Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

09.03.01 Информатика и вычислительная техника

код и наименование направления подготовки

**ОТЧЕТ**

по преддипломной практике

на кафедре прикладной математики и кибернетики   
института информатики и вычислительной техники

Выполнил:

студент гр. ИП-017 \_\_\_\_\_\_\_\_\_\_\_\_\_ /Санфиров К.В./

«26» мая 2024 г.

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

оценка

Руководитель практики от университета

Старший преподаватель каф. ПМиК \_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Ефимов А.В./

«26» мая 2024 г.

Новосибирск 2024

**План-график проведения производственной практики**

Вид практики

Тип практики: преддипломная практика

Способ проведения практики: стационарная

Форма проведения практики: дискретно по периодам проведения практик

Выдано обучающемуся Санфирову Константину Витальевичу

Фамилия Имя Отчество студента

Направление: 09.03.01 – Информатика и вычислительная техника

Код – Наименование направления

Профиль: Программное обеспечение средств вычислительной техники и автоматизированных систем

Курс 4, группа ИП-017

Тема ВКР: Разработка веб-приложения с программным интерфейсом взаимодействия с базой данных медиаконтента.

Содержание практики

|  |  |
| --- | --- |
| Наименование видов деятельности | Дата (начало – окончание) |
| *Постановка задачи на практику, определение конкретной индивидуальной темы, формирование плана работ* | 22.04.2024 – 25.04.2024 |
| *Работа с библиотечными фондами, сбор и анализ материалов по теме практики* | 26.04.2024 – 29.04.2024 |
| *Выполнение работ в соответствии с составленным планом:*  *1. Анализ предметной области*  *2. Изучение аналогичных продуктов*  *3. Анализ и выбор инструментов разработки* | 30.04.2024 – 06.05.2024  07.05.2024 – 12.05.2024  13.05.2024 – 21.05.2024 |
| *Анализ полученных результатов и произведенной работы, составление отчета по практике* | 22.05.2024 – 26.05.2024 |

Согласовано:

Руководитель практики от СибГУТИ \_\_\_\_\_\_\_\_\_\_\_\_ /Ефимов А.В./

**СОДЕРЖАНИЕ**

[ЗАДАНИЕ НА ПРАКТИКУ 3](#_Toc167274256)

[ВВЕДЕНИЕ 4](#_Toc167274257)

[1. ПОСТАНОВКА ЗАДАЧИ 6](#_Toc167274258)

[1.1 Объект и предмет исследования 6](#_Toc167274259)

[1.2 Цели и задачи 6](#_Toc167274260)

[1.3 Анализ предметной области 7](#_Toc167274261)

[1.3.1 Основные термины 7](#_Toc167274262)

[1.3.2 Целевая аудитория 7](#_Toc167274263)

[1.3.3 Анализ аналогичных продуктов 8](#_Toc167274264)

[2. ИНСТРУМЕНТЫ РАЗРАБОТКИ 10](#_Toc167274265)

[2.1 Язык программирования 10](#_Toc167274266)

[2.2 СУБД 11](#_Toc167274267)

[2.3 Используемые библиотеки 13](#_Toc167274268)

[2.4 Среда разработки 14](#_Toc167274269)

[2.5 Система контроля версий 16](#_Toc167274270)

[ЗАКЛЮЧЕНИЕ 17](#_Toc167274271)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 18](#_Toc167274272)

ЗАДАНИЕ НА ПРАКТИКУ

Изучение принципов разработки веб-приложений с интерфейсом взаимодействия с базой данных медиаконтента, изучение документации Kinopoisk API. Анализ, сравнение и последующий выбор инструментов и библиотек для разработки приложения. Исследование и анализ информации о целевой аудитории.

ВВЕДЕНИЕ

В современную цифровую эпоху потребление мультимедиа находится на рекордно высоком уровне: бесчисленное количество пользователей ежедневно смотрят, просматривают и ищут медиаконтент. Веб-приложение, которое взаимодействует с базой данных медиаконтента, такой как Kinopoisk, предоставляет этим пользователям бесценный ресурс, предлагая организованную и удобную платформу для поиска и изучения фильмов, телешоу и других форм развлечений. Это приложение не только упрощает процесс поиска, но и обогащает пользовательский опыт, предоставляя подробную информацию, обзоры, рейтинги и рекомендации в одном месте.

