МИНИСТЕРСТВО ОБРАЗОВАНИЯ ОРЕНБУРГСКОЙ ОБЛАСТИ

**ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«ОРЕНБУРГСКИЙ КОЛЛЕДЖ ЭКОНОМИКИ И ИНФОРМАТИКИ»**

**(ГАПОУ «ОКЭИ»)**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К ДИПЛОМНОМУ ПРОЕКТУ**

*ОКЭИ 09.02.07. 1025. 08 ПЗ*

*Разработка веб-приложения «УчительPRO» для обмена образовательными ресурсами в МОАУ «СОШ №71»*

На тему \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

Пояснительной записки \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*15.06.2025*

Дата готовности проекта к защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*группы 4вб2 Васильева Валерия Павловна*

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О., группа)

*Морозова Маргарита Вячеславовна*

Руководитель проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Неделькина А.Е.*

Нормоконтроль\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Рецензент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дипломный проект допущен к защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Председатель ГЭК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Консультант по экономической части\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Содержание

[Введение 3](#_Toc15472)

[1 Технико-экономическое обоснование 6](#_Toc6072)

[2 Техническое задание 7](#_Toc8900)

[3 Технический проект 12](#_Toc19750)

[3.1 Документация функциональной части 12](#_Toc17626)

[3.1.1 Описание постановки задачи 12](#_Toc11490)

[3.1.2 Описание функций 12](#_Toc11341)

[3.2 Документация обеспечивающей части 12](#_Toc22710)

[3.2.1 Информационное обеспечение 12](#_Toc8178)

[3.2.2 Техническое обеспечение 12](#_Toc16265)

[3.2.3 Программное обеспечение 12](#_Toc12498)

[4 Рабочий проект 13](#_Toc7421)

[4.1. Разработка веб-приложения 13](#_Toc18396)

[4.2 Логика работы веб-приложения 13](#_Toc10956)

[4.3 Руководство системному программисту 13](#_Toc577)

[4.4 Руководство пользователя 13](#_Toc14912)

[4.5 Ревьюирование программного кода 18](#_Toc22937)

[4.6 Рефакторинг программного кода 23](#_Toc24123)

[4.7 Программа и методика испытания веб-приложения 23](#_Toc15935)

[5 Техника безопасности и пожарная безопасность 24](#_Toc21158)

[Заключение 26](#_Toc15754)

[Список использованных источников 28](#_Toc29767)

[Приложение А(обязательное) Расчёт экономической эффективности 30](#_Toc23240)

[Приложение Б(обязательное) Входные документы 31](#_Toc29768)

[Приложение В(обязательное) Выходные документы 32](#_Toc14639)

[Приложение Г(обязательное) Диаграмма потоков данных DFD 33](#_Toc3318)

[Приложение Д(обязательное) Диаграмма прецедентов 34](#_Toc26134)

[Приложение Е(обязательное) Функциональная модель 35](#_Toc27015)

[Приложение Ж(обязательное) Информационная модель 36](#_Toc20021)

[Приложение И(обязательное) Схема работы системы 37](#_Toc16604)

[Приложение К(обязательное) ER-диаграмма 38](#_Toc32197)

[Приложение Л(обязательное) Контрольный пример 39](#_Toc25506)

[Приложение М(обязательное) SQL-скрипты 40](#_Toc2736)

[Приложение Н(обязательное) Код реализации веб-приложения 41](#_Toc16454)

[Приложение П(обязательное) Акт о внедрении результатов 42](#_Toc16692)

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 «PostgreSQL», 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;
* разработать адаптивную версию для мобильных устройств;
* 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 Технико-экономическое обоснование

Муниципальное общеобразовательное автономное учреждение средняя общеобразовательная школа № 71 (МОАУ «СОШ № 71») расположена по адресу: г. Оренбург, ул. Джангильдина, 6. Директором учреждения является Бодина Татьяна Михайловна. Основная деятельность школы включает реализацию образовательных программ, воспитательную работу и обеспечение условий для осуществления образовательной деятельности.

Школа была основана в 1986 году и за 38 лет своей работы выпустила более 2000 учащихся с аттестатом об окончании средней школы, среди которых 78 медалистов, а также более 4000 учащихся с аттестатом об окончании основной школы, из них 73 с отличием. С 2012 года школа имеет статус автономного учреждения. Педагогический коллектив состоит из квалифицированных специалистов, среди которых 3 почетных работника образования и 18 педагогов с высоким уровнем профессиональной квалификации. Средний возраст преподавателей составляет 40 лет, что способствует внедрению инновационных методов обучения.

Школа обладает развитой структурой управления, включающей администрацию, педагогический коллектив, техническую службу и учебные подразделения.

Основные цели школы:

* формирование общей культуры личности, ее адаптация к жизни в обществе;
* создание основы для осознанного выбора и освоения профессиональных образовательных программ;
* воспитание трудолюбия, уважения к правам и свободам человека, любви к природе, родине и семье;
* формирование здорового образа жизни.

Программа развития школы направлена на создание условий для получения качественного, конкурентоспособного образования, способствующего социальной адаптации учащихся.

Образование – это многогранная область, включающая обучение, преподавание, управление и оценку знаний. Центральными фигурами образовательного процесса являются учащиеся и преподаватели. Учебный план, инфраструктура и управление играют ключевую роль в обеспечении эффективности образовательного процесса.

На сегодняшний день школа использует устаревшие методы обмена учебными материалами, что приводит к низкой скорости доступа к ресурсам, дублированию информации и недостаточной автоматизации процессов. В рамках внедрения веб-приложения предполагается автоматизация хранения и обмена учебными материалами, что позволит снизить затраты времени на подготовку и поиск информации, повысить качество учебных материалов и обеспечить доступ к ним в любой точке и в любое время.

В условиях современного образования, когда учебные материалы постоянно обновляются, систематизируются и требуют быстрого доступа, Нужно сделать так, чтобы работа внутри компании шла быстрее и лучше. Для этого нужно внедрить автоматизацию.

Внедрение веб-приложения позволит создать единую платформу для хранения, поиска и обмена учебными ресурсами. Это значительно упростит работу педагогов и повысит качество образовательного процесса.

Проблема, с которой сталкиваются учителя, – отсутствие единой платформы для обмена учебными материалами, методическими разработками, играми и тестами. Существующие ресурсы разрознены, что затрудняет поиск качественной информации и обмен опытом между педагогами. Это приводит к значительным временным затратам на подготовку материалов, сокращая время для непосредственного взаимодействия с учениками.

Для решения этой проблемы было разработано веб-приложение «УчительPRO», которое является платформой для обмена образовательными ресурсами.

Основные пользователи веб-приложения:

* учителя и преподаватели, которые будут загружать и делиться материалами;
* ученики и студенты, использующие платформу для получения учебных ресурсов;
* администрация школы, осуществляющая управление платформой и контроль за содержимым;
* родители, получающие доступ к учебным материалам для контроля за учебным процессом.

Диаграмма прецедентов для веб-приложения «УчительPRO» иллюстрирует взаимодействие различных типов пользователей и администратора с функционалом разрабатываемого сайта. На диаграмме выделяются 4 основных актора, каждый из которых имеет свои особенности и возможности.

Гость – незарегистрированный посетитель, который может просматривать главную страницу с учебными материалами и осуществлять поиск.

Педагог – зарегистрированный учитель с расширенными возможностями: настройка материалов, загрузка новых файлов, просмотр профиля и списка загруженных материалов.

Учащийся – зарегистрированный ученик, который может просматривать свой профиль и список скачанных материалов.

Администратор – управляющий сайтом, имеющий полный доступ: удаление или изменение материалов, добавление новых пользователей, анализ статистики использования сайта.

На сайте реализованы две ключевые связи:

«include» (включает) – автоматическое отображение каталога материалов при просмотре главной страницы.

«extend» (расширяет) – предоставление дополнительных возможностей при необходимости, например, загрузка файлов при публикации материала.

Для детального представления пользователей и их возможностей можно ознакомиться с диаграммой прецедентов в приложении Д.

Приложение предоставит следующие функции:

* авторизация и регистрация пользователей;
* персонализированный профиль;
* поиск и фильтрация материалов по ключевым словам, предметам и тегам;
* просмотр и скачивание ресурсов;
* публикация собственных разработок с возможностью загрузки файлов различных форматов (PDF, DOCX, PPTX, JPG и др.);
* подписка на рассылку новых материалов.

Согласно расчету экономической эффективности, представленному в приложении А, разработка веб-приложения для обмена образовательными ресурсами является экономически выгодным. Экономический эффект в 1,2 раза превышает затраты на разработку, а срок окупаемости проекта равен 0,81 года.

Ожидается, что внедрение платформы позволит:

* сократить временные затраты учителей на поиск материалов;
* повысить эффективность образовательного процесса;
* улучшить взаимодействие между участниками образовательного процесса.

Реализация проекта для обмена образовательными ресурсами «УчительPRO» соответствует стратегическим целям школы и современным тенденциям цифровизации образования и соответсвенно является экономически обоснованной и актуальной мерой, направленной на оптимизацию рабочих процессов учителей, упрощение обмена ресурсами и повышение качества образования за счет систематизации материалов и использования лучших практик. Это важный шаг в развитии школы и адаптации ее к современным образовательным стандартам.

Создание веб-приложения для МОАУ «СОШ №71» является актуальной задачей, которая позволит повысить качество образовательного процесса, снизить операционные затраты и обеспечить современный уровень автоматизации. Внедрение системы создаст условия для более эффективного использования учебных ресурсов, расширит возможности взаимодействия участников образовательного процесса и повысит конкурентоспособность учреждения в условиях цифровой трансформации образования.

# 2 Техническое задание

Организация ставит задачу разработать и внедрить веб-приложение, предназначенное для публикации и обмена образовательными ресурсами среди преподавателей и участников образовательного процесса. В рамках проекта создается платформа, которая станет удобным онлайн-ресурсом для загрузки, поиска и обмена учебными материалами, методическими разработками, тестами и планами уроков. Это специализированная веб-платформа, где пользователи смогут делиться своими разработками, находить необходимые материалы по предметам, тегам и форматам для повышения качества и эффективности образовательного процесса.

Создание платформы «УчительPRO» направлено на автоматизацию обмена образовательными ресурсами, повышение эффективности педагогической деятельности, расширение возможностей для самообразования и профессионального развития учителей и учеников. Внедрение системы должно обеспечить удобный доступ к учебным материалам, упростить их публикацию и обмен, а также повысить качество образовательного процесса.

В роли заказчика выступает организация МОАУ «СОШ №71».

Адрес: 460052, Россия, Оренбургская обл., г. Оренбург, ул. Джангильдина, 6.

Номер телефона/Электронная почта: +7 (3532) 62-51-30, 71@orenschool.ru.

В роле исполнителя выступает: Васильева Валерия Павловна, студент ГАПОУ «ОКЭИ».

Плановые сроки начала и окончания работ: 21.04.25 – 15.06.25.

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

Условное обозначение веб-приложения: «УчительPRO».

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.

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

Сервер должен иметь многопроцессорную архитектуру с процессорами «Intel Xeon» или «AMD EPYC» (тактовая частота от 2,5 ГГц). Рекомендуемый объем оперативной памяти – 16 Гб, с возможностью увеличения до 32 Гб и более.

Дистрибутив должен соответствовать технологиям и поддерживаемому ПО. Жесткий диск – минимум 500 Гб (рекомендуется SSD для скорости).

Сетевое соединение – не менее 100 Мбит/с с резервированием. Виртуализация и системы мониторинга обязательны для оптимизации и контроля. Регулярное обновление ПО и систем безопасности – необходимо.

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 дизайна и прототипирования, позволяющая 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.

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

HTML (язык гипертекстовой разметки) – язык для создания структуры и контента веб-страниц. Он позволяет описывать внешний вид и поведение элементов, таких как текст, изображения, таблицы и формы. HTML работает с CSS и JavaScript для оформления и добавления интерактивности. Он также обеспечивает совместимость с различными веб-браузерами.

CSS стилизует страницы, определяя цвета, шрифты, расположение и другие визуальные элементы. Разделение стилей от структуры улучшает читаемость кода и упрощает его поддержку. Использование CSS-фреймворков ускоряет разработку и обеспечивает единообразие стиля.

JavaScript делает веб-приложения интерактивными, обрабатывая данные на стороне клиента, валидируя формы и обрабатывая события. Он создает динамические и отзывчивые интерфейсы, улучшая пользовательский опыт. JavaScript-фреймворки повышают эффективность разработки и качество кода.

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

«PostgreSQL» – свободно распространяемая объектно-реляционная система управления базами данных (СУБД) с открытым исходным кодом. Легко интегрируется с «JavaScript-кодом». Обеспечивает автономность и портативность приложений.

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.

Для организации доступа пользователей должен использоваться протокол презентационного уровня HTTP и его расширение HTTPS.

Для разработки веб-приложения выделены следующие этапы:

* анализ требований и определение технического задания;
* проектирование архитектуры и функциональной модели;
* разработка макетов интерфейса и дизайна;
* реализация клиентской части приложения с использованием HTML и CSS;
* реализация серверной части приложения с использованием Node.js и PostgreSQL;
* интеграция клиентской и серверной частей;
* тестирование функционала и производительности;
* внедрение и публикация веб-приложения;
* обучение пользователей и предоставление технической поддержки.

Доступ к базе данных должен быть разграниченным и контролируемым.

Для каждого типа пользователя будут установлены роли и права, позволяющие выполнять разрешенные операции.

Реализуются механизмы авторизации и аутентификации для защиты данных от несанкционированного доступа.

Веб-приложение будет иметь различные уровни доступа в зависимости от статуса пользователя.

Неавторизованные пользователи смогут просматривать общедоступные разделы, но для редактирования и управления им потребуется пройти авторизацию.

После авторизации пользователи получат доступ к редактированию контента, добавлению материалов и управлению личными данными, с ограничениями по уровню доступа.

Администратор будет иметь полные права для управления пользователями, настройками системы и контентом.

Вход будет осуществляться через логин и пароль, защищенный от несанкционированного доступа.

Пользователи смогут восстановить пароль через электронную почту.

Ключевые действия пользователей будут регистрироваться в журнале для аудита и повышения безопасности.

Логика разграничения прав будет реализована на уровне системы, исключая обход прав доступа и обеспечивая целостность данных.

Общая структура страниц сайта для клиента состоит из следующих логических страниц:

* главная страница с краткой информации об услугах и навигационным меню;
* страница с каталогом где находятся все загруженные пользователями ресурсы;
* карточки разработок, с информации о ресурсе, включая название, изображение, описание, теги и категорию;
* личный профиль, с возможностью просмотра и редактирования загруженных разработок и истории скачиваний пользователя;
* страница входа/регистрации системы, включающая формы регистрации и входа;
* дополнительные страницы с формами для восстановления пароля пользователя;
* страница добавления разработки с формой для заполнения информации о ресурсе и загрузкой файла;
* ознакомительная страница с общей информацией про функции сайта, миссию и цели с контактной информацией.

Следующим этапом в разработке сайта являются проектирование и разработка дизайна. Необходимо разработать структуру информации и создать макеты страниц, разработать дизайн, учитывая ранее поставленные требования и UX/UI принципы.

На рисунке 1, представлены «Шапка» и «Подвал» сайта, они имеют функцию навигации и контактной информации соответственно, они дублируются на каждой странице.



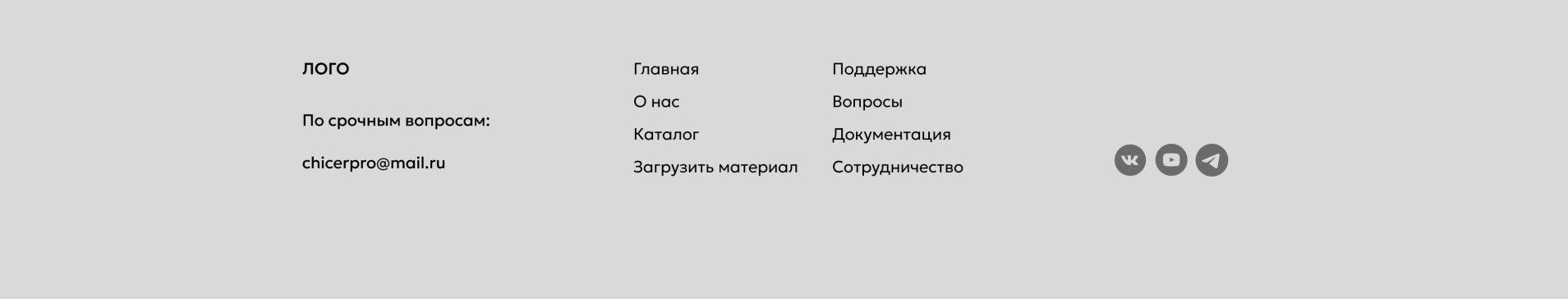


Рисунок 1 – Эскиз «Шапки» и «Подвала» сайта

На рисунке 2 представлен эскиз контента главной страницы, визуализирующий ее ключевые структурные элементы.

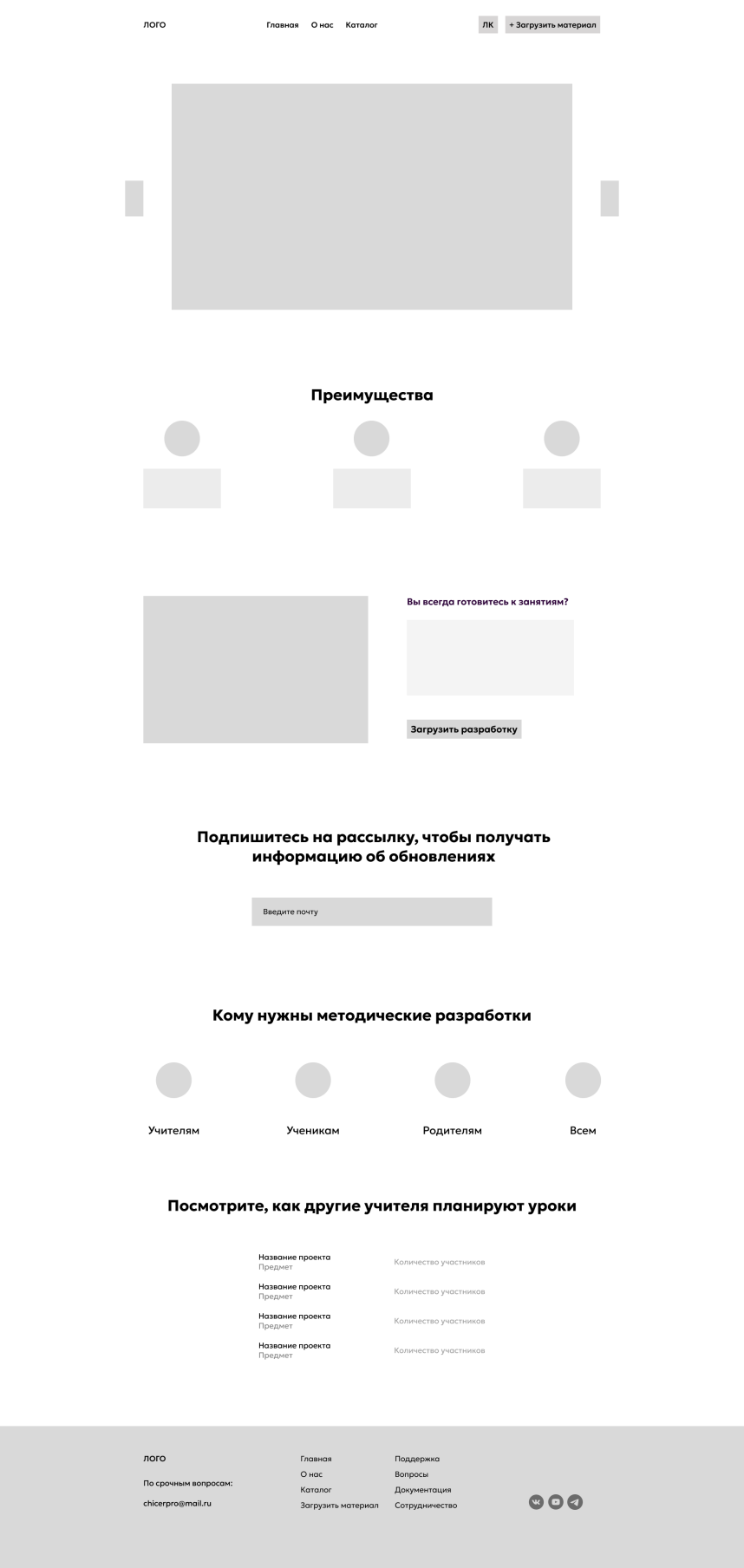
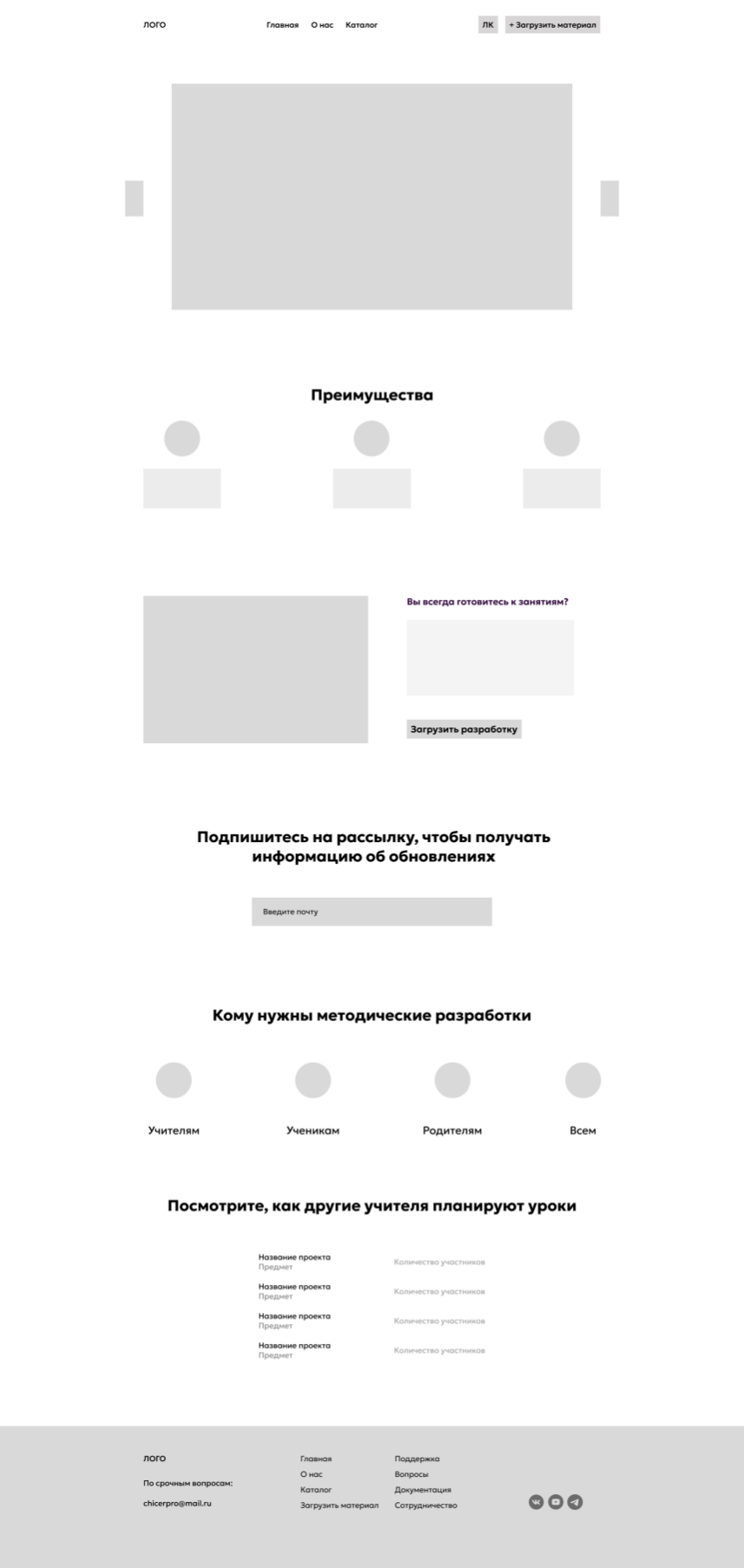
 

