

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра Прикладная математика и информатика
(наименование)

02.03.03 Математическое обеспечение и администрирование информационных систем
(код и наименование направления подготовки / специальности)

Мобильные и сетевые технологии
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка и администрирование web-приложения анимационной студии»

Обучающийся

А.В. Марахтанов
(И.О. Фамилия)

(личная подпись)

Руководитель

Канд. тех. наук, Д.Г. Токарев
(ученая степень, звание, И.О. Фамилия)

Консультант

Канд. пед. наук, доцент С.А. Гудкова
(ученая степень, звание, И.О. Фамилия)

Тольятти 2024

Аннотация

Бакалаврская работа выполнена на тему Разработка и администрирование web-приложения анимационной студии.

Объектом исследования является процесс управления анимационными проектами в психолого-педагогическом центре, где дети занимаются созданием анимации под руководством специалистов.

Предметом исследования является система управления анимационными проектами через web-приложение, включающее функциональность для администрирования, планирования и отслеживания проектов, а также поддержки креативного процесса участников.

Целью работы является разработка web-приложения для анимационной студии, которое обеспечит удобный доступ к творческим работам детей, улучшит взаимодействие между администрацией, родителями и детьми, а также повысит эффективность работы центра.

Выпускная квалификационная работа состоит из введения, трёх глав, заключения и списка литературы, включающего 20 источников. Объем работы составляет 53 страницы, содержит 19 рисунков, 4 таблицы, 3 формулы и 1 диаграммы.

Во введении описываются актуальность темы, объект и предмет исследования, формулируются цели и задачи работы. Первая глава посвящена анализу предметной области, определению требований к функциональности системы и постановке задачи на разработку. Вторая глава описывает выбор технологий и инструментов для разработки web-приложения, проектирование логической модели.

Третья глава охватывает вопросы разработки и администрирования web-приложения, включая структуру проекта, разработку административной панели, интерфейса пользователя и тестирование системы.

В заключении описываются основные выводы, сделанные в ходе выполнения выпускной квалификационной работы.

Abstract

The title of the graduation work is «Development and Administration of a Web Application for an Animation Studio».

The graduation work consists of an introduction, three chapters, a conclusion, tables, a list of references including foreign sources, and graphical content on several pages.

The key issue of the thesis is the development of software to automate the processes of managing and administering an animation studio operating within a psychological and pedagogical center.

The application is intended for specialists of the center who organize and support the process of creating animation projects by children.

The aim of the work is to develop a web application for the animation studio that provides easy access to the creative works of children, enhances interaction between the administration, parents, and children, and increases the efficiency of the center's operations.

The graduation work is divided into several logically interconnected parts: analysis of the subject area and defining system requirements; selection of technologies and tools for web application development, system design, and hardware-software requirements; and the development and administration of the web application, including the project structure, administrative panel, user interface, and system testing.

Finally, the study presents conclusions on the successful implementation of the web application, emphasizing its relevance for improving the management of animation projects in psychological and pedagogical centers and suggesting that similar technological and constructive solutions can be applied to other educational and creative institutions.

This work is relevant not only for addressing the needs of animation studios in psychological and pedagogical centers but also for other institutions requiring effective project management and administration systems.

Оглавление	
Введение.....	6
Глава 1 Анализ предметной области и определение требований к функциональности	8
1.1 Техничко-экономическая характеристика предметной области	8
1.2 Концептуальное моделирование предметной области	10
1.2.2 Анализ модели бизнес-процесса.....	10
1.2.3 Обоснование необходимости автоматизированного варианта решения и формирование требований к проектируемой информационной системе.....	11
1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям	12
1.4 Постановка задачи на разработку проекта создания автоматизированной информационной системы	16
Глава 2 Проектирование web-приложения.....	19
2.1 Выбор технологий и инструментов разработки web-приложения	19
2.1.1 Выбор языка программирования	19
2.1.2 Выбор фреймворка	20
2.1.3 Выбор системы управления базами данных.....	21
2.1.4 Выбор средств разработки	23
2.2 Разработка логической модели автоматизированной информационной системы	26
2.3.1 Диаграмма прецедентов.....	26
2.3.2 Схема развертывания	27
2.3.3 Диаграмма активности.....	29
2.3.4 Схемы совместной работы	31
2.3 Требования к аппаратно-программному обеспечению автоматизированной информационной системы	32
Глава 3 Разработка и администрирование web-приложения.....	35
3.1 Структура проекта	35
3.2 Разработка административной панели	37
3.3 Разработка интерфейса пользователя	42
3.4 Тестирование веб-приложения.....	47
3.4.1 Функциональное тестирование.....	48
3.4.2 Тестирование производительности	48
3.4.3 Тестирование безопасности	49

3.4.4 Пользовательское тестирование	49
3.5 Расчет экономической эффективности.....	50
Заключение	53
Список используемой литературы	55

Введение

В современном мире цифровые технологии активно влияют на все аспекты жизнедеятельности человека, включая образование и творчество. Особенно значимо их применение в области педагогики и психологии, где цифровые инструменты могут использоваться как средство развития и самовыражения детей. Анимационные студии, созданные в образовательных учреждениях, позволяют детям осваивать основы мультипликации, редактирования видео и создания анимационных фильмов, что способствует развитию их творческого потенциала, коммуникативных навыков и технических компетенций.

Цель работы.

Целью данной ВКР является разработка и администрирование web-приложения для анимационной студии, расположенной в психолого-педагогическом центре. Приложение предназначено для использования специалистами центра, которые организуют и поддерживают процесс создания анимационных проектов детьми.

Объект исследования.

Объектом исследования выступает процесс управления анимационными проектами в психолого-педагогическом центре, где дети занимаются созданием анимации под руководством специалистов.

Предмет исследования.

Предметом исследования является система управления анимационными проектами через web-приложение, которое включает функциональность для администрирования, планирования и отслеживания проектов, а также поддержки креативного процесса участников.

Задачи исследования.

Для достижения поставленной цели были определены следующие задачи:

- изучение предметной области;

- анализ проблем в существующем бизнес-процессе;
- анализ существующих решений на рынке;
- обоснование выбора архитектуры информационной системы;
- обоснование выбора технологий разработки;
- обоснование выбора СУБД;
- тестирование разработанной информационной системы

Web-приложение предоставит возможность не только визуализировать и хранить результаты детского творчества, но и организовывать совместную работу над проектами, взаимодействовать с преподавателями и родителями, а также получать обратную связь. Это станет значительным вкладом в образовательный процесс и поможет поддерживать интерес детей к анимации как к форме современного искусства.

Глава 1 Анализ предметной области и определение требований к функциональности

1.1 Техничко-экономическая характеристика предметной области

В данном разделе представлена технико-экономическая характеристика психолого-педагогического центра, где функционирует анимационная студия. Центр занимается образовательной и воспитательной деятельностью, при этом анимационная студия является одним из ключевых направлений, направленных на развитие творческих способностей детей.

Описание организации: Психолого-педагогический центр — это учреждение, ориентированное на комплексное развитие личности ребенка, включая его интеллектуальное, творческое и социальное воспитание. Центр предлагает разнообразные образовательные программы и курсы, в том числе по анимации и медиа.

Основные направления деятельности организации:

- образовательные программы: различные учебные курсы и классы, направленные на развитие креативных и аналитических навыков.
- психологическая поддержка: Консультации и сессии с детскими психологами для поддержки эмоционального благополучия учащихся.
- анимационный отдел: предоставляет детям возможность создавать короткие видеоролики, которые демонстрируют их деятельность в центре, такие как игры, творческие мастер-классы или учебные проекты. Отдел оснащён необходимым оборудованием для съёмки и базового монтажа видео, что позволяет участникам освоить начальные навыки в области видеопроизводства и медиа. Программа направлена на развитие творческих и коммуникативных навыков, а также на формирование способности к сотрудничеству и работе в команде

- социально-развивающие мероприятия: организация мероприятий, направленных на социальное взаимодействие и коммуникативное развитие детей.

Организационная структура представлена на рисунке 1.

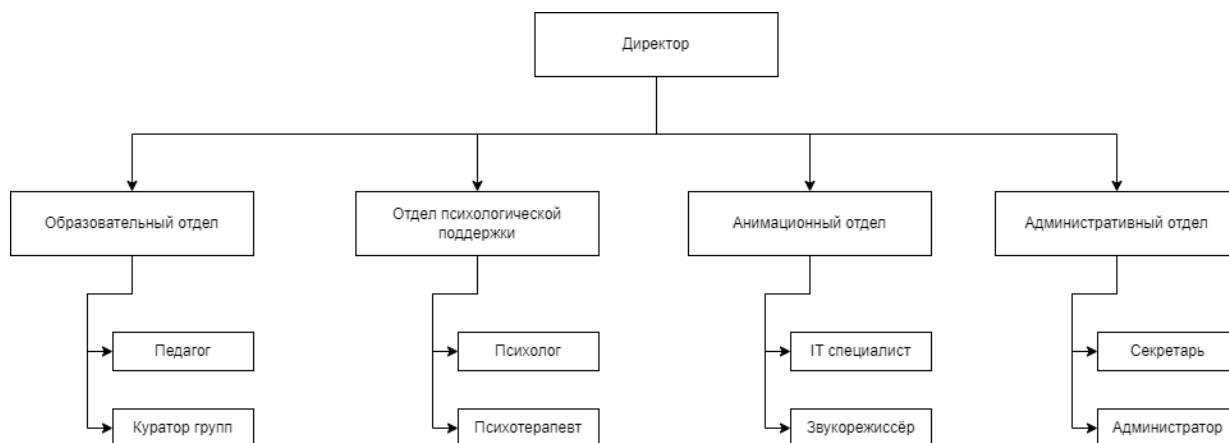


Рисунок 1 – Организационная структура

В качестве предметной области выступает анимационная студия, которая представляет собой отдельный отдел в структуре психолого-педагогического центра. Этот отдел отвечает за разработку и проведение анимационных проектов, которые помогают детям развивать творческие способности и коммуникативные навыки. Процесс создания анимации включает в себя планирование сценария, подбор анимационных техник, реализацию проекта и его презентацию.

Подробно рассмотрим процесс планирования анимационного проекта. На этой стадии специалисты студии определяют тему и сценарий будущего анимационного проекта, формируют команду молодых аниматоров и распределяют роли и задачи. Коммуникация с участниками проекта осуществляется через регулярные встречи и обсуждения, а также с помощью электронной почты и онлайн-платформ для совместной работы.

После описания технико-экономической характеристики деятельности перейдем к следующему этапу — процессу концептуального моделирования

предметной области, в рамках которого будет выбрана наиболее подходящая технология для моделирования и реализации анимационного проекта.

1.2 Концептуальное моделирование предметной области

1.2.2 Анализ модели бизнес-процесса

Анализ модели бизнес-процесса включает в себя изучение текущего состояния процесса, выявление узких мест, определение возможностей для улучшения эффективности и проектирование оптимальных изменений. В наше время, когда информационные технологии проникают в различные сферы жизни, существует необходимость в инновационных подходах к организации и управлению процессами. Одной из таких сфер является психолого-педагогический центр с анимационной студией, где дети могут проявить свой творческий потенциал и развиваться как личности. Однако, несмотря на значимость этого учреждения, существует серьезная проблема, которая затрудняет взаимодействие между администрацией, родителями и детьми - отсутствие удобного способа доступа к творческим работам детей.

В настоящее время, чтобы родители могли увидеть творчество своих детей, им приходится лично обращаться в центр и запрашивать видеозаписи процесса творчества или результатов. Этот процесс не только занимает много времени и сил у родителей, но и создает дополнительную нагрузку на администрацию центра, которая вынуждена заниматься организацией и предоставлением этих материалов.