Разработка этого веб-приложения с использованием React обеспечивает современный, отзывчивый и динамичный пользовательский интерфейс. Компонентная архитектура React позволяет эффективно управлять сложными пользовательскими интерфейсами и обеспечивает удобство работы с пользователем. Используя API Kinopoisk, приложение может получить доступ к обширной и надежной базе данных медиаконтента, предоставляя пользователям актуальную информацию о последних выпусках, популярных изданиях и исторические медиаданные.(1) Такое сочетание надежной внешней среды и обширной базы данных делает приложение ценным инструментом для энтузиастов мультимедиа.

Более того, проект служит практическим применением навыков веб-разработки, предлагая возможность углубиться в интеграцию API, управление состоянием и асинхронную обработку данных. Этот практический опыт имеет решающее значение для оттачивания технических навыков и понимания сложностей разработки реальных приложений. Кроме того, процесс исследования и анализа целевой аудитории поможет адаптировать приложение к потребностям пользователей, гарантируя, что конечный продукт будет одновременно функциональным и привлекательным.

С точки зрения бизнеса такое приложение может привлечь широкую базу пользователей, предлагая потенциальные возможности монетизации за счет рекламы, премиальных подписок и партнерства с медиадистрибьюторами. Информация, полученная в результате взаимодействия и предпочтений пользователей, также может предоставить ценные данные для дальнейшего улучшения приложения и расширения его функций. На конкурентном рынке предоставление уникальной и комплексной платформы медиаконтента может выделить приложение, стимулируя привлечение и удержание пользователей.

Таким образом, разработка этого веб-приложения удовлетворяет явные потребности в сфере цифровых медиа, использует современные технологии, совершенствует практические навыки и открывает многочисленные возможности для взаимодействия с пользователями и роста бизнеса. Это проект, который обещает принести значительную пользу как пользователям, так и разработчикам.

1. ПОСТАНОВКА ЗАДАЧИ
   1. Объект и предмет исследования

Объектом исследования в ходе выполнения данного дипломного проекта является разработка веб-приложения

Предметом исследования соответственно является взаимодействие веб-приложения и базы данных Kinopoisk. Сюда входят технические аспекты интеграции API Kinopoisk, управления поиском данных и представления медиаконтента в удобной для пользователя форме.

* 1. Цели и задачи

Целью моей дипломной работы является разработка комплексного веб-приложение, которое позволит пользователям искать, исследовать и взаимодействовать с обширной базой данных медиаконтента.

Для достижения поставленной цели, необходимо будет выполнить ряд задач, среди которых:

* Проанализировать аудиторию.
* Изучить аналогичные продукты.
* Проанализировать и сравнить доступные инструменты и библиотеки для разработки на основе React.
* Спроектировать и реализовать пользовательский интерфейс с помощью React, гарантируя, что он будет интуитивно понятным, отзывчивым и визуально привлекательным.
* Выбрать подходящую СУБД для хранения информации о пользователе и медиаконтенте, а также спроектировать и реализовать базу данных.
* Реализовать функциональные возможности для отображения подробной информации, такой как краткое содержание сюжета, сведения об актерах, и рейтинги.
* Защитить пользовательские данные и обеспечьте безопасную связь между веб-приложением и API Кинопоиска.

Выполнив данные задачи, можно достичь поставленной в текущем дипломном проекте цели.

* 1. Анализ предметной области
     1. Основные термины

Веб-приложение — это программное приложение, которое работает на веб-сервере и к которому можно получить доступ через веб-браузер через Интернет.

React — популярная библиотека JavaScript с открытым исходным кодом для создания пользовательских интерфейсов, особенно одностраничных приложений.

API Кинопоиска — это интерфейс, предоставляемый Кинопоиском, популярной российской онлайн-базой данных фильмов и телешоу.

* + 1. Целевая аудитория

Целевая аудитория веб-приложения в основном состоит из медиа-энтузиастов, которые регулярно потребляют и ищут информацию о фильмах, телешоу и другом медиаконтенте. В эту группу входят случайные зрители, которым нравится быть в курсе последних выпусков, заядлые фанаты, следящие за конкретными жанрами, актерами или режиссерами, а также критики, которые анализируют и рецензируют медиаконтент. Предоставляя подробную информацию, обзоры и персонализированные рекомендации, приложение стремится удовлетворить разнообразные потребности и предпочтения этих пользователей.

