Минобрнауки России

Юго-Западный государственный университет

Кафедра программной инженерии

ОТЧЕТ

о преддипломной практике

наименование вида и типа практики

на (в) ООО ”Тахион”

наименование предприятия, организации, учреждения

Студента 4 курса, группы ПО-01б

курса, группы

Сентищева Максима Владимировича

фамилия, имя, отчество

Руководитель практики от предприятия, организации, учреждения

Оценка _____

_____

должность, звание, степень

_____ _____

фамилия и. о. подпись, дата

Руководитель практики от университета

Оценка _____

_____

должность, звание, степень

_____ _____

фамилия и. о. подпись, дата

Члены комиссии

_____ _____

подпись, дата фамилия и. о.

_____ _____

подпись, дата фамилия и. о.

Курск, 2024 г.

СОДЕРЖАНИЕ

1 Анализ предметной области

1.1 Описание предметной области

В современном мире визуальный контент, такой как фотографии и картинки, становится все более значимым и доступным благодаря развитию интернета, социальных сетей и мобильных устройств. Однако, с каждым годом количество создаваемых фотографий продолжает стремительно расти. Социальные сети и платформы для обмена медиафайлами, такие как VK, Одноклассники и Instagram, стимулируют пользователей к ежедневному созданию и публикации новых изображений, делая фотографию неотъемлемой частью повседневной коммуникации и самовыражения. По оценкам исследований, ежегодно производится несколько триллионов новых фотографий, и этот объем продолжает увеличиваться, что подчеркивает важность развития более эффективных инструментов для управления, хранения и поиска изображений в этом бесконечно растущем массиве визуальной информации. Традиционные методы, такие как простые текстовые запросы и ручной поиск фотографий, не всегда способны полноценно описать то, что человек хочет найти или увидеть за разумное время. В таких случаях важную роль играет контекст, который может включать в себя время, местоположение, социальные связи, а также предпочтения и интересы пользователей.

Контекстный поиск и рекомендация изображений представляют собой подходы, направленные на улучшение процесса поиска и выбора изображений путем учета контекстуальных факторов. Вместо того чтобы рассматривать изображения изолированно, эти системы анализируют контекст, в котором они используются или запрашиваются, и адаптируют результаты под конкретные потребности пользователя.

Для достижения этой цели в области контекстного поиска и рекомендации изображений активно применяются технологии машинного обучения. Методы классификации изображений, извлечения признаков, а также алгоритмы коллаборативной и контентной фильтрации используются для анали-

за и обработки визуального контента, а также для адаптации результатов под контекст запроса или использования.

1.1.1 Исторический контекст и эволюция визуального контента

Исторически визуальный контент всегда играл важную роль в передаче информации и культуры. От наскальных рисунков древних людей до современных цифровых фотографий, изображения служили средством сохранения и передачи знаний, эмоций и художественных идей. С развитием технологий способы создания и распространения изображений значительно изменились.

Доцифровой период

В доцифровую эпоху основными средствами создания визуального контента были живопись, скульптура и графика. Наскальные рисунки, созданные древними людьми, считаются одними из первых форм визуального искусства. Эти изображения позволяли передавать информацию о повседневной жизни, религиозных ритуалах и охоте. В античности и средневековье визуальный контент также играл важную роль в культуре и религии, служа средством записи исторических событий, мифов и легенд.

XIX век: изобретение фотографии

В XIX веке изобретение фотографии революционизировало способ документирования событий и людей. Первая фотография была создана в 1826 году Жозефом Нисефором Ньепсом. Со временем технология совершенствовалась, и фотографии стали важным инструментом в журналистике, науке и искусстве. Фотографии использовались для фиксации исторических событий, научных исследований и создания художественных произведений. Они позволили запечатлеть моменты, которые ранее можно было только описать словами.

XX век: развитие кино и телевидения

В XX веке развитие кино и телевидения добавило новые измерения к визуальному контенту, позволяя передавать движение и звук. Киноиндустрия начала развиваться с изобретением кинематографа братьями Люмьер

в 1895 году. Кинофильмы стали популярным средством развлечения и культурного влияния, а также мощным инструментом для передачи историй и идей. Телевидение, появившееся в 1930-х годах, предоставило возможность массового распространения визуального контента, сделав его доступным для широкой аудитории.

Конец XX века: цифровая революция

С появлением цифровых технологий в конце XX века создание и распространение визуального контента стало еще более доступным. Цифровые камеры, а затем и камеры в мобильных устройствах, сделали фотографию повседневной практикой для миллионов людей. Этот период ознаменовался быстрым развитием технологий, таких как компьютерная графика и цифровое редактирование изображений, что значительно расширило возможности создания и обработки визуального контента.

XXI век: эра социальных сетей и онлайн-платформ

В XXI веке социальные сети и онлайн-платформы предоставили новые возможности для обмена и публикации изображений, делая их глобально доступными в считанные секунды. Платформы, такие как Instagram, Facebook, и Pinterest, позволяют пользователям делиться своими фотографиями и видео с широкой аудиторией, получать обратную связь и взаимодействовать с другими пользователями. Это привело к созданию огромных массивов визуального контента, который требует эффективных инструментов для управления, поиска и анализа.

Будущее визуального контента

С развитием технологий, таких как искусственный интеллект, машинное обучение и дополненная реальность, будущее визуального контента обещает быть еще более захватывающим. Эти технологии открывают новые возможности для создания, анализа и взаимодействия с визуальным контентом. Например, искусственный интеллект уже используется для автоматической классификации и поиска изображений, а дополненная реальность позволяет создавать новые формы визуального опыта.

1.2 Технологии машинного обучения в контекстном поиске и рекомендации изображений

В последние годы, поиск и рекомендация изображений, значительно продвинулись благодаря использованию технологий машинного обучения. Эти технологии позволяют системам не только распознавать и классифицировать визуальный контент на заранее указанные группы, но и адаптироваться к предпочтениям и контексту пользователя. Основная задача таких систем заключается в обеспечении более глубокого понимания содержания изображений, что позволяет осуществлять более точные и персонализированные рекомендации. Эффективность этих систем зависит от выбора подходящих алгоритмов и архитектур нейронных сетей, которые могут адаптироваться и реагировать на различные контекстные сигналы. На данный момент ключевыми являются три следующие архитектуры:

- Convolutional Neural Networks (CNN) - является одной из основных архитектур для анализа изображений. Нейронные сети на данной архитектуре способны выделять ключевые признаки изображений на разных уровнях абстракции, что делает их эффективными для задач классификации, детекции и сегментации объектов. CNN успешно используются для извлечения признаков изображений и создания их векторных представлений.