Рисунок 2 – Эскиз страницы «Главная»

На рисунке 3 представлен эскиз контента страницы о компании, где находится рекламный баннер, история с миссией и главной целью сайта, а также ее ценности и контактная информация для связи.

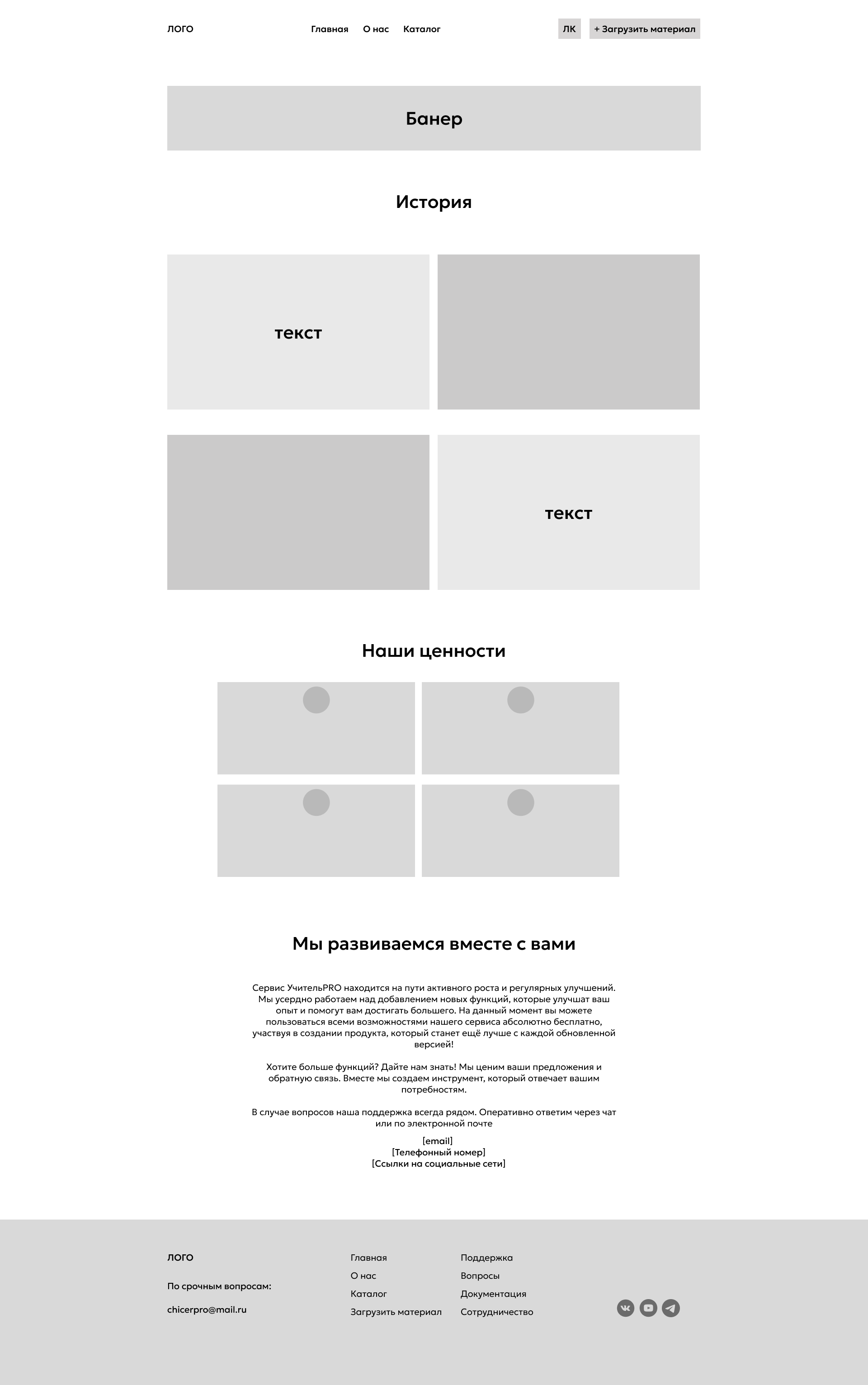
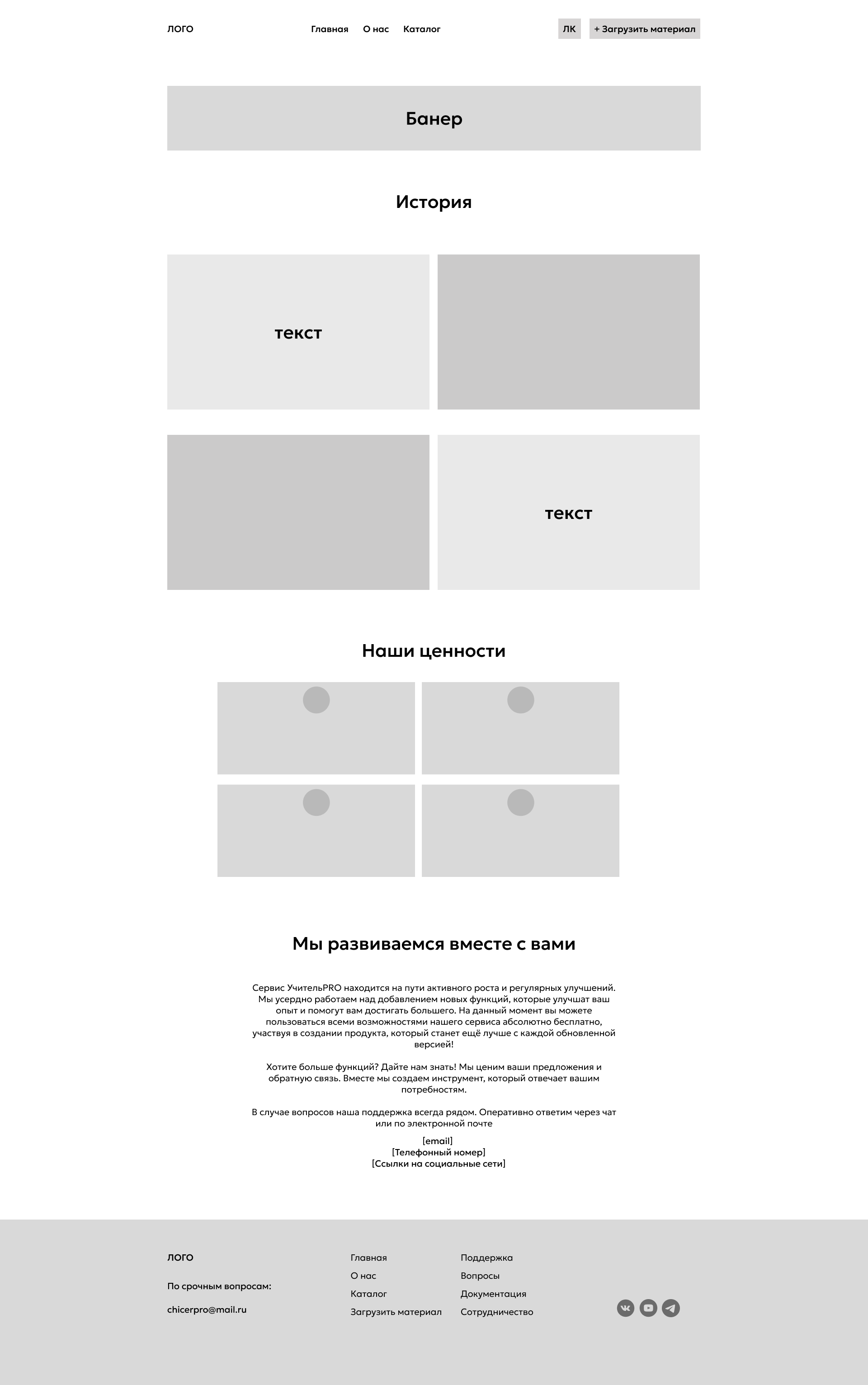
 

Рисунок 3 – Эскиз страницы «О нас»

На рисунке 4 представлен эскиз контента страницы с каталогом, где находится рекламный баннер, поиск и фильтрация, сам каталог с разработанными материалами других пользователей.

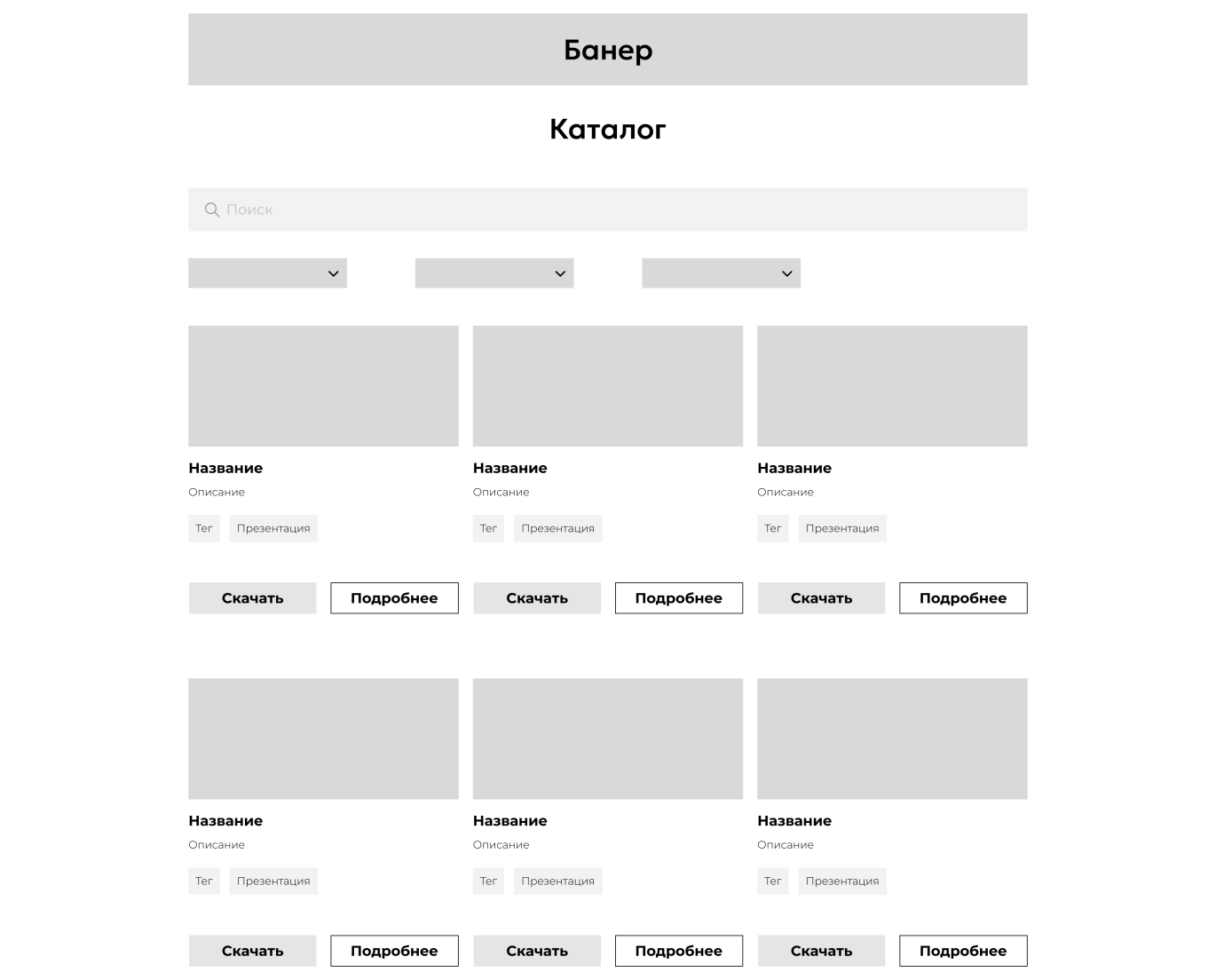


Рисунок 4 – Эскиз страницы «Каталог»

А на рисунке 5 изображен эскиз страницы, где хранятся загруженные личные разработки пользователя, так же можно будет увидеть все скаченные пользователем ресурсы.



Рисунок 5 – Эскиз страницы «Ваши разработки»

На рисунке 6 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.

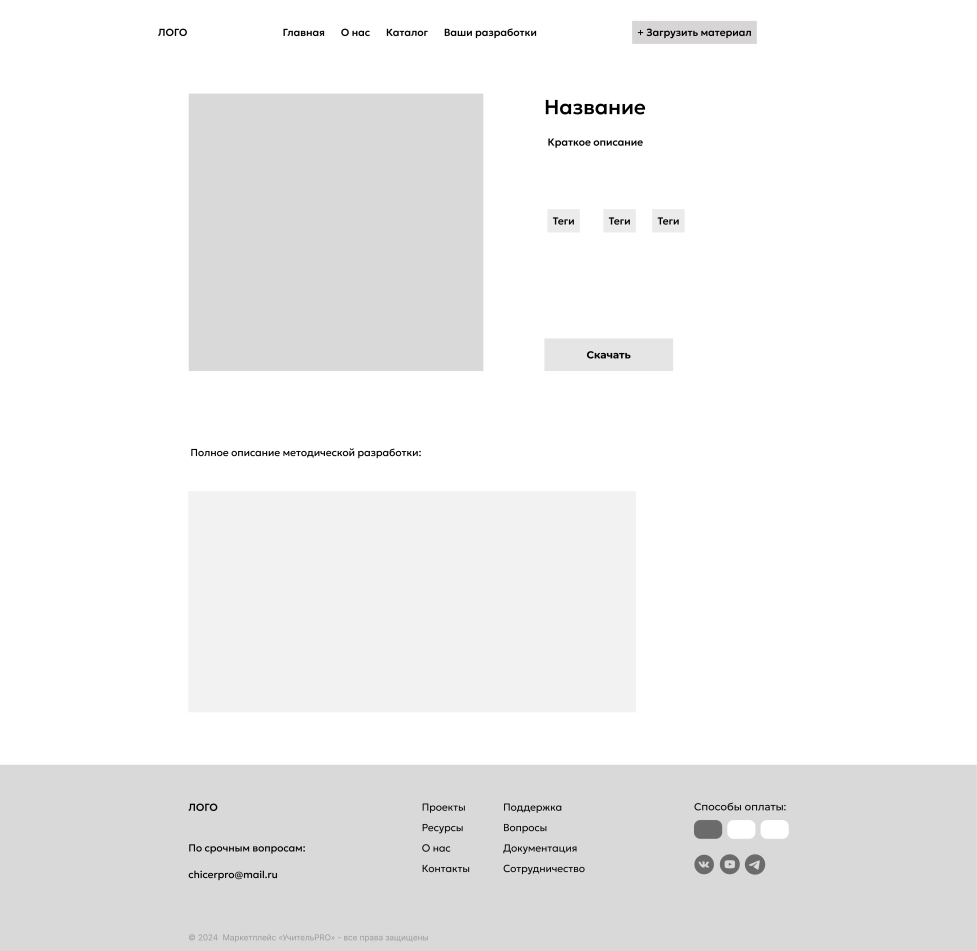


Рисунок 6 – Эскиз страницы «Страница продукта»

Во время создания системы, необходимо опираться на представленные базовые элементы, такие как: цветовая палитра, типографика, колоночная сетка, которые задают цветовую гамму, текстовое оформление и расположение объектов для дальнейшего дизайна системы.

К интерфейсу и дизайну веб-сайта предъявляется ряд общих требований, а именно:

* 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 легкости 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.

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.

В качестве СУБД для разработки базы данных будет использоваться PostgreSQL.

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

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

Эффективность функционирования веб-приложения после его публикации в сети оценивается по следующим критериям:

* время загрузки страниц не должно превышать 5 секунд;
* приложение должно работать стабильно без сбоев и ошибок;
* интерфейс должен быть интуитивно понятным и удобным для пользователей;
* должны быть обеспечены меры защиты данных пользователей и транзакций;
* приложение должно легко адаптироваться к увеличению числа пользователей и объема данных.

3 Технический проект

3.1 Документация функциональной части

### 3.1.1 Описание постановки задачи

Основная задача системы – обеспечить эффективный обмен образовательными ресурсами между педагогами, учениками и администрацией школы. В рамках решения задачи пользователь может загружать, просматривать, искать и скачивать учебные материалы, а также управлять своим профилем и контентом.

Цепочка действий начинается с момента входа пользователя в систему, после чего он может выбрать необходимую функцию: загрузить материал, выполнить поиск, просмотреть профиль или скачать ресурс. Ввод информации осуществляется через формы, расположенные на интерфейсе, а вывод – через страницы с результатами поиска, профилями и списками загруженных материалов.

Диаграмма прецедентов использования (use-case diagram) – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей. Данная диаграмма изображена в приложении Д.

DFD (Data Flow Diagram) или диаграмма потоков данных – это графическое представление потока данных через информационную систему. DFD показывает, как данные входят в систему, как они обрабатываются и какие выходные данные генерируются. Данная диаграмма представлена в приложении Г.

Взаимодействие с пользователем реализуется через графические формы, включающие следующие макеты:

* форма регистрации и входа: поля для ввода электронной почты, пароля, подтверждения пароля;
* форма загрузки ресурса: поля для выбора файла, указания названия, описания, категории, ключевых слов, предмета;
* форма поиска: поля для ввода ключевых слов, выбора категории, предмета, формата файла;
* профиль пользователя: разделы для редактирования публикаций, просмотра истории загрузок и скачиваний.

