContent.toLowerCase();

                return title.includes(searchTerm);

            });

            filteredCards.forEach(card => {

                catalogContainer.appendChild(card);

            });

            if (filteredCards.length === 0) {

                const noResults = document.createElement('p');

                noResults.textContent = 'Ничего не найдено';

                catalogContainer.appendChild(noResults);

            }

        } else {

            cards.forEach(card => {

                catalogContainer.appendChild(card);

            });

        }

    });

});

Код реализации админ-панели:

<a href="/index"><span class="close" id="closeConfirm">← На главную</span></a>

    <div class="admin-container">

        <h1>Админ панель</h1>

            <% if (user) { %>

              <p>Добро пожаловать, <%= user.fullName %>! <a href="/logout">Выйти</a></p>

          <% } else { %>

              <p>Вы не авторизованы.</p>

              <p><a href="/login">Авторизоваться</a></p>

          <% } %>

            <div id="developments" class="tab-content active">

                <div class="form-section">

                    <h2>Добавить тег</h2>

                      <div class="form-group">

                            <label for="tagName">Название тега:</label>

                            <input type="text" id="tagName" name="tagName" required>

                        </div>

                        <div class="button-container">

                            <button onclick="addTag()">Добавить тег</button>

                        </div>

                    <div id="tagMessage"></div>

               </div>

               <div class="form-section">

                   <h2>Добавить категорию</h2>

                     <div class="form-group">

                        <label for="categoryName">Название категории:</label>

                         <input type="text" id="categoryName" name="categoryName" required>

                    </div>

                    <div class="button-container">

                       <button onclick="addCategory()">Добавить категорию</button>

                     </div>

                    <div id="categoryMessage"></div>

                </div>

               <div class="form-section">

                    <h2>Добавить разработку</h2>

                   <form id="addDevelopmentForm" method="post" enctype="multipart/form-data">

                       <div class="form-group">

                           <label for="developmentTitle">Название разработки:</label>

                            <input type="text" id="developmentTitle" name="title" required>

                       </div>

                     <div class="form-group">

                            <label for="developmentDescription">Описание:</label>

                            <textarea id="developmentDescription" name="description" rows="4"></textarea>

                     </div>

                       <div class="form-group">

                            <label for="developmentFile">Файл разработки:</label>

                           <input type="file" id="developmentFile" name="file\_path" accept=".pdf,.docx,.pptx,.mp4" required></div>

                         <div class="form-group">

                            <label for="developmentPreview">Превью:</label>

                             <input type="file"  id="developmentPreview" name="preview" accept=".jpg,.jpeg,.png"  required onchange="previewImage(event)">

                         </div>

                        <div class="preview-container">

                           <img id="preview-img" src="#" alt="Превью" style="display:none;">

                       </div>

                       <div class="form-group">

                          <label for="developmentCategoryId">Категория:</label>

                            <select id="developmentCategoryId" name="category\_id" required></select>

                      </div>

                     <div id="tagsContainer"></div>

                       <div id="errorContainer" class="error-message"></div>

                         <button type="submit">Загрузить</button>

                   </form>

                    <div id="developmentMessage"></div>

                 </div>

         </div>

            <div class="form-section">

                 <h2>Редактировать и удалить разработки</h2>

                 <div id="developmentsList" class="development-list">

                  </div>

            </div>

            <div id="users" class="tab-content">

                 <div class="form-section">

                     <h2>Список пользователей</h2>

                     <p>Всего пользователей: <%= userCount %></p>

                         <div class="user-list">

                            <table>

                                  <tr>

                                      <th>ID</th>

                                       <th>ФИО</th>

                                      <th>Почта</th>

                                  </tr>

                                    <% if(users) {

                                        users.forEach(user => { %>

                                        <tr>

                                            <td><%= user.id %></td>

                                           <td><%= user.fullName %></td>

                                              <td><%= user.email %></td>

                                          </tr>

                                  <% });} %>

                          </table>

                    </div>

            </div>