Такая система имеет ряд недостатков. Во-первых, она неудобна для родителей, которым приходится тратить свое время на посещение центра и ожидание предоставления материалов. Это особенно неудобно для родителей, у которых ограниченное количество свободного времени из-за работы или других обязанностей. Во-вторых, такая система неэффективна с точки зрения использования ресурсов центра, поскольку администрация тратит время на обработку запросов и предоставление материалов, вместо

того чтобы сосредотачиваться на более важных задачах, таких как организация занятий и взаимодействие с клиентами. В-третьих, отсутствие централизованной системы доступа к творческим работам детей ограничивает возможности их продвижения и обмена опытом между родителями.

Таким образом, проблема отсутствия удобного способа доступа к творческим работам детей в анимационной студии психолого-педагогического центра не только затрудняет взаимодействие между администрацией, родителями и детьми, но и создает дополнительные трудности в организации работы центра и эффективном использовании его ресурсов. Для решения этой проблемы необходимо разработать и внедрить новую модель бизнес-процесса, которая будет обеспечивать удобный и эффективный доступ к творческим работам детей.

1.2.3 Обоснование необходимости автоматизированного варианта решения и формирование требований к проектируемой информационной системе

На основании выявленных проблем с доступом к творческим работам детей в анимационной студии психолого-педагогического центра, необходимо обосновать и утвердить необходимость разработки автоматизированного варианта решения данной проблемы. Автоматизированное решение позволит значительно улучшить процесс взаимодействия между администрацией, родителями и детьми, а также повысить эффективность работы центра в целом.

Обоснование необходимости автоматизированного решения:

- увеличение удобства для родителей и детей: автоматизированное решение позволит родителям и детям получать доступ к творческим работам онлайн, в любое удобное для них время, без необходимости личного посещения центра. Это значительно сэкономит время и силы родителей и сделает процесс взаимодействия более комфортным

- оптимизация ресурсов центра: внедрение автоматизированного решения сократит временные и человеческие ресурсы, затрачиваемые на обработку запросов и предоставление видеоматериалов. администрация сможет перераспределить эти ресурсы на более важные задачи, такие как организация занятий и развитие творческих программ для детей
- улучшение коммуникации и вовлеченности: автоматизированное решение позволит родителям активнее участвовать в творческом процессе своих детей, оставлять комментарии и оценки к их работам, а также легко обмениваться опытом и идеями с другими родителями
- создание централизованной базы данных: внедрение информационной системы позволит создать централизованную базу данных с творческими работами всех детей, что облегчит их архивирование, поиск и управление.

1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям

Для обеспечения эффективной реализации проекта разработки автоматизированной информационной системы, необходимо проанализировать существующие разработки на предмет их соответствия сформулированным требованиям. Данный анализ поможет определить наиболее подходящее решение и учесть опыт других проектов при разработке новой системы.

Проведем сравнительный анализ наиболее известных видеохостингов. Это необходимо для того, чтобы более точно определить дальнейшие направления проектирования будущей информационной системы, а также для того чтобы не повторять чужих ошибок при разработке. Далее рассмотрим существующие аналоги:

Vimeo, пользовательский интерфейс представлен на рисунке 2:

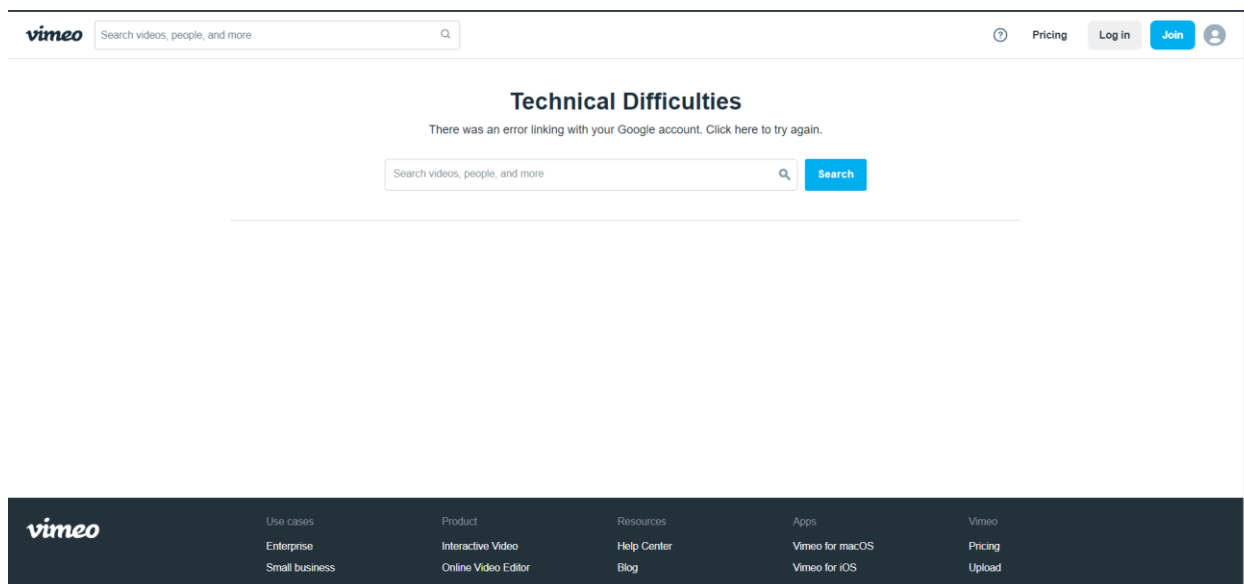


Рисунок 2 – Vimeo

Преимущества:

- высокое качество видео и профессиональное сообщество - позволяют создателям контента представлять свои работы в наилучшем качестве и получать обратную связь от качественной аудитории.
- Отсутствие рекламы на платформе
- более активная поддержка авторов контента - компания активно взаимодействует с создателями, предоставляя инструменты и ресурсы для развития.

Недостатки:

- ограниченные возможности социальной сети и взаимодействия - платформа ориентирована в первую очередь на хранение и публикацию контента, что снижает социальные возможности для пользователей
- ограничения на бесплатном тарифе - некоторые функции и возможности могут быть доступны только при подписке на платные тарифы.

Twitch, пользовательский интерфейс представлен на рисунке 3.

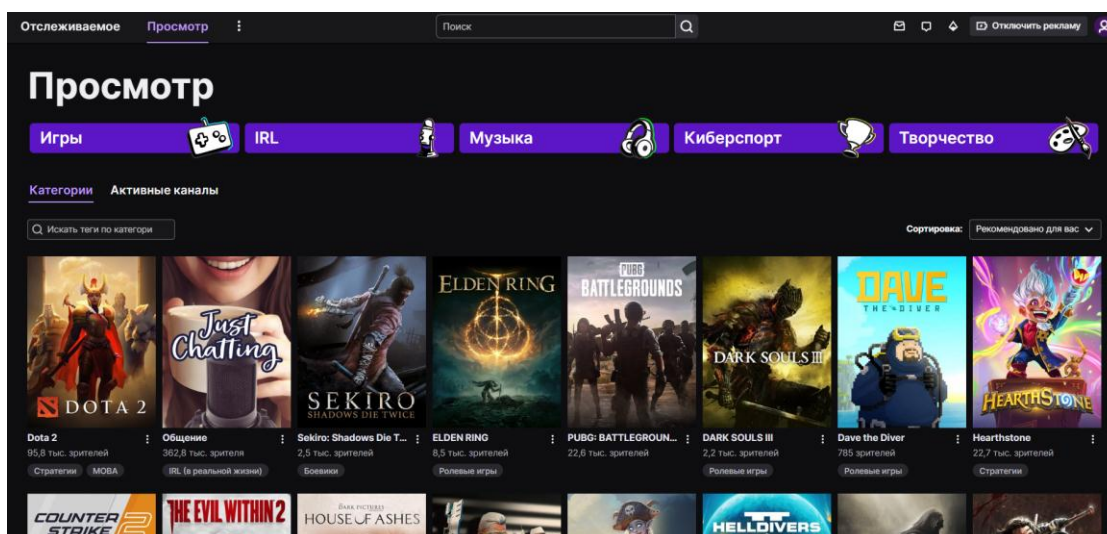


Рисунок 3 – Twitch

Преимущества:

- ориентация на стриминг и интерактивное взаимодействие - позволяет стримерам взаимодействовать с аудиторией в реальном времени, создавая уникальные и захватывающие контентные форматы
- широкая аудитория, особенно в сфере игрового контента - платформа привлекает миллионы зрителей, заинтересованных в различных игровых трансляциях и контенте.

Недостатки:

- ограничения контента вне игровой сферы - платформа традиционно ориентирована на игровой контент, что может ограничить аудиторию для создателей, не связанных с играми
- ограниченные возможности для неигровых контент-создателей - функциональность и инструменты Twitch могут быть менее подходящими для создания контента, не связанного с играми
- некоторое ограничение в доле рынка за пределами игровой аудитории

– несмотря на популярность, Twitch имеет ограниченную аудиторию в сравнении с другими видеоплатформами.

Netflix, пользовательский интерфейс представлен на рисунке 4.

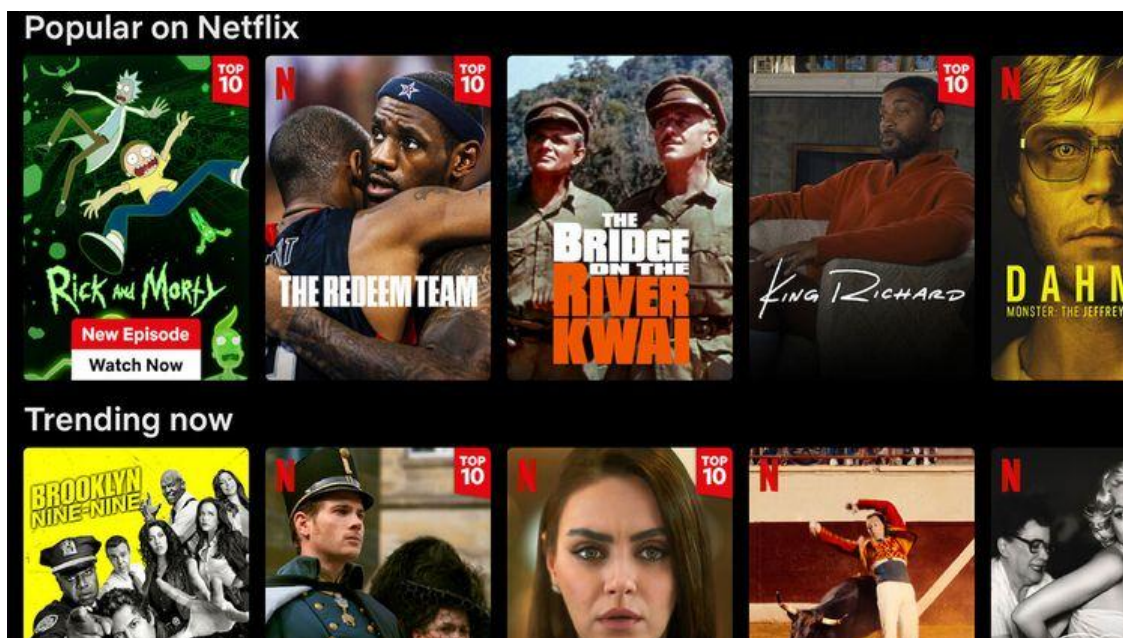


Рисунок 4 – Netflix

Преимущества:

- огромная библиотека фильмов и сериалов - предоставляет пользователям широкий выбор контента для просмотра в различных жанрах и настроениях
- персонализированные рекомендации на основе предпочтений пользователя - помогают пользователям находить контент, который соответствует их вкусам и интересам
- высокое качество потокового видео и возможность просмотра в высоком разрешении - обеспечивают приятный и качественный просмотр для пользователей.

Недостатки:

- подписочная модель может быть дорогой для некоторых пользователей
- ежемесячные платежи могут быть значительными для пользователей с ограниченным бюджетом
- ограничения в доступности контента в разных регионах - некоторые фильмы и сериалы могут быть недоступны в определенных странах или регионах.

Результат анализа информационных систем видеохостинга представлен в таблице 1.