Макеты интерфейсов представлены в виде прототипов на рисунках 7-10, демонстрирующих расположение элементов и последовательность взаимодействий.

Основными формами взаимодействия пользователя с системой на первом этапе будут формы авторизации и регистрации. Их эскизы представлены на рисунке 7.

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

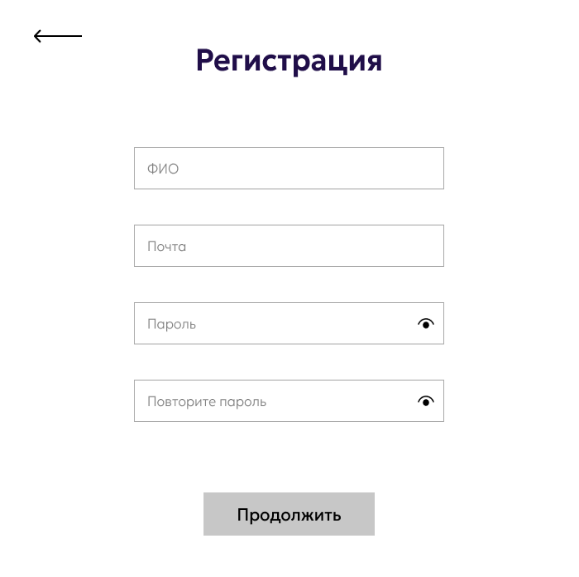
 

Рисунок 7 – Эскизы формы регистрации и авторизации

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

На рисунке 8 представлен эскиз формы, где пользователь может загрузить свои разработки, оформить ресурс для удобного поиска, выбрать категорию и тег, выбрать привлекательную обложку и информативное описание.

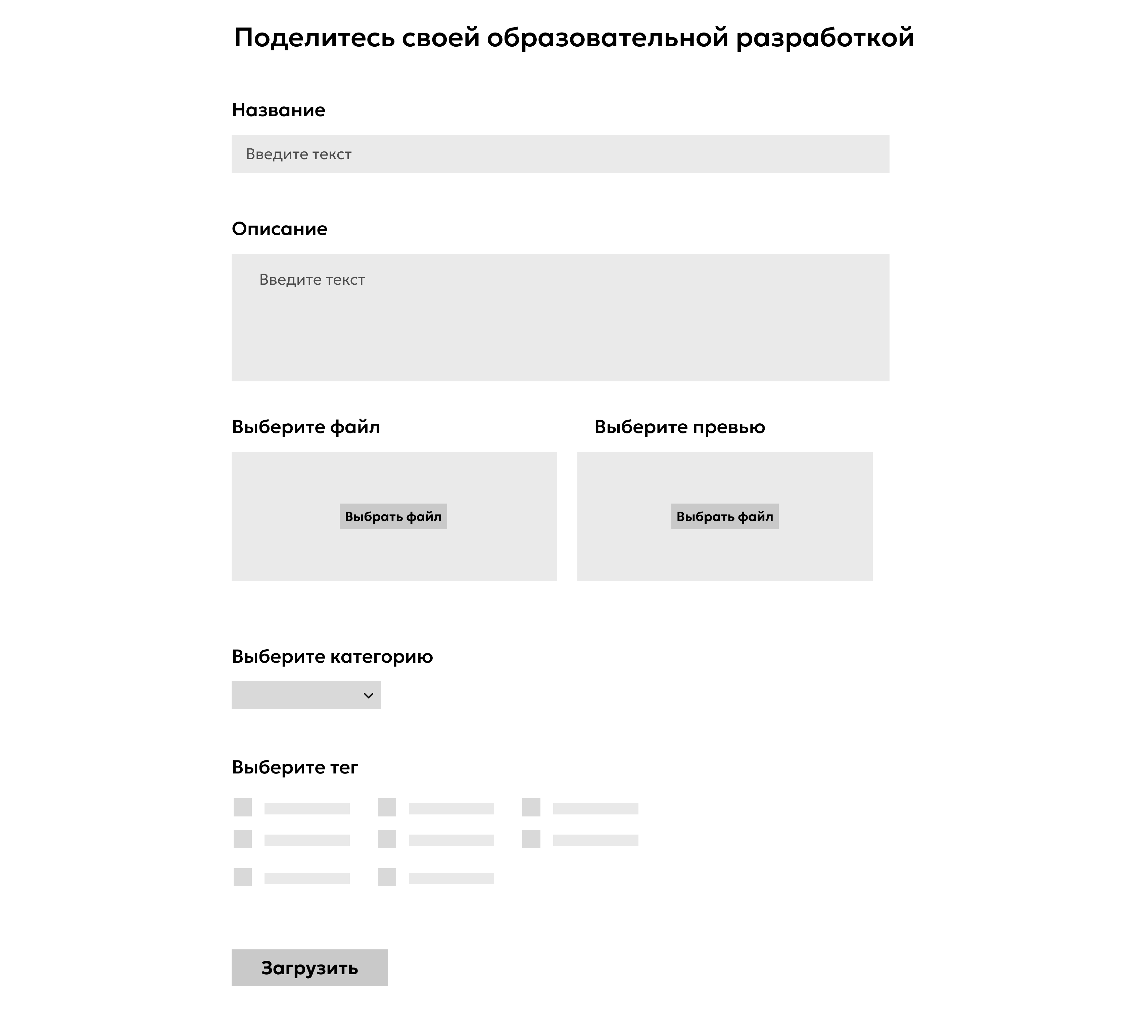


Рисунок 8 – Эскиз формы загрузки материала

А на рисунке 9 изображен эскиз формы для редактирования уже опубликованной разработки, данная форма находится в личном профиле пользователя.

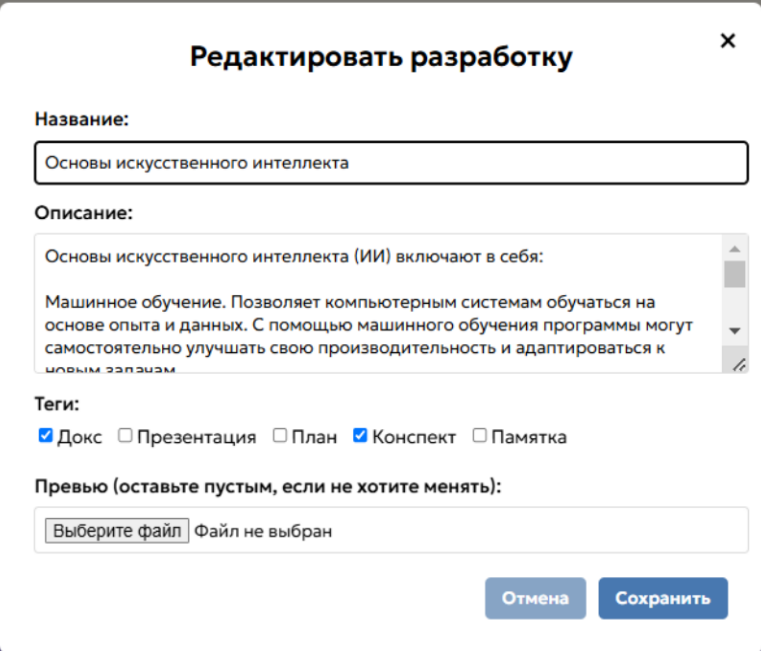


Рисунок 9 – Эскиз формы редактирования публикации

Рисунок 10 иллюстрирует форму поиска: поля для ввода ключевых слов, выбора фильтра категории и тега.



Рисунок 10 – Эскиз формы поиска и фильтров

Процесс обработки данных включает следующие этапы:

1. Ввод данных пользователем через формы.
2. Валидация введенной информации (например, проверка формата файла, уникальности email).
3. Обработка запросов на сервере, включая поиск по базе данных, загрузку и сохранение файлов.
4. Формирование ответных сообщений и отображение результатов на экране.

Для оценки эффективности работы системы можно использовать показатели времени отклика, точности поиска и успешности загрузки.

Время отклика системы можно представить как сумму времени обработки серверных операций и времени передачи данных по сети:

(1)

где Tобраб – время выполнения серверных операций, мс;

Tперед – время передачи данных по сети, мс.

Допустим, при выполнении запроса на поиск образовательных материалов:

Время обработки серверных операций (Tобраб) составляет 200 миллисекунд (мс).

Время передачи данных (Tперед) зависит от объема передаваемой информации. Например, при передаче файла размером 2 МБ по сети со скоростью 100 Мбит/с:

(2)

Итак, общее время отклика:

(3)

Точность поиска характеризуется как отношение количества релевантных найденных результатов к общему количеству запросов:

(4)

где Нрелев – количество релевантных найденных результатов;

Нвсе – общее количество запросов.

Например, за день было выполнено 100 поисковых запросов, из которых 96 вернули релевантные материалы, то:

(5)

Это означает, что 96% запросов возвращали релевантные результаты.

Модули системы включают:

* модуль аутентификации: взаимодействует с базой данных пользователей;
* модуль управления ресурсами: обеспечивает загрузку, хранение и поиск материалов;
* модуль профилей: управляет информацией о пользователях;
* модуль поиска: реализует алгоритмы поиска и фильтрации.

Взаимодействие между модулями осуществляется через внутренние API. Внешние системы, такие как системы хранения данных (облачные хранилища) или системы авторизации, могут интегрироваться через REST API.

Входная информация включает:

* данные пользователей: электронная почта, пароль, личные данные (имя, роль).
* образцы учебных материалов (файлы в форматах PDF, DOCX, PPTX и др.);
* метаданные ресурсов (название, описание, категория, тег);
* запросы поиска (ключевые слова, фильтры по категориям, формату файла);
* данные профиля (личная информация, история загрузок и скачиваний).

Эта информация поступает через формы интерфейса или API-запросы.

Выходная информация включает:

* сообщения об успешных операциях (подтверждение загрузки, сохранения данных);
* результаты поиска (список ресурсов, соответствующих запросу, с кратким описанием и возможностью скачать);
* профиль пользователя (отображение личных данных, истории активности);
* страницы с результатами (формы для просмотра, редактирования и управления ресурсами);
* ошибки и предупреждения (сообщения о неправильных вводных данных, ошибках поиска).

Входные данные представлены в приложении Б.

Выходные данные представлены в приложении В

### 3.1.2 Описание функций

В функционале сайта должны присутствовать такие возможности, как:

* 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;
* возможность удалять и редактировать опубликованные ресурсы.

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

В рамках разработки веб-приложения выделены основные функциональные модули, представленные в таблицах 1-6. Для структурирования работы системы использован предметно-функциональный принцип, где каждый модуль отвечает за выполнение определенной функции.

Таблица 1 – Регистрация и аутентификация пользователей

|  |  |  |  |
| --- | --- | --- | --- |
| Назначение режима | Обеспечить безопасный вход пользователей в систему и регистрацию новых участников платформы. | | |
| Входные данные | | Выходные данные | Процесс выполнения: |
| Логин, пароль, email, дополнительные данные пользователя. Источники информации – пользователь вводит данные через интерфейс. | | Подтверждение успешной авторизации или регистрации, отображение личного кабинета. В случае ошибок – сообщение о причине (например, неправильный пароль или уже существующий логин). | 1. Пользователь вводит данные для регистрации или входа. 2. Веб-приложение проверяет корректность данных (например, проверка формата email, уникальности логина). 3. В случае регистрации – создается новая учетная запись, данные сохраняются в базе данных. 4. В случае входа – происходит проверка данных и создание сессии пользователя. 5. После успешной аутентификации пользователь перенаправляется в личный кабинет. |

Таблица 2 – Публикация образовательных ресурсов

|  |  |  |  |
| --- | --- | --- | --- |
| Назначение режима | Позволить пользователю загружать и публиковать учебные материалы, указывать их параметры. | | |
| Входные данные | | Выходные данные | Процесс выполнения: |
| Данные о ресурсе – файл, описание, ключевые слова, категория, предмет. Источники – пользовательский ввод. | | Публикация ресурса на платформе, отображение его в общем списке с возможностью поиска и сортировки. | 1. Пользователь выбирает файл и заполняет метаданные (описание, теги, категория). 2. Веб-приложение проверяет формат файла, валидирует метаданные. 3. Загружает файл на сервер, сохраняет информацию в базе данных. 4. Обновляет список доступных ресурсов. |

Таблица 3 – Поиск и фильтрация ресурсов

|  |  |  |  |
| --- | --- | --- | --- |
| Назначение режима | Обеспечить пользователю возможность быстро находить нужные материалы по ключевым словам, категориям, предметам. | | |
| Входные данные | | Выходные данные | Процесс выполнения: |
| Параметры поиска – ключевые слова, категория, предмет, формат. Источники – пользовательский ввод. | | Вывод списка ресурсов, соответствующих критериям поиска, с возможностью сортировки по различным параметрам. | 1. Пользователь вводит параметры поиска. 2. Веб-приложение формирует запрос к базе данных. 3. Получает и отображает список ресурсов, сортируя их по выбранным критериям. |

Таблица 4 – Загрузка и скачивание ресурсов

|  |  |  |  |
| --- | --- | --- | --- |
| Назначение режима | Обеспечить пользователю возможность скачивать учебные материалы и загружать новые. | | |
| Входные данные | | Выходные данные | Процесс выполнения: |
| Запрос на скачивание или загрузка файла, метаданные файла. Источники – пользовательский запрос. | | Передача файла пользователю или сохранение файла на сервере. | 1. Пользователь выбирает ресурс для скачивания или загружает новый файл. 2. Веб-приложение проверяет права доступа и формат файла. 3. Для скачивания – формирует и передает файл. 4. Для загрузки – сохраняет файл и обновляет базу данных. |

Таблица 5 – Администрирование системы

|  |  |  |  |
| --- | --- | --- | --- |
| Назначение режима | Обеспечить управление системой, контроль за содержимым и пользователями. | | |
| Входные данные | | Выходные данные | Процесс выполнения: |
| Данные администраторских команд, отчёты, лог-файлы. | | Обновление данных, управление пользователями, просмотр логов и статистики. | 1. Администратор входит в панель управления. 2. Выполняет операции по управлению пользователями, контентом или настройками системы. 3. Веб-приложение обрабатывает команды, обновляет базу данных и отображает текущий статус. |

Все описанные модули и режимы работы системы представлены на схеме, которая размещена в приложении И. Эта схема отображает последовательность взаимодействий между пользователями и системой, а также внутренние связи между модулями.

## 3.2 Документация обеспечивающей части

### 3.2.1 Информационное обеспечение

Следующий этап проектирования – формирование физической модели базы данных. Созданная база данных моделирует систему для управления веб-приложения для обмена образовательными ресурсами для организации МОАУ «СОШ №71».

База данных для веб-приложения интернет-магазина построена по реляционной модели данных. Основные принципы проектирования включают:

* нормализация данных, включающая разделение данных на логически связанные таблицы для минимизации избыточности и предотвращения аномалий при обновлении;
* обеспечение целостности данных, включающее использование первичных и внешних ключей для обеспечения согласованности и связи между таблицами.
* безопасность данных, включающее управление доступом и разграничение прав пользователей с различными ролями;
* производительность, включающая оптимизацию структуры таблиц и индексов для ускорения выполнения запросов и обеспечения масштабируемости системы.

ER-диаграмма (Entity-Relationship Diagram) – это визуальное представление структуры базы данных, которое иллюстрирует сущности в базе данных и отношения между ними. Эта диаграмма помогает в проектировании и анализе баз данных, показывая, как данные связаны друг с другом. ER-диаграмма в приложении К.

В таблицах 6-15 представлены технические спецификации реляционных таблиц, сформулированные на языке определения данных SQL.

Таблица 6 – Техническое описание таблицы «users»

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

Таблица 7 – Техническое описание таблицы «roles»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Ключ | Тип, длина | Обязательность значения | Логическое ограничение | Примеры данных |
| id | PK | int | not null | Unique | 1, 2, 3 |
| name |  | varchar  (50) | not null |  | user, admin |

Таблица 8 – Техническое описание таблицы «developments»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Ключ | Тип, длина | Обязательность значения | Логическое ограничение | Примеры данных |
| development\_id | PK | int | not null | Unique | 1, 2, 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 | references Categories(id) | 1, 2, 3 |
| preview |  | int | not null |  | математика.jpg |
| profession |  | varchar  (225) | null |  | «Учитель информатики» |

Таблица 9 – Техническое описание таблицы «tags»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Ключ | Тип, длина | Обязательность значения | Логическое ограничение | Примеры данных |
| tag\_id | PK | int | not null | Unique | 1, 2, 3 |
| name |  | varchar  (50) | not null |  | docx, конспект |

Таблица 10 – Техническое описание таблицы «development\_tag»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Ключ | Тип, длина | Обязательность значения | Логическое ограничение | Примеры данных |
| development\_id | FK | int | not null | references Developments(id) | 1, 2, 3 |
| tag\_id | FK | int | not null | references Tags(id) | 1, 2, 3 |

Таблица 11 – Техническое описание таблицы «download\_history»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Ключ | Тип, длина | Обязательность значения | Логическое ограничение | Примеры данных |
| downloadhistory\_id | PK | int | not null | Unique | 1, 2, 3 |
| download\_date |  | date | not null |  | 20.05.25 |
| development\_id | FK | int | not null | references Developments(id) | 1, 2, 3 |
| user\_id | FK | int | not null | references Users(id) | 1, 2, 3 |

Таблица 12 – Техническое описание таблицы «categories»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Ключ | Тип, длина | Обязательность значения | Логическое ограничение | Примеры данных |
| id | PK | int | not null | Unique | 1, 2, 3 |
| name |  | varchar  (50) | not null |  | математика, физика |

Таблица 13 – Техническое описание таблицы «profiles»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Ключ | Тип, длина | Обязательность значения | Логическое ограничение | Примеры данных |
| profile\_id | PK | int | not null | Unique | 1, 2, 3 |
| user\_id |  | int | not null | references Users(id) | 1, 2, 3 |

Таблица 14 – Техническое описание таблицы «password\_reset\_tokens»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Ключ | Тип, длина | Обязательность значения | Логическое ограничение | Примеры данных |
| id | PK | int | not null | Unique | 1, 2, 3 |
| token |  | varchar  (50) | not null |  | \dvd56dv6dvs2d |
| user\_id |  | int | not null | references Users(id) | 1, 2, 3 |
| expiresAt |  | datetime | not null |  | 02-05-25-20-34 |

Таблица 15 – Техническое описание таблицы «subscriptions»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Ключ | Тип, длина | Обязательность значения | Логическое ограничение | Примеры данных |
| id | PK | int | not null | Unique | 1, 2, 3 |
| email |  | varchar  (100) | not null |  | ivan@mail.ru |

### 3.2.2 Техническое обеспечение

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;
* объем оперативного запоминающего устройства 1024 Мбайт или выше;
* объем на жестком 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.

### 3.2.3 Программное обеспечение

Для создания веб-сайта используются различные технологии, инструменты и языки программирования.

HTML – язык разметки, определяющий структуру и содержимое веб-страниц. Он прост, широко распространен и обеспечивает интуитивно понятный интерфейс. HTML поддерживает кросс-браузерную совместимость.

CSS – язык стилей, отвечающий за внешний вид элементов: цвета, шрифты, расположение. CSS позволяет разделять структуру и оформление, что упрощает поддержку кода и его изменение. Использование CSS-фреймворков ускоряет разработку и обеспечивает единообразие стиля.

JavaScript – язык программирования для интерактивности веб-приложений. С его помощью реализуются динамическая обработка данных, валидация форм и другие функции. JavaScript улучшает пользовательский опыт и позволяет создавать динамичные интерфейсы. Использование JavaScript-фреймворков повышает эффективность разработки.

Node.js – серверная платформа, позволяющая использовать JavaScript на сервере. Это обеспечивает высокую производительность благодаря асинхронной обработке запросов.

PostgreSQL – объектно-реляционная СУБД с открытым исходным кодом. Позволяет создавать, изменять и удалять записи, а также отправлять транзакции – наборы последовательных запросов на языке SQL..

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.

4 Рабочий проект

4.1. Разработка 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 четкую 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 верстки сайта.

«section» – это 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» этот 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 11 и 13, 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).

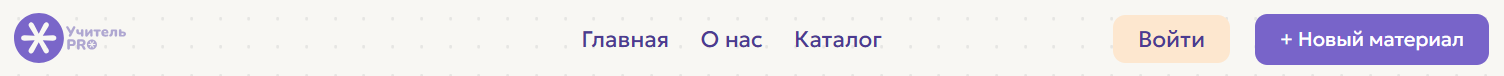


Рисунок 11 – Шаблон «шапки» сайта

Данный шаблон хедера реализован фрагментом кода, изображенном на рисунке 12.