- Vision Transformer (ViT) - другая важная архитектура нейронных сетей,, которая была представлена в 2017 году и адаптирована для обработки визуальных данных в 2020 году. Особенностью ViT является способ обработки данных, который заключается в разбивании изображения на патчи и анализа их с помощью трансформерных блоков, создавая векторные представления изображений.

- Contrastive Language-Image Pre-training (CLIP) - одна из современных архитектур для анализа изображений, разработанная компанией OpenAI в 2021 году. Она подразумевает обучение визуальных данных под контролем нейронной сети естественного языка. Данная архитектура устраняет разрыв между текстовыми и визуальными данными путем совместного обучения мо-

дели на крупномасштабном наборе данных, содержащем изображения и соответствующие им текстовые описания.

1.3 Существующие решения

В области поиска и рекомендации изображений существуют как заграничные, так и российские платформы, предоставляющие возможность поиска изображений, расположенных как на самой платформе, так и в интернете в целом, среди которых выделяются такие крупные компании, как Pinterest, Google и Yandex. Эти платформы демонстрируют различные подходы к индексации, поиску и визуализации визуального контента.

Pinterest - это социальная сеть, основанная на принципах визуального поиска и коллекционирования изображений. Платформа позволяет пользователям создавать и управлять тематическими коллекциями изображений, таких как интерьеры, мода, питание и многое другое. Одной из ключевых особенностей Pinterest является использование современных алгоритмов машинного обучения для рекомендации изображений, основанных на предпочтениях пользователя и визуальном сходстве. Это достигается благодаря разработке собственных технологий по анализу изображений и их контексту, что позволяет предлагать пользователю наиболее релевантные и интересные изображения, расположенные на платформе.

Google Картинки представляет собой мощный поисковик изображений, который позволяет пользователям находить визуальный контент по всему интернету. Система использует сложные алгоритмы индексации и ранжирования, которые включают анализ текстовой информации, связанной с изображением, и контекста страницы, на которой это изображение размещено.

Yandex Картинки также как и Google Картинки, позволяет пользователям находить визуальный контент по всему интернету и фильтровать его на основе заданных параметров.

Однако, несмотря на наличие российских платформ, таких как Yandex, на текущий момент отсутствуют решения, которые бы позволяли рекомендовать изображения исключительно из собственной базы данных, не выходя за рамки платформы, подобно тому, как это делает Pinterest. В дополнение к потребностям общего пользователя, такая система могла бы найти применение в корпоративной сфере, облегчая хранение и поиск рабочих файлов в рамках группы или организации. Это становится особенно важным в контексте увеличения объемов цифровых данных и необходимости их структурирования для удобства доступа и использования. Внедрение системы, ориентированной на работу с внутренней базой данных изображений, может значительно повысить эффективность работы команд, ускоряя процессы поиска необходимых материалов и обеспечивая более тесное взаимодействие в рамках коллективных проектов.

2 Техническое задание

2.1 Основание для разработки

Основанием для разработки веб-сервиса для контекстного поиска и рекомендации изображений на основе машинного обучения является задание на выпускную квалификационную работу приказ ректора ЮЗГУ от « » 2024 года № 0000-0 «Об утверждении тем выпускных квалификационных работ и руководителей выпускных квалификационных работ».

2.2 Назначение разработки

Функциональное назначение разрабатываемого веб-сервиса заключается в предоставлении пользователям мощного инструмента для поиска и загрузни новых изображений. Платформа предназначена для фотографов, дизайнеров, маркетологов, исследователей и компаниям, которым необходим эффективный способ доступа и организации визуального контента.

Предполагается, что данным веб-сервисом будут пользоваться как профессионалы для управления своими проектами и архивами изображений, так и обычные пользователи, заинтересованные в поиске визуальной информации по введенному запросу.

Задачами разработки данным веб-сервисом являются:

- создание обширной и легко доступной базы изображений;
- разработка алгоритмов контекстного поиска, позволяющих анализировать и находить изображения по визуальным и текстовым запросам;
- разработка функции добавления новых фотографий в базу изображений;
- обучение нейронной сети для контекстного поиска.

Кроме того, платформа должна выполнять следующие функции:

1. Интеграция с социальными сетями и другими платформами: Важно обеспечить возможность интеграции с популярными социальными сетями

и другими онлайн-платформами, чтобы пользователи могли легко делиться найденными изображениями или использовать их в своих проектах.

2. Поддержка различных форматов изображений: Веб-сервис должен поддерживать широкий спектр форматов изображений, обеспечивая их корректное отображение и обработку. Это повысит гибкость использования сервиса для различных нужд пользователей.

3. Безопасность и конфиденциальность данных: Необходимо гарантировать безопасность и конфиденциальность данных пользователей. Это включает защиту личной информации и обеспечивание надежного хранения загруженных изображений.

Таким образом, данный веб-сервис предоставит пользователям всесторонние возможности для работы с изображениями, делая процесс поиска, управления и использования визуального контента более удобным и результативным.

2.3 Требования к веб-сервису

2.3.1 Требования к данным веб-сервиса

Входными данными для веб-сервиса являются фотографии со следующими расширениями:

- *.jpg;
- *.png;
- *.webp.

Выходными данными для веб-сервиса являются отображаемые фотографии по запросу или рекомендации с возможностью их быстрого скачивания.

На рисунке 2.1 представлены концептуальные классы веб-сервиса.

