**Изображение выглядит как логотип

Автоматически созданное описаниеМИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ   
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ**(национальный исследовательский университет)»

**Программа стратегического академического лидерства «Приоритет – 2030»**

**ПРОЕКТ «ЦИФРОВАЯ КАФЕДРА»**

**Дополнительная профессиональная программа профессиональной переподготовки  
«Методы искусственного интеллекта в задачах обработки результатов дистанционного зондирования Земли»**

**ОТЧЁТ О ХОДЕ ВЫПОЛНЕНИЯ ИТОГОВОЙ АТТЕСТАЦИОННОЙ РАБОТЫ**

на тему: *«*Создание системы хранения и обработки датасетов»

Руководитель: Стрижак Сергей Владимирович (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

Консультант: Буреева Полина Сергеевна (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

**К защите допустить**

Руководитель ДПП ПП

Булакина Мария Борисовна (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 года

Москва 2025

**КОМАНДА ПРОЕКТА:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | ФИО | Группа по ООП | Роль в команде | Подпись |
| 1 | Кириллова Елена Константиновна | М8О-203Б-23 | Team lead, frontend- разработчик |  |
| 2 | Борисов Денис Сергеевич | М8О-209Б-23 | Backend-разработчик |  |
| 3 | Ветошкина София Владимировна | М8О-203Б-23 | Frontend-разработчик |  |
| 4 | Власко Михаил Михайлович | М8О-208Б-23 | Технический писатель, Backend-разработчик |  |
| 5 | Жуховицкий Александр Дмитриевич | М8О-203Б-23 | Backend-разработчик |  |
| 6 | Михайлов Александр Денисович | М8О-203Б-23 | Fullstack-разработчик |  |
| 7 | Никитцев Антон Валерьевич | М8О-203Б-23 | Fullstack-разработчик |  |
| 8 | Слободин Никита Алексеевич | М8О-203Б-23 | ML-инженер |  |
| 9 | Штыхно Илья Алексеевич | М8О-209Б-23 | ML-инженер |  |
| 10 | Юсупов Артём Маратович | М8О-209Б-23 | ML-инженер |  |
| 11 | Яковлев Вадим Дмитриевич | СМ-31 | Frontend-разработчик |  |
| Группа ДПП | | | ИО204-24 | |

Руководитель работы:

Доцент кафедры 806, кандидат технических наук

Подпись С. В. Стрижак

Консультант:

Сотрудник кафедры 806

Подпись

П. С. Буреева

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](file:///C:\\Users\\Михаил\\Downloads\\наш%20диплом%20(1).docx" \l "_Toc138116175)

[АРХИТЕКТУРА ПРИЛОЖЕНИЯ 7](file:///C:\Users\Михаил\Downloads\наш%20диплом%20(1).docx#_Toc138116177)

[ПРОМЕЖУТОЧНЫЕ ИТОГИ РАБОТЫ](file:///C:\Users\Михаил\Downloads\наш%20диплом%20(1).docx#_Toc138116178) 10

ЗАКЛЮЧЕНИЕ 13

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 14

# **ВВЕДЕНИЕ**

В рамках выполнения проектной работы разрабатывается система (приложение) для хранения и обработки наборов данных (датасетов), используемых в процессе создания дипломных работ студентами Цифровой кафедры Московского авиационного института [1].

Создание системы такого вида является актуальной задачей в рамках совершенствования образовательного процесса Цифровой кафедры, так как в её отсутствие студентам приходится изыскивать датасеты, необходимые в процессе разработки программных продуктов (в особенности частей, связанных с машинным обучением), в сторонних источниках, что сопряжено с трудностями поиска и излишнего взаимодействия с преподавательским составом. Наличие централизованной системы позволит преподавателям размещать и структурировать все необходимые для работы студентов датасеты, а студентам – с помощью средств системы своевременно осуществлять их поиск и выгрузку.

# **АРХИТЕКТУРА ПРИЛОЖЕНИЯ**

Основная логика приложения будет представлена следующим функционалом:

* Загрузка в систему и обработка датасетов, создание групп пользователей (организаций) и управление ими (для администраторов);
* Поиск, просмотр и выгрузка датасетов пользователями.

Отдельные элементы системы будут отвечать за отправку и получение уведомлений (о статусе загрузки датасетов, добавлении пользователя в организацию и др.). Также будет функционировать подсистема, основанный на технологии машинного обучения, которая упростит поиск датасетов и реализует возможность рекомендаций конкретных датасетов для определённой темы дипломной работы. Фактическая архитектура приложения представлена на рисунке 1.