Рисунок 12 – Код реализации шаблона «шапки» сайта

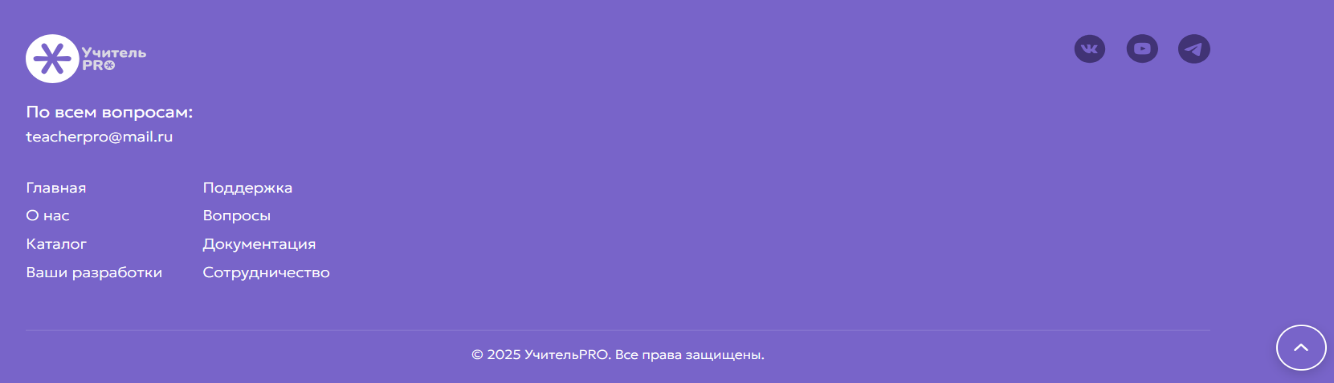


Рисунок 13 – Шаблон «подвала» сайта

Данный шаблон футера реализован кодом, изображенном на рисунке 14.



Рисунок 14 – 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 15.

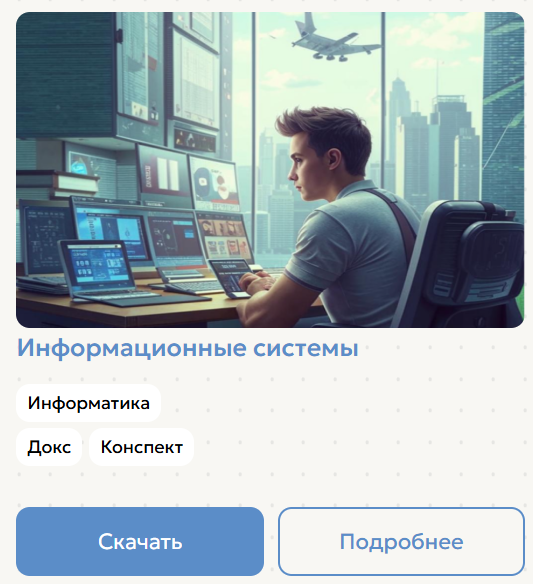


Рисунок 15 – Шаблон карточки разработки в каталоге

Данный шаблон карточки реализован фрагментом кода, изображенном на рисунке 16.



Рисунок 16 – Код реализации шаблона карточки

Детальная страница разработки, изображенная на рисунке 17, 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 17 – Шаблон детальной карточки

Данный шаблон реализован фрагментом кода, изображенном на рисунке 18.



Рисунок 18 – Код реализации шаблона подробной карточки

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 19.

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

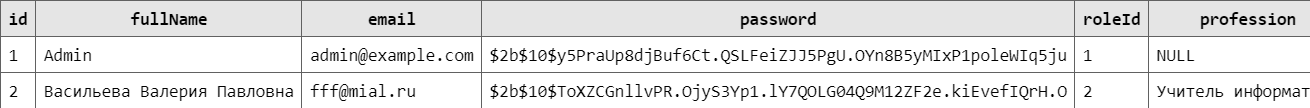


Рисунок 19 – Демонстрация хеширования пароля

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 20.



Рисунок 20 – 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 21.

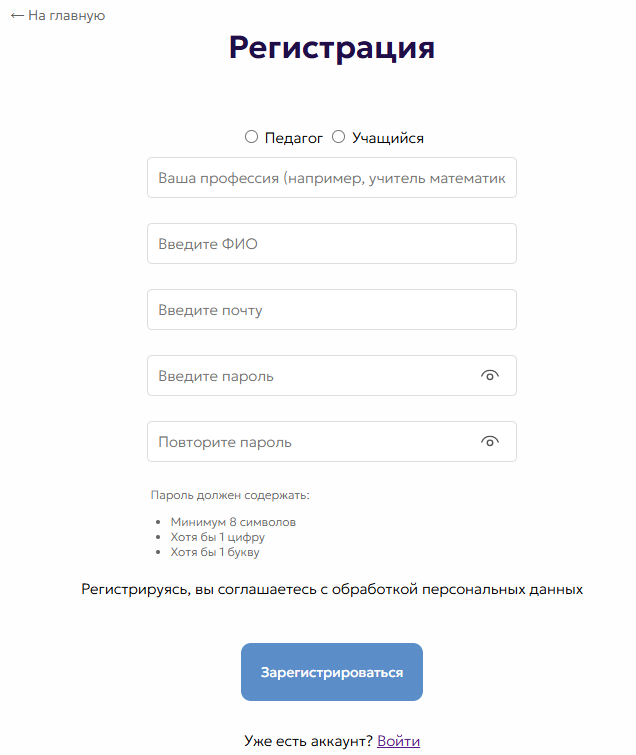


Рисунок 21 – Форма регистрации

На рисунке 22 представлен код реализации отображения разработок в личном кабинете пользователя.

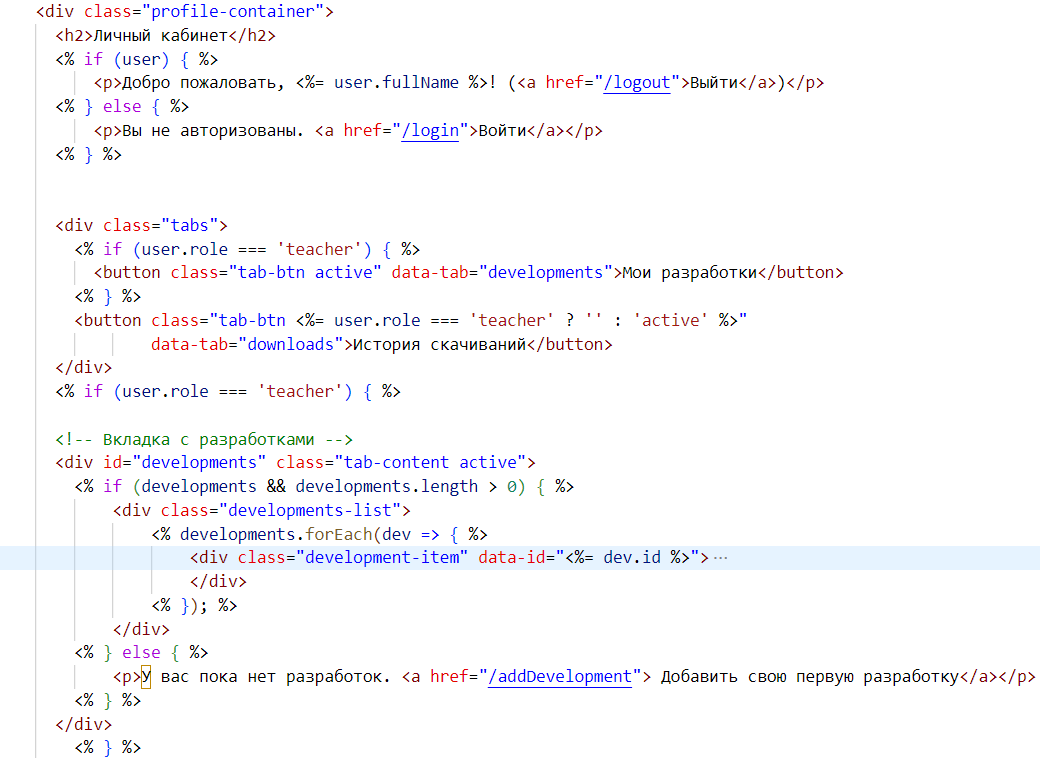


Рисунок 22 – Код реализации отображения разработок в личном кабинете

Рисунок 23 демонстрирует реализацию макета личного кабинета, где представлены основные разделы, позволяющие пользователям просматривать свои ресурсы.

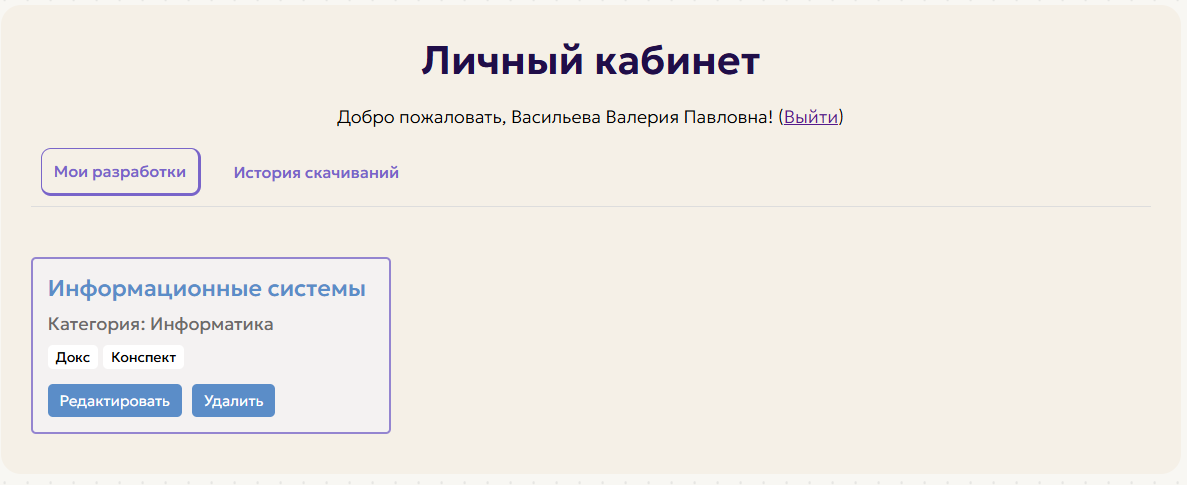


Рисунок 23 – Реализация макета личного кабинета

На рисунке 24 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 24 – Код реализации отображения каталога

На рисунке 25 представлена структура и дизайн пользовательского интерфейса, который обеспечивает доступ к образовательным ресурсам. Он включает в себя: функцию поиска, фильтры и сортировку, список ресурсов.

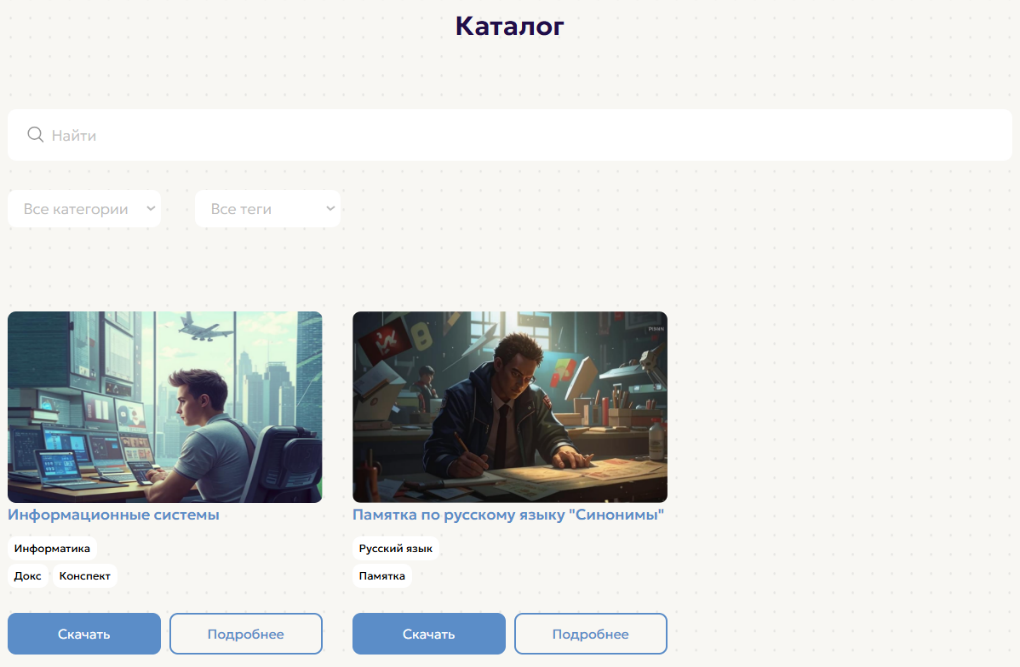


Рисунок 25 – Реализация макета каталога

Учитывая, что веб-приложение может открываться через мобильное устройство, все страницы сайта должны быть адаптированы и корректно отображаться на телефонах. На рисунке 34 представлен каталог в мобильном варианте.

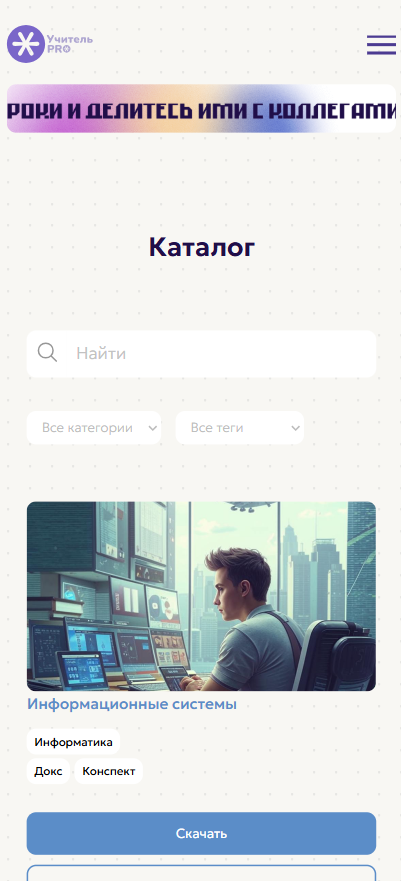


Рисунок 26 – Реализация мобильной версии каталога

На рисунке 27 представлен фрагмент кода добавления ресурса через форму, он структурирован для обработки и валидации данных.



Рисунок 27 – Код реализации добавления ресурса через форму

Рисунок 28 демонстрирует форму для загрузки ресурсов с текстовыми полями, кнопками и инструкциями.

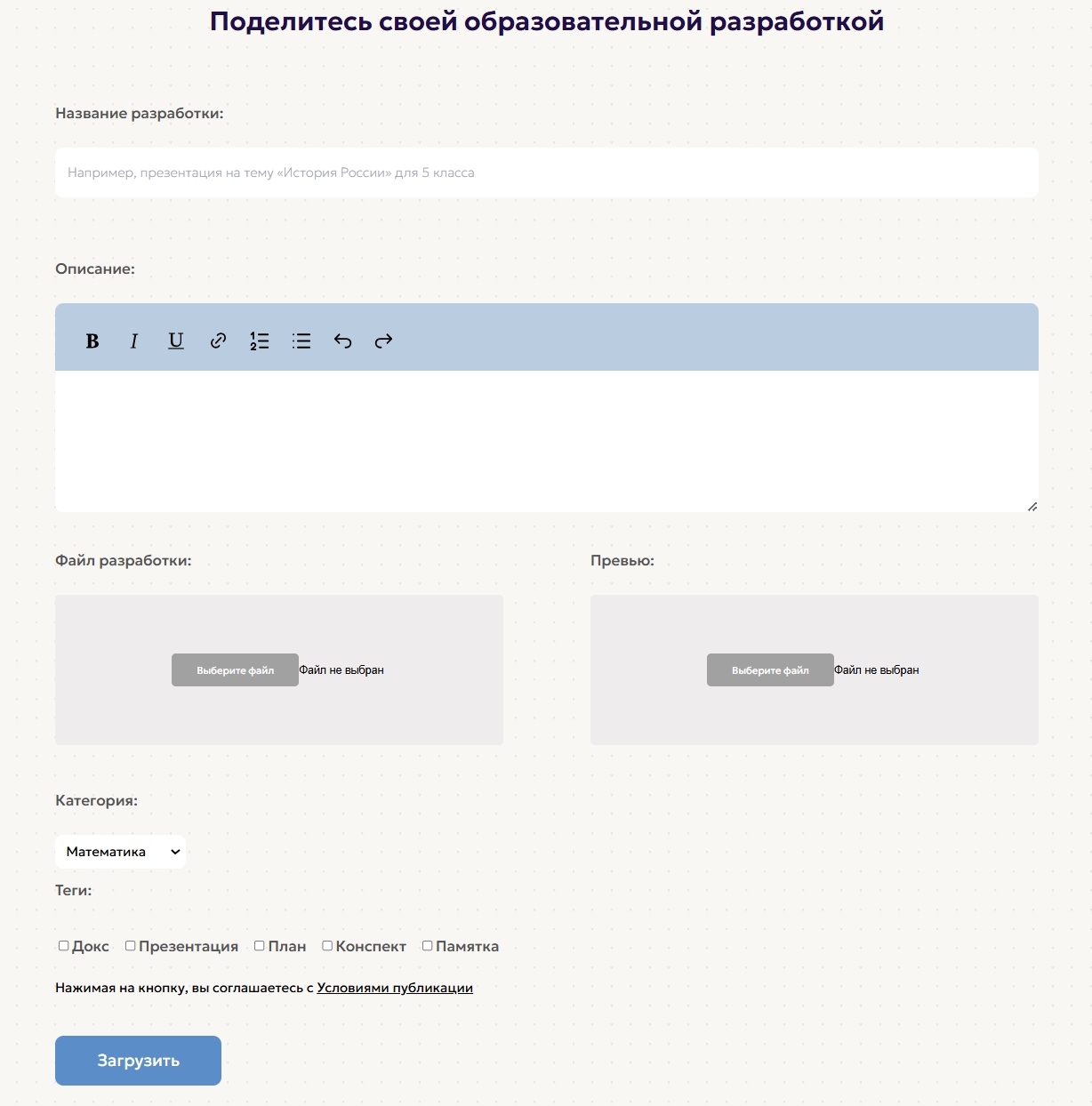


Рисунок 28 – Реализация макета формы для загрузки ресурсов

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 29. 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.

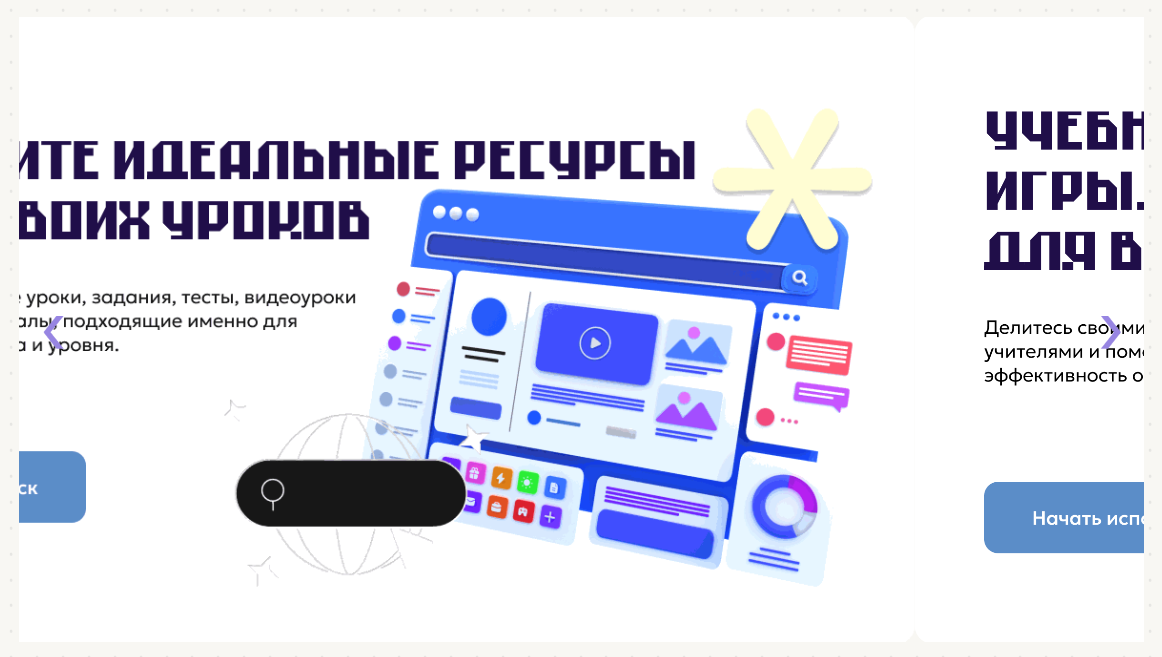


Рисунок 29 – Реализация слайдера на сайте

Для реализации слайдера была выбрана одна из популярных библиотек, таких как Slick, что обеспечивает высокую производительность и гибкость, код представлен на рисунке 30. 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 30 – 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 31.