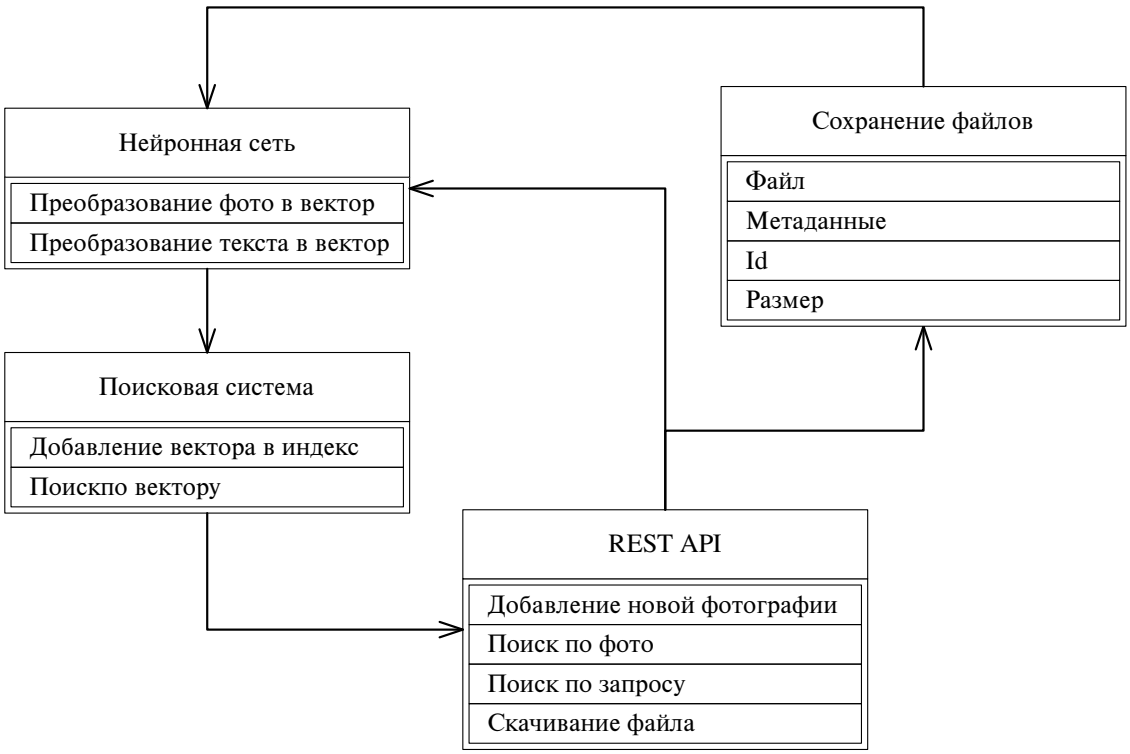

Рисунок 2.1 – Концептуальные классы веб-сервиса

2.3.2 Функциональные требования к веб-сервису

На основании анализа предметной области в разрабатываемом веб-сервисе должны быть реализованы следующие функции:

- поиск фотографий по запросу;
- добавление новой фотографии;
- просмотр фотографии;
- скачивание выбранного фото;
- поиск фото по фотографии;
- выдача похожих фотографий.

На рисунке 2.2 представлены функциональные требования к системе в виде диаграммы прецедентов нотации UML.

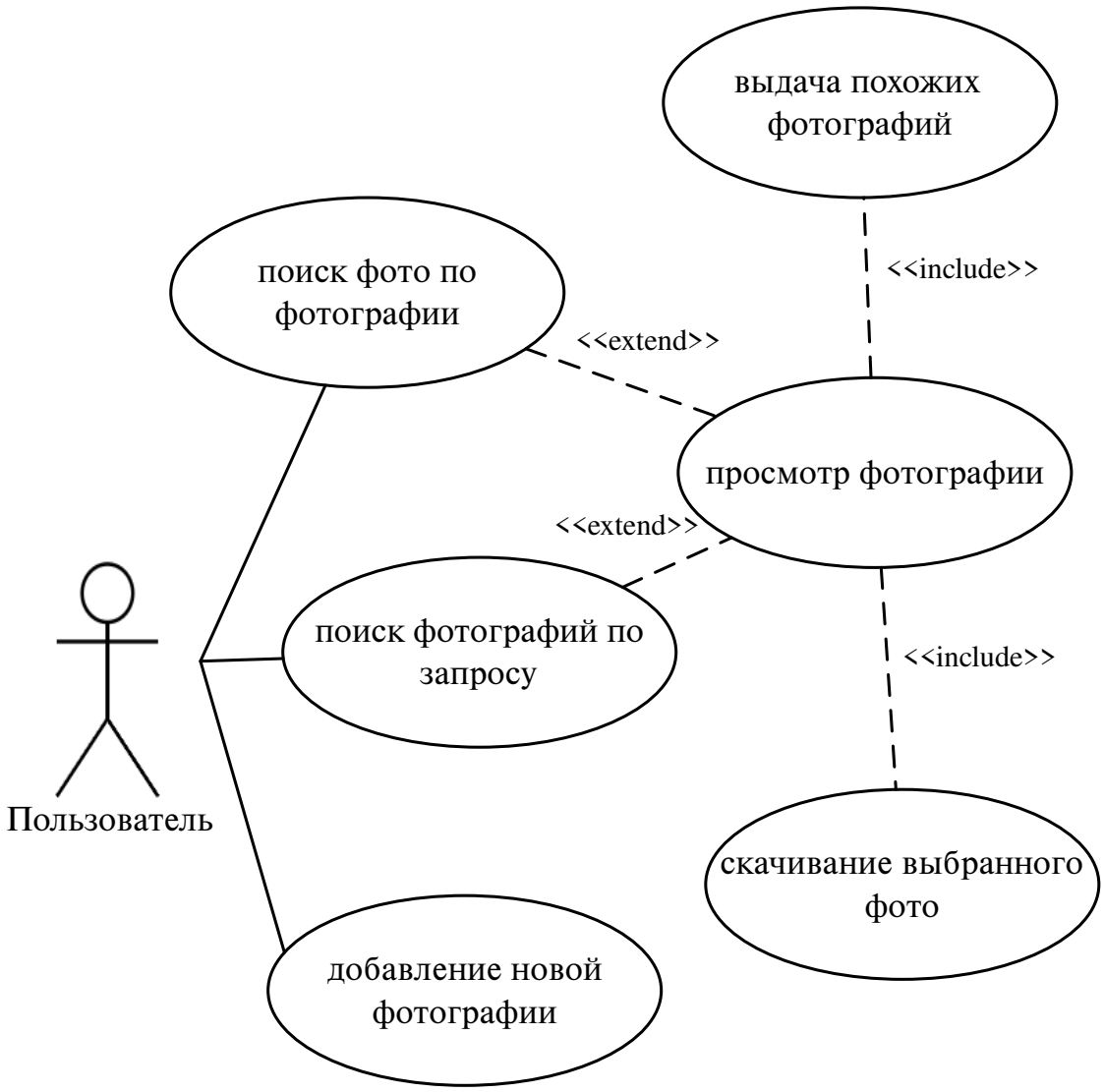

Рисунок 2.2 – Диаграмма прецедентов

2.3.2.1 Вариант использования «Поиск фотографий по запросу»

Заинтересованные лица и их требования: Пользователь веб-сервиса, который хочет выполнить поиск фотографии по текстовому запросу.

Предусловие: Пользователь загружает главную страницу веб-сервиса.

Постусловие: Пользователь просматривает предложенные по запросу фотографии.

Основной успешный сценарий:

1. Пользователь заходит на главную страницу.
2. Пользователь вводит запрос в поле поиска.
3. Пользователь нажимает ”Enter”или кнопку поиска.

4. Система отображает фотографии которые больше всего подходят для данного запроса.

2.3.2.2 Вариант использования «Добавление новой фотографии»

Заинтересованные лица и их требования: Пользователь веб-сервиса, который хочет загрузить новую фотографию в базу фотографий.

Предусловие: Пользователь загружает главную страницу сайта.

Постусловие: Пользователь загрузил новую фотографию в веб-сервис.

Основной успешный сценарий:

1. Пользователь заходит на главную страницу.
2. Пользователь выбирает кнопку добавления новой фотографии.
3. Пользователь загружает новую фотографию.
4. Система загружает обработанную фотографию и отображает информацию на сайте.