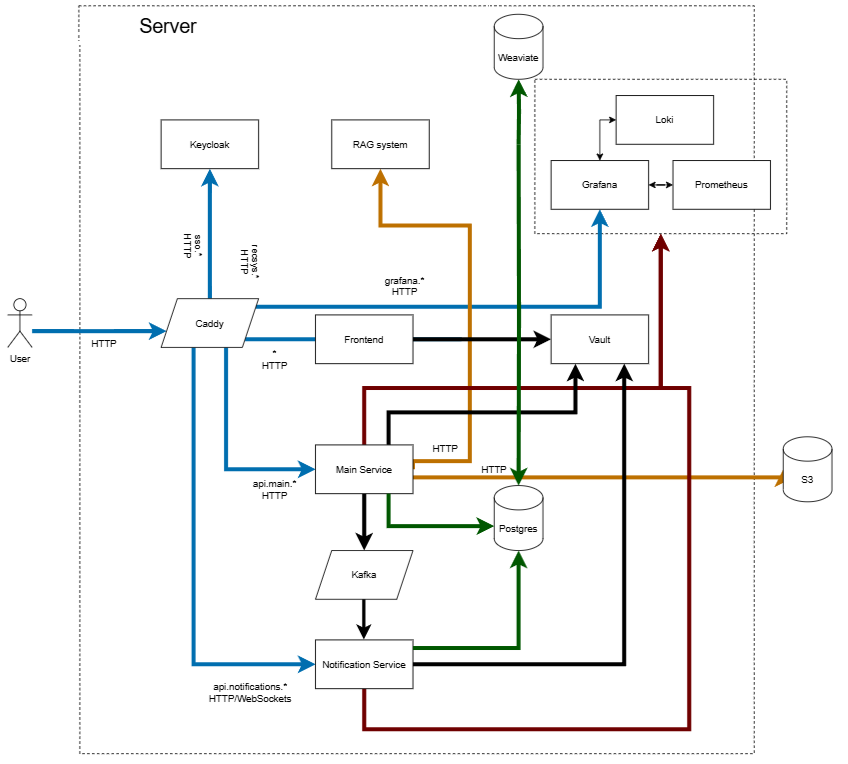


Рисунок 1 – Архитектура приложения

Представленные на схеме компоненты будут выполнять следующие функции:

* Main Service: Обеспечивает загрузку и обработку датасетов, управление правами доступа и группами пользователей, менеджмент базы данных, поиск и получение датасетов пользователями;
* Notification Service: Обеспечивает отправку и получение пользователями уведомлений о статусе загрузки датасетов и изменения принадлежности к организациям;
* ML Service: Обеспечивает непосредственное взаимодействие с нейросетью в процессе углублённого поиска и рекомендации датасетов;
* Kafka: Реализует очередь задач для ML Service, а также очередь уведомлений для Notification Service;
* S3 (Minio): Обеспечивает хранение датаестов;
* PostgreSQL (база данных): Обеспечивает хранение общих данных приложения;
* Authentik: Обеспечивает авторизацию пользователей;
* Vault: Обеспечивает хранение конфиденциальных данных приложения;
* Grafana и Prometheus: Обеспечивает сбор логов и метрик приложения;
* Caddy: Обеспечивает проксирование трафика на определённые сервисы.

Схема основной базы данных представлена на рисунке 2. Представленные на схеме таблицы служат следующим целям:

* users, organizations, datasets, notifications – пользователи, организации, датасеты и уведомления соответственно;
* organization\_users, dataset\_users – пользователи, получившие доступ к конкретным организациям или датасетам;
* organization\_roles, dataset\_roles – роли (права), которые могут быть назначены пользователю в рамках конкретной организации или датасета;
* metadatas, tags, data\_language, data\_task, data\_format, data\_type – метаданные датасетов и система тегов, обеспечивающие поиск датасетов и их разделение на соответствующие категории;
* invites – приглашения пользователя по вступлению в организацию или получению доступа к датасету.