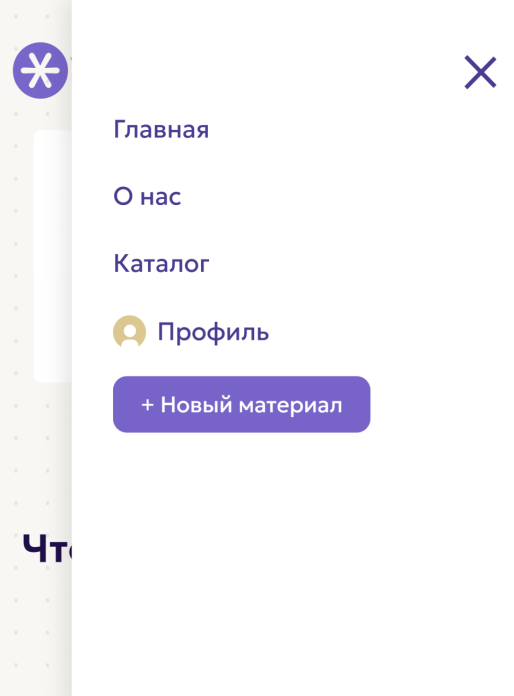


Рисунок 31 – Реализация меню на мобильном устройстве

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 32, представлен скрипт для работы меню в мобильной версии.



Рисунок 32 – Скрипт реализации меню на мобильном устройстве

## 4.2 Логика работы веб-приложения

Веб-приложение состоит из нескольких модулей и функциональных файлов, реализующих основные функции системы. Архитектура приложения построена на принципе клиент-серверной модели, где серверная часть реализована на базе Node.js, а фронтенд – с использованием HTML, CSS, JavaScript и современных библиотек. Взаимодействие между модулями осуществляется через маршруты и API, обеспечивая динамическое обновление данных и взаимодействие с пользователем.

Основные файлы и функции веб-приложения:

Главная страница (index.ejs) – центральный модуль, который отображает такие блоки, как слайдер с баннерами, информация об услугах и перечень преимуществ платформы, раздел с информацией об услугах, форма подписки на рассылку, целевая аудитория пользователей и примеры разработок.

Каждая страница вызывает вызывает частичные шаблоны header и footer.

Страницы авторизации и регистрации (login.ejs, register.ejs, forgot-password.ejs, reset-password.ejs) – обеспечивают вход, регистрацию и восстановление пароля. Эти страницы вызывают общие части интерфейса.

Профиль пользователя (profile.ejs) – отображает личные данные, историю активности и позволяет редактировать информацию.

Каталог(catalog.ejs) – отображающий список образовательных ресурсов, с возможностью поиска и фильтрации.

Добавление разработки (addDevelopment.ejs) – модуль для загрузки новых образовательных материалов.

Административная панель (admin.ejs) – управленческий модуль для администраторов, включает управление пользователями и контентом.

Файлы скриптов и стилей (script.js, CSS) подключаются к страницам через шаблоны и обеспечивают динамическое взаимодействие и оформление.

Клиентская часть веб-приложения имеет следующую структуру файлов, представленную на рисунке 33.

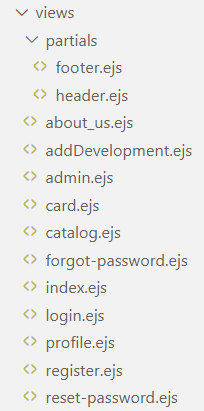


Рисунок 33 – Структура клиентской части веб-приложения

Создано серверное приложение на Node.js с использованием Express.js.

Важной частью является создание обработчика запросов для аутентификации (auth) и регистрации (reg) пользователей в системе.

Аутентификация (auth). Этот обработчик используется для проверки аутентификационных данных пользователя (телефон и пароль). При получении запроса извлекается телефон из тела запроса и выполняется запрос к базе данных для поиска пользователя по этому номеру телефона. Если пользователь не найден или пароль не совпадает с хэшированным паролем в базе данных, возвращается соответствующее сообщение об ошибке. Если аутентификация проходит успешно, генерируется токен доступа с помощью функции generateAccessToken и возвращается вместе с информацией о пользователе.

Регистрация (reg). Этот обработчик используется для регистрации новых пользователей в системе. При получении запроса извлекаются имя, телефон и пароль из тела запроса. Выполняется запрос к базе данных для проверки, существует ли уже пользователь с указанным телефоном. Если пользователь уже существует, возвращается сообщение об ошибке. В противном случае пароль хэшируется с помощью bcrypt, и данные о пользователе добавляются в базу данных. Также создается запись в таблице users для нового пользователя. После успешной регистрации генерируется токен доступа и возвращается вместе с информацией о пользователе и сообщением о успешной регистрации.

Также важной частью серверной части является создание обработчиков маршрутов для осуществления запросов к сервеной части через фронтенд часть системы.

GET /index – отображение главной страницы сайта.

GET /login – страница входа пользователя.

GET /catalog – просмотр каталога образовательных ресурсов.

GET /card – страница с подробной информацией о конкретном ресурсе.

GET /download/:id – скачивание ресурса по уникальному идентификатору.

GET /about\_us – страница с информацией о компании или проекте.

GET /register – форма регистрации нового пользователя.

GET /addDevelopment – страница для добавления нового образовательного ресурса.

GET /user/developments/:userId – просмотр всех разработок пользователя по его ID.

GET /admin – административная панель управления сайтом.

GET /user/developments/edit/:id – страница редактирования конкретной разработки.

GET /user/categories – просмотр категорий ресурсов пользователя.

GET /profile – страница профиля текущего пользователя.

POST /subscribe – подписка на рассылки или уведомления.

POST /user/add/development – отправка формы для добавления нового ресурса.

POST /user/developments/delete/:id – удаление ресурса по ID.

POST /user/developments/update/:id – обновление информации о ресурсе.

POST /register – обработка данных регистрации нового пользователя.

POST /login – обработка входа пользователя.

POST /forgot-password – запрос на восстановление пароля.

POST /reset-password – установка нового пароля после восстановления.

Серверная часть веб-приложения имеет следующую структуру файлов, представленную на рисунке 34.

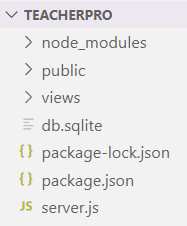


Рисунок 34 – Структура серверной части веб-приложения

Код разрабатываемой системы представлен в приложении Н.

## 4.3 Руководство системному программисту

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 – PostgreSQL.

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 для 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.

Данные для контрольного примера:

* пользователь: test\_user;
* пароль: Test1234;
* файл для загрузки: sample\_material.pdf.

Ожидаемый результат: успешная регистрация, вход в систему, загрузка файла, его отображение в личном кабинете, возможность скачивания.

Действия:

1. Зарегистрировать нового пользователя.
2. Войти в систему под этим пользователем.
3. Загрузить тестовый файл.
4. Проверить отображение файла в личном кабинете.
5. Скачать файл и убедиться в его целостности.

Результаты выполнения действий фиксируются в логах и скриншотах. В случае успешного выполнения – они будут прикреплены в приложении Л. В случае ошибок – будут зафиксированы сообщения об ошибках и предприняты необходимые меры по устранению.

Таблица 16 – Вывод сообщений для системного программиста

|  |  |  |
| --- | --- | --- |
| Этап | Сообщение | Действия при сообщении |
| Установка ОС | «Операционная система успешно установлена.» | Проверить корректность установки, выполнить обновление системы. |
| Установка зависимостей | «Все необходимые пакеты установлены.» | Проверить наличие всех зависимостей, устранить ошибки. |
| Настройка веб-сервера | «Веб-сервер успешно настроен и запущен.» | Проверить работу сайта через браузер, убедиться в корректной работе SSL. |
| Развертывание приложения | «Веб-приложение успешно запущено.» | Проверить доступность сайта, выполнить тестовые сценарии. |
| Ошибки при запуске | «Ошибка при запуске сервера: [описание ошибки].» | Проанализировать лог-файлы, устранить проблему, повторить запуск. |

Это руководство поможет системному программисту обеспечить стабильную работу и своевременное сопровождение веб-приложения «УчительPRO», а также быстро реагировать на возможные сбои и ошибки.

## 4.4 Руководство пользователя

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 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.

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».

Для запуска веб-сайта «УчительPRO» необходимо в браузере перейти по адресу, полученному в ходе запуска интерпретатора локально или от администратора, развернувшего сайт глобально. После ввода адреса на экране появится главная страница, представленная на рисунке 35, на которой находится главный слайдер с банерами, информация которых, сообщает о назначении сайта и его функциях.



Рисунок 35 – Главная страница

Далее у пользователя есть выбор, авторизоваться или продолжить как гость, но с ограниченными функциями. Попытки добавить новый материал, используя кнопку в шапке или открыть подробную информацию о разработке в каталоге, приведут пользователя к формам авторизации и регистрации с целью расширения функциональных возможностей, заполненная форма авторизации представлена на рисунке 36.

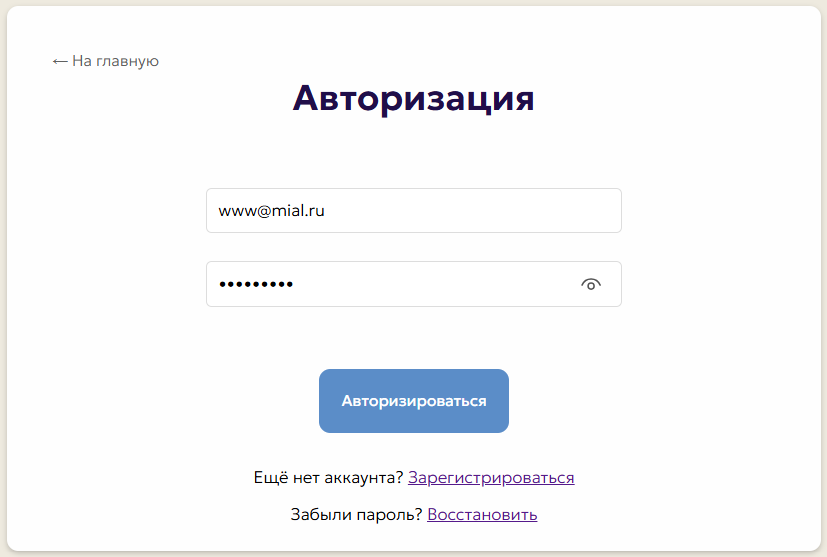


Рисунок 36 – Форма входа

В случае ввода неправильных данных, форма выдает сообщение об ошибке, оно показано на рисунке 37.

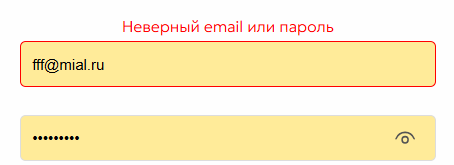


Рисунок 37 – Сообщение об ошибке

После авторизации, или, если нет аккаунта, регистрации, пользователь попадает в личный кабинет. Стоит помнить, что доступ к некоторым функциям зависит от роли пользователя, которую пользователь выбирает при регистрации. Профили педагога и учащегося почти не отличаются, пример профиля педагога показан на рисунке 38.

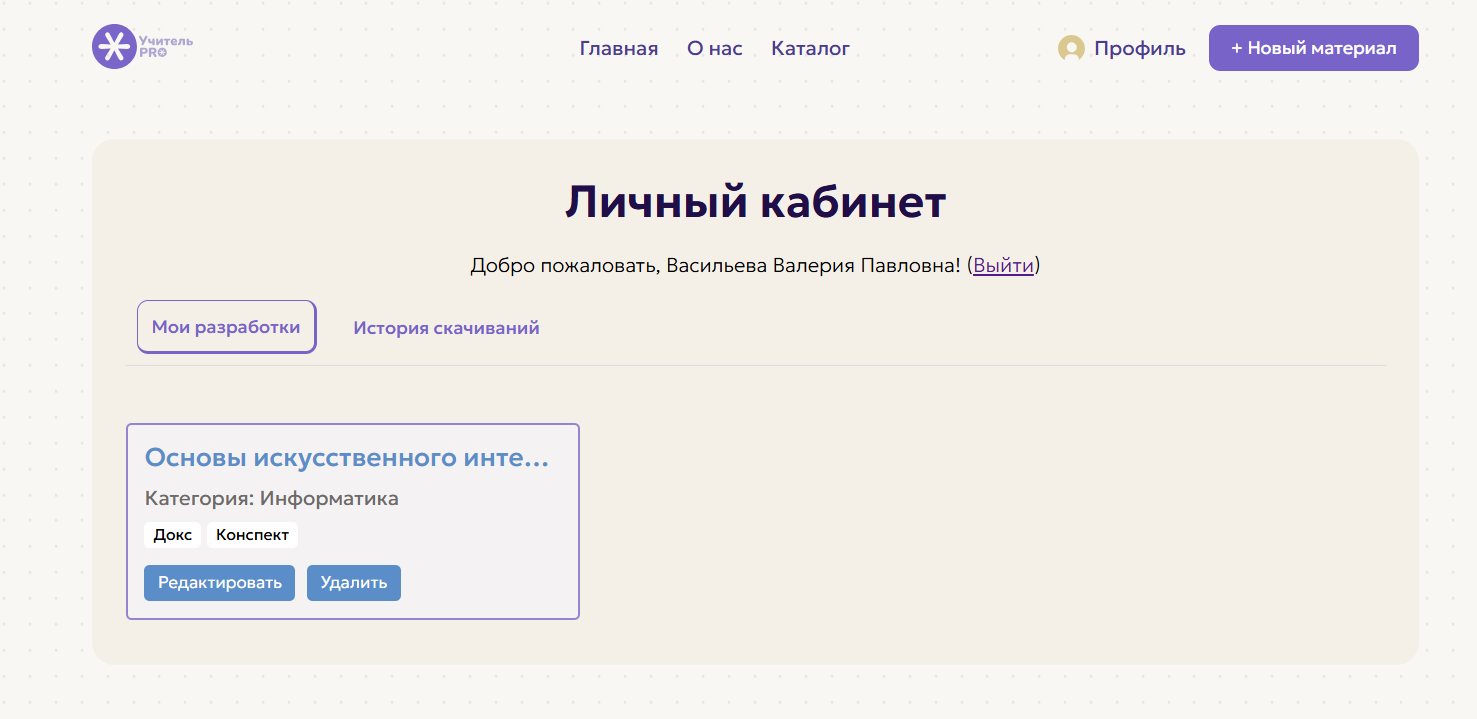


Рисунок 38 – Личный профиль пользователя-педагога

В случае профиля учащегося, у него отсутствует функция публикации собственных разработок и соответственно отсутствует раздел «Мои разработки». Если ученик попытается использовать функцию публикации, то увидит сообщение об ограничении функционала, представленное на рисунке 39.

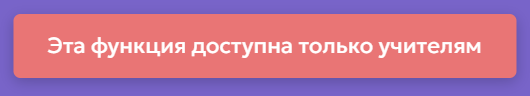


Рисунок 39 – Сообщение об ограничении

Для всех типов пользователей доступен раздел «История скачиваний», в котором отражается список скаченных разработок с датой и с возможностью скачать файл повторно, данный раздел представлен на рисунке 40.

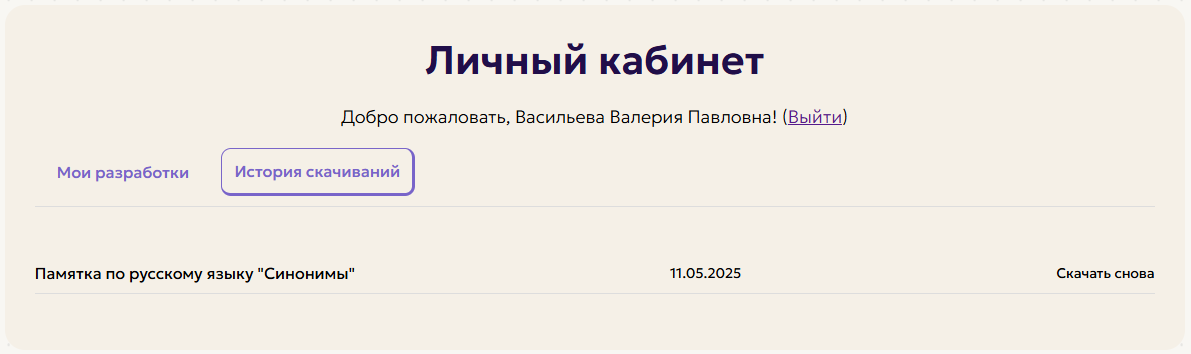


Рисунок 40 – Раздел «История скачиваний»

Также всем пользователям доступна функция просмотра каталога и скачивания файла разработки. Для удобства предусмотрены фильтры и поиск. Каталог продемонстрирован на рисунке 41.

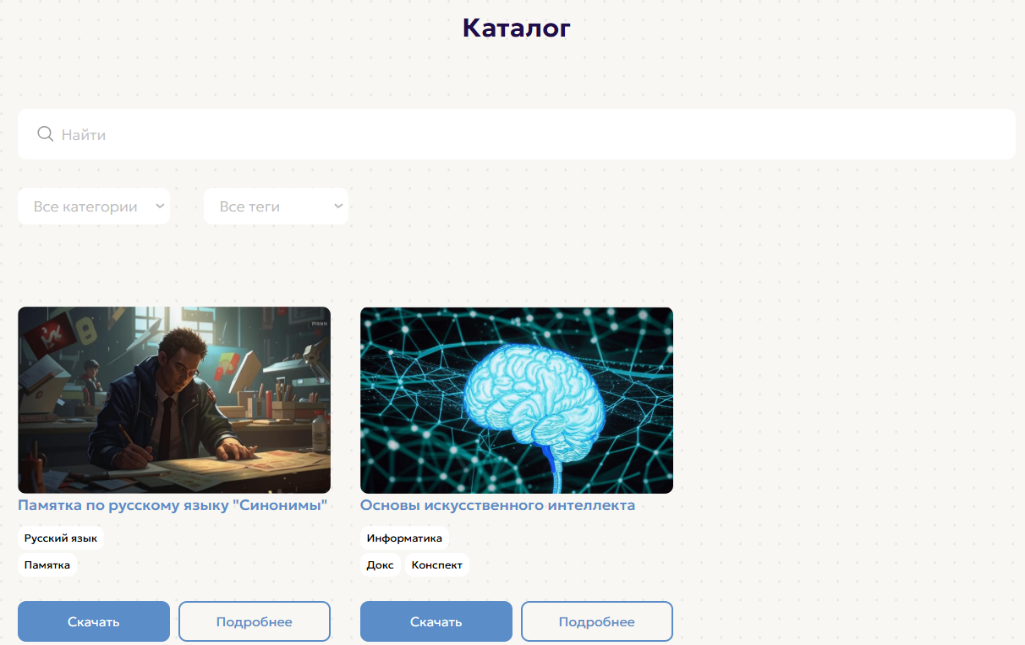


Рисунок 41 – Страница каталога

В карточке разработки есть кнопка «Подробнее», которая ведет на страницу с подробным описанием содержимого файла, неавторизованные пользователи могут скачать файл, но для просмотра подробной информации необходимо авторизоваться, функция доступна как педагогу, так и ученику, данная траница показана на рисунке 42.