2.3.2.3 Вариант использования «Просмотр фотографии»

Заинтересованные лица и их требования: Пользователь веб-сервиса, который хочет просмотреть предложенную фотографию из списка.

Предусловие: Пользователь выполнил поиск фотографии.

Постусловие: Пользователь открывает фотографию для просмотра.

Основной успешный сценарий:

1. Пользователь выполнил поиск фотографии.
2. Пользователь находит понравившуюся фотографию из предложенного списка.
3. Пользователь нажимает на понравившуюся фотографию.
4. Выбранная фотография открывает на полный экран.

2.3.2.4 Вариант использования «Скачивание выбранного фото»

Заинтересованные лица и их требования: Пользователь веб-сервиса, который хочет скачать найденную фотографию.

Предусловие: Пользователь выполнил поиск фотографии.

Постусловие: Пользователь скачивает фотографию на свой компьютер.

Основной успешный сценарий:

1. Пользователь выбирает требуемую фотографию.
2. Пользователь нажимает кнопку ”Сохранить”.
3. Фотография загружается на компьютер пользователя.

2.3.2.5 Вариант использования «Поиск фото по фотографии»

Заинтересованные лица и их требования: Пользователь веб-сервиса, который хочет выполнить поиск фотографии по своей фотографии.

Предусловие: Пользователь загружает главную страницу веб-сервиса.

Постусловие: Пользователь просматривает предложенные по запросу фотографии.

Основной успешный сценарий:

1. Пользователь заходит на главную страницу.
2. Пользователь нажимает кнопку камеры в поле поиска.
3. Пользователь загружает свою фотографию.
4. Пользователь нажимает ”Enter”или кнопку поиска.
5. Система отображает фотографии которые больше всего подходят для данной фотографии.

2.3.2.6 Вариант использования «Выдача похожих фотографий»

Заинтересованные лица и их требования: Пользователь веб-сервиса, хочет просмотреть похожие фотографии к выбранному фото.

Предусловие: Пользователь выполнил поиск по запросу или по фотографии.

Постусловие: Пользователь просматривает похожие фотографии.

Основной успешный сценарий:

1. Пользователь выбирает интересующую фотографию.
2. Пользователь нажимает кнопку ”Похожие”.
3. Система отображает похожие фото для выбранной фотографии.

2.3.3 Требования пользователя к интерфейсу веб-сервиса

В веб-сервисе должны присутствовать следующие графические интерфейсы взаимодействия с пользователем:

1. Страница с полем для поиска фотографий.
2. Окно для загрузки файлов.
3. Окно для отображения фотографии.

Все страницы должны иметь адаптивную верстку для взаимодействия с платформой с разных устройств. Каждая страница должна соответствовать основным принципам UI/UX дизайна, что включает в себя:

1. Простота и понятность: Интерфейс должен быть интуитивно понятным, чтобы пользователи могли легко ориентироваться и находить нужные функции.

2. Согласованность: Все элементы интерфейса должны быть согласованы по стилю, цветовой гамме и расположению, что способствует созданию единого визуального образа веб-сервиса.

3. Эффективность: Время, затрачиваемое пользователем на выполнение задач, должно быть минимизировано. Это можно достичь за счет удобного размещения элементов управления и быстрой реакции системы на действия пользователя.

4. Доступность: Интерфейс должен быть доступен для всех пользователей, включая людей с ограниченными возможностями.

5. Адаптивность: Все страницы должны корректно отображаться на различных устройствах и разрешениях экранов. Это включает в себя использование гибких макетов и адаптивной верстки, которые автоматически подстраиваются под размеры экрана.

Кроме того, должны быть предусмотрены уведомления и подсказки для улучшения пользовательского опыта. Система должна предоставлять пользователям своевременные уведомления и подсказки для упрощения работы и информирования о важных событиях.

2.4 Нефункциональные требования к программной системе

2.4.1 Требования к архитектуре

Веб-сервис должен быть выполнен в клиент-серверной архитектуре с возможностью докеризации для обеспечения гибкости, масштабируемости и легкого запуска.

2.4.2 Требования к надежности

При использовании веб-сервиса должна быть обеспечена стабильная работа серверов, а также регулярное создание резервных копий данных. Все возникающие во время работы приложения ошибки должны быть корректно обработаны. Для удобства пользователей ошибки должны выводиться в виде всплывающих окон на сайте, предоставляя пользователю краткую информацию о проблеме.

2.4.3 Требования к программному обеспечению

Для реализации клиентской части веб-сервис должен быть использован язык JavaScript, библиотека компонентов Material UI и фреймворк React.

Для реализации серверной части веб-сервиса должен использоваться язык программирования Python.

2.4.4 Требования к аппаратному обеспечению

Для открытия веб-сервиса потребуется устройство с доступом в сеть интернет, а так же любой из доступных браузеров с поддержкой выполнения JavaScript.

Для работы серверной части веб-сервиса необходим сервер на операционной системе Linux с установленным Docker. Так же дисковое пространство не менее 2 Гб, свободная оперативная память в размере не менее 4 Гб, видеокарта с не менее 512 Мб видеопамяти.

2.4.5 Требования к оформлению документации

Разработка программной документации и программного изделия должна производиться согласно ГОСТ 19.102-77 и ГОСТ 34.601-90. Единая система программной документации.

3 Технический проект

3.1 Общие сведения о программной системе

Необходимо спроектировать и разработать веб-сервис для контекстного поиска и рекомендации изображений на основе машинного обучения.

Разрабатываемая программная система предназначена для предоставления мощного инструмента для фотографов, дизайнеров, маркетологов, исследователей и компаний, которым необходим эффективный способ доступа и организации визуального контента. Платформа предоставит пользователям возможность загружать, искать и скачивать изображения, а также получать похожие фотографии на основе контекстного поиска.

Основной принцип работы системы заключается в комплексной обработке загруженных изображений для извлечения их метаданных и визуальных характеристик, что позволяет пользователям легко находить нужные изображения по различным критериям. Система будет использовать машинное обучение для улучшения процесса поиска и предоставления персонализированных рекомендаций.

3.2 Проектирование архитектуры программной системы

3.2.1 Выбор архитектурного стиля и паттернов проектирования

Для разработки веб-сервиса для контекстного поиска и рекомендации изображений на основе машинного обучения была выбрана клиент-серверная архитектура. Этот подход позволяет разделить систему на две основные части: клиентскую (frontend) и серверную (backend), что обеспечивает гибкость в разработке, развертывании и масштабировании компонентов.