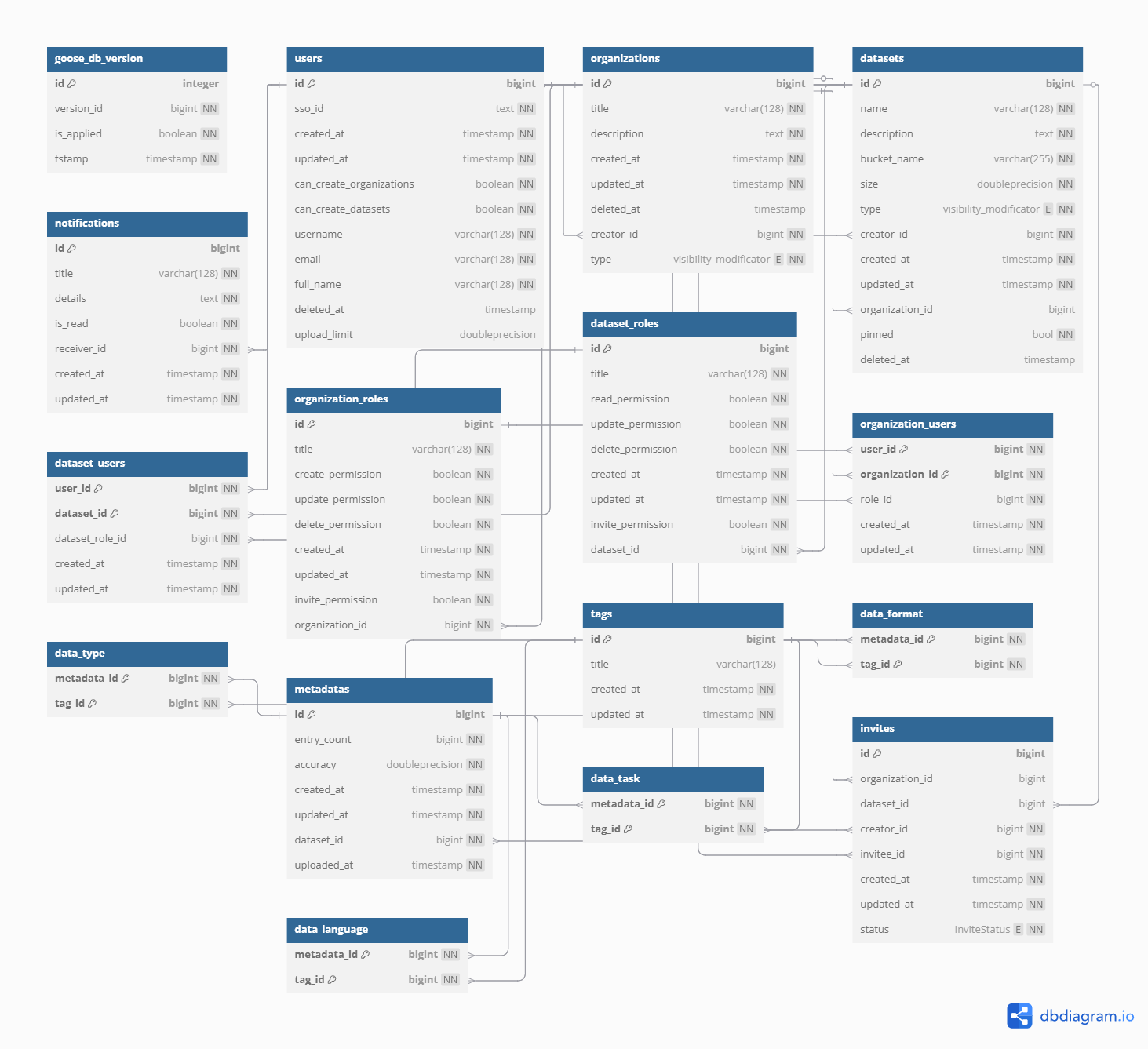


Рисунок 2 – Схема базы данных

# **ПРОМЕЖУТОЧНЫЕ ИТОГИ РАБОТЫ**

**1 ML**

В рамках выполнения задач, связанных с машинным обучением, начата разработка сервиса для взаимодействия с ML-моделями. Интерфейс, использующий FastAPI, реализует внешние эндпоинты для поиска по датасетам и функционирование чата, в котором пользователь, с помощью большой языковой модели (LLM) сможет взаимодействовать с системой, а также внутренние, необходимые для синхронизации с общей базой данных (PostgreSQL). Реализована бизнес-логика по обработке запросов пользователей в чат и дальнейшему взаимодействию с LLM, поиску датасетов. Реализовано взаимодействие с внешней и векторной (Weaviate [2]) базами данных, а также.

Промежуточное представление сервиса подразумевает возможность производить поиск датасетов как с помощью интерфейса поиска, так и с помощью чата. Ответ системы будет иметь следующий формат:

* Для поиска: возврат всех атрибутов объектов из векторной базы данных;
* Для RAG-чата: текстовый ответ по шаблону:
  + В случае успеха: «Найдены следующие датасеты: [список найденных объектов с их идентификаторами и названиями].»;
  + В случае отсутствия результатов: «К сожалению, подходящих датасетов не найдено».

Схема таблицы векторной базы данных согласуется с таблицей datasets (рис. 2):

* id (идентификатор объекта, соответствует идентификатору датасета в PostgreSQL);
* name (название датасета);
* description (описание датасета);
* embedding (векторное представление, вычисленное по name и description);
* Метаданные:
  + upload\_date (дата загрузки);
  + tags: data\_language, data\_format, data\_task, data\_type;
  + size (размер датасета, число записей).

Общая схема сервиса представлена на рисунке 3.

Также был проведён сбор и EDA-анализ более 100 датасетов различных типов (image-classification, text-classification и др.) с портала Huggung Face [3], целью которого было собрать и систематизировать как можно большее количество специфических данных о датасетах в векторной базе данных для последующего использования в процессе обучения моделей и обеспечении процедуры поиска датасетов.