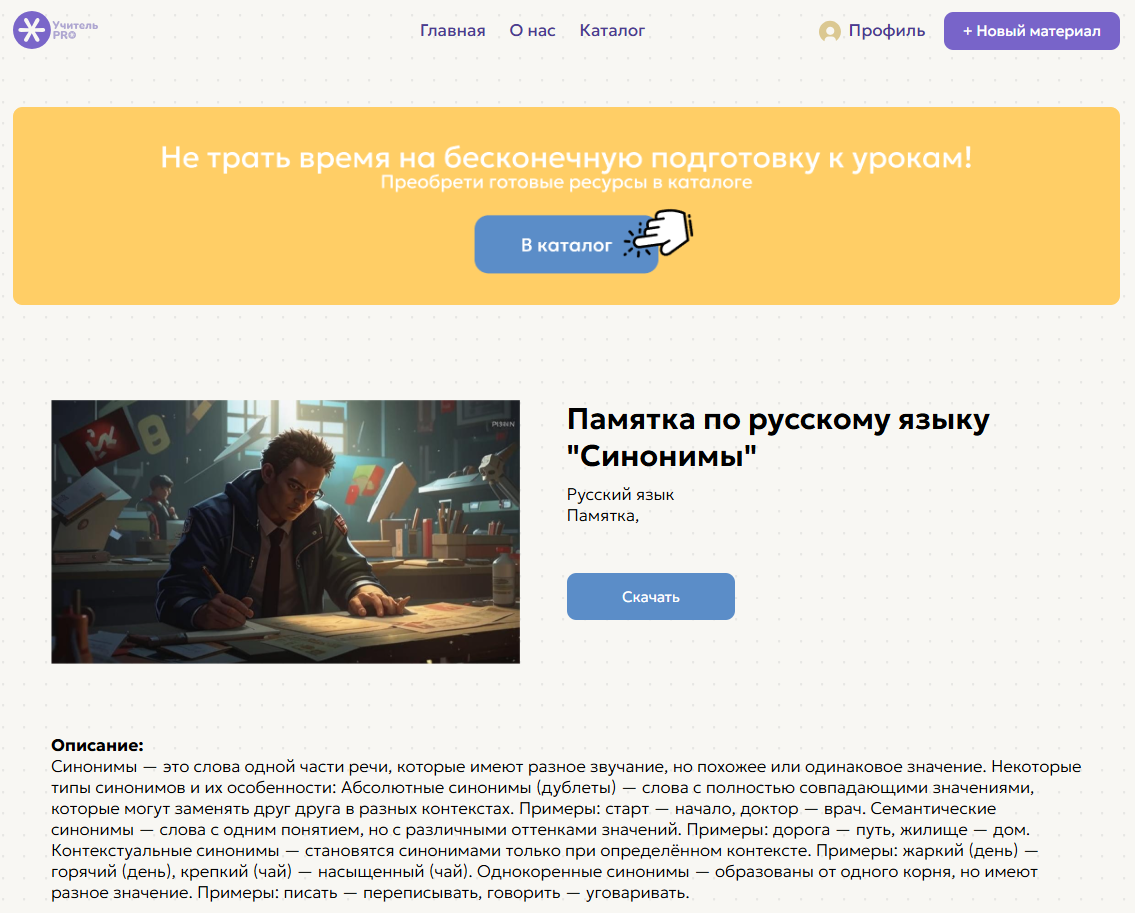


Рисунок 42 – Страница разработки

## 4.5 Ревьюирование программного кода

Был проанализирован следующий фрагмент программного кода, представленного на рисунке 43. Данный код обрабатывает отправку формы для добавления разработки. Он использует fetch для отправки данных на сервер и отображает сообщения об успехе или ошибке в зависимости от ответа сервера. Код написан с использованием асинхронных функций, что делает его более современным и удобным для чтения.



Рисунок 43 – Фрагмент кода добавления разработки до ревьюирования

В результате ревьюирования были сделаны выводы по разным аспектам, данные выводы приведены далее.

Код в целом работает корректно, однако есть несколько моментов, которые можно улучшить. Например, не обрабатываются случаи, когда ответ от сервера не является JSON, что может привести к ошибкам.

Код работает эффективно для небольших форм. Однако, если форма будет содержать большое количество данных, стоит рассмотреть возможность оптимизации, например, путем использования async/await для обработки нескольких запросов.

Структура кода логична и последовательна. Обработчик событий четко разделяет логику обработки формы и отображения сообщений. Однако, можно выделить отдельные функции для обработки успеха и ошибки, что улучшит читаемость и тестируемость кода.

Стиль кода в целом соответствует современным стандартам JavaScript. Используются шаблонные строки для формирования HTML, что делает код более читаемым. Однако, стоит избегать использования innerHTML для вставки данных, чтобы предотвратить возможные уязвимости, связанные с XSS.

В результате ревьюирования программный код был приведен к следующему виду, представленному на рисунке 44.



Рисунок 44 – Фрагмент кода добавления разработки после ревьюирования

Код выполняет свою задачу, но есть возможности для улучшения. Рекомендуется выделить функции для обработки успеха и ошибок, улучшить безопасность, а также добавить более детализированную обработку ошибок. Эти изменения сделают код более надежным и удобным для дальнейшей работы.

Данный, на рисунке 45, код реализует маршрут для получения разработок пользователя по его идентификатору. Он использует асинхронную функцию для обработки запроса и возвращает данные в формате JSON. В случае ошибки, код обрабатывает исключение и возвращает статус 500 с сообщением об ошибке.

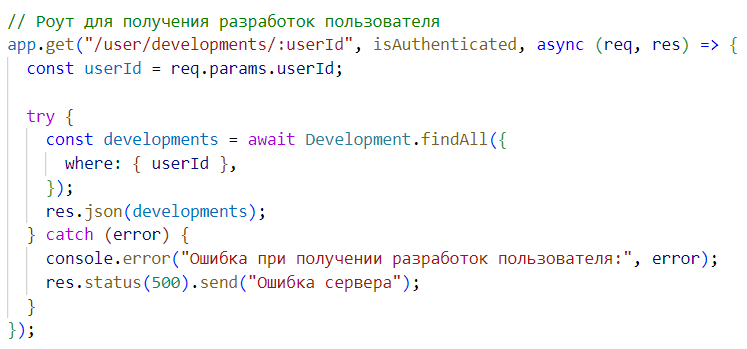


Рисунок 45 – Фрагмент кода получения разработки до ревьюирования

Код в целом работает корректно, однако есть несколько моментов, которые можно улучшить. Например, не проверяется, существует ли пользователь с данным идентификатором, что может привести к возврату пустого массива, если пользователь не найден.

Структура кода ясная и логичная. Однако, для повышения модульности, можно вынести логику получения разработок в отдельную функцию. Это упростит тестирование и повторное использование кода.

Читаемость кода хорошая, но можно улучшить, добавив более подробные комментарии и используя более явные имена переменных. Например, вместо userId можно использовать userIdentifier, чтобы сделать код более самодокументируемым.

Код использует middleware isAuthenticated, что является хорошей практикой для защиты маршрута. Однако, стоит также проверить, что userId является допустимым идентификатором, чтобы избежать возможных уязвимостей.

В результате ревьюирования программный код был приведен к следующему виду, представленному на рисунке 46.

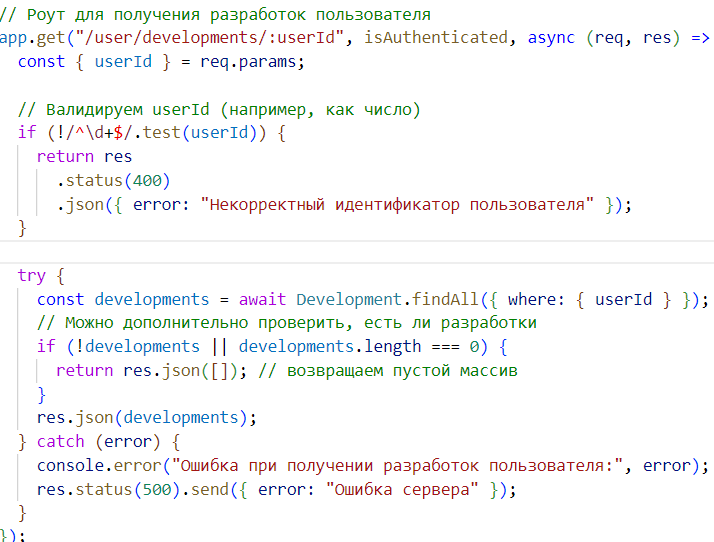


Рисунок 46 – Фрагмент кода получения разработки после ревьюирования

В целом, код выполняет свою задачу, но есть возможности для улучшения.

Данный, на рисунке 47, код реализует маршрут для страницы каталога, который извлекает данные о разработках, категориях и тегах из базы данных и отображает их на странице. Код написан с использованием асинхронных функций и включает обработку ошибок.

Код корректно обрабатывает ошибки, однако, в случае возникновения ошибки, пользователю отправляется общее сообщение «Ошибка сервера». Это может затруднить диагностику проблемы. Рекомендуется предоставить более детальную информацию о типе ошибки в логах, но не раскрывать ее пользователю.



Рисунок 47 – Фрагмент кода получения каталога до ревьюирования

Структура кода логична, но можно выделить отдельные функции для получения категорий и тегов, чтобы уменьшить количество асинхронных вызовов в одном месте. Это улучшит читаемость и тестируемость кода.

Код не содержит явных уязвимостей, однако, стоит убедиться, что данные, получаемые от пользователя (например, req.session.user), корректно проверяются и валидируются, чтобы избежать возможных атак.

В результате ревьюирования программный код был приведен к следующему виду, представленному на рисунке 48.



Рисунок 48 – Фрагмент кода получения каталога после ревьюирования

Код маршрута для страницы каталога написан с использованием современных подходов, однако, его можно улучшить с точки зрения читаемости, производительности и обработки ошибок.

## 4.6 Рефакторинг программного кода

Цель рефакторинга: улучшение поддерживаемости, производительности и читаемости кода без изменения внешнего поведения приложения.

Рефакторинг кода модуля загрузки файлов.

Исходный код (проблемы):

* смешение логики (проверки, логирование, генерация имени файла);
* дублирование кода обработки ошибок;
* отсутствие реиспользуемых компонентов;
* прямая работа с файловой системой внутри роута.

## 4.7 Программа и методика испытания веб-приложения

Объектом испытаний является веб-приложение «УчительPRO», предназначенное для публикации, обмена образовательными ресурсами.

Целью проведения испытаний является подтверждение соответствия системы техническому заданию, обеспечение надежности, безопасности, производительности и удобства использования веб-приложения.

Задачи включают проверку функциональности, отзывчивости интерфейса, безопасности, совместимости, производительности и полноты документации. В ходе испытаний необходимо выявить и устранить возможные дефекты, обеспечить выполнение требований ТЗ и подготовить систему к эксплуатации.

Испытания включают следующие этапы:

1. Подготовительный этап – сбор тестовых данных, подготовка тестовой среды, настройка автоматизированных средств тестирования.
2. Функциональное тестирование – проверка выполнения всех функций системы согласно ТЗ, включая регистрацию, авторизацию, загрузку и скачивание ресурсов, поиск и фильтрацию.
3. Тестирование интерфейса – оценка отзывчивости и удобства дизайна, соответствия макетам и прототипам.
4. Тестирование производительности – нагрузочное тестирование системы для определения пределов пропускной способности и времени отклика.
5. Тестирование безопасности – проверка защиты данных, устойчивости к атакам, уязвимостей.
6. Тестирование совместимости – проверка работы системы на различных браузерах и устройствах.
7. Проверка документации – оценка полноты и актуальности технической и пользовательской документации.

Испытания проводятся в специально подготовленной тестовой среде, которая максимально имитирует рабочие условия эксплуатации. Продолжительность испытаний – 2 недели. Испытания начинаются после завершения этапа разработки и подготовки тестовых данных. Перед началом каждого этапа проводится согласование условий и целей. Завершение этапов фиксируется актами испытаний, а результаты документируются в отчётах.

Общий порядок проведения испытаний предполагает последовательное выполнение всех этапов с обязательным контролем и документированием результатов. После завершения всех тестов проводится итоговая оценка готовности системы к эксплуатации и подготовка отчета о выполнении программы испытаний.

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.

В таблице 17 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 17 – Проверка корректной регистрации нового пользователя

|  |  |  |  |
| --- | --- | --- | --- |
| №1 | Проверка корректной регистрации нового пользователя | | |
| Ссылка на ТЗ | Васильева В. П. | Высокий | Регистрация |
| 1. Выбрать роль: педагог или учащийся, в случае педагога также ввести профессию (например, учитель физики). 2. Ввести в поля ФИО и почту корректные данные (например, «Анна Петрова», «anna.petrova @mail.ru»). 3. Ввести пароль (например, «Пароль123»). 4. Ввести пароль повторно для проверки. 5. 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; * после авторизации может пользоваться личным профилем. |

На рисунке 47 представлена работа формы регистрации пользователя, на которой видно, как пользователь с ролью педагога вводит свои личные данные.

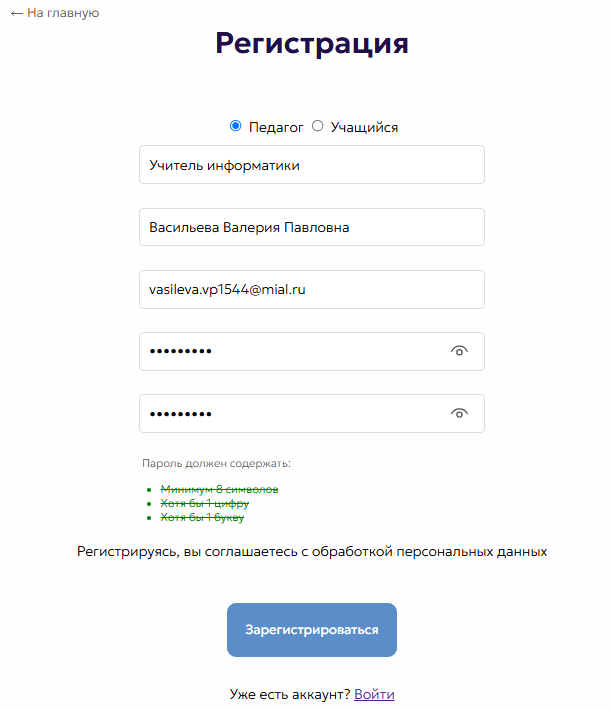


Рисунок 47 ‒ Работа формы регистрации пользователя

На рисунке 48 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.

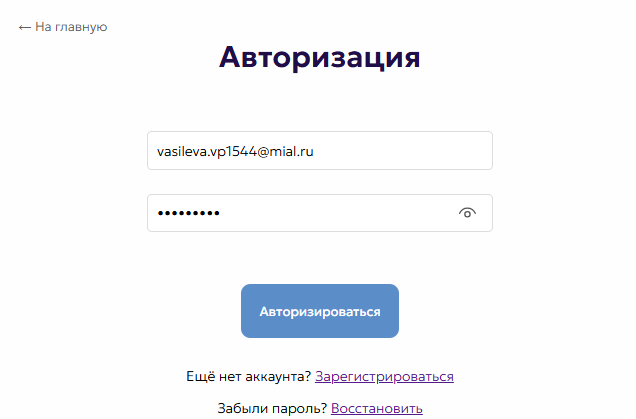


Рисунок 48 ‒ Форма авторизации

Рисунок 49 демонстрирует раздел отчетности о пользователях в административной панели. После регистрации, как упоминалось ранее, данные пользователя сохраняются в базе данных и используются для авторизации.

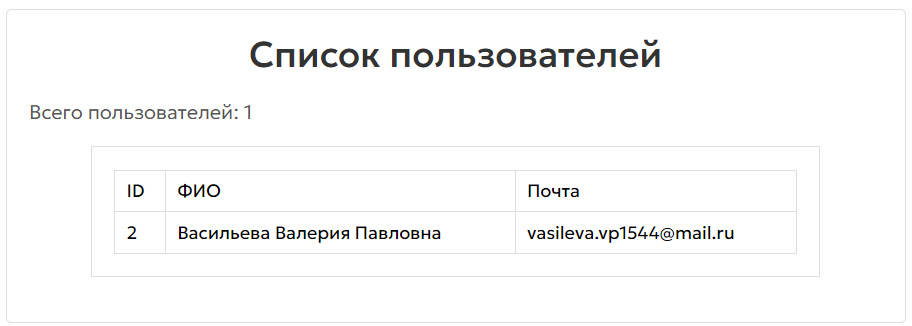


Рисунок 49 ‒ Отчетность о пользователях в админ-панели

В таблице 18 представлена информация о процессе проверки формы добавления разработки через администратора.

Таблица 18 – Проверка работы формы добавления разработки

|  |  |  |  |
| --- | --- | --- | --- |
| №2 | Проверка работы формы добавления разработки | | |
| 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. Получить сообщение об успешном добавлении, с предложением перейти в профиль. | | 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 корректно добавлены в базу данных; * карточка разработки корректно отображается в каталоге и профиле (превью, заголовок, категория, теги и кнопки); * все данные коректно обображаются на странице «подробнее» * загруженный файл успешно скачивается. |

Рисунок 50 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.

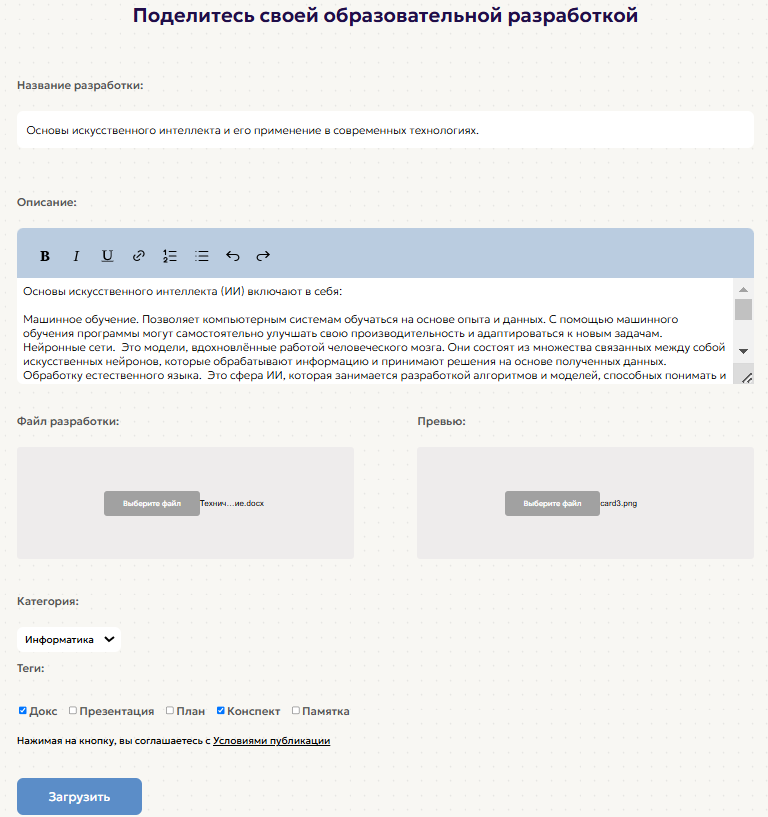


Рисунок 50 ‒ Работа формы для публикации образовательной разработки

После нажатия на кнопку «Загрузить», будет выведено соответствующее сообщение, как показано на рисунке 51.

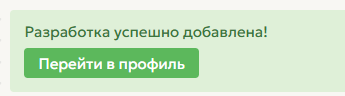


Рисунок 51 ‒ Работа формы для публикации образовательной разработки

Данные, были добавлены в базу данных, это представлено на рисунке 52.

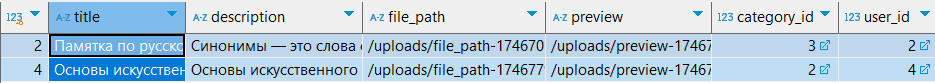


Рисунок 52 ‒ Отображение добавленных разработок в базе данных

На рисунке 53 представлен фрагмент каталога с отображением недавно добавленной разработки. Здесь можно увидеть, как новая разработка интегрируется в общую структуру каталога.

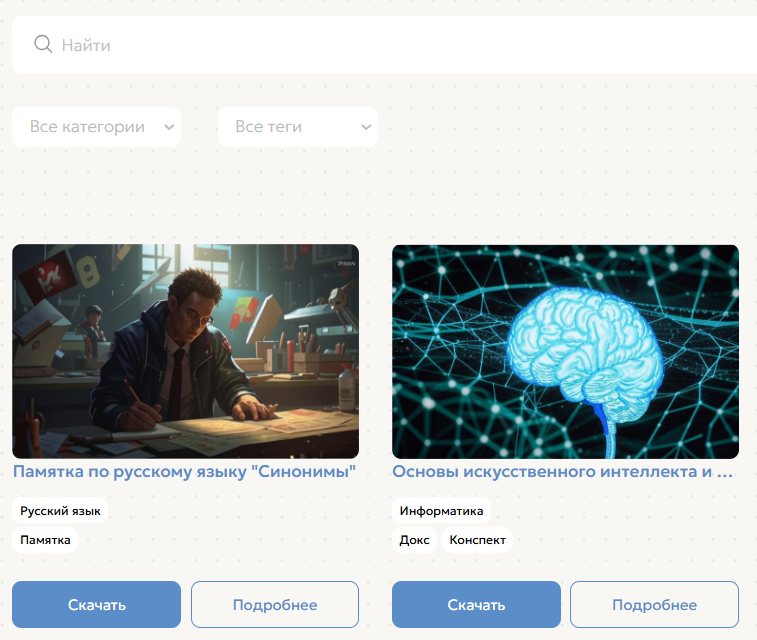


Рисунок 53 ‒ Отображение добавленной разработки в каталоге и профиле

В таблице 19 представлены результаты проверки функциональности удаления опубликованных разработок.