Другой значительный сегмент целевой аудитории составляют молодые люди и подростки, которые активно пользуются социальными сетями и часто делятся, и обсуждают медиаконтент со своими сверстниками. Эта группа людей ценит современный, интуитивно понятный и интерактивный пользовательский интерфейс, который приложение предоставляет благодаря использованию React. Такие функции, как профили пользователей, обмен информацией в социальных сетях и интерактивные рекомендации, предназначены для привлечения этой технически подкованной аудитории, побуждая их проводить больше времени на платформе и делиться своим опытом с друзьями.

Кроме того, приложение предназначено для профессионалов медиаиндустрии, таких как кинематографисты, продюсеры и маркетологи, которым для своей работы требуется доступ к полным и надежным данным. Благодаря интеграции API Кинопоиска приложение предлагает обширную базу данных медиаконтента, что делает его ценным инструментом для профессионалов отрасли для исследования фильмографии, анализа рыночных тенденций и сбора информации. Эта профессиональная аудитория получает выгоду от точной и актуальной информации, предоставляемой приложением, что улучшает процессы принятия решений.

Таким образом, целевая аудитория веб-приложения включает случайных зрителей, преданных фанатов, молодых людей и профессионалов. Учитывая конкретные потребности и предпочтения этих групп, приложение призвано создать комплексную и увлекательную платформу для поиска и взаимодействия с медиаконтентом.

* + 1. Анализ аналогичных продуктов

Поскольку рынок веб-приложений постоянно развивается, очевидно, что подобные продукты уже существуют. Поэтому полезно изучить эти существующие ресурсы.

В частности, основное внимание уделяется пониманию функций, предлагаемых этими платформами, и используемых ими стилей дизайна. Исследования выявили несколько подобных приложений, которые предоставляют подробную информацию о медиаконтенте, отзывы пользователей и персонализированные рекомендации. Однако определить такие всеобъемлющие платформы было непросто, поскольку не все сервисы предлагают целостный пользовательский опыт, адаптированный к разнообразной аудитории.

Проанализируем основные аналоги продуктов.

IMDb — одна из самых популярных и полных баз данных фильмов, телешоу, актеров, режиссеров и других специалистов киноиндустрии. Он предоставляет подробную информацию о фильмографии, отзывах пользователей, рейтингах, мелочах и многом другом.

Сильные стороны:

* Обширная база данных с миллионами записей.
* Удобный интерфейс с надежными возможностями поиска и фильтрации.
* Контент, созданный сообществом, с отзывами и рейтингами пользователей.
* Интеграция с IMDb Pro для профессионалов отрасли.

Недостатки:

* Для некоторых функций требуется платная подписка IMDb Pro.
* Пользовательский интерфейс может быть ошеломляющим из-за огромного количества доступной информации.

Кинопоиск — ведущая российская онлайн-база данных фильмов и сериалов, аналогичная IMDb, но ориентированная в первую очередь на российское и зарубежное кино. Он предоставляет подробную информацию, отзывы пользователей, рейтинги и рекомендации.

Сильные стороны:

* Обширная база данных российского и международного медиаконтента.
* Удобный интерфейс, адаптированный для русскоязычной аудитории.
* Активное участие сообщества с отзывами и рейтингами пользователей.
* Комплексный контент, включая фильмографии, трейлеры и новости.

Недостатки:

* Пользовательский интерфейс может быть сложным и подавляющим для новых пользователей

Поскольку в проекте используете API Кинопоиска, приложение по своей сути будет опираться на его сильные стороны. В дипломном проекте следует сосредоточиться на улучшении пользовательского опыта с помощью современного пользовательского интерфейса и дополнительных функций.

1. ИНСТРУМЕНТЫ РАЗРАБОТКИ
   1. Язык программирования

В связи с необходимостью разработки данного веб-приложения был проведен анализ и изучение доступных языков программирования, предлагающих широкие возможности для реализации задуманного проекта.

Язык программирования должен быть способен реализовать такие аспекты, как:

* Создание современного, адаптивного и интерактивного пользовательского интерфейса.
* Эффективное получение данных из API Кинопоиска и управление асинхронными операциями.
* Управление состоянием приложения, включая пользовательские данные, результаты поиска и настройки.
* Реализация плавной навигации внутри веб-приложения.
* Обеспечение высокой производительности и быстрой загрузки.

Для реализации данных аспектов JavaScript и React предоставляют мощную, гибкую и эффективную среду, которая повышает производительность разработки, обеспечивает высококачественный пользовательский опыт и обеспечивает прочную основу для создания, масштабируемого и удобного в обслуживании веб-приложения.