Таблица 1 – Анализ ИС на рынке

Сервис	Vimeo	Twitch	Netflix
Качество видео	+	-	+
База контента	-	-	+
Загрузка видео	+	-	-
Удаление видео	+	-	-
Узкость аудитории	-	+	+
Итого	3\5	1\5	3\5

Исходя из анализа таблицы 1, делаем вывод, что каждая из существующих ИС имеет определенные недостатки.

После того, как был проведен сравнительный анализ существующих ИС, необходимо сделать вывод о необходимости проектирования и разработки новой узконаправленной ИС, а также описать цель ее создания и основные требования к будущей информационной системе.

1.4 Постановка задачи на разработку проекта создания автоматизированной информационной системы

Исходя из анализа, проведенного в пункте 1.3 делаем вывод, что ни одна из рассмотренных уже существующих информационных систем в данной предметной области не удовлетворяет всем необходимым

требованиям. Таким образом, еще раз подтверждается необходимость проектирования и разработки новой ИС.

Требования к проектируемой информационной системе:

- удобный пользовательский интерфейс: система должна иметь интуитивно понятный интерфейс, позволяющий родителям и детям легко находить необходимую информацию и взаимодействовать с платформой
- безопасность данных: информационная система должна обеспечивать высокий уровень защиты конфиденциальности данных родителей и детей, а также предотвращать несанкционированный доступ к информации
- возможность загрузки, хранения, редактирования и удаления видеоматериалов: Система должна поддерживать загрузку, хранение, редактирования и удаления видеозаписей творческих работ детей, обеспечивая быстрый доступ к материалам
- возможность комментирования и обратной связи: Платформа должна предоставлять возможность родителям оставлять комментарии и оценки к работам своих детей
- мобильная совместимость: Система должна быть адаптирована для использования на мобильных устройствах, чтобы обеспечить доступность для пользователей в любое время и из любого места.

Разработка и внедрение информационной системы, соответствующей указанным требованиям, будет эффективным и эффективным способом решения проблемы доступа к творческим работам детей в анимационной студии психолого-педагогического центра.

Выводы по главе 1.

В первой главе проведен анализ предметной области, начиная с технико-экономической характеристики и заканчивая анализом существующих разработок. В ходе работы было осуществлено концептуальное моделирование предметной области, включая анализ бизнес-процесса и

определение необходимости автоматизированного варианта решения. Были сформулированы требования к проектируемой информационной системе на основе выявленных потребностей и проблем.

Данный этап позволил четко сформулировать постановку задачи на разработку проекта создания автоматизированной информационной системы, учитывая требования предметной области и анализ существующих решений. В результате первой главы были сделаны первоначальные шаги к созданию эффективного и соответствующего потребностям решения, что является ключевым для успешной реализации проекта.

Глава 2 Проектирование web-приложения.

2.1 Выбор технологий и инструментов разработки web-приложения

Выбор технологий и инструментов для разработки web-приложения является ключевым этапом процесса проектирования. В данной главе будут рассмотрены основные аспекты выбора языка программирования, фреймворка и системы управления базами данных (СУБД), которые определяют архитектуру и функциональность будущего приложения [9].

2.1.1 Выбор языка программирования

В данном разделе проводится сравнительный анализ различных языков программирования с учетом их синтаксиса, производительности, сообщества и поддержки сторонних библиотек .

Результат сравнительного анализа языков программирования представлен в таблице 2.

Таблица 2 – Сравнительный анализ языков программирования

Характеристика	Python	JavaScript	Java
Простота и читаемость кода	Python известен своим простым и лаконичным синтаксисом, что делает код более читаемым и поддерживаемым	JavaScript может быть менее простым в понимании из-за своей гибкости и множества вариантов написания кода	Java обычно требует более объемного и строгого синтаксиса, что может делать код менее читаемым в сравнении с Python
Широкая стандартная библиотека	Python поставляется с богатой стандартной библиотекой, которая включает множество модулей и инструментов для разработки различных типов приложений	JavaScript имеет стандартную библиотеку, но она может быть менее обширной и полезной в сравнении с Python	Java также имеет обширную стандартную библиотеку, но она может быть менее гибкой и универсальной в сравнении с Python

Продолжение таблицы 2

Характеристика	Python	JavaScript	Java
Кроссплатформенность	Python является кроссплатформенным языком программирования, что означает, приложения работают на разных ОС	JavaScript также кроссплатформенный язык, но его применение на серверной стороне часто требует использования Node.js	Java также кроссплатформенный язык, но требует установки Java Virtual Machine (JVM) на целевых устройствах
Обширное сообщество и экосистема	Python имеет большое и активное сообщество разработчиков, что обеспечивает поддержку, обмен опытом и доступ к множеству библиотек и фреймворков	JavaScript также имеет обширное сообщество разработчиков, но его экосистема может быть менее структурированной и предсказуемой в сравнении с Python	Java также имеет широкое сообщество разработчиков и экосистему инструментов, но она может быть менее динамичной и инновационной в сравнении с Python

Исходя из таблицы 2, делаем выводы, что Python обладает рядом преимуществ, таких как простота и читаемость кода, широкая стандартная библиотека, кроссплатформенность и обширное сообщество разработчиков, что делает его привлекательным выбором для разработки различных типов приложений [1].

2.1.2 Выбор фреймворка

Фреймворк предоставляет набор инструментов и шаблонов для ускорения разработки web-приложений. В данном разделе рассматривается выбор фреймворка, учитывая его функциональные возможности, расширяемость, удобство использования и поддержку сообщества.

Результат сравнительного анализа фреймворков на Python в таблице 3.

Таблица 3 – Сравнительный анализ фреймворков

Характеристика	Django	Flask	Ruby on Rails
ORM (Object-Relational Mapping)	Полноценный ORM встроенный в фреймворк, обеспечивает удобное взаимодействие с базой данных, автоматическое создание SQL-запросов на основе моделей данных и миграции	Отсутствует встроенный ORM, требует использования сторонних библиотек для работы с базой данных	Встроенный ORM ActiveRecord обеспечивает мощное взаимодействие с базой данных, но может быть менее гибким в сравнении с Django ORM
Встроенная аутентификация и авторизация	Имеет встроенные средства для аутентификации и авторизации пользователей, что упрощает создание безопасных приложений	Отсутствует встроенная аутентификация и авторизация, требует использования сторонних библиотек или ручной реализации	Имеет собственные механизмы аутентификации и авторизации, но может потребовать дополнительной настройки и ручной работы
Административный интерфейс	Поставляется с готовым административным интерфейсом для управления данными приложения без дополнительной разработки	Отсутствует готовый административный интерфейс, требует разработки собственного или использования сторонних решений	Предоставляет готовый административный интерфейс, который может быть менее гибким в сравнении с Django
Сообщество и документация	Имеет большое активное сообщество разработчиков и обширную документацию, что облегчает разработку и решение проблем	Имеет активное сообщество и документацию, но может быть менее распространенным и обширным в сравнении с Django	Имеет свое сообщество и документацию, которые могут быть менее доступными или обширными в сравнении с Django

Таблица 3 демонстрирует, что Django обладает рядом преимуществ, таких как встроенный ORM, аутентификация и авторизация, административный интерфейс и обширная документация, что делает его предпочтительным выбором для разработки web-приложений.

2.1.3 Выбор системы управления базами данных

Система управления базами данных (СУБД) играет важную роль в хранении и обработке данных приложения. Перед выбором СУБД необходимо учитывать требования к производительности,

масштабируемости и надежности данных. В данном разделе проводится сравнительный анализ различных СУБД, выявляя их основные преимущества и недостатки для оптимального выбора [16].

Результат сравнительного анализа СУБД в таблице 4.

Таблица 4 – Сравнительный анализ СУБД

Характеристика	SQLite3	PostgreSQL	MySQL
Простота установки и использования	Простая установка и использование, не требует настройки отдельного сервера базы данных	Требует установки и настройки сервера базы данных, что может быть более сложным процессом	Требует установки и настройки сервера базы данных, но может быть менее сложным в сравнении с PostgreSQL
Эффективность для небольших и средних проектов	Идеально подходит для небольших и средних проектов, обеспечивает высокую производительность при работе с небольшими объемами данных	Подходит для проектов любого размера, но может быть избыточным для небольших приложений из-за дополнительных функций и настроек	Подходит для проектов любого размера, но может потребовать более сложной настройки для оптимальной производительности
Не требует настройки сервера базы данных	SQLite3 хранит всю базу данных в одном файле, что упрощает развертывание и управление приложением	Требует настройки и управления отдельным сервером базы данных, что может потребовать дополнительных ресурсов и знаний	Требует настройки и управления отдельным сервером базы данных, но может быть более гибким в сравнении с PostgreSQL

Продолжение таблицы 4

Производительность в условиях ограниченных ресурсов	Обеспечивает высокую производительность при работе с ограниченными ресурсами, так как не требует дополнительных процессов и служб, работающих в фоновом режиме	Может быть более ресурсоемким в сравнении с SQLite3 из-за необходимости в дополнительных процессах и службах, работающих на сервере	Может потреблять больше ресурсов, чем SQLite3, из-за дополнительных функций и возможностей
Поддержка SQL-стандартов	SQLite3 поддерживает большую часть SQL-стандартов, что обеспечивает совместимость с множеством инструментов и библиотек	PostgreSQL соблюдает множество SQL-стандартов и предоставляет обширные возможности для работы с данными и расширений	MySQL также соблюдает множество SQL-стандартов, но может отличаться в реализации некоторых функций
Распространенность и поддержка	SQLite3 широко распространен и поддерживается сообществом разработчиков, что обеспечивает обновления и поддержку вопросов безопасности	PostgreSQL также имеет широкое распространение и активное сообщество разработчиков, обеспечивающее поддержку и развитие	MySQL является одной из самых популярных СУБД и имеет обширное сообщество разработчиков, но может быть более сложным в некоторых аспектах в сравнении с SQLite3

Из проведенного сравнения видно, что SQLite3 обладает рядом значительных преимуществ перед другими системами управления базами данных (СУБД), особенно в контексте небольших и средних проектов, а также в условиях ограниченных ресурсов. Простота установки и использования, эффективность при работе с небольшими объемами данных, отсутствие необходимости в настройке сервера базы данных и высокая производительность в условиях ограниченных ресурсов делают SQLite3 привлекательным выбором для множества приложений [18].

После того как были выбраны технологии перейдем к разработке логической модели информационной системы.

2.1.4 Выбор средств разработки

Перед началом разработки необходимо определиться с правильным выбором IDE. Интегрированная среда разработки (IDE) представляет собой готовый набор инструментов, необходимых для создания программного обеспечения. В процессе работы в IDE программист на Python использует множество средств, включая редакторы, библиотеки, платформы для запуска, отладки и тестирования кода. Благодаря таким средам разработчики могут не только сэкономить время, но и улучшить качество и читаемость кода [4].

Также представим основные возможности IDE в виде удобного списка:

- редактор кода с подсветкой синтаксиса и автодополнением
- отладчик для поиска и устранения ошибок в коде
- интеграция с системами контроля версий для управления исходным кодом
- поддержка различных языков программирования и технологий
- возможность создания проектов и управления проектными файлами
- инструменты для автоматизации и ускорения процесса разработки, такие как генераторы кода и шаблоны проектов
- интеграция с базами данных для управления данными приложений;
- встроенные средства для тестирования кода
- Поддержка различных операционных систем и платформ
- возможность настройки среды разработки в соответствии с требованиями конкретного проекта или разработчика .

В качестве среды разработки я решил выбрать PyCharm [2]. Главная страница сайта представлена на рисунке 5.