Таблица 19 – Проверка работы редактирования разработки

|  |  |  |  |
| --- | --- | --- | --- |
| №3 | Проверка работы редактирования разработки | | |
| Ссылка на ТЗ | Васильева В. П. | Высокий | Редактирование разработок |
| 1. Перейти в личный профиль, предварительно авторизовавшись; 2. В разделе «Мои разработки» выбрать опубликованную разработку для редактирования. 3. Нажать кнопку «редактировать». 4. В появившейся форме, исправить необходимые данные(например, название). 5. Нажать кнопку «Сохранить» 6. Получить уведомление об изменении данных. | | Ожидаемый результат: Разработка успешно отредактирована, данные в базе данных поменялись и корректно отображаются в каталоге и на всех связанных страницах. | Фактический результат:   * данные базы данных переписаны; * разработка корректно отображается в каталоге и связанных страницах. |

На рисунке 54 представлена форма редактирования, где для примера меняется название разработки

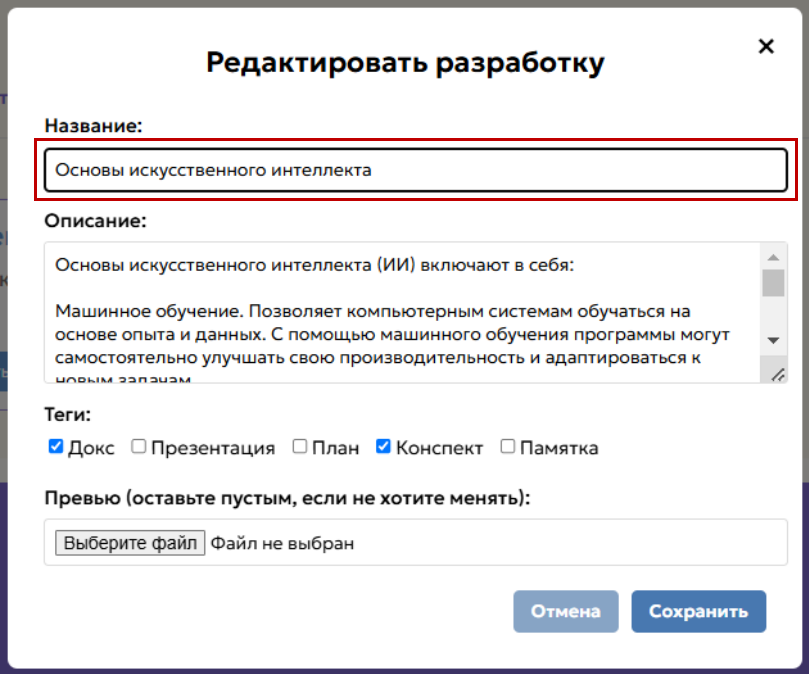


Рисунок 54‒ Форма редактирования разработки

После попытки отредактировать разработку в веб-приложении появляется окно с уведомлением об сохранении изменений, представлено на рисунке 55.

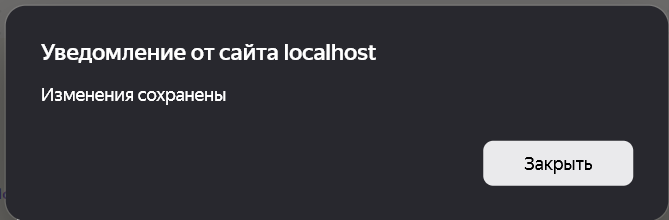


Рисунок 55‒ Уведомлении об изменениях

После редактирования данных происходит изменение самой разработки в каталоге, что представлено на рисунке 56.



Рисунок 56 ‒ Отсутствие удаленной разработки в каталоге

В таблице 20 представлены результаты проверки работы функции поиска и фильтрации в каталоге. Данная проверка была проведена для оценки эффективности и удобства использования этой ключевой функции в веб-приложении.

Таблица 20 – Проверка работы поиска и фильтров в каталоге

|  |  |  |  |
| --- | --- | --- | --- |
| №4 | Проверка работы поиска и фильтров в каталоге | | |
| Ссылка на ТЗ | Васильева В. П. | Высокий | Поиск |
| 1. Перейти в каталог разработок. 2. Ввести ключевое слово в строку поиска (например, «математика»). 3. Или выбрать любой из пунктов фильтров. 4. Нажать кнопку «Поиск». | | Ожидаемый результат: Список разработок фильтруется по введенному ключевому слову или фильтру 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 57.

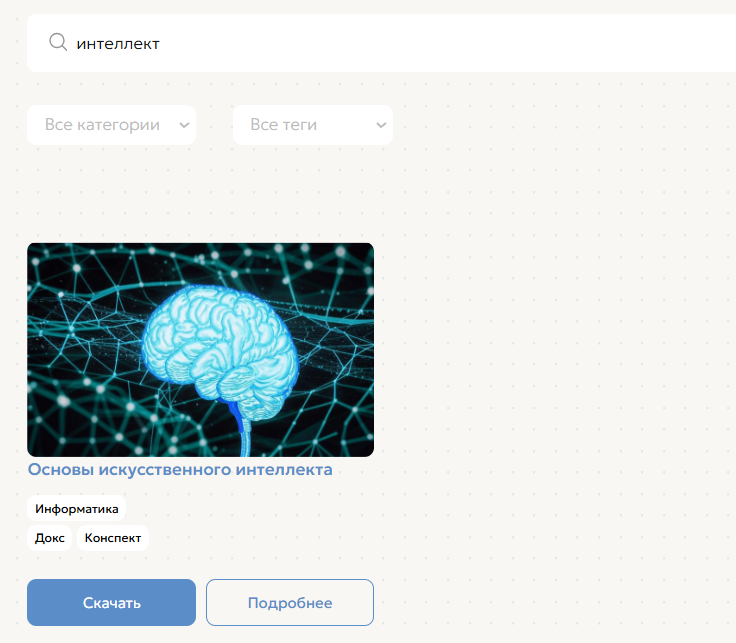


Рисунок 57 ‒ Отображение разработки, найденной по ключевому слову

Работа фильтров реализуется по тому же принципу, выбирая категорию или тег, остаются только релевантные разработки, представлено на рисунке 58.

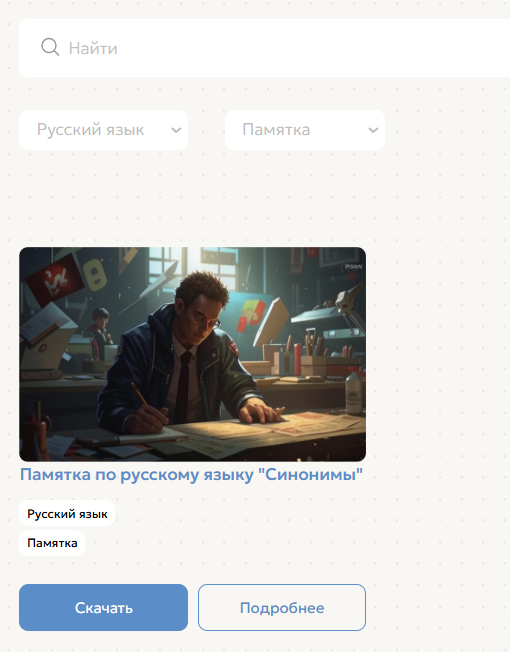


Рисунок 58 ‒ Демонстрация работы фильтров

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 Техника безопасности и пожарная безопасность

Обеспечение безопасности при установке, наладке, эксплуатации, обслуживании и ремонте технических средств является важнейшей составляющей безопасной работы информационных систем. В процессе эксплуатации серверного оборудования, сетевых устройств и другого технического оборудования необходимо соблюдать следующие требования:

* защита от воздействий электрического тока;
* защита от электромагнитных полей;
* защита от акустических шумов;
* допустимые уровни освещенности;
* вибрационные и шумовые нагрузки;
* обеспечение безопасных условий работы.

Все электроприборы должны быть подключены к заземленным розеткам, а кабели – правильно изолированы и закреплены, чтобы исключить риск короткого замыкания и поражения электрическим током. Использование автоматических выключателей и устройств защитного отключения (УЗО) обязательно для предотвращения аварийных ситуаций.

Технические средства должны располагаться в специально оборудованных помещениях, где уровень электромагнитных излучений не превышает допустимых норм, установленных санитарными стандартами. Это предотвращает влияние электромагнитных полей на здоровье сотрудников и работу оборудования.

Оборудование, создающее значительный уровень шума, должно располагаться в специально оборудованных помещениях или в шумоизоляционных шкафах. Уровень шума не должен превышать допустимых значений, установленных санитарными нормами, чтобы избежать вредных воздействий на здоровье работников.

В помещениях, где расположено оборудование и осуществляется обслуживание, должна поддерживаться оптимальная освещенность – не менее 300 люкс. Это обеспечивает комфортные условия для работы и предотвращает утомляемость глаз.

Оборудование должно устанавливаться на устойчивых и виброизоляционных основаниях, чтобы снизить вибрационные нагрузки и обеспечить стабильную работу техники. Вибрационные уровни не должны превышать установленные нормативами пределы.

В помещениях должны быть организованы безопасные пути эвакуации, наличие противопожарных средств (огнетушителей, автоматических систем пожаротушения), а также инструкции по безопасной эксплуатации оборудования.

Пожарная безопасность очень важна при работе с компьютерами и серверными комнатами. В помещениях с вычислительным оборудованием необходимо применять сертифицированные средства пожаротушения, например, порошковые или углекислотные огнетушители. Рекомендуется также внедрять автоматические системы пожаротушения, которые активируются при обнаружении дыма или повышения температуры.

Для предотвращения перегрева оборудования нужно поддерживать оптимальные параметры температуры (в диапазоне от 18 до 25 °C) и влажности воздуха (в пределах от 40 до 60 %). Перегрев может служить причиной возгорания, поэтому важно обеспечить надлежащую вентиляцию и кондиционирование.

Проведение периодических технических осмотров электрических сетей, кабелей и оборудования позволяет своевременно выявлять и устранять потенциальные источники коротких замыканий и перегрева. Все сотрудники должны проходить регулярные инструктажи по вопросам пожарной безопасности, знать местоположение средств пожаротушения и порядок действий при возникновении пожара.

Запрещается хранение легковоспламеняющихся материалов вблизи серверных и вычислительных помещений. Необходимо избегать перегрузки электрических сетей и некорректной эксплуатации оборудования. В помещениях должны быть разработаны и утверждены планы эвакуации, установлены соответствующие указатели и инструкции по действиям в случае пожара. Соблюдение требований пожарной безопасности и техники безопасности обеспечивает создание безопасных условий для работы персонала, а также надёжную защиту оборудования, что способствует бесперебойной и эффективной эксплуатации информационных систем.

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 – 02.05.2025г.
13. Данильчук, Е. В., Куликова, Н. Ю., Чернышова, М. В., Волков, Д. В. Обучение информатике в условиях виртуализации образовательного пространства
14. Современные проблемы науки и образования. 2019. № 6. [Электронный ресурс]. – Режим доступа: http://www.science-education.ru/ru/article/view?id=29323 – 22.04.2025г.
15. Helpiks.org: Достоинства и недостатки языка [Электронный ресурс]. – Режим доступа: https://helpiks.org/6-21879.html – 10.05.2025г.
16. Логическая структура БД. [Электронный ресурс]. – Режим доступа: https://intuit.ru/studies/courses/5/5/lecture/124?page=4/ – 05.05.2025г.
17. Методология IDEF0. [Электронный ресурс]. – Режим доступа: https://www.sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6\_2 – 05.05.2025г.
18. МОАУ «СОШ № 71» город Оренбург – Текст : электронный // orenschool71 : [Электронный ресурс]. – [Электронный ресурс]. – https://orenschool71.gosuslugi.ru/ – 24.04.2025г.
19. 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 – 23.04.2025г.
20. 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 – 29.04.2025г.
21. UML Use Case Diagrams [Электронный ресурс]. – Режим доступа: uml-diagrams.org – 25.04.2025г.

https://handyhost.ru/help/hosting/vremya-otveta-servera.html

**Приложение А**

***(обязательное)***

**Расчёт 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.

Сокращенные обозначения при расчете:

Зм – затраты на материалы;

Здоп – доподнительная заработная плата;

Зосн – основная заработная плата;

P – расчет электропотребления;

Змаш – затраты на машинное время;

Дн – дни разработки;

Ц – цена;

CиPaсх – расходы системы;

ЗатрДо – затраты до;

ЗатрП – затраты после;

Э – экономический результат;

Ток – точный срок окупаемости.

График выполнения работ представлен в таблице А.1.

Таблица А.1 – График выполнения работ по разработке «УчительPRO»

|  |  |  |
| --- | --- | --- |
| Наименование работ | Длительность работы | |
| Дни | Часы |
| Разработка технического задания | 1 | 6 |
| Планирование | 2 | 12 |
| Разработка бд | 3 | 18 |
| Рабочее проектирование | 15 | 90 |
| Отладка и тестирование | 5 | 30 |
| Обобщение и оценка результатов | 2 | 12 |

Таким образом, на разработку «УчительPRO» для МОАУ «COШ №71» было затрачено 28 дней или 168 часов (при шестичасовом 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;
* накладные расходы;
* затраты на машинное время (затраты на электроэнергию).

Перечень затрат на материалы и покупные изделия приведен в таблице А.2.

Таблица А.2 – 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:

(А.1)

К этой статье относится основная и дополнительная заработная плата разработчика веб-приложения. Результаты расчета фонда заработной платы представлены в таблице А.3.

Таблица А.3 – Расчет фонда заработной платы

|  |  |  |  |
| --- | --- | --- | --- |
| Должность: программист | Количество проработанных дней | Размер ежедневной платы, руб. | Заработная плата в месяц, руб |
| Основная заработная плата | 28 | 470 | 13160 |
| Дополнительная заработная плата (80% от основной) |  |  | 10528 |
| Итого |  |  | 23688 |

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.

(А.2)

Следовательно, разработчику сайта всего начислено 23688 рублей.

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:

(А.3)

Как следует из данных таблицы А.1, на разработку и последующую отладку веб-приложения «УчительPRO» для МОАУ «COШ №71» потребовалось 28 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 4 рубля 72 копейки.

При проведении расчетов в проекте необходимо в расчеты брать существующие на дату расчета тарифы. Суммарная мощность энергопотребителей для программиста складывается из мощности, потребляемой системным блоком персонального компьютера, монитором, принтером и другим периферийным оборудованием, которая составляет 1,5 кВт. Следовательно, за 6 часов работы программиста суммарное энергопотребление за день составит:

. (А.4)

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

(А.5)

Затраты на машинное время учитываются как затраты на электроэнергию.

В результате выше произведенных расчетов мы получили итоговые затраты на разработку, представленные в таблице А.4.

Таблица А.4 – Итоговая смета затрат

|  |  |
| --- | --- |
| Наименование статей расходов | Сумма, руб |
| Стоимость материалов и покупных изделий | 1170 |
| Основная заработная плата | 13160 |
| Дополнительная заработная плата | 10528 |
| Накладные расходы (20% от п.2.5) | 4737,6 |
| Затраты на машинное время (затраты на электроэнергию) | 1184,4 |
| Итого | 30780 |

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

(А.6)

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, влияющих на отправление в ремонт устройства, до и после размещения веб приложения.

Усредненные данные о реализации программного обеспечения за период с 21 апреля по 17 мая 2025 года (до размещения веб-приложения) по сравнению с аналогичными данными за период с 1 июня по 29 июня 2025 года (после размещения веб-приложения) приведены в таблице А.5.

Таблица А.5 – Усредненные данные о результатах реализации программного обеспечения «УчительPRO» до и после размещения веб-приложения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Затраты на диагностику устройств | | Затраты на ремонт устройств | | Итого |
| До | После | До | После |  |
|  |  | 50000 | 1200 | -96,7% |
| 250 | 2500 |  |  | +100% |
| Общие затраты до | 50250 | Общие затраты после | 3700 | 100% - 7,4% |

Из приведенных данных в таблице А.5 можно сделать вывод, что благодаря реализации программного продукта общие затраты на обслуживание устройств снизились на 92,6%:

(А.7)

Цена программного продукта составляет 37886,4 рублей в то время как расходы, которые несет компания, при внедрении «УчительPRO» несет на 46550 рублей меньше.

(А.8)

Таким образом, экономический эффект в 1,2 раза превышает затраты на разработку вет-сайта.

. (А.9)

Срок окупаемости проекта равен 0,81 года.

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

**Приложение Б**

***(обязательное)***

**Входные документы**

Таблица Б.1 – Описание входной информации

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название данных | Тип данных | Описание | Формат/  Пример | Особенности обработки |
| Регистрационные данные пользователя | Текстовые | Информация для регистрации нового пользователя | ФИО, электронная почта, пароль, роль | Ввод через форму регистрации, проверка уникальности email |
| Данные для входа | Текстовые | Логин и пароль для авторизации | Email, пароль | Ввод через форму входа, шифрование пароля |
| Загруженные файлы | Файлы | Образцы учебных материалов, презентации, тесты | PDF, DOCX, PPTX, JPG, MP4 | Загрузка через форму, проверка формата и размера |
| Метаданные материалов | Таблица | Информация о загруженных ресурсах | Название, описание, предмет, теги, дата загрузки | Ввод при публикации, автоматическая генерация даты |
| Запросы поиска | Текстовые | Ключевые слова, фильтры для поиска материалов | «Математика», «Тесты», «2019» | Ввод через поисковую строку, фильтрация данных |
| Данные профиля | Таблица | Информация о пользователе | Имя, фамилия, роль, активность | Автоматическое отображение из базы данных |

**Приложение В**

***(обязательное)***

**Выходные документы**

Таблица В.1 – Описание выходной информации

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название данных | Тип данных | Описание | Формат/  Пример | Особенности обработки |
| Страница профиля | Веб-форма | Информация о пользователе, история активности | Имя, список загруженных и скачанных материалов | Таблица, карточки, кнопки редактирования |
| Результаты поиска | Список карточек | Перечень материалов, соответствующих запросу | Название, краткое описание, предмет, дата | Карточки с изображениями, кнопки «скачать», «подробнее» |
| Детальная страница ресурса | Веб-форма | Полное описание выбранного материала | Название, описание, формат файла, дата загрузки | Текст, изображения, кнопки скачивания |
| Сообщения системы | Всплывающие окна / уведомления | Уведомления о успешных действиях или ошибках | «Файл успешно загружен», «Ошибка при входе" | Всплывающие окна, статус-бар |
| Страница загрузки файла | Веб-форма | Интерфейс для загрузки новых материалов | Поля для выбора файла, описание, теги | Форма с кнопкой «Загрузить» |
| Страница администратора | Веб-интерфейс | Управление пользователями и контентом | Таблицы пользователей, журнал действий | Таблицы, фильтры, кнопки редактирования |