Для взаимодействия между клиентом и сервером будет использоваться REST API (Representational State Transfer Application Programming Interface). REST API обеспечивает стандартизированный и легковесный способ обмена данными между различными компонентами системы. Основные принципы REST API включают:

- **Использование HTTP-методов:** REST API использует стандартные HTTP-методы (GET, POST, PUT, DELETE) для выполнения операций над ресурсами.

- **Структурирование URI:** В REST API ресурсы идентифицируются с помощью унифицированных URI (Uniform Resource Identifier). Это позволяет однозначно определять ресурсы и выполнять над ними операции.

- **Безопасность данных:** Все данные, передаваемые между клиентом и сервером, будут защищены с использованием HTTPS. HTTPS обеспечивает шифрование данных, что гарантирует конфиденциальность и целостность передаваемой информации, а также защиту от атак.

- **Формат данных:** REST API поддерживает обмен данными в различных форматах, наиболее распространенным из которых является JSON (JavaScript Object Notation). JSON обеспечивает легкость чтения и обработки данных как на клиентской, так и на серверной стороне.

- **Статусные коды:** В ответах API используются стандартные HTTP-статусные коды для информирования клиентов о результате их запросов.

Все данные, передаваемые между клиентом и сервером, будут защищены с использованием HTTPS. Это обеспечивает конфиденциальность, целостность передачи данных и защиту от атак. HTTPS гарантирует, что информация, передаваемая между пользователями и системой, будет защищена от перехвата и подделки.

Для интеграции различных компонентов системы и обеспечения совместимости будет использоваться паттерн проектирования ”Адаптер”. Этот паттерн позволяет создавать интерфейсы для взаимодействия несовместимых компонентов, обеспечивая гибкость и возможность масштабирования системы. Паттерн ”Адаптер”будет применяться для взаимодействия с базами данных и библиотеками.

Архитектура всей системы представлена на рисунке 3.1.

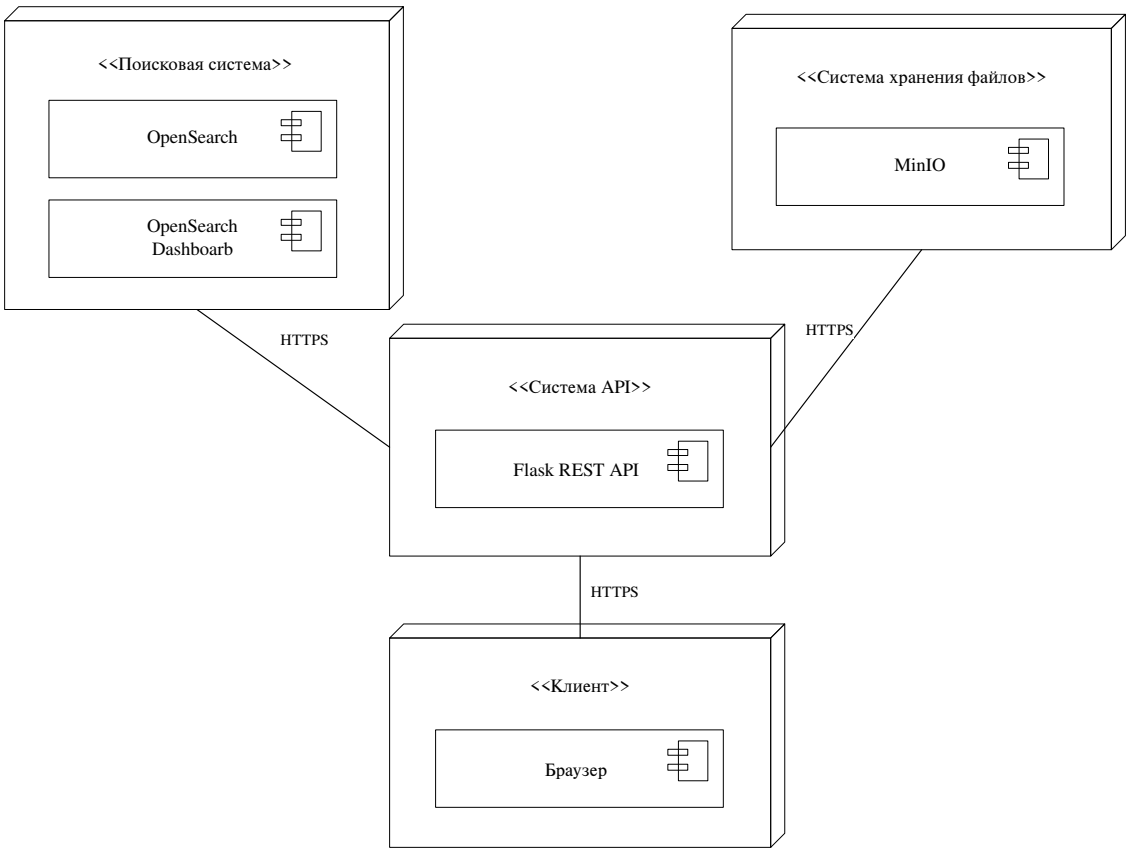

Рисунок 3.1 – Архитектура системы

3.2.2 Используемые базы данных

Для обеспечения хранения и управления данными в веб-сервисе для контекстного поиска и рекомендации изображений на основе машинного обучения будут использоваться следующие базы данных: MinIO и OpenSearch. Эти системы обеспечат высокую производительность, масштабируемость и надежность работы сервиса.

3.2.2.1 MinIO

MinIO — это высокопроизводительная объектная система хранения, совместимая с Amazon S3. MinIO предоставляет гибкий и масштабируемый способ хранения больших объемов данных, таких как изображения, видео и другие файлы. Основные особенности использования MinIO в данном проекте включают:

- Объектное хранение данных: MinIO позволяет хранить данные в виде объектов, каждый из которых состоит из самого файла, метаданных и уни-

кального идентификатора. Это упрощает управление большими объемами неструктурированных данных.

- Высокая производительность: MinIO оптимизирован для высокой производительности и может обрабатывать большие объемы данных с минимальными задержками. Это особенно важно для приложений, работающих с мультимедийным контентом.

- Масштабируемость: MinIO поддерживает горизонтальное масштабирование, что позволяет увеличивать емкость и производительность системы по мере роста объема данных. Это достигается за счет добавления новых узлов в кластер MinIO.

- Безопасность: MinIO обеспечивает высокий уровень безопасности, включая поддержку шифрования данных как в процессе передачи, так и в состоянии покоя. Это помогает защитить данные пользователей от несанкционированного доступа и утечек.

3.2.2.2 OpenSearch

OpenSearch — это распределенная поисковая и аналитическая система, основанная на Elasticsearch распространяемая под лицензией Apache-2.0 и являющаяся поисковой системой RESTful с открытым исходным кодом. OpenSearch используется для индексирования и поиска данных, предоставляя мощные возможности для работы с текстовыми и структурированными данными. Основные особенности использования OpenSearch в данном проекте включают:

- Индексирование данных: OpenSearch позволяет индексировать большие объемы данных, обеспечивая быстрый и эффективный поиск по различным критериям. Это особенно важно для задач контекстного поиска и рекомендаций изображений.