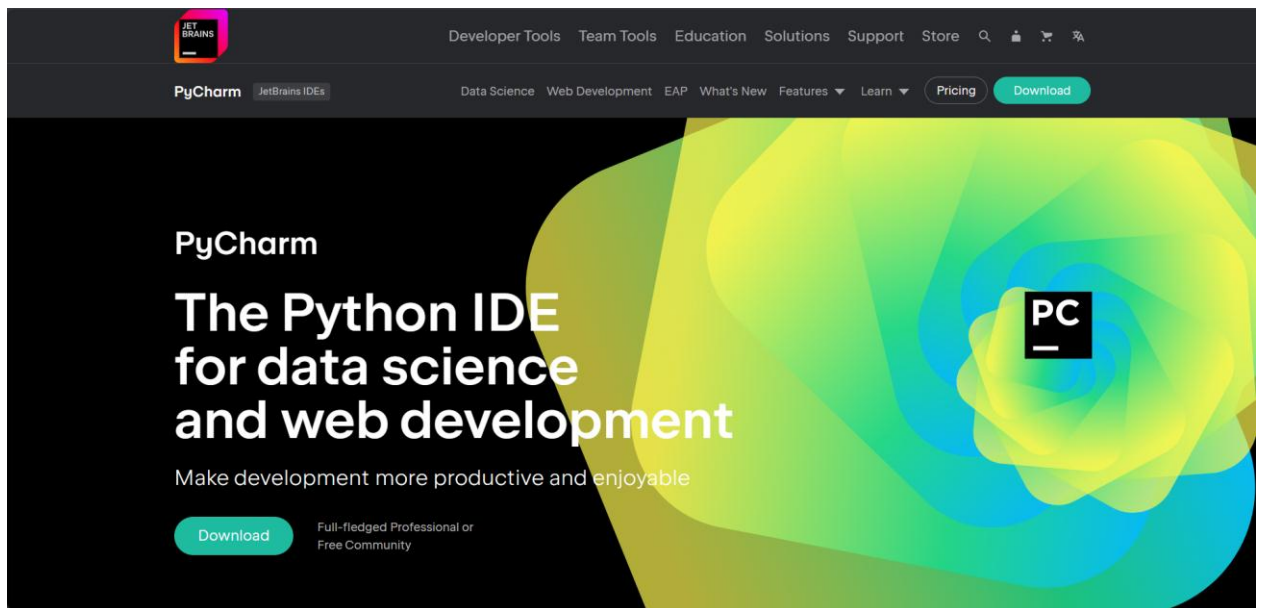


Рисунок 5 – Pycharm главная страница сайта

PyCharm - это интегрированная среда разработки (IDE) для языка программирования Python, разработанная компанией JetBrains. Она предоставляет широкий набор инструментов для разработки, отладки, тестирования и управления проектами на Python. PyCharm является одной из самых популярных IDE для Python и широко используется в сообществе разработчиков на этом языке.

Преимущества PyCharm:

- Широкий набор инструментов для разработки на Python;
- Автодополнение кода и подсветка синтаксиса;
- Интеграция с системами контроля версий;
- Поддержка виртуальных окружений;
- Отладка кода;
- Поддержка различных фреймворков;
- Доступность в двух версиях: Community Edition и Professional Edition.

Недостатки PyCharm:

- Некоторые функции могут быть сложными для новичков;

- Professional Edition платная;
- Не все функции доступны в Community Edition;
- Некоторые пользователи могут считать интерфейс PyCharm слишком громоздким.

На основании вышеизложенного, я выбрал IDE PyCharm.

2.2 Разработка логической модели автоматизированной информационной системы

В данном разделе осуществляется разработка логической модели автоматизированной информационной системы (АИС) с применением нотации Unified Modeling Language (UML). Логическая модель включает в себя диаграмму прецедентов, схему развертывания, диаграмму активности и схемы совместной работы .

2.3.1 Диаграмма прецедентов

Диаграмма прецедентов является одним из ключевых инструментов в методологии объектно-ориентированного анализа и проектирования программного обеспечения. Она позволяет наглядно представить функциональные требования системы через взаимодействие с внешними пользователями и иллюстрирует поведение системы в ответ на внешние воздействия. Диаграмма прецедентов предоставляет ясное и структурированное представление о том, какие функции должна выполнять система и как они будут использованы, что делает её неотъемлемым компонентом процесса разработки программного обеспечения [15].

На Рисунке 6 представлена диаграмма прецедентов разрабатываемой АИС.

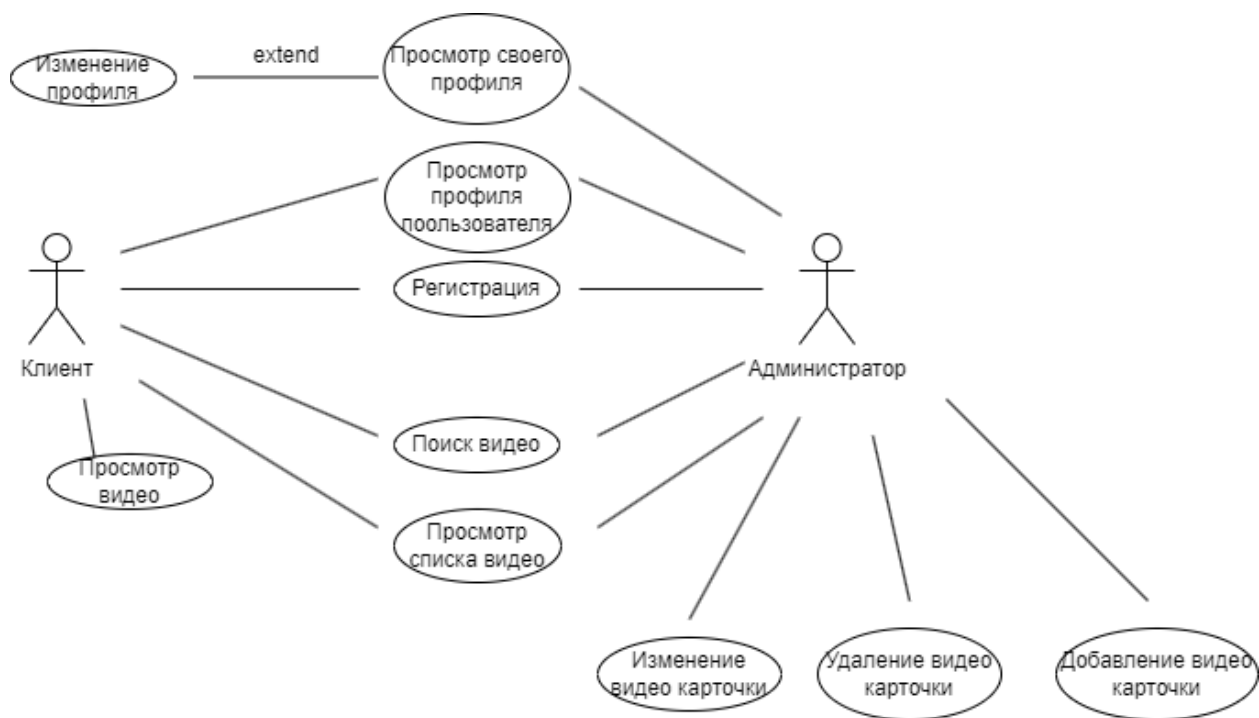


Рисунок 6 – Диаграмма прецедентов

Диаграмма включает два типа пользователей «Клиент» и «Администратор», каждый из которых взаимодействует с системой через определённые функции (прецеденты).

Клиент:

- регистрация: прецедент, позволяющий пользователю зарегистрироваться в системе для получения доступа к её функциям
- просмотр своего профиля: прецедент, предоставляющий пользователю возможность просмотра своего профиля. Включает в себя возможность изменения профиля
- изменение профиля: прецедент, позволяющий пользователю изменять свои персональные данные в профиле. Расширяет (extend) функционал прецедента «Просмотр своего профиля»
- просмотр профиля пользователя: прецедент, позволяющий пользователю просматривать профили других пользователей

- поиск видео: прецедент, предоставляющий пользователю возможность поиска видеоконтента в системе
- просмотр списка видео: прецедент, позволяющий пользователю просматривать список доступных видео
- просмотр видео: прецедент, предоставляющий пользователю возможность просмотра выбранного видео из списка.

Администратор:

- изменение видео карточки: прецедент, позволяющий администратору изменять метаданные (название, описание, теги) видеоконтента
- удаление видео карточки: прецедент, позволяющий администратору удалять видеоконтент из системы
- добавление видео карточки: прецедент, предоставляющий администратору возможность добавлять новый видеоконтент в систему.

Взаимодействия и зависимости:

- прецедент «Просмотр своего профиля» включает возможность «Изменение профиля» (отношение extend), что указывает на дополнительный функционал, доступный при просмотре собственного профиля
- основные функции для пользователя включают взаимодействие с профилем и видеоконтентом, тогда как функции администратора сосредоточены на управлении видеоконтентом.

2.3.2 Схема развертывания

Схема развертывания описывает физическую архитектуру системы, включая распределение компонентов и их размещение на аппаратном обеспечении.

На Рисунке 6 показана схема развертывания разрабатываемой АИС.

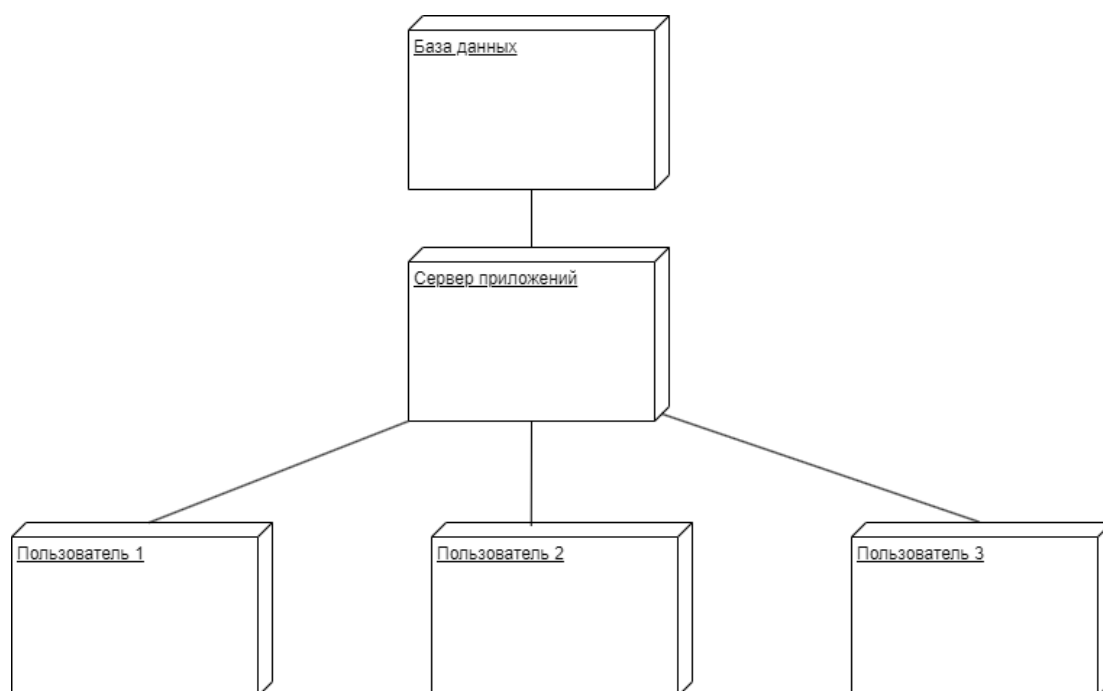


Рисунок 7 – Схема развертывания

В верхней части диаграммы находится «База данных», которая является хранилищем всей информации, используемой системой. База данных связана с «Сервером приложений», расположенным ниже, который отвечает за обработку данных и выполнение логики приложений. Сервер приложений, в свою очередь, взаимодействует с тремя пользователями: «Пользователь 1», «Пользователь 2» и «Пользователь 3». Пользователи подключаются к серверу приложений для выполнения различных операций, таких как запросы к базе данных, получение результатов обработки данных и взаимодействие с приложением.

2.3.3 Диаграмма активности

Диаграмма активности представляет собой последовательность действий или процессов в системе, позволяя визуализировать шаги, необходимые для выполнения определенной задачи или достижения цели. На Рисунке 7 приведена диаграмма активности, иллюстрирующая основные процессы в рамках разрабатываемой АИС.