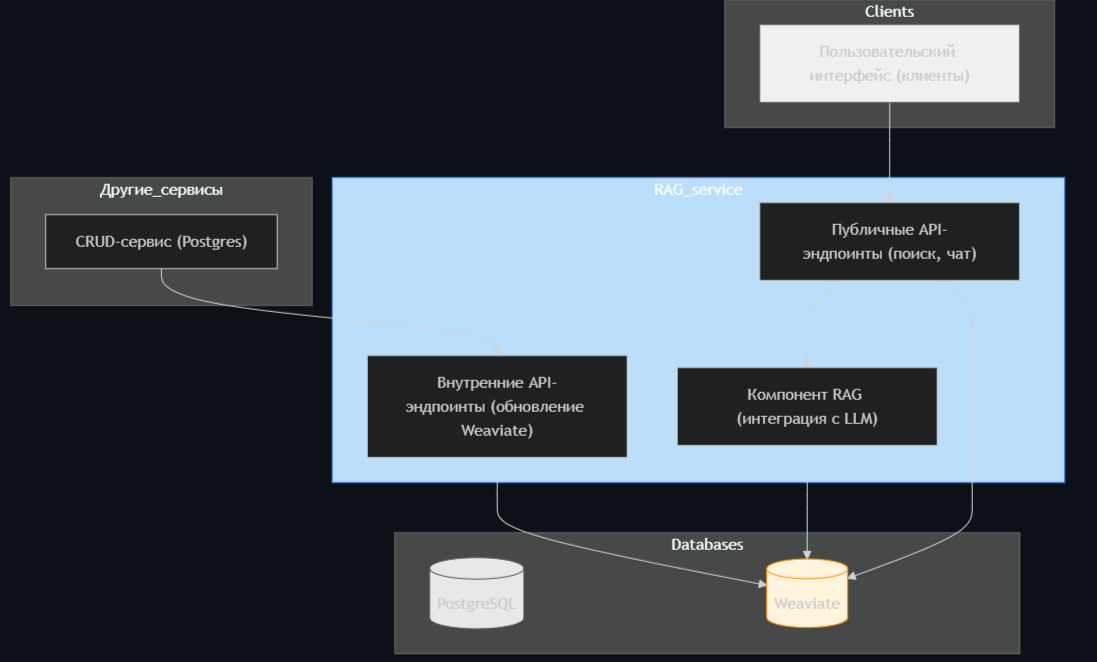


Рисунок 3 – Схема ML-сервиса

**2 Backend**

В рамках разработки серверной части приложения подготовлены атрибуты для развёртывания системы на локальном или удалённом сервере, подготовлена спецификация API и начато написание кода main service и notification service с использованием технологий C# (ASP.NET, Entity framework и др.).

**3 Frontend**

В рамках разработки пользовательского интерфейса была создана базовая структура репозитория для последующей работы с next.js, а также проработан дизайн и начата реализация в коде основных страниц приложения:

* страница настроек пользователя (рисунок 4);
* страница регистрации (рисунок 5);
* страница входа (рисунок 6);
* страницы поиска и просмотра датасетов (рисунки 7 и 8);
* страницы просмотра и создания организаций (рисунки 9 и 10);
* страница просмотра уведомлений (рисунок 11).