- Полнотекстовый поиск: OpenSearch предоставляет мощные возможности для полнотекстового поиска, включая поддержку сложных запросов,

ранжирование результатов и анализ текста. Это позволяет улучшить точность и релевантность результатов поиска.

- Аналитика и визуализация: OpenSearch включает встроенные инструменты для аналитики и визуализации данных, такие как OpenSearch Dashboards. Это позволяет создавать интерактивные отчеты и дашборды для анализа данных и принятия обоснованных решений.

- Распределенная архитектура: OpenSearch поддерживает горизонтальное масштабирование, что позволяет обрабатывать большие объемы данных и обеспечивать высокую производительность системы. Распределенная архитектура также повышает отказоустойчивость и доступность системы.

- Интеграция с другими системами: OpenSearch легко интегрируется с другими системами и инструментами, что позволяет использовать его в различных сценариях и улучшать функциональность веб-сервиса.

3.2.3 Взаимодействие MinIO и OpenSearch

Использование MinIO и OpenSearch в данном проекте обеспечивает эффективное хранение, управление и поиск данных, что позволяет создавать мощный и удобный инструмент для работы с изображениями. Взаимодействие этих компонентов внутри системы должно происходить следующим образом:

Загрузка и хранение изображений:

1. Загрузка изображений: Когда пользователь загружает изображение в систему, оно отправляется на сервер, где происходит его первичная обработка. После этого изображение сохраняется в MinIO, где каждому объекту присваивается уникальный идентификатор и добавляются метаданные, такие как название файла, размер, тип изображения и дата загрузки.

2. Извлечение и индексирование метаданных: После загрузки изображения в MinIO, изображение обрабатывается нейронной сетью и векторное представление обработанного изображения отправляются в OpenSearch для индексирования. Это позволяет создавать поисковые индексы, которые будут

использоваться для быстрого поиска и фильтрации изображений по различным параметрам.

Поиск и рекомендации изображений:

1. Запросы на поиск: Когда пользователь выполняет поиск изображений на платформе, запрос отправляется в OpenSearch. OpenSearch использует свои индексы для быстрого нахождения релевантных результатов на основе заданных критериев в векторном пространстве среди всех изображений. Это могут быть текстовые запросы, фильтры по дате, автору, тегам и другим метаданным.

2. Контекстный поиск и рекомендации: Для улучшения качества поиска и предоставления персонализированных рекомендаций, система будет использовать алгоритмы машинного обучения. Эти алгоритмы анализируют поведение пользователя, его предпочтения и контекст запроса, чтобы предложить наиболее релевантные изображения. Результаты поиска и рекомендации будут динамически обновляться в зависимости от контекста и истории взаимодействий пользователя с платформой.

3.2.4 Выбор модели нейронной сети

Для реализации контекстного поиска и рекомендации изображений в веб-сервисе будут использоваться модели нейронных сетей, специализирующиеся на мультиформатном понимании данных. Для данной задачи выбраны две модели: sentence-transformers/clip-ViT-B-32-multilingual-v1 и sentence-transformers/clip-ViT-B-32. Эти модели должны быть совместно интегрированы с помощью фреймворка Haystack для обеспечения высококачественного поиска и рекомендаций.

Модель sentence-transformers/clip-ViT-B-32-multilingual-v1 основана на архитектуре CLIP (Contrastive Language-Image Pre-training), которая была разработана компанией OpenAI. Эта модель позволяет преобразовывать текстовые запросы в векторные представления, что делает её идеальной для задач, связанных с мультиформатным поиском. Модель поддерживает более

50 языков, что позволяет использовать её для текстовых запросов на различных языках в разных странах. Данная нейросеть способна быстро обрабатывать текстовые запросы и преобразовывать их в векторные представления, что ускоряет процесс поиска и улучшает пользовательский опыт при взаимодействии с веб-системой.

Модель sentence-transformers/clip-ViT-B-32 также основана на архитектуре CLIP и предназначена для работы с изображениями. Она используется для преобразования изображений в векторные представления, что позволяет сравнивать их с текстовыми запросами, представленными в векторной форме.

3.3 Обоснование выбора технологий проектирования и моделей нейронных сетей

3.3.1 Выбор используемых технологий и языков программирования

3.3.1.1 Python

Python - это высокоуровневый язык программирования, известный своей простотой и читаемостью. Он широко используется в разработке веб-приложений и машинного обучения благодаря богатой экосистеме библиотек и фреймворков. большое количество библиотек для машинного обучения, такие как TensorFlow или PyTorch, упрощают и ускоряют разработку сложных приложений.

3.3.1.2 Flask

Flask - это легковесный веб-фреймворк для Python, который позволяет быстро и просто создавать веб-приложения. Flask не включает много встроенных функций, что делает его гибким и позволяет разработчикам добавлять только необходимые компоненты для реализации API.

3.3.1.3 React

React - это JavaScript-библиотека для создания пользовательских интерфейсов, разработанная Facebook. React позволяет создавать быстрые и интерактивные веб-приложения благодаря созданию модульных и повторно используемых компонентов, что упрощает разработку и поддержку кода. Так же React обеспечивает высокую производительность за счет минимизации манипуляций с реальным DOM при обновлении компонентов.

3.3.1.4 Material-UI

Material-UI - это библиотека компонентов для React, основанная на принципах Material Design, разработанных Google. Она предоставляет набор готовых, стилизованных компонентов, таких как кнопки, формы, карточки и модальные окна. Все эти компоненты легко настраиваются под потребности конкретного проекта и обеспечивают консистентный и современный внешний вид приложения, что улучшает пользовательский опыт.

3.3.1.5 Haystack

Haystack - это фреймворк для построения поисковых систем на основе машинного обучения. Он поддерживает интеграцию с различными моделями и базами данных и обеспечивает высококачественный поиск и рекомендации, включая поддержку мультиформатных данных.

3.3.1.6 MinIO

MinIO - это высокопроизводительная объектная система хранения, совместимая с Amazon S3, которая позваляет эффективно хранить и управлять большими объемами неструктурированных данных, обеспечивая быструю обработку данных с минимальными задержками.

3.3.1.7 OpenSearch

OpenSearch - это распределенная поисковая и аналитическая система, основанная на Elasticsearch обеспечивающая быстрое и эффективное индексирование больших объемов данных. Так же включает встроенные инструменты для аналитики и визуализации данных, такие как OpenSearch Dashboards.