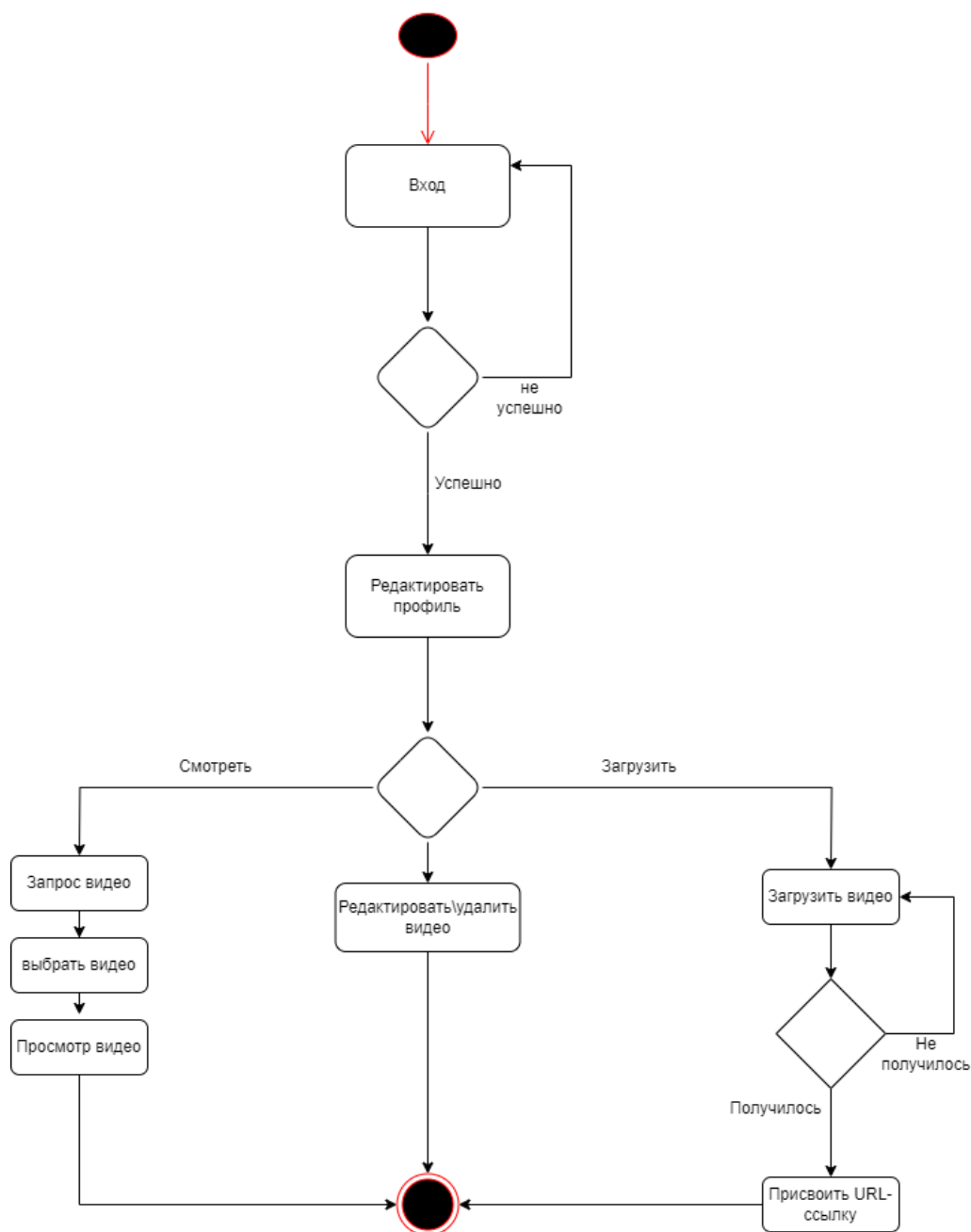


Рисунок 8 – Диаграмма активности

Диаграмма начинается с начальной точки входа, обозначенной черным кружком. Первым шагом процесса является «Вход» в систему. В случае неуспешного входа, пользователь возвращается к повторной попытке входа. Если вход успешен, пользователь переходит к этапу «Редактировать профиль».

После успешного редактирования профиля, пользователь имеет два основных пути дальнейших действий: «Смотреть» видео или «Загрузить» видео. В случае выбора просмотра видео, процесс включает три шага: «Запрос видео», «Выбрать видео» и «Просмотр видео». Каждый из этих шагов логически следует друг за другом, обеспечивая пользователю последовательный доступ к видеоконтенту.

Если пользователь выбирает загрузку видео, процесс включает шаг «Загрузить видео», после которого проверяется успешность загрузки. В случае неудачной загрузки, система возвращает пользователя к повторной попытке загрузки видео. Если загрузка видео успешна, следующий шаг - «Присвоить URL-ссылку», что завершает процесс загрузки.

На этом этапе обе ветки процесса - просмотр видео и загрузка видео - сходятся на этапе «Редактировать/удалить видео». Это позволяет пользователю внести изменения или удалить ранее загруженные видео, завершая таким образом взаимодействие с системой. Конечная точка процесса обозначена черным кружком, заключающим блок-схему.

Таким образом, данная блок-схема подробно описывает последовательность действий пользователя при взаимодействии с системой видеоконтента, охватывая основные функциональные возможности, такие как вход в систему, редактирование профиля, просмотр видео, загрузка видео, а также редактирование или удаление загруженного контента.

2.3.4 Схемы совместной работы

Схемы совместной работы представляют процессы совместной работы пользователей в рамках системы. На Рисунках 8 и 9 показаны схемы совместной работы, связанные с просмотром и загрузкой видео в разрабатываемой АИС.

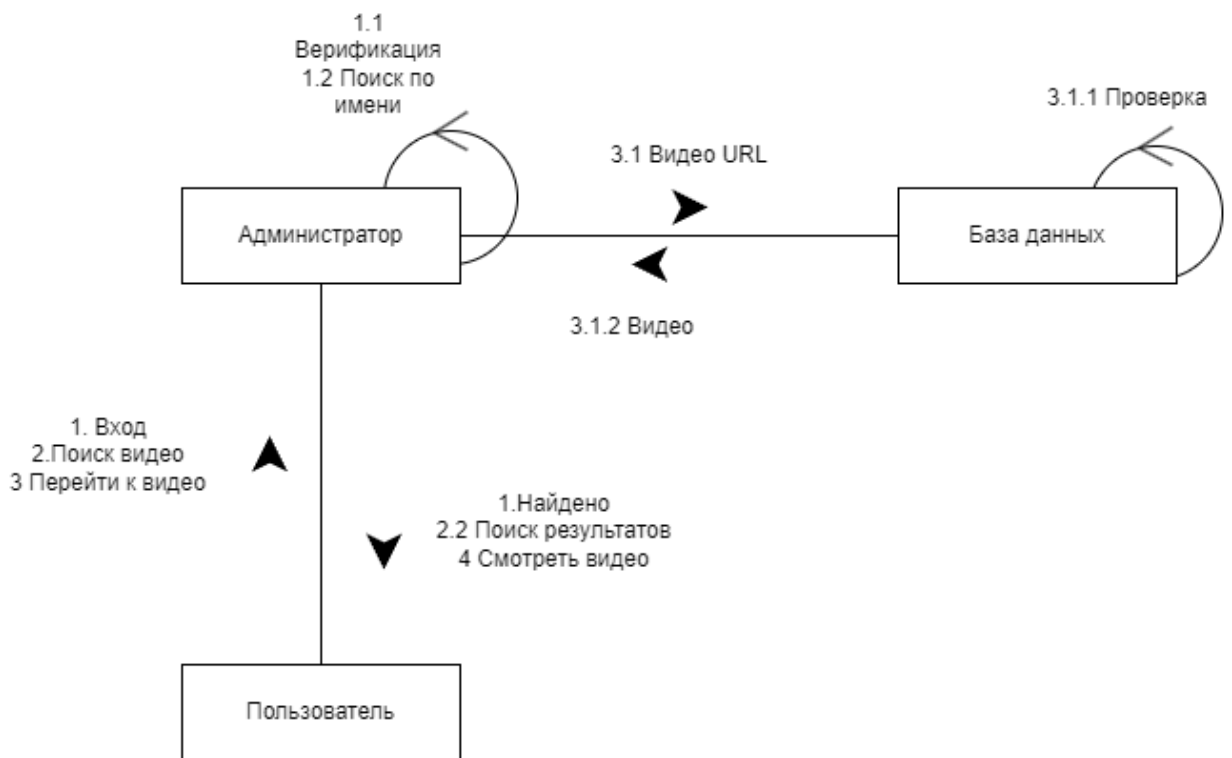


Рисунок 9 – Схема совместной работы. Просмотр видео



Рисунок 10 – Схема совместной работы. Загрузить видео

Эти логические модели, разработанные с использованием UML, играют ключевую роль в проектировании и анализе функциональности и архитектуры разрабатываемой АИС. Они предоставляют фундаментальные основы для последующей разработки и реализации системы.

2.3 Требования к аппаратно-программному обеспечению автоматизированной информационной системы

Техническое обеспечение системы должно максимально и наиболее эффективно использовать существующие на предприятии технические средства.

В состав технического комплекса должны входить следующие технические средства:

- сервер СУБД и АИС
- ПК пользователей.

Минимальные требования к серверу СУБД и АИС:

- процессор – частота не менее 2.4 ГГц
- оперативная память – не менее 4 Гб
- свободное место на жестком диске - не менее 25 Гб
- подключение к сети Интернет
- операционная система - Windows.

Минимальные требования к ПК пользователей:

- процессор – частота не менее 2 ГГц
- оперативная память – не менее 2 Гб
- подключение к сети Интернет
- ПК под управлением любой популярной ОС (Windows, MacOS, Linux и др.)
- любой из популярных браузеров (Google Chrome, Mozilla Firefox, Opera и др.).

После того как были сформулированы и описаны требования к аппаратно-программному обеспечению информационной системы, перейдем к подведению итогов второй главы

Выводы по главе 2.

В ходе второй главы был проведен комплексный анализ и проектирование web-приложения, включая выбор оптимальных технологий,

разработку логической модели АИС и определение требований к аппаратно-программному обеспечению.

Выбор технологий и инструментов разработки был выполнен с учетом требований проекта и современных тенденций веб-разработки. Это включало выбор языка программирования, фреймворка и системы управления базами данных, обеспечивая оптимальную основу для дальнейшей реализации системы.

Разработка логической модели АИС с использованием нотации Unified Modeling Language (UML) позволила визуализировать структуру и функциональность системы. Это помогло уяснить основные компоненты и процессы в системе, что является важным шагом для последующей разработки и внедрения.

Определение требований к аппаратно-программному обеспечению было выполнено с целью обеспечения эффективной работы web-приложения. Это включало оценку вычислительных ресурсов, поддерживаемых операционных систем и других технических параметров, необходимых для обеспечения стабильной и надежной работы системы.

В целом, вторая глава играет ключевую роль в подготовке к разработке и реализации web-приложения, предоставляя необходимые основы и рекомендации для успешного завершения проекта.

Глава 3 Разработка и администрирование web-приложения.

3.1 Структура проекта

Рассмотрим структуру проекта, представленного на изображениях, и опишем назначение основных файлов и папок.

На рисунке 11 представлена главная директория проекта.

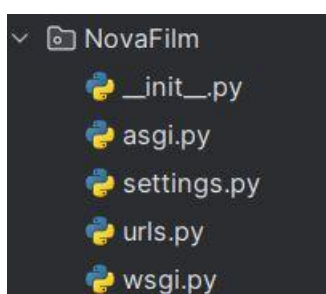


Рисунок 11-Главная директория проекта

`__init__.py`: Этот файл обозначает директорию как пакет Python. Он может быть пустым или содержать инициализационный код для пакета.

`asgi.py`: Этот файл содержит настройки для ASGI (Asynchronous Server Gateway Interface), которые позволяют проекту обрабатывать асинхронные запросы. ASGI используется для поддержки веб-сокетов и других асинхронных протоколов.