**Приложение Г**

***(обязательное)***

**Диаграмма потоков данных DFD**

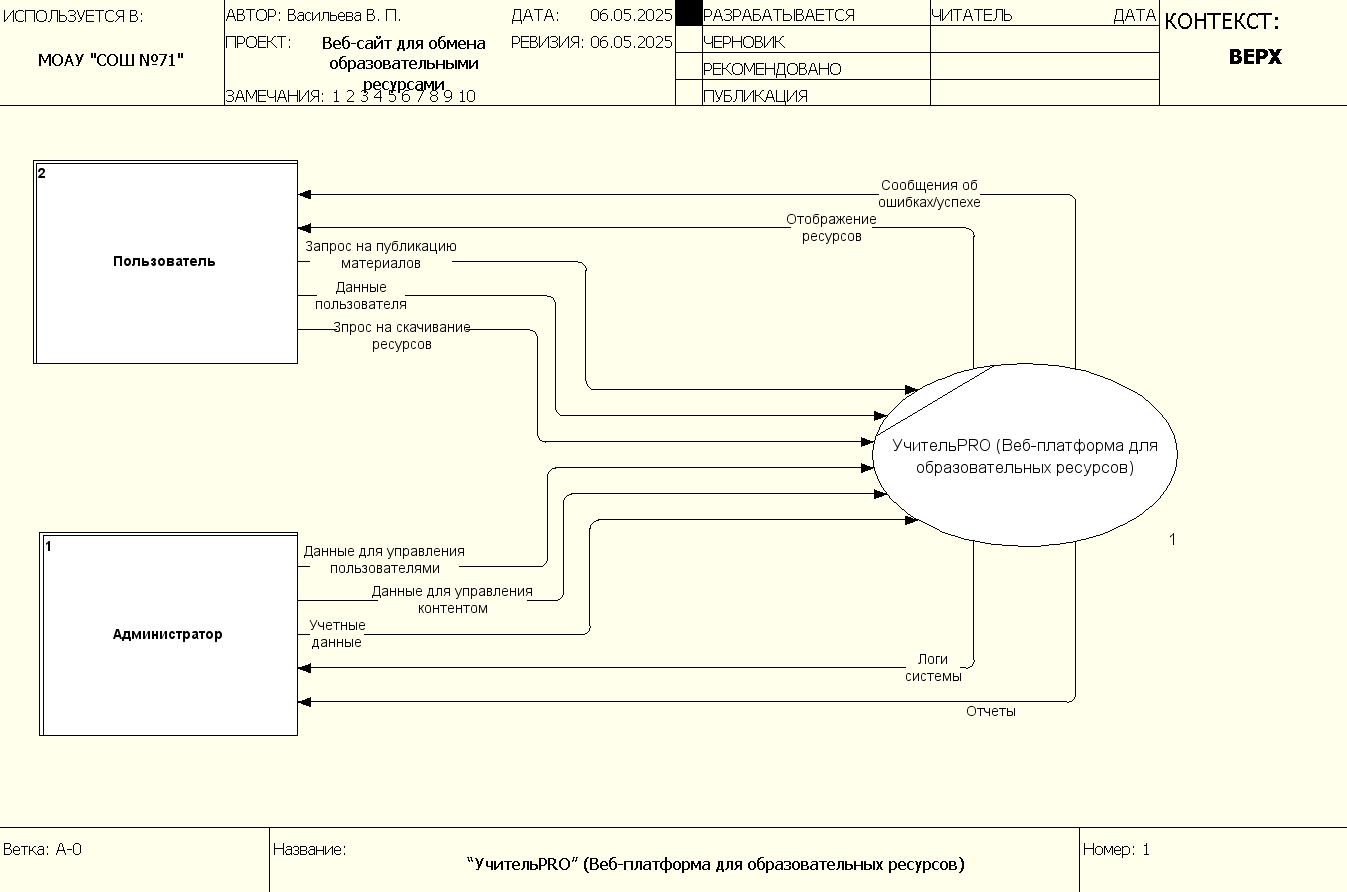


Рисунок Г.1 – Диаграмма потоков данных DFD

**Приложение Д**

***(обязательное)***

**Диаграмма прецедентов**

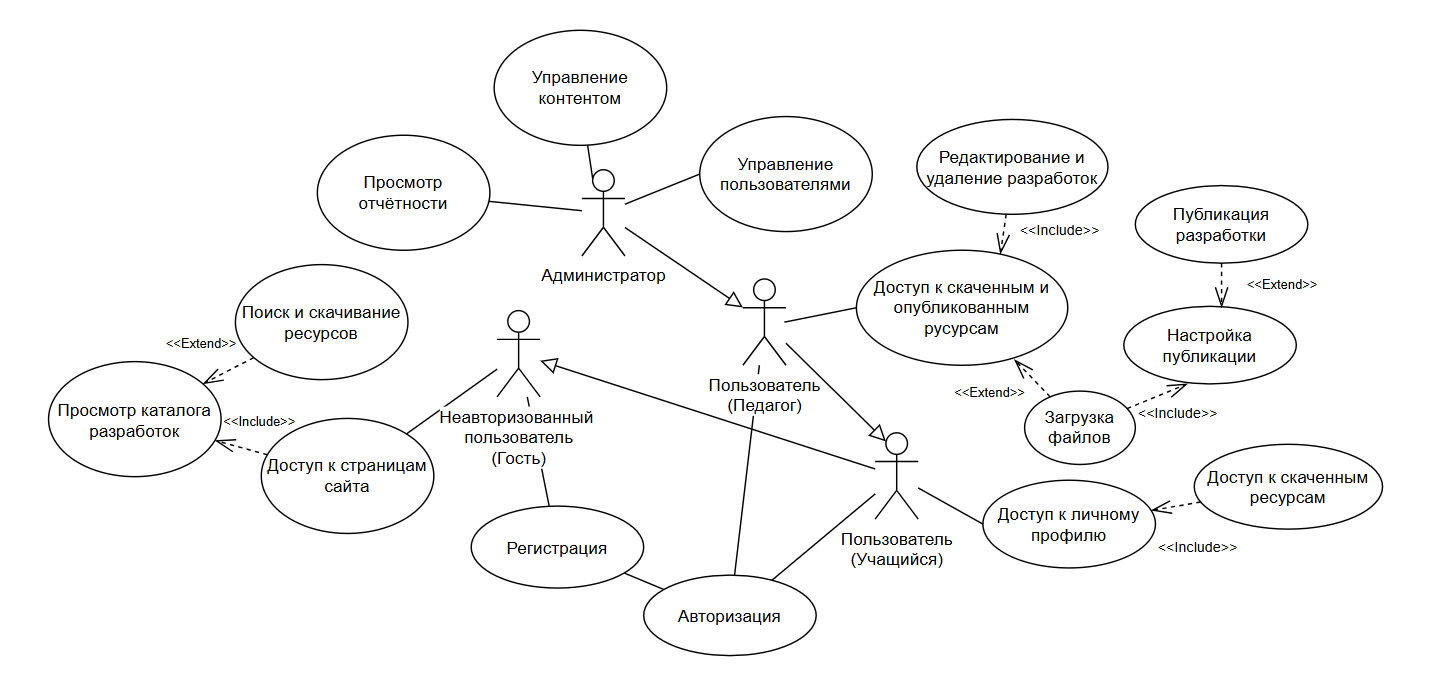


Рисунок Д.1 – Диаграмма прецедентов

**Приложение** **Е**

***(обязательное)***

**Функциональная модель**



Рисунок Е.1 – Функциональная модель

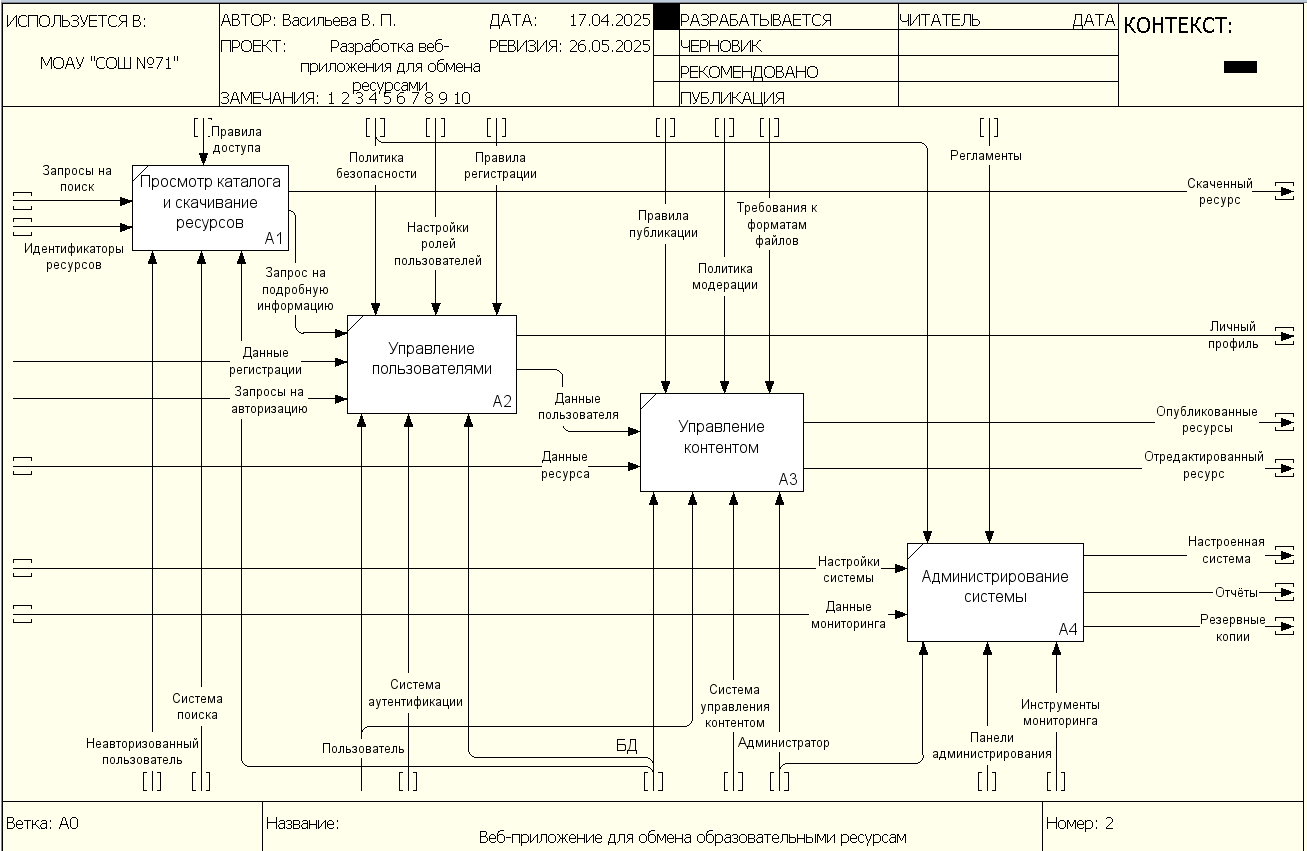


Рисунок Е.2 – Функциональная модель (детализация)

**Приложение** **Ж**

***(обязательное)***

**Информационная модель**

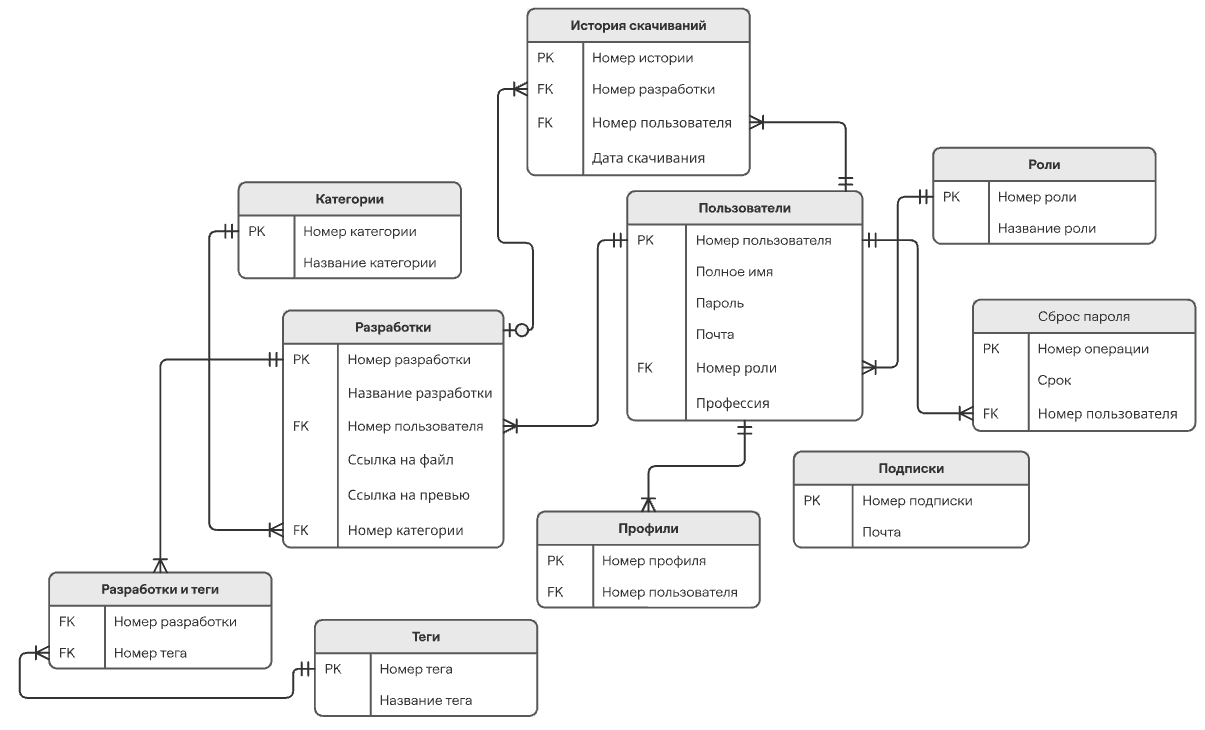


Рисунок Ж.1 – Информационная модель

**Приложение И**

***(обязательное)***

**Схема работы системы**

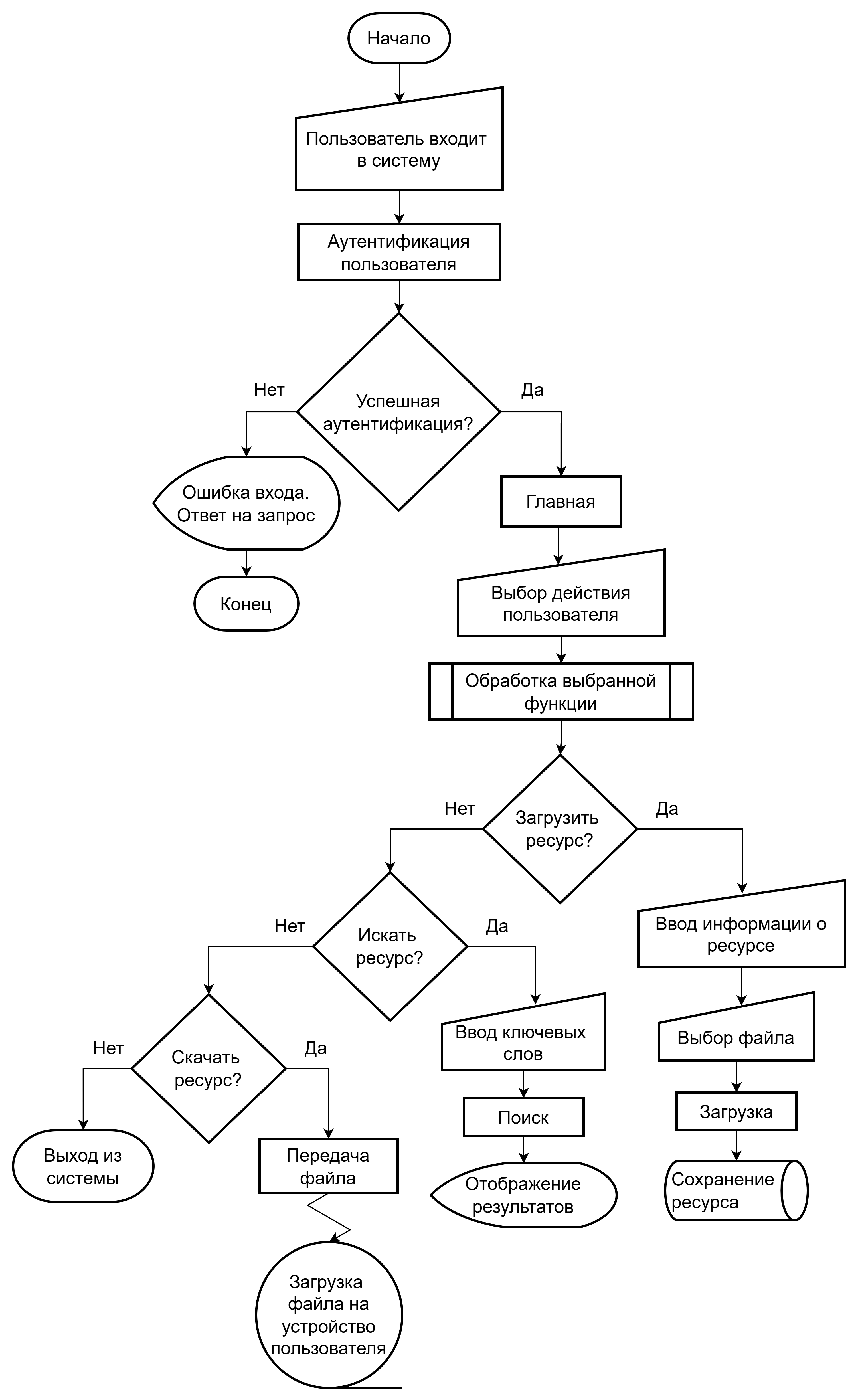


Рисунок И.1 – Схема работы системы

**Приложение К**

***(обязательное***

**ER-диаграмма**



Рисунок К.1 – ER-диаграмма

**Приложение Л**

***(обязательное)***

**Контрольный пример**

Код до рефакторинга:

// Роут для скачивания файлов

app.get("/download/:id", isAuthenticated, async (req, res) => {

  try {

    const developmentId = req.params.id;

    const userId = req.session.user.id;

    // Находим разработку

    const development = await Development.findByPk(developmentId);

    if (!development) {

      return res.status(404).send("Разработка не найдена");

    }

    // Записываем в историю скачиваний

    await DownloadHistory.create({

      userId: userId,

      developmentId: development.id,

    });

    // Получаем путь к файлу

    const filePath = path.join(\_\_dirname, "public", development.file\_path);

    // Проверяем существование файла

    if (!fs.existsSync(filePath)) {

      return res.status(404).send("Файл не найден на сервере");

    }

    // Получаем расширение файла

    const fileExt = path.extname(development.file\_path);

Код после рефакторинга:

module.exports = { getSafeFileName, fileExists };

const express = require('express');

const router = express.Router();

const { Development, DownloadHistory } = require('../models');

const { getSafeFileName, fileExists } = require('../utils/fileHelpers');

const path = require('path');

const fs = require('fs');

const handleDownload = async (req, res) => {

  try {

    const { id } = req.params;

    const userId = req.session.user.id;

    const development = await Development.findByPk(id);

    if (!development) {

      return res.status(404).json({ error: 'Разработка не найдена' });

    }

    await logDownload(userId, development.id);

    const filePath = getFilePath(development.file\_path);

    if (!fileExists(filePath)) {

      return res.status(404).json({ error: 'Файл не найден' });

    }

    const downloadFileName = getDownloadFileName(development);

    setDownloadHeaders(res, downloadFileName);

    streamFile(res, filePath, downloadFileName);

  } catch (error) {

    console.error('Download error:', error);

    res.status(500).json({ error: 'Ошибка сервера' });

  }

};

**Приложение М**

***(обязательное)***

**SQL-скрипты**

Определение «Developments»:

CREATE TABLE `Developments` (`development\_id` INTEGER PRIMARY KEY AUTOINCREMENT, `title` VARCHAR(255) NOT NULL, `description` TEXT, `file\_path` VARCHAR(255) NOT NULL, `preview` VARCHAR(255), `category\_id` INTEGER NOT NULL REFERENCES `categories` (`category\_id`) ON DELETE CASCADE ON UPDATE CASCADE, `user\_id` INTEGER NOT NULL REFERENCES `users` (`id`) ON DELETE CASCADE ON UPDATE CASCADE);

Определение «Users»:

CREATE TABLE `Users` (`id` INTEGER PRIMARY KEY AUTOINCREMENT, `fullName` VARCHAR(255) NOT NULL, `email` VARCHAR(255) NOT NULL UNIQUE, `password` VARCHAR(255) NOT NULL, `roleId` INTEGER NOT NULL REFERENCES `Roles` (`id`) ON DELETE CASCADE ON UPDATE CASCADE, `profession` VARCHAR(255));

Определение «Roles»:

CREATE TABLE `Roles` (`id` INTEGER PRIMARY KEY AUTOINCREMENT, `name` VARCHAR(255) NOT NULL UNIQUE);

Определение «tags»:

CREATE TABLE `tags` (`tag\_id` INTEGER PRIMARY KEY AUTOINCREMENT, `name` VARCHAR(255) NOT NULL UNIQUE);

Определение «development\_tags»:

CREATE TABLE `development\_tags` (`id` INTEGER PRIMARY KEY AUTOINCREMENT, `developmentId` INTEGER NOT NULL REFERENCES `Developments` (`development\_id`) ON DELETE CASCADE ON UPDATE CASCADE, `tagId` INTEGER NOT NULL REFERENCES `tags` (`tag\_id`) ON DELETE CASCADE ON UPDATE CASCADE, UNIQUE (`developmentId`, `tagId`));

Определение «download\_history»:

CREATE TABLE `download\_history` (`download\_history\_id` INTEGER PRIMARY KEY AUTOINCREMENT, `download\_date` DATETIME NOT NULL, `user\_id` INTEGER NOT NULL REFERENCES `users` (`id`) ON DELETE CASCADE ON UPDATE CASCADE, `development\_id` INTEGER REFERENCES `Developments` (`development\_id`) ON DELETE SET NULL ON UPDATE CASCADE);

Определение «password\_reset\_tokens»:

CREATE TABLE `password\_reset\_tokens` (`id` INTEGER PRIMARY KEY AUTOINCREMENT, `token` VARCHAR(255) NOT NULL UNIQUE, `userId` INTEGER NOT NULL REFERENCES `Users` (`id`) ON DELETE CASCADE ON UPDATE CASCADE, `expiresAt` DATETIME NOT NULL);

**Приложение** **Н**

***(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>

**Приложение П**

***(обязательное)***

**Акт о внедрении результатов**