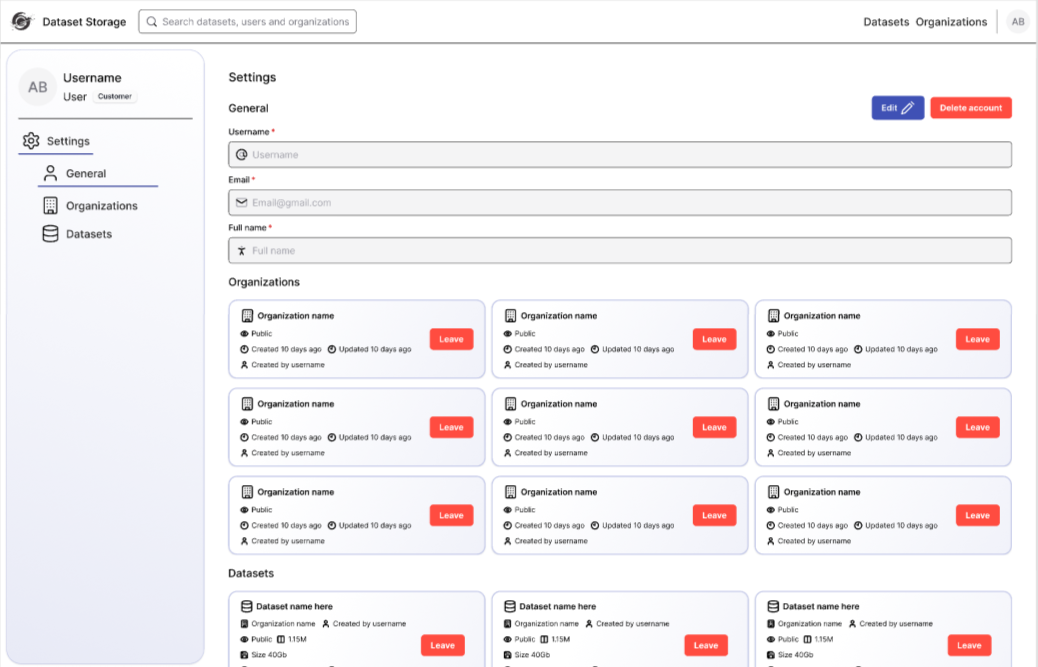


Рисунок 4 – Главная страница

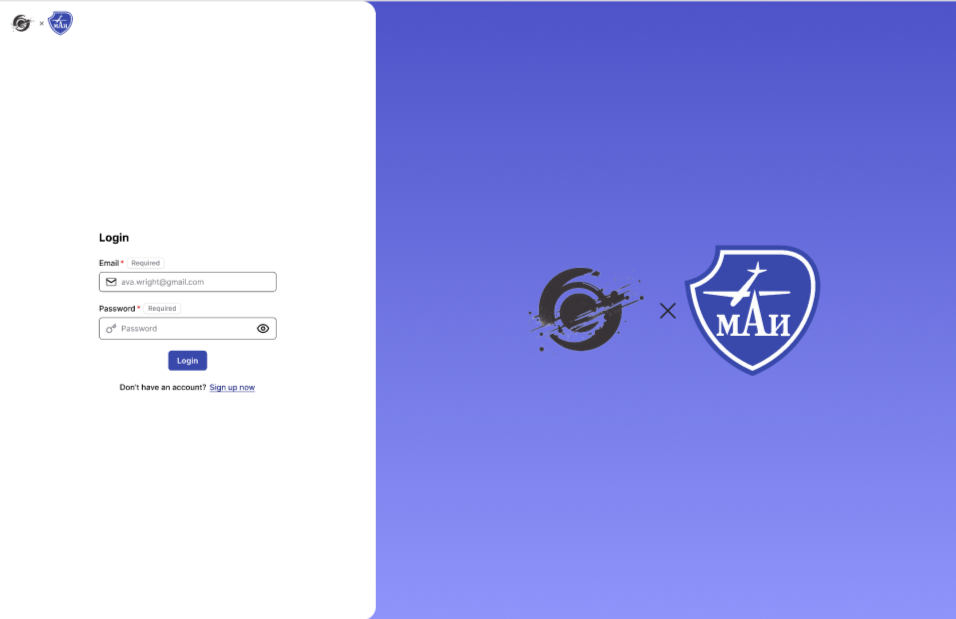


Рисунок 5 – Страница регистрации

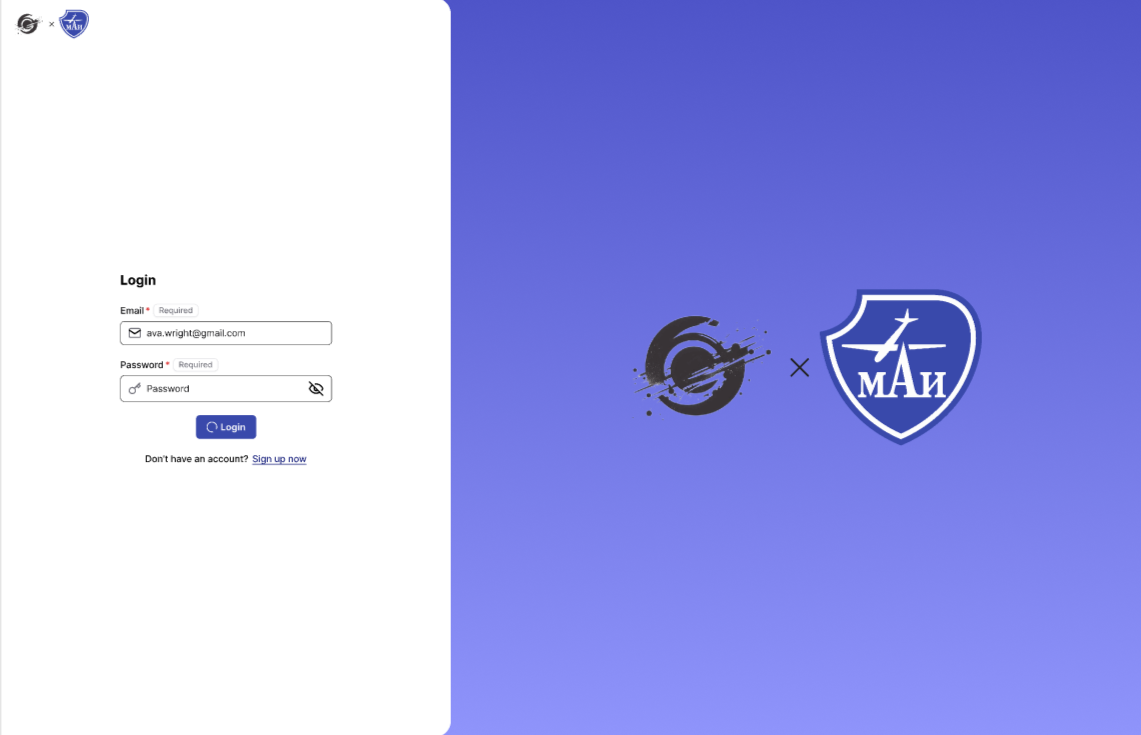


Рисунок 6 – Страница входа

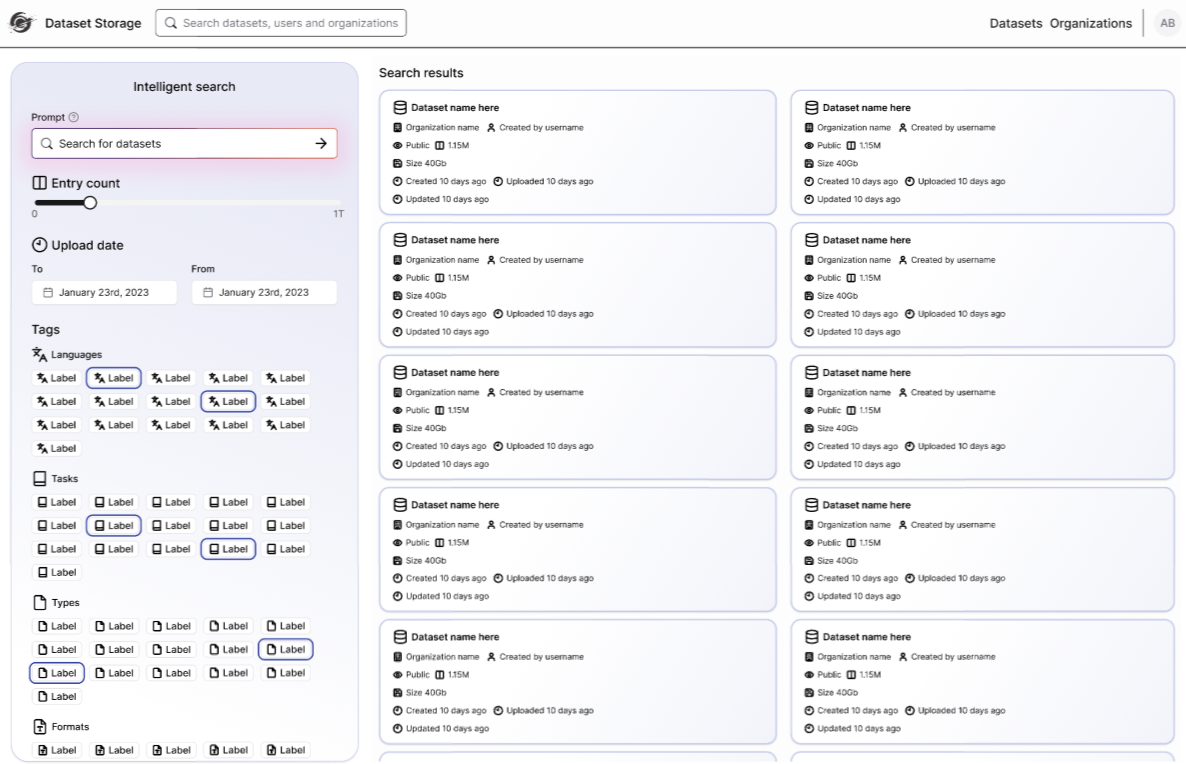


Рисунок 7 – Страница поиска датасетов

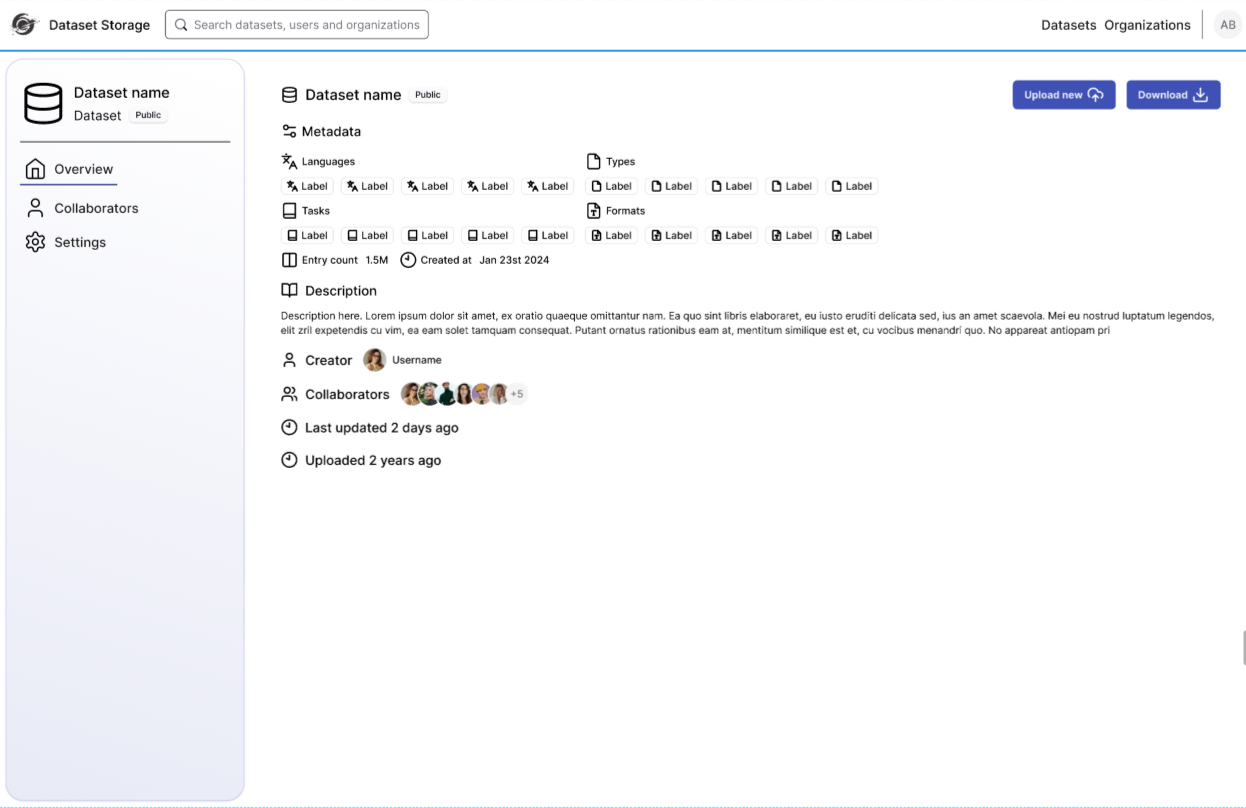


Рисунок 8 – Страница просмотра датасета

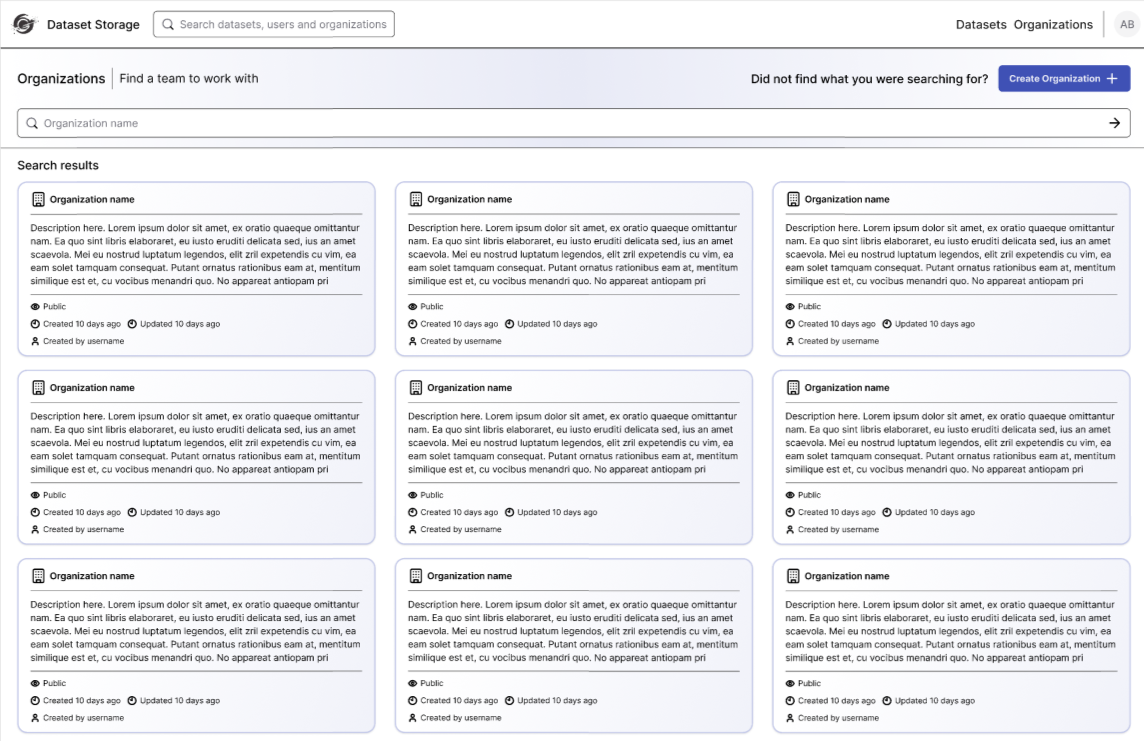


Рисунок 9 – Страница поиска организаций