`settings.py`: Файл с основными настройками проекта Django. Здесь хранятся конфигурации базы данных, настройки безопасности, приложения, зарегистрированные в проекте, и другие параметры.

`urls.py`: Этот файл содержит маршрутизацию URL-адресов проекта. Здесь определены все пути, по которым могут обращаться пользователи, и указано, какие представления (views) обрабатывают эти пути.

wsgi.py: Этот файл содержит настройки для WSGI (Web Server Gateway Interface), которые позволяют проекту обрабатывать синхронные запросы. WSGI используется для развертывания Django приложений на серверах.

Теперь рассмотрим директорию users на рисунке 12.

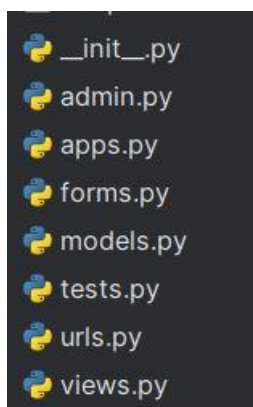


Рисунок 12-Директория приложения users

__init__.py: Как и в главной директории, этот файл обозначает папку как пакет Python.

admin.py: В этом файле регистрируются модели для административной панели Django. Админка позволяет легко управлять данными через веб-интерфейс.

apps.py: Этот файл содержит конфигурацию приложения. Здесь можно определить некоторые параметры приложения, такие как его имя и метки.

forms.py: Этот файл используется для определения форм Django. Формы позволяют удобно обрабатывать ввод данных от пользователей.

models.py: В этом файле определяются модели базы данных. Модели представляют структуру данных и являются основой для взаимодействия с базой данных в Django.

tests.py: Этот файл содержит тесты для приложения. Тесты помогают удостовериться, что приложение работает правильно, и изменения в коде не приводят к неожиданным ошибкам.

`urls.py`: Этот файл содержит маршрутизацию URL-адресов конкретного приложения. Он аналогичен файлу `urls.py` в главной директории, но маршруты в нем относятся только к данному приложению.

`views.py`: В этом файле определяются представления (`views`) приложения. Представления обрабатывают запросы пользователей и возвращают соответствующие ответы, такие как HTML-страницы или JSON-данные [19], [20].

3.2 Разработка административной панели

Административная панель на сайте «NOVAFilm» разработана с использованием фреймворка Django. Одним из ключевых преимуществ использования Django является автоматическое создание интерфейса для административной панели. Это позволяет существенно упростить управление данными сайта. Благодаря этому функционалу, модели данных, такие как `HomePageModel`, `Genre`, `Director`, `Serial`, `Season`, `Episode`, `comments` и `comments_serial`, после регистрации в административной панели становятся доступны для администраторов сайта.

Автоматически сгенерированный интерфейс предоставляет базовые формы для ввода данных и таблицы для отображения записей, что позволяет администратору без дополнительных усилий выполнять операции по созданию, редактированию и удалению записей. Таким образом, интерфейс админки обеспечивает удобное и эффективное управление контентом сайта, позволяя администратору сосредоточиться на содержании и структуре данных без необходимости разработки пользовательских интерфейсов для каждого действия.

На рисунке 13 изображён фрагмент кода, написанного на языке программирования Python с использованием фреймворка Django, который предназначен для создания модели базы данных.

```

# Create your models here.
# Movies
19 usages
class HomePageModel(models.Model):
    title = models.CharField(max_length=200)
    director = models.ForeignKey('Director', on_delete=models.CASCADE, default=None)
    release_date = models.CharField(max_length=70, default='None')
    short_intro = models.TextField(max_length=700)
    IMDb_RATING = models.CharField(max_length=50, default=None)
    genre = models.ManyToManyField('Genre')
    poster = models.ImageField(upload_to='Posters/')
    movie_page_poster = models.ImageField(upload_to='Posters/MoviePage/', null=True, blank=True)
    summary = models.TextField(max_length=1600)
    trailer = models.CharField(max_length=650, null=True, blank=True)
    download_link_1080 = models.CharField(max_length=650, null=True, blank=True)
    download_link_720 = models.CharField(max_length=650, null=True, blank=True)
    download_link_480 = models.CharField(max_length=650, null=True, blank=True)
    page_view = models.IntegerField(default=1)
    created = models.DateTimeField(auto_now_add=True)
    id = models.UUIDField(default=uuid.uuid4, unique=True, primary_key=True, editable=False)

    def __str__(self):
        return self.title

class Meta:
    ordering = ['-created']

```

Рисунок 13 – Фрагмент кода Movies

Данная модель, названная `HomePageModel`, используется для хранения информации о фильмах и включает в себя различные атрибуты, необходимые для полноценного представления данных о фильме на веб-сайте.

Основные поля модели включают в себя название фильма (`title`), внешний ключ для связи с моделью `Director` (`director`), дату выхода фильма (`release_date`), краткое описание (`short_intro`), рейтинг IMDb (`IMDb_RATING`) и жанры фильма, которые реализованы с помощью Many-to-Many связи с моделью `Genre` (`genre`). Также в модели предусмотрены поля для хранения изображений: постера фильма (`poster`) и постера для страницы фильма (`movie_page_poster`).

Для более подробного представления фильма предусмотрены текстовые поля: длинное описание (`summary`) и ссылка на трейлер фильма (`trailer`). Кроме того, включены поля для хранения ссылок на скачивание фильма в различных разрешениях: 1080p (`download_link_1080`), 720p (`download_link_720`) и 480p (`download_link_480`). Для отслеживания популярности фильма имеется поле с количеством просмотров страницы (`page_view`).

Дополнительно, каждую запись в базе данных сопровождает автоматическое поле с датой и временем её создания (`created`). Уникальный идентификатор записи (`id`) генерируется автоматически при её создании и является первичным ключом.

Метод `__str__` был переопределён для удобного отображения названия фильма при преобразовании объекта модели в строку. Класс `Meta` задаёт порядок сортировки записей по дате их создания в обратном порядке.

Таким образом, данная модель обеспечивает полноценное представление информации о фильмах, позволяя гибко управлять данными и легко интегрировать их в веб-сайт, предоставляя пользователю исчерпывающие сведения о каждом фильме [3].

На рисунке 14 представлена административная панель Django, предназначенная для управления пользователями в системе аутентификации и авторизации.



Рисунок 14 – Административная панель пользователя

Интерфейс данной панели демонстрирует страницу управления пользователями, где администратор имеет возможность просматривать и изменять информацию о зарегистрированных пользователях.

В верхней части экрана отображается уведомление о том, что данные пользователя с именем «М» были успешно изменены. Ниже расположена основная таблица с перечнем пользователей, включающая следующие колонки: имя пользователя (`USERNAME`), адрес электронной почты (`EMAIL ADDRESS`), имя (`FIRST NAME`), фамилия (`LAST NAME`) и статус сотрудника (`STAFF STATUS`). В данной таблице показаны семь пользователей с различными именами и адресами электронной почты. Для некоторых пользователей указаны имена и фамилии, а также статус сотрудника, который представлен в виде зелёной или красной иконки, обозначающих активный или неактивный статус соответственно.

Слева от каждого пользователя находится флажок, позволяющий администратору выбрать одного или нескольких пользователей для выполнения различных действий. Выше таблицы расположены элементы управления, включающие поле поиска для фильтрации пользователей по различным критериям и выпадающее меню с возможными действиями.

Справа на странице расположена панель фильтров, позволяющая фильтровать пользователей по статусу сотрудника, статусу суперпользователя и активности. Эта панель предоставляет быстрый доступ к наиболее часто используемым фильтрам, упрощая процесс управления пользователями.

В правом верхнем углу экрана присутствует кнопка «ADD USER», позволяющая добавить нового пользователя в систему. В верхней части страницы также находятся ссылки для просмотра сайта, изменения пароля и выхода из системы, предоставляя администратору быстрый доступ к этим функциям.

Таким образом, данная страница административной панели Django предоставляет полный набор инструментов для управления пользователями, обеспечивая удобный интерфейс для выполнения различных операций, включая просмотр, изменение, фильтрацию и добавление новых пользователей.

На рисунке 15 изображена главная страница административной панели Django, используемая для управления различными аспектами веб-сайта.

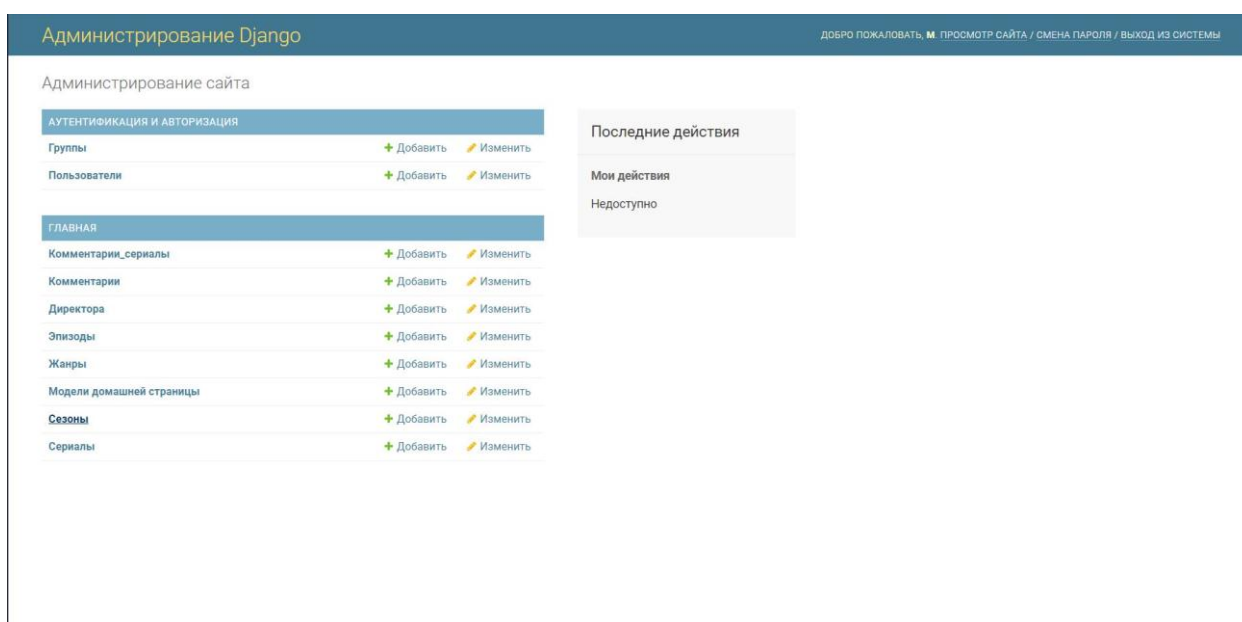


Рисунок 15 – Административная панель

Административная панель предоставляет доступ к различным категориям и функциям сайта, позволяя администраторам добавлять, изменять и управлять данными.

В левой части экрана расположены основные секции управления сайтом, разделённые на две категории: «Аутентификация и авторизация» и «Главная». В разделе «Аутентификация и авторизация» присутствуют ссылки для управления группами и пользователями. Каждая из этих ссылок имеет два варианта действий: «Добавить» и «Изменить», что позволяет

администратору либо создать новую группу или пользователя, либо редактировать уже существующие записи.

Раздел «Главная» включает в себя множество ссылок, каждая из которых соответствует различным моделям данных сайта. Эти модели включают комментарии к сериалам, комментарии, директоров, эпизоды, жанры, модели домашней страницы, сезоны и сериалы. Каждая ссылка также снабжена опциями «Добавить» и «Изменить», предоставляя аналогичные возможности для управления этими данными.

В правой части экрана находится раздел «Последние действия», который отображает последние выполненные действия пользователя. В данном случае отображается надпись «Мои действия - Недоступно», что указывает на отсутствие недавно выполненных действий.

В верхней части страницы, справа, расположены элементы навигации и управления сеансом: приветствие пользователя (в данном случае, «Добро пожаловать, М.»), ссылка на просмотр сайта, смена пароля и выход из системы. Эти элементы позволяют пользователю быстро перейти к основным функциям и управлять своей учётной записью.