3.4 Проектирование пользовательского интерфейса программной системы

3.4.1 Макеты пользовательского интерфейса

На основании требований к пользовательскому интерфейсу, представленных в пункте 2.3 технического задания, был разработан графический интерфейс, используя React с использованием библиотеки Material UI. Разработанный интерфейс ориентирован на обеспечение легкости в использовании и удобного поиска фотографий в системе.

На рисунке 3.2 представлен макет главной страницы.

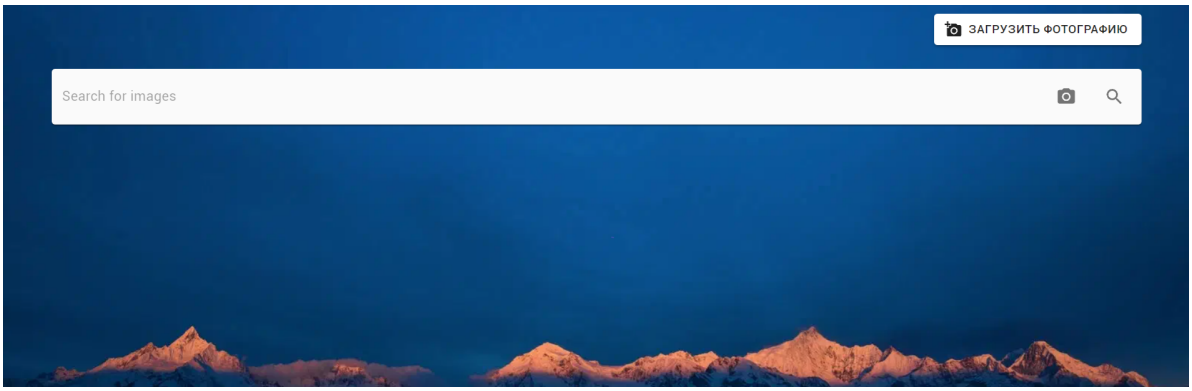

Рисунок 3.2 – Макет главной страницы

Макет содержит следующие элементы:

- Кнопку добавления новой фотографии.
- Поле для ввода запроса.
- Кнопку поиска.
- Кнопку поиска по фотографии.

На рисунке 3.3 представлен макет окна добавления файла.

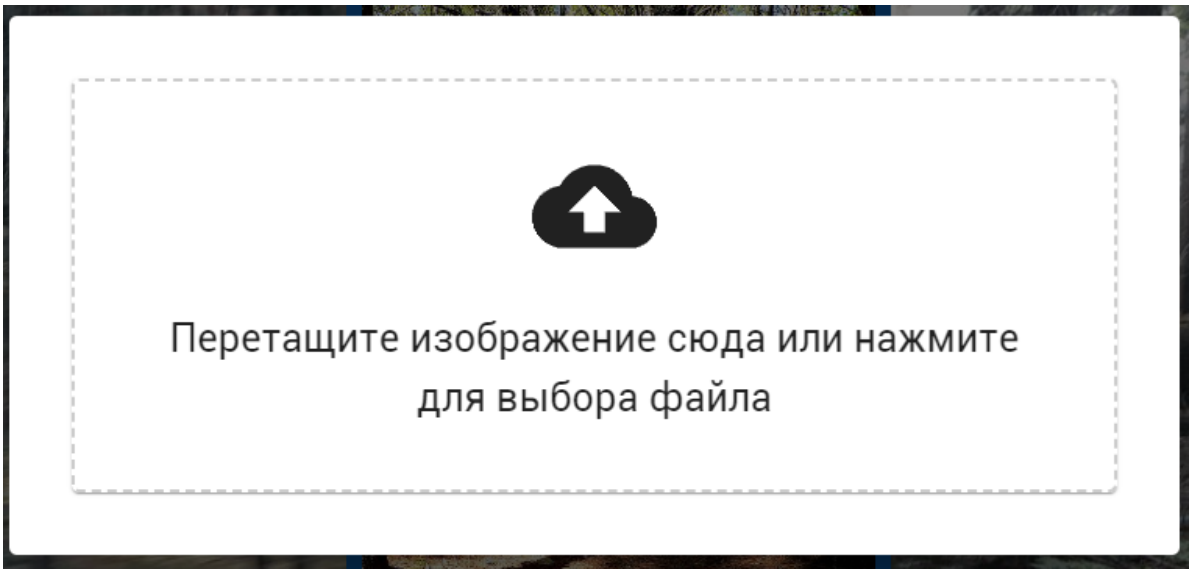

Рисунок 3.3 – Макет окна добавления файла

Макет содержит следующие элементы:

- Заголовок страницы.
- Поле для перетаскивания файлов (drag-and-drop).
- Кнопка для выбора файлов с компьютера.
- Прогресс-баром загрузки.

На рисунке 3.4 представлен макет окна отображения фотографий.

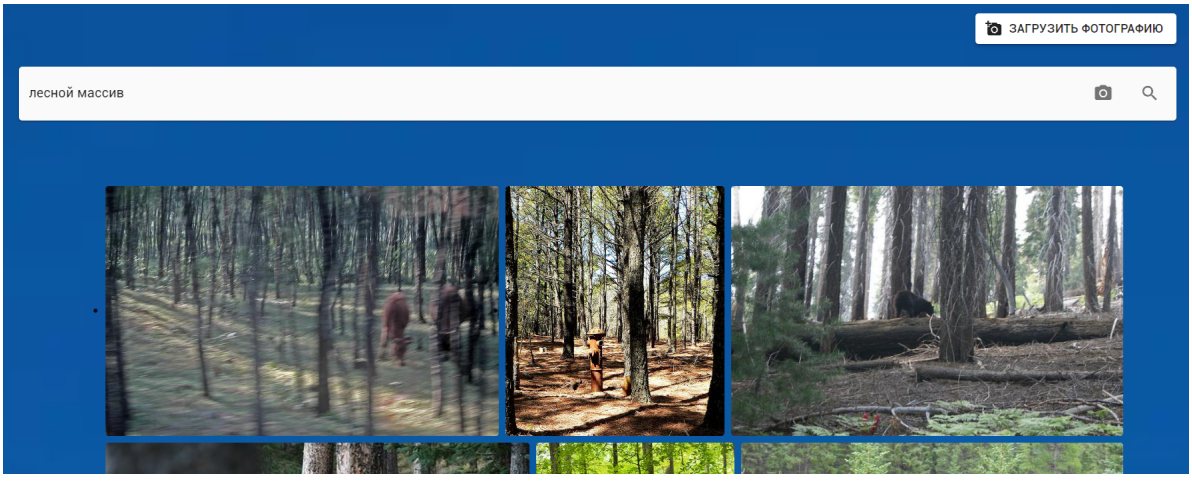

Рисунок 3.4 – Макет окна отображения фотографий

Макет содержит следующие элементы:

- Поле для поиска фотографий.