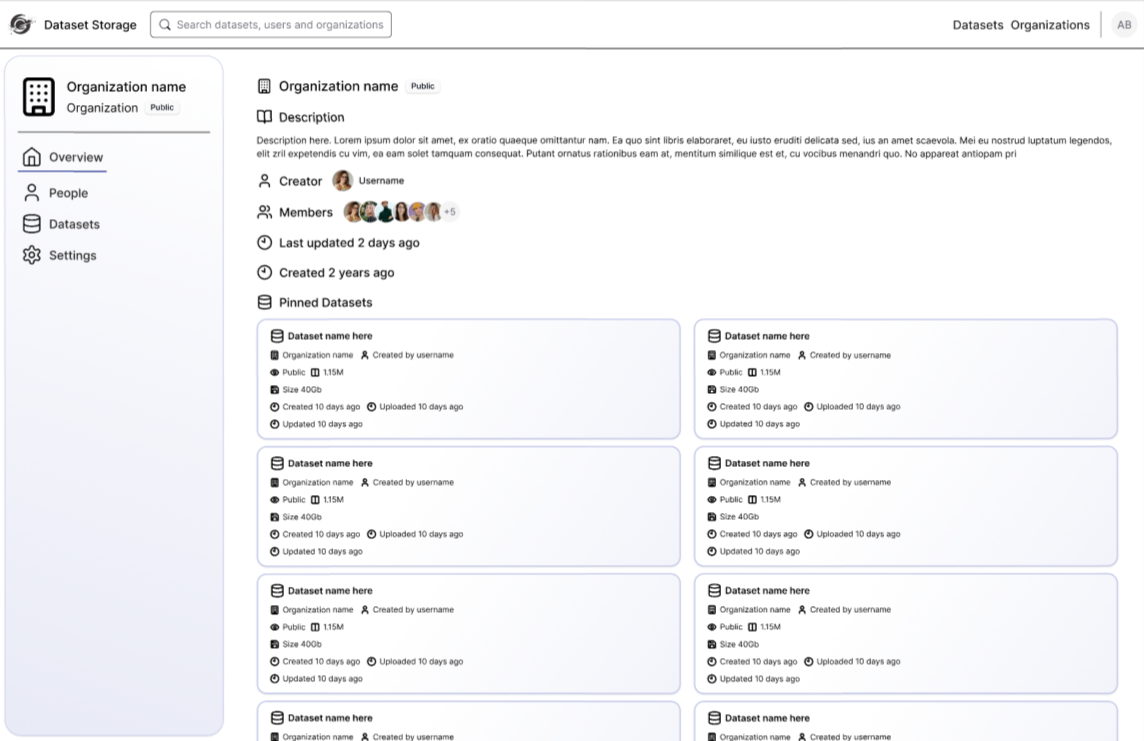


Рисунок 10 – Страница просмотра организации

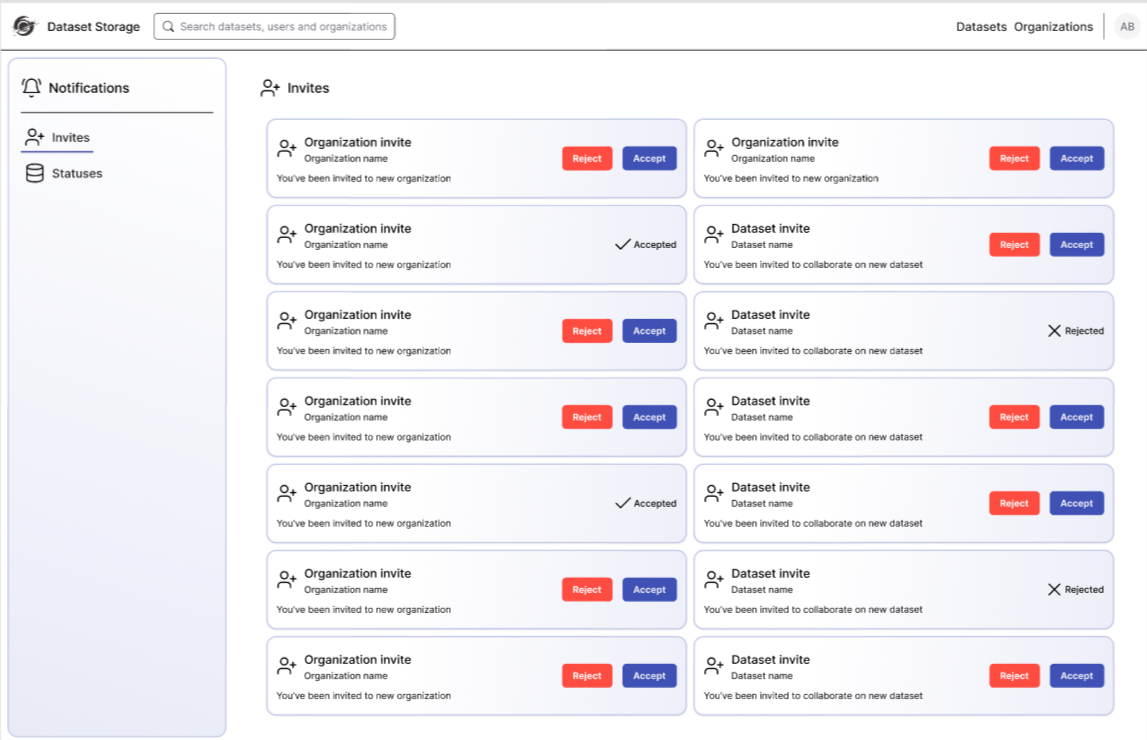


Рисунок 11 – Страница просмотра уведомлений

# **ЗАКЛЮЧЕНИЕ**

Таким образом, разработка приложения для хранения и обработки датасетов начата и успешно продвигается: определены общие принципы и архитектура приложения, проведён анализ данных для последующего обучения и функционирования системы поиска, разработана значительная часть сервиса поиска и рекомендаций датасетов, частично или полностью разработаны некоторые сервисы серверной части приложения, разработаны концепции и часть реализации пользовательского интерфейса.

В рамках дальнейшей работы над проектом будут доработаны backend-сервисы, будет обучена на основе подобранных данных и интегрирована в приложение модель для поиска и рекомендации датасетов, окончательно реализован и интегрирован пользовательский интерфейс, а также будет разработана сопутствующая пользовательская документация и составлен окончательный отчёт об итоговой аттестационной работе.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Цифровая кафедра МАИ //Московский авиационный институт – URL: https://digital.mai.ru/ (дата обращения: 23.03.2025)
2. The AI-native database developers love | Weaviate // Weaviate, B.V. – URL: https://weaviate.io (дата обращения: 23.03.2025)
3. Hugging Face – The AI community building the future // Hugging Face – URL: https://huggingface.co/ (дата обращения: 23.03.2025)