Таким образом, данная главная страница административной панели Django предоставляет администраторам удобный интерфейс для доступа к различным разделам и функциям сайта, обеспечивая эффективное управление данными и настройками сайта [7].

3.3 Разработка интерфейса пользователя

На рисунке 16 показана страница сайта «NOVAFilm», отображающая раздел с популярными фильмами.

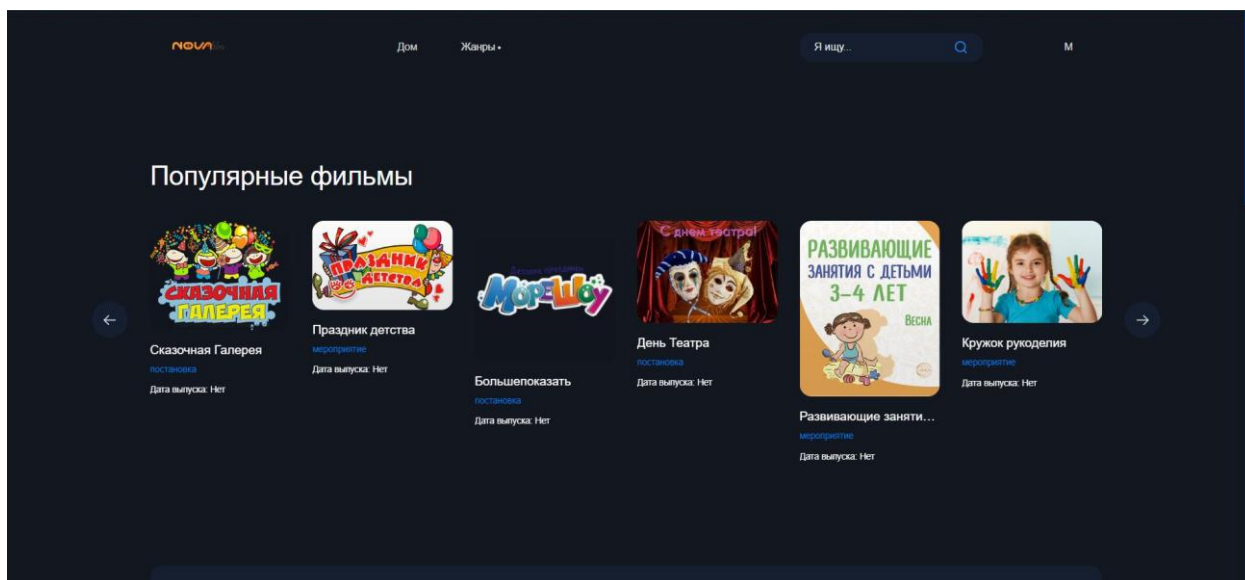


Рисунок 16 – Главная страница

В верхней части интерфейса находится логотип сайта, основное навигационное меню с вкладками «Дом» и «Жанры», а также строка поиска и иконка профиля пользователя. Основная часть страницы посвящена популярным фильмам, представленным в виде карусели с постерами. В данной карусели отображены следующие фильмы: «Сказочная Галерея», «Праздник детства», «Море шоу», «День Театра», «Развивающие занятия с детьми 3-4 лет» и «Кружок рукоделия». Для каждого фильма указана категория (постановка или мероприятие) и сообщение о том, что дата выпуска отсутствует. Слева и справа от карусели находятся стрелки для навигации, позволяющие пролистывать доступные фильмы. Интерфейс направлен на предоставление пользователю удобного доступа к популярному контенту, а также включает функции поиска и навигации по жанрам [10], [11], [8].

На рисунке 17 показана страница настроек профиля пользователя на сайте «NOVAFilm».

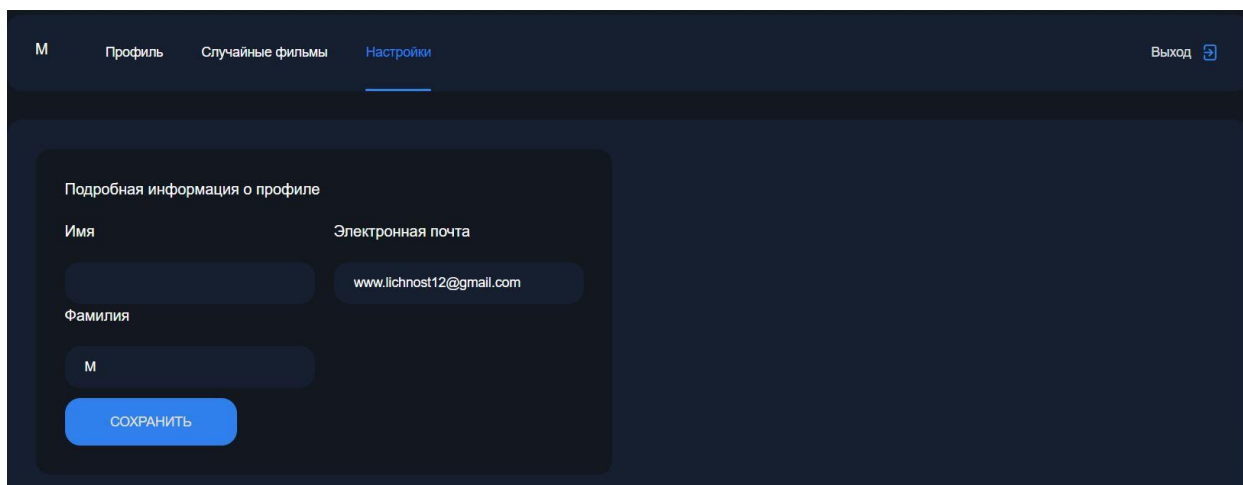


Рисунок 17 – Настройки профиля

В верхней части интерфейса находится навигационное меню с вкладками «Профиль», «Случайные фильмы» и «Настройки», а также кнопка выхода из аккаунта. На основной части страницы отображается форма для редактирования личной информации пользователя. В этой форме можно увидеть поля для ввода имени, фамилии и электронной почты. Поля для имени и фамилии пока пусты, а в поле для электронной почты уже введён адрес пользователя. Под этими полями расположена синяя кнопка «СОХРАНИТЬ», предназначенная для сохранения изменений, внесённых в профиль. Таким образом, интерфейс обеспечивает пользователю удобный способ обновления личных данных и управления информацией своего профиля [12], [17].

На рисунке 18 представлена форма авторизации на сайте «NOVAFilm», предназначенная для ввода учетных данных пользователя с целью входа в систему.

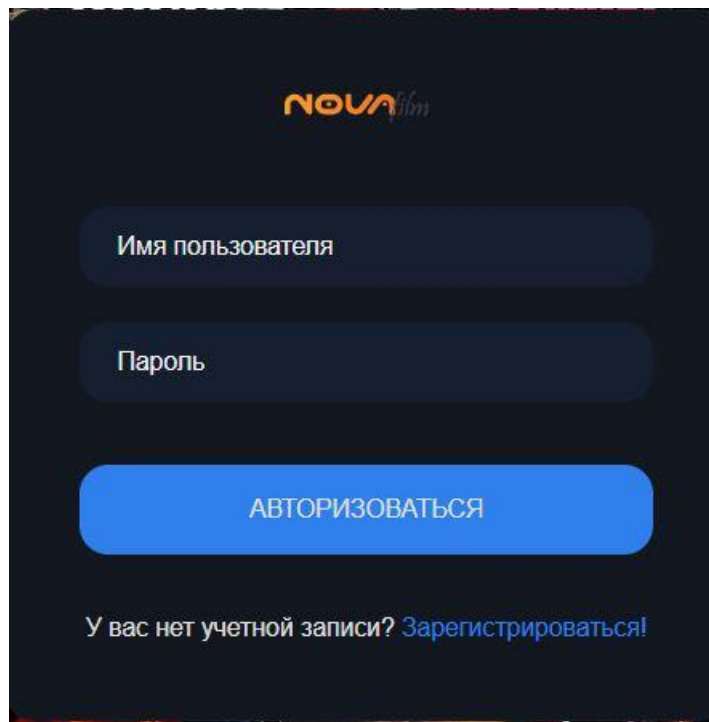


Рисунок 18 – Авторизация пользователя

В верхней части формы находится логотип сайта, позволяющий пользователю сразу идентифицировать ресурс. Далее расположены поля для ввода имени пользователя и пароля, при этом поле пароля скрывает вводимые символы для обеспечения конфиденциальности. Под полями ввода размещена синяя кнопка «АВТОРИЗОВАТЬСЯ», которая активирует процесс аутентификации, отправляя введенные данные на сервер для проверки. В случае успешной проверки пользователь получает доступ к своему профилю и другим защищенным разделам сайта. Если учетной записи у пользователя нет, он может воспользоваться ссылкой «Зарегистрироваться!», расположенной под кнопкой авторизации, для перехода на страницу регистрации. Данный интерфейс обеспечивает удобство использования и безопасность, контролируя доступ к ресурсу [13], [6].

На рисунке 19 представлен фрагмент регистрации в разработке интерфейса пользователя.

```

def registerUser(request):
    # use this variable in front end
    page = 'register'
    # use register form model
    form = Register()

    # if method is post, get the data
    if request.method == 'POST':
        form = Register(request.POST)
        # check the data
        if form.is_valid():
            # save but don't commit new data
            user = form.save(commit=False)
            # make username lower letter and then save the data
            user.username = user.username.lower()
            user.save()

            messages.success(request, 'User account was created!')

            # redirect the user to profile page
            login(request, user)
            return redirect('profile')

        else:
            messages.error(
                request, 'An error has occurred during registration')

    # send data to front end
    context = {'page' : page, 'form' : form}
    return render(request, 'users/login_register.html', context)

```

Рисунок 19 – Фрагмент кода. Регистрация новых пользователей

Этот фрагмент кода представляет собой обработчик (view) для регистрации новых пользователей на сайте.

Общее описание того, как работает этот фрагмент кода:

- инициализация: когда пользователь обращается к странице регистрации, функция `registerUser` инициализируется. В начале создаётся переменная `page` со значением `register`, которая

используется для отображения информации на фронтенде. Также создаётся объект формы `form` на основе модели `Register`

- обработка POST-запроса: если запрос к этой странице является POST-запросом (т.е., когда пользователь отправляет данные формы регистрации), данные из формы передаются в объект `form` для дальнейшей обработки
- Валидация данных: проверяется, является ли форма валидной.

Если данные корректны:

- Создаётся пользователь, но пока не сохраняется в базе данных (`commit=False`)
- Приводится имя пользователя к нижнему регистру
- Сохраняется пользователь в базу данных
- Отправляется сообщение об успешной регистрации
- Пользователь автоматически логинится (функция `login`)
- Пользователь перенаправляется на страницу профиля
- обработка ошибок: если данные формы невалидны, отправляется сообщение об ошибке
- отображение формы: независимо от метода запроса (POST или GET), на фронтенд отправляются переменные `page` и `form` для рендеринга шаблона `login_register.html`.

Этот обработчик эффективно управляет процессом регистрации пользователя, начиная с отображения формы, валидации введённых данных, сохранения пользователя и завершения авторизацией и перенаправлением на страницу профиля. Если происходят ошибки, пользователю отправляется соответствующее сообщение [5], [14].

3.4 Тестирование веб-приложения

Для обеспечения стабильности, безопасности и соответствия функциональным требованиям веб-приложения необходимо тщательное

тестирование. В данной подглаве подробно рассмотрены этапы и методики тестирования, примененные при разработке веб-приложения для анимационной студии.

3.4.1 Функциональное тестирование

Функциональное тестирование проводилось для проверки корректности работы всех функций приложения. Оно включало следующие этапы:

- тестирование пользовательской аутентификации и авторизации: проверка корректности регистрации, входа в систему, изменения пароля и восстановления пароля. Тестирование прав доступа для различных ролей пользователей (администратор, сотрудник, пользователь)
- тестирование административной панели: проверка возможности добавления, редактирования и удаления данных о пользователях, фильмах, жанрах и других сущностях. Оценка корректности работы фильтров и сортировок
- тестирование пользовательского интерфейса: Проверка доступности всех страниц и разделов сайта, корректности отображения данных и работы интерактивных элементов (формы, кнопки, ссылки)
- тестирование функциональности загрузки и отображения мультимедийного контента: проверка возможности загрузки и отображения постеров, трейлеров и других мультимедийных элементов.