- Сетка фотографий с предпросмотром.
- Блок загрузки новых фотографий.

На рисунке 3.5 представлен макет окна отображения выбранной фотографии.

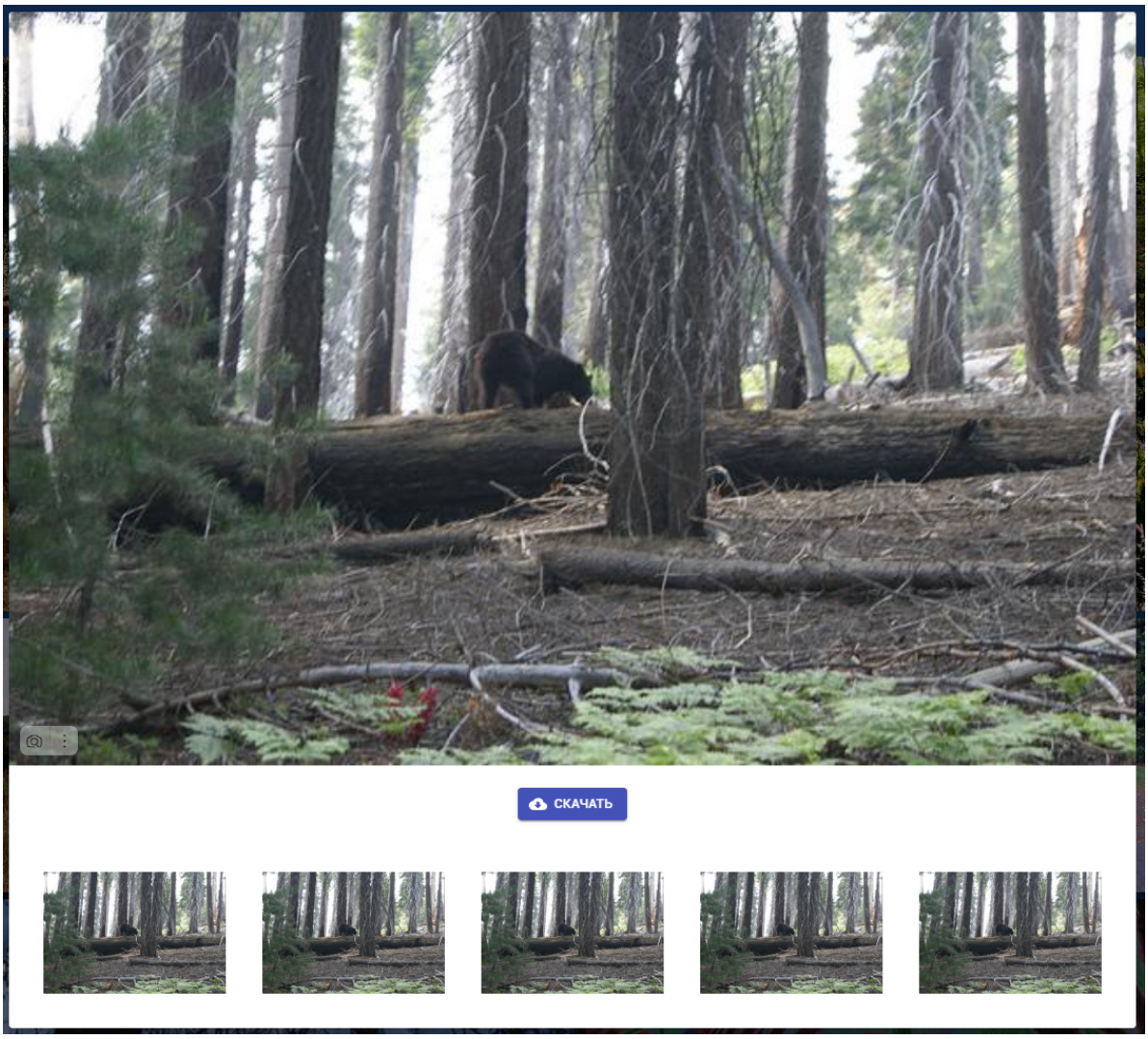

Рисунок 3.5 – Макет окна отображения выбранной фотографии

Макет содержит следующие элементы:

- Изображение в полном размере.
- Кнопка для скачивания фотографии.
- Список похожих на эту фотографий.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Порселло, Б. React. Современные шаблоны для разработки приложений / Б. Порселло. – Санкт-Петербург : Питер, 2022. – 320 с. – ISBN 978-5-4461-1492-4. – Текст : непосредственный.

2. Мартин, Р. Чистый код. Создание, анализ и рефакторинг / Р. Мартин. – Санкт-Петербург : Питер, 2020. – 464 с. – ISBN 978-5-4461-0960-9. – Текст : непосредственный.

3. Haystack : Haystack documentation : сайт. - URL: https://docs.haystack.deepset.ai/docs/intro (дата обращения: 10.04.2024). – Текст : электронный.

4. Arxiv : Learning Transferable Visual Models From Natural Language Supervision : сайт. - URL: https://arxiv.org/abs/2103.00020 (дата обращения: 12.04.2024). – Текст : электронный.

5. Python : Python 3.12.3 documentation : сайт. - URL: https://docs.python.org/3/index.html (дата обращения: 28.03.2024). – Текст : электронный.

6. Скотт, А. Разработка на JavaScript. Построение кроссплатформенных приложений с помощью GraphQL, React / А. Скоттарланд. – Санкт-Петербург : Питер, 2021. – 320 с. – ISBN 978-5-4461-1462-7. – Текст : непосредственный.

7. Material UI : Material UI components : сайт. - URL: https://mui.com/material-ui/all-components/ (дата обращения: 14.04.2024). – Текст : электронный.

8. Хеллман, Д. Стандартная библиотека Python 3. Справочник с примерами / Д. Хеллман. – Москва : Диалектика, 2020. – 1376 с. – ISBN 978-5-6040043-8-8. – Текст : непосредственный.

9. Рашка, С, Мирджалили, В. Python и машинное обучение. Машинное и глубокое обучение с использованием Python, scikit-learn / С. Рашка, В. Мирджалили. – Москва : Диалектика, 2020. – 848 с. – ISBN 978-5-907203-57-0. – Текст : непосредственный.

10. Docker : Docker Docs : сайт. - URL: https://docs.docker.com/ (дата обращения: 29.04.2024). – Текст : электронный.

11. Лутц, М. Программирование на Python. Том 1 / М. Лутц. – Москва : Вильямс, 2024. – 768 с. – ISBN 978-5-907705-28-9. – Текст : непосредственный.

12. Flask :Flask documentation : сайт. - URL: https://flask.palletsprojects.com/en/3.0.x/ (дата обращения: 09.04.2024). – Текст : электронный.