Из преимуществ языка JavaScript следует отметить следующие характеристики:

1. Фреймворки и библиотеки. JavaScript, особенно вместе с React, предоставляет мощные инструменты для создания динамических и отзывчивых пользовательских интерфейсов. Компонентная архитектура React позволяет повторно использовать элементы пользовательского интерфейса, что делает разработку более эффективной и удобной в обслуживании.

Виртуальный DOM: библиотеки JavaScript, такие как React, используют виртуальный DOM для оптимизации обновлений пользовательского интерфейса, обеспечивая высокую производительность и удобство работы с пользователем.

1. Асинхронные операции: встроенная поддержка асинхронного программирования в JavaScript с использованием обещаний и синтаксиса async/await упрощает обработку вызовов API и получение данных.

Fetch API и Axios. JavaScript предлагает надежные инструменты, такие как Fetch API и Axios, для выполнения HTTP-запросов, что обеспечивает плавную интеграцию с внешними API.

1. Разделение кода: React поддерживает разделение кода посредством динамического импорта, что позволяет выполнять отложенную загрузку компонентов. Это сокращает время начальной загрузки и повышает производительность.

Оптимизированный рендеринг: эффективный алгоритм сравнения React в виртуальном DOM гарантирует, что повторно визуализируются только необходимые части пользовательского интерфейса, что повышает производительность.

* 1. СУБД

Когда речь идет о хранении информации, всегда возникает вопрос, как и в каком формате ее хранить. Поэтому было решено использовать СУБД для хранения информации, которую веб-приложение будет использовать для взаимодействия с базой данных медиаконтента. В частности, необходимо хранить в базе данных детали медиаконтента и пользовательские данные.

Выбор был среди трёх самых популярных СУБД: SQLite, MySQL и PostgreSQL. Каждая из этих систем имеет свои сильные и слабые стороны, которые были выявлены в процессе оценки. В процессе принятия решения в первую очередь учитывались следующие аспекты:

* Возможность выполнения асинхронных запросов с минимальной блокировкой таблицы.
* Совместимость требуемого функционала приложения с предоставляемым СУБД.
* Наличие удобных инструментов для редактирования конфигураций и данных, хранящихся в базе данных.
* Легкость и простота развертывания СУБД на локальном или удаленном сервере.

После анализа данных СУБД были сделаны следующие выводы:

SQLite:

* Легкий вес: занимает мало места и легко встраивается в приложения.
* Автономность: установка или настройка сервера не требуется, что делает его идеальным для небольших приложений и сред разработки.
* Портативность: файлы базы данных можно легко копировать и перемещать между различными системами.
* Простота: простота установки и использования с минимальными административными затратами.

MySQL:

* Производительность: оптимизирован для операций с большим объемом чтения, что делает его быстрым и надежным для веб-приложений.
* Простота использования: простая настройка и обширная документация, а также большое сообщество поддержки.
* Совместимость: широко используется и поддерживается многими службами веб-хостинга и платформами.
* Репликация: поддерживает репликацию «главный-подчиненный» для масштабируемости и избыточности данных.

PostgreSQL:

* Расширенные функции: богатый набор функций, включая поддержку сложных запросов, полнотекстового поиска и пользовательских типов данных.
* Расширяемость: высокая расширяемость с поддержкой пользовательских функций и расширений.
* Соответствие ACID: строгое соблюдение принципов ACID, обеспечение целостности и надежности данных.
* Открытый исходный код: активное сообщество разработчиков и обширная документация, обеспечивающая надежную поддержку и постоянные улучшения.

В заключение был сделан вывод, что MySQL — отличный выбор для проекта веб-приложения благодаря своей высокой производительности, простоте использования, масштабируемости и надежным функциям безопасности. Его совместимость с различными платформами и языками делает его надежным и экономически эффективным решением.(5)

Хотя SQLite легок и прост в использовании, ему не хватает расширенных функций, таких как надежное индексирование, хранимые процедуры и комплексная поддержка транзакций, которые могут быть необходимы для сложных запросов и манипулирования данными в проекте. PostgreSQL, хотя и обладает богатым набором функций и расширяемостью, может быть более сложным в настройке и управлении, а его расширенные функции могут оказаться ненужными для требований вашего проекта. MySQL обеспечивает правильный баланс между производительностью, простотой и функциональностью, что делает его наиболее подходящим.