3.4.2 Тестирование производительности

Тестирование производительности проводилось для оценки быстродействия и устойчивости веб-приложения при различных нагрузках. Включало следующие этапы:

- нагрузочное тестирование: оценка работы приложения при большом количестве одновременных пользователей и запросов.

Использование инструментов для симуляции нагрузки и анализа полученных данных

- стресс-тестирование: проверка устойчивости приложения при пиковых нагрузках, превышающих обычные условия эксплуатации. Оценка поведения системы в условиях ограниченных ресурсов.

3.4.3 Тестирование безопасности

Безопасность веб-приложения была проверена на наличие уязвимостей и обеспечение защиты данных. Включало следующие этапы:

- тестирование на уязвимости типа SQL-инъекций и XSS (межсайтовый скриптинг): проверка возможности внедрения вредоносного кода через пользовательские входные данные.
- тестирование защиты паролей и данных пользователей: оценка механизмов шифрования и хранения паролей, проверка защиты личной информации пользователей.

3.4.4 Пользовательское тестирование

На завершающем этапе проводилось тестирование с участием реальных пользователей. Включало следующие этапы:

- бета-тестирование: привлечение ограниченной группы пользователей для тестирования приложения в реальных условиях. Сбор обратной связи и выявление возможных проблем.
- тестирование пользовательского опыта: оценка удобства и интуитивности интерфейса, выявление проблем, с которыми могут столкнуться конечные пользователи.

Результаты тестирования

В результате проведенных тестов были выявлены и устранены многочисленные ошибки и уязвимости, что позволило значительно повысить качество и надежность веб-приложения. Все функциональные требования были подтверждены, а приложение успешно прошло проверку на производительность и безопасность. Тестирование показало, что

разработанное приложение соответствует требованиям и готово к развертыванию в реальных условиях эксплуатации.

Тестирование веб-приложения является критически важным этапом, обеспечивающим его высокое качество и стабильность работы. В ходе тестирования были применены разнообразные методики и инструменты, позволившие выявить и устранить проблемы на ранних этапах, что существенно сократило риски и повысило удовлетворенность пользователей.

3.5 Расчет экономической эффективности

Сайт, должен облегчать работу сотрудников, сокращать время и повышать эффективность работы.

Чтобы узнать эффективность от внедрения, нужно рассчитать показатели трудовых затрат.

Без использования информационной системы время, которое уходило на поиск видео и отправка пользователю и просмотра, составляло около 30 минут. После внедрения затраты сократились до 2 минут.

Абсолютное снижение временных затрат рассчитывается по формуле 1.

Формула 1 – Абсолютное снижение временных затрат

$$\Delta T = T_0 - T_1 = 30 - 2 = 28(\text{мин})$$

Где:

- T_0 - это количество времени, которое необходимо было потратить на осуществление операций в первоначальном варианте
- T_1 - это время на выполнение таких же операций после внедрения информационной системы.

Коэффициент относительного снижения рассчитан по формуле 2.

Формула 2 – Коэффициент относительного снижения

$$K_T = \frac{\Delta T}{T_0} * 100\% = \frac{28}{30} * 100\% = 93.3\%$$

Индекс снижения трудовых затрат рассчитан по формуле 3.

Формула 3 – Индекс снижения трудовых затрат

$$Y_T = \frac{T_0}{T_1} = \frac{30}{2} = 15$$

Для того чтобы показать разницу в трудоемкости до и после внедрения информационной системы, построена диаграмма 1.

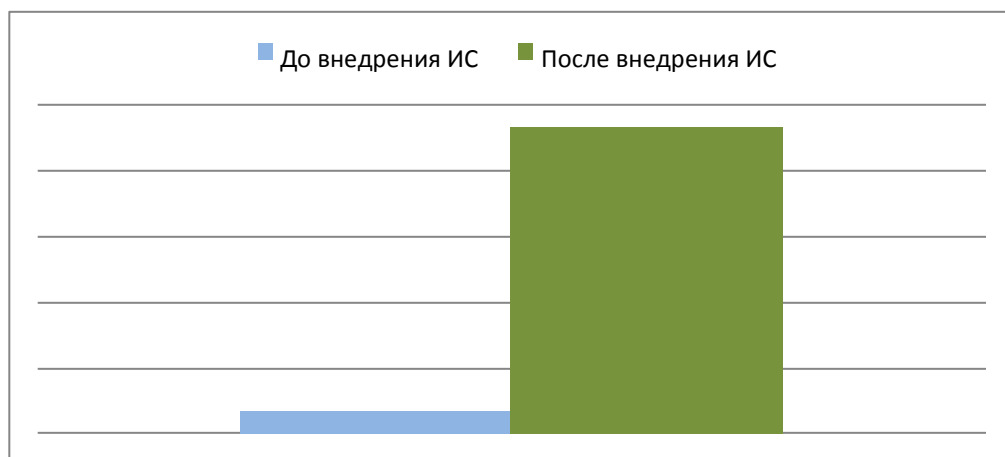


Диаграмма 1 – Изменение трудозатрат

Эта диаграмма демонстрирует, что время, необходимое для поиска видео и просмотра, после внедрения информационной системы сократилось примерно в 15 раз.

Данный факт подтверждает, что разработанная система является эффективным решением для оптимизации трудозатрат и улучшения качества работы студии.

Выводы по главе 3.

В третьей главе была проведена разработка и администрирование web-приложения. Была определена структура проекта, что является основой для дальнейшей разработки. Разработка административной панели была

выполнена с учетом необходимости управления контентом и функционалом сайта. Разработка интерфейса пользователя была направлена на обеспечение удобства и простоты использования веб-приложения.

Тестирование веб-приложения было проведено в нескольких направлениях: функциональное тестирование, тестирование производительности, тестирование безопасности и пользовательское тестирование. Это позволило обеспечить стабильность работы приложения, его безопасность и удобство использования.

Расчет экономической эффективности позволил оценить целесообразность разработки данного веб-приложения. В результате проведенных исследований было выявлено, что разработка и внедрение данного веб-приложения является экономически целесообразной.

Таким образом, третья глава представляет собой важный этап в разработке веб-приложения, включающий в себя как техническую, так и экономическую составляющие.

Заключение

В ходе работы была рассмотрена разработка и администрирование web-приложения для анимационной студии. Работа включала анализ предметной области, определение требований к функциональности системы, проектирование и разработку web-приложения с учетом всех необходимых аспектов.

В первой главе была проведена всесторонняя технико-экономическая характеристика предметной области. Рассмотрены бизнес-процессы анимационной студии, выявлены ключевые потребности и проблемы, требующие автоматизации. Обоснована необходимость создания автоматизированной информационной системы для повышения эффективности управления студией. На основании проведенного анализа были сформированы требования к функциональности будущей системы и проведен анализ существующих решений на рынке, что позволило выявить недостатки и преимущества аналогичных разработок.

Во второй главе был проведен выбор технологий и инструментов для разработки web-приложения. Основываясь на требованиях и специфике проекта, был выбран язык программирования Python, фреймворк Django и система управления базами данных SQLite3. Кроме того, определены средства разработки и требования к аппаратно-программному обеспечению. В результате была разработана логическая модель системы, описаны ее основные компоненты и взаимосвязи.

Третья глава была посвящена непосредственной разработке web-приложения. Были описаны структура проекта, разработка административной панели и интерфейса пользователя. Административная панель была реализована с учетом требований к функциональности, предоставляя удобные инструменты для управления пользователями, контентом и другими ключевыми аспектами системы. Интерфейс пользователя был разработан с акцентом на удобство и интуитивность, что

обеспечило простоту взаимодействия с приложением для конечных пользователей. В завершение главы проведен расчет экономической эффективности внедрения разработанного web-приложения, который показал значительное снижение операционных затрат и повышение эффективности управления анимационной студией.

Результаты проведенной работы подтверждают, что разработанное web-приложение соответствует всем поставленным требованиям и обладает необходимой функциональностью для эффективного управления анимационной студией. Использование современных технологий и инструментов разработки обеспечило высокую производительность и надежность системы. Внедрение автоматизированной информационной системы позволило оптимизировать бизнес-процессы студии, улучшить качество управления и повысить экономическую эффективность деятельности. Данная работа может служить основой для дальнейших исследований и разработок в области автоматизации бизнес-процессов в индустрии анимации и других творческих сферах.

Список используемой литературы

1. Python API для создания и редактирования документов Microsoft Word [Электронный ресурс] URL: <https://products.fileformat.com/ru/word-processing/python/python-docx/>
2. PyCharm IDE для профессиональной разработки на Python [Электронный ресурс] URL: <https://www.jetbrains.com/ru-ru/pycharm/>
3. Криволапов С. Я. Введение в анализ данных. Поиск структуры данных с применением языка Python. М. : НИЦ ИНФРА-М, 2024. 177 с.
4. Лучшие инструменты Python: IDE и редакторы кода [Электронный ресурс] URL: <https://eternalhost.net/blog/razrabotka/python-ide>
5. Шелудько В. М. Основы программирования на языке высокого уровня Python : ЮФИ, 2017. 146 с.
6. Немцова, Т. И. Практикум по информатике. Компьютерная графика и web-дизайн : учебное пособие / Т.И. Немцова, Ю.В. Назарова ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2023. — 288 с. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование). - ISBN 978-5-8199-0800-6. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/1908342>
7. Вепрев, С. Б. Введение в Django : краткий учебный курс / С. Б. Вепрев, В. В. Крылов. - Москва : ИНТУИТ, 2016. - 261 с. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2138301>
8. Гультяев, А. К. Проектирование и дизайн пользовательского интерфейса / А. К. Гультяев, В. А. Машин. – СПб. : Корона-Принт, 2018. 350 с.
9. Чистов, Д. В. Проектирование информационных систем. Учебник и практикум / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук. — М.: Юрайт, 2016. — 260 с.

10. Основы работы с HTML : краткий курс / . - Москва : ИНТУИТ, 2016. - 164 с. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2152338>

11. Маркин, А. В. Web-программирование : учебник / А. В. Маркин. — Москва : Ай Пи Ар Медиа, 2021. — 286 с. — ISBN 978-5-4497-1002-4. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/104883.html>

12. Буренин, С. Н. Web-программирование и базы данных : учебный практикум / С. Н. Буренин. — Москва : Московский гуманитарный университет, 2014. — 120 с. — ISBN 978-5-906768-17-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/39683.html>

13. Маркин, А. В. Web-программирование : учебник / А. В. Маркин. — Москва : Ай Пи Ар Медиа, 2021. — 286 с. — ISBN 978-5-4497-1002-4. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/104883.html>

14. Лысенкова, С. Н. Распределенные базы данных. Основы языка SQL : методическое пособие для студентов направления подготовки 09.04.03. Прикладная информатика / С. Н. Лысенкова. — Брянск : Брянский государственный аграрный университет, 2022. — 36 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/138511.html>

15. Леоненков, А. В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose : учебное пособие / А. В. Леоненков. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 317 с. — ISBN 978-5-4497-0667-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/97554.html>

Литература на иностранном языке

16. Welling, L., Thomson, L. PHP and MySQL Web Development. Addison-Wesley Professional, 2008. 1008 p.
17. Robbins, J. N. Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics. O'Reilly Media, 2012. 624 p.
18. Owens, M. The Definitive Guide to SQLite. Apress, 2010. 544 p.
19. Greenfeld, D., RoyGreenfeld, A. Two Scoops of Django: Best Practices for Django. Two Scoops Press, 2020. 529 p.
20. Bennett, A., Elwood, M. Django for Beginners: Build Websites with Python and Django. William S. Vincent, 2018. 292 p.