* 1. Используемые библиотеки

При разработке для достижения поставленной цели были использованы различные сторонние библиотеки:

* React — популярная библиотека JavaScript для создания пользовательских интерфейсов, в первую очередь для одностраничных приложений. Разработанный и поддерживаемый Facebook.
* Next.js — это платформа React, которая обеспечивает рендеринг на стороне сервера и генерацию статических сайтов.
* GraphQL — это язык запросов для API и среда выполнения для выполнения этих запросов, позволяющая клиентам запрашивать определенные данные.
* Material-UI — это популярная библиотека компонентов React, которая реализует рекомендации Google по дизайну материалов и предоставляет набор повторно используемых и настраиваемых компонентов.
* Express.js — это минимальная и гибкая платформа веб-приложений Node.js, предоставляющая надежный набор функций для веб-приложений и мобильных приложений.

Для проекта веб-приложения, включающего базу данных медиаконтента, выбор правильных технологий имеет решающее значение для создания надежной, масштабируемой и удобной для пользователя платформы. Выбранный стек должен облегчить разработку интерактивного и динамического пользовательского интерфейса, обеспечить эффективные возможности извлечения и манипулирования данными, а также обеспечить плавный и отзывчивый пользовательский интерфейс.(3)

Для достижения этой цели важно использовать инструменты, поддерживающие компонентную архитектуру, позволяющие создавать повторно используемые и поддерживаемые элементы пользовательского интерфейса. Такой подход обеспечивает единообразный внешний вид всего приложения и упрощает управление и масштабирование по мере роста проекта. Кроме того, рендеринг на стороне сервера и создание статического сайта могут значительно повысить производительность и SEO, гарантируя быструю загрузку вашего информации и его легкость обнаружения поисковыми системами.

Эффективная обработка данных является еще одним важным аспектом. Использование языка запросов, который позволяет клиентам запрашивать именно те данные, которые им нужны, сводит к минимуму передачу данных и повышает производительность. Особое внимание к целостности и согласованности данных также важно для обеспечения надежного и точного взаимодействия с базой данных медиаконтента.

Стек разработки также должен включать надежную внутреннюю поддержку с промежуточным программным обеспечением и возможностями маршрутизации, упрощающими управление HTTP-запросами и ответами. Это гарантирует, что серверная логика хорошо организована и способна выполнять различные операции, необходимые для управления информацией и взаимодействием с пользователем.(2)

Наконец, включение полного набора настраиваемых компонентов пользовательского интерфейса, основанных на современных принципах дизайна, гарантирует, что приложение будет визуально привлекательным и обеспечит удобство работы с пользователем на разных устройствах.

* 1. Среда разработки

При выборе среды разработки мой выбор сразу упал на Visual Studio Code, так как у меня имеется большой опыт работы с ним. Visual Studio Code (VS Code) — это отличная интегрированная среда разработки для работы со стеком технологий благодаря своим обширным функциям и поддержке веб-разработки.(4) Вот несколько причин, по которым VS Code особенно хорошо подходит для проекта:

1. Расширяемость и плагины

Широкий спектр расширений: VS Code имеет обширный рынок расширений, расширяющих его функциональность, в том числе для React, Next.js, GraphQL, Material-UI и Express.js. Эти расширения предоставляют подсветку синтаксиса, фрагменты кода и инструменты интеграции, адаптированные к стеку.

Настройка: возможность адаптировать IDE в соответствии с вашими конкретными потребностями и рабочим процессом, повышая производительность и эффективность.

2. Встроенная интеграция с Git.

Контроль версий: VS Code имеет встроенную поддержку Git, что позволяет легко управлять контролем версий и сотрудничать с членами команды. Возможность выполнять стандартные операции Git непосредственно в среде IDE, например, фиксации, ветвление и слияние.

Визуальный интерфейс: помогает понять изменения и историю проекта, упрощая управление контролем версий.

3. Мощные функции редактирования кода.

IntelliSense: обеспечивает интеллектуальное завершение кода, информацию о параметрах, краткую информацию и списки участников, что особенно полезно для разработки JavaScript и TypeScript.

Навигация по коду: такие функции, как «Перейти к определению», «Найти все ссылки» и «Просмотреть определение», упрощают навигацию по базе кода, понимание зависимостей и улучшение рабочего процесса.

Инструменты рефакторинга: включает встроенную поддержку рефакторинга, упрощающую поддержку и улучшение вашей кодовой базы с течением времени.

4. Интегрированный терминал

Встроенный терминал: позволяет запускать команды оболочки непосредственно в среде IDE, что позволяет легко запускать сценарии сборки, запускать сервер разработки или взаимодействовать с базой данных без переключения контекстов.

Несколько терминалов: возможность открыть несколько терминалов на разных вкладках, что полезно для одновременного запуска таких процессов, как сервер разработки React и внутренний сервер Express.

5. Возможности отладки

Углубленная отладка: VS Code предоставляет мощные инструменты отладки для JavaScript, Node.js и клиентских приложений. Возможность устанавливать точки останова, проверять переменные и поэтапно выполнять код для диагностики и устранения проблем.

Гибкость конфигурации: легко настраивайте параметры отладки для различных сред и платформ, что обеспечивает беспрепятственную отладку всего стека.

6. Производительность и скорость

Легкость: несмотря на свои надежные функции, VS Code легок и хорошо работает даже на менее мощных машинах, обеспечивая плавность разработки.

7. Сообщество и поддержка

Активное сообщество: большое и активное сообщество означает частые обновления, множество учебных пособий и множество ресурсов поддержки. Можно легко найти решения распространенных проблем или узнать новые советы и рекомендации сообщества.

Документация: Полная и хорошо поддерживаемая документация гарантирует максимальную эффективность использования возможностей VS Code.

Используя эти функции, Visual Studio Code повышает производительность, упрощает процесс разработки и обеспечивает надежную среду для работы с выбранным стеком технологий, что делает его хорошим выбором для проекта веб-приложения.

* 1. Система контроля версий

Очевидным выбором системы контроля версий является Git и GitHub. GitHub выделяется как ведущая платформа для совместного контроля версий, предлагая централизованный репозиторий для хранения, управления и совместной работы над исходным кодом вашего проекта. Он упрощает рабочие процессы благодаря надежным функциям, таким как ветвление, слияние и запросы на извлечение, которые имеют решающее значение для параллельной разработки и поддержания качества кода.

Интеграция с Visual Studio Code (VS Code) еще больше расширяет возможности GitHub, обеспечивая удобство разработки непосредственно в IDE. Разработчики могут клонировать репозитории, создавать ветки и управлять запросами на включение, не выходя из VS Code, благодаря встроенной поддержке Git и интеграции с GitHub. Эта интеграция обеспечивает совместную работу в режиме реального времени и эффективную проверку кода, поскольку разработчики могут просматривать, комментировать и утверждать запросы на включение непосредственно в редакторе.

Исходя из этих преимуществ, выбор был сделан в пользу системы контроля версий Git, который лежит в основе сервиса GitHub. Основные преимущества включают удобство использования и широкую поддержку IDE с интеграцией GitHub. Это позволяет создавать и настраивать новые репозитории, фиксировать и отправлять изменения в несколько кликов, все это без необходимости покидать среду разработки.

ЗАКЛЮЧЕНИЕ

Таким образом, в результате прохождения преддипломной практики, были проанализированы потребности целевой аудитории и аналогичные продукты, сформированы план разработки и функционал продукта.

Был произведен анализ, сравнение и выбор доступных инструментов разработки, с помощью которых будет спроектирована база данных и будет реализовываться программная часть приложения.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Документация по API кинопоиска. [Электронный ресурс] // Документация для неофициального API кинопоиска (kinopoisk.dev). URL : https://api.k inopoisk.dev/documentation/ (дата обращения: 04.05.2024).
2. Документация по модулю Next.js [Электронный ресурс] // The React Framework for the Web. URL : <https://nextjs.org/docs/> (дата обращения: 06.05.2024).
3. UI-библиотеки для React [Электронный ресурс] // Хабр. URL : <https://habr.com/ru/articles/731964/> (дата обращения: 06.05.2024).
4. Топ-11 лучших IDE и редакторов кода JavaScript [Электронный ресурс] // Обучающая платформа онлайн-курсов GeekBrains. URL : <https://blog.geekbrains.by/top-11-luchshih-ide-i-redaktorov-koda-javascript/> (дата обращения: 15.05.2024).
5. Как выбрать СУБД для нового проекта [Электронный ресурс] // IT-World: Мир цифровых и информационных технологий. URL : <https://www.it-world.ru/tech/choice/188005.html/> (дата обращения: 19.05.2024).