

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ
Школа бакалавриата

ОТЧЕТ

По проекту
«Разработка модуля для анализа новостей в составе NEO Lab Manager»
по дисциплине «Проектный практикум»

Заказчик: Чернышов Ю. Ю.

Куратор: Чернышов Ю. Ю.

Доцент, заведующий учебной лабораторией

Студенты команды Нереальные МЛщики

Голубев И. Д.

Хабибулин Т. Р.

Алексеев Н. Р.

Меслер Р. А.

Ибатулов С. В.

Екатеринбург, 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Отчет о работе. Тимлид / аналитик – Голубев Илья.....	6
2 Backend-разработчик – Меслер Роман	8
3 Backend-разработчик – Алексеев Никита	9
4 ML-разработчик – Хабибулин Тимур	11
5 ML-разработчик – Ибатулов Степан.....	12
6 Требования заказчика	13
7 Анализ и сопоставление аналогов разрабатываемого продукта	14
7.1 Feedly news reader	14
7.2 Google Alerts	15
7.3 Talkwalker	17
7.4 Dzen.....	18
8 Обзор архитектуры программного продукта. Описание основных компонентов и связей между ними.....	19
8.1 Скрапер	19
8.2 LLM-обработчик.....	19
8.3 База данных	19
8.4 News client	19
8.5 Пошаговый процесс	20
8.5.1 Сбор информации	20
8.5.2 Обработка в LLM.....	20
8.5.3 База данных PostgreSQL.....	20
8.5.4 Формирование дайджеста.....	20
8.5.5 Доставка клиенту.....	20
8.6 Итог.....	21
9 Методология разработки и процесс тестирования. Мониторинг качества. Распределение задач между участниками команды разработчиков.....	22

9.1 Методология разработки	22
10 Результаты проекта	24
ЗАКЛЮЧЕНИЕ	25
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	27
ПРИЛОЖЕНИЕ А.....	29

ВВЕДЕНИЕ

Описание: в лаборатории кибербезопасности NEO Lab в ИРИТ-РТФ УрФУ разрабатывается программное обеспечение NEO Lab Manager, имеющее модульную структуру. В рамках проекта предполагается создание одного из модулей для анализа новостей в различных открытых источниках в Интернет – новостные сайты, сайты производителей. Тематика – информационная безопасность.

Функции системы:

- краулинг и парсинг информации;
- ранжирование информации по актуальности (обнаружение «новых» новостей, новых технологий);
- хранение полученной информации для последующего анализа;
- формирование пользовательских рассылок по актуальным новостям (еженедельная основа).

Уровни результата:

- минимальный: реализован базовый функционал краулинга и парсинга;
- базовый: добавлены функции ранжирования и хранения данных;
- оптимальный: полностью готовый модуль с рассылками и интеграцией LLM.

Основная цель: создание программного обеспечения в составе существующего комплекса (MVP).

Требуемый результат: прототип, опубликованный в git.

Критерии оценки: протестированное программное обеспечение, опубликованное в git. Доклад на митапе ИЦ UDV Group. Заготовка статьи на habr.

Стек разработки:

- бэкенд: Java (Kotlin);
- ML: Python.

Описание решения:

1. Аналитика и проектирование:
 - 1.1. Обзор аналогов и решений;
 - 1.2. Проектирование архитектуры.
2. Первый этап разработки:
 - 2.1. Написание скрапера – собирает данные с новостного ресурса;
 - 2.2. Написание LLM модели – выбираем модель, настраиваем API-взаимодействие. Реализуем функционал суммаризации;
 - 2.3. Сбор датасета.
3. Второй этап разработки:
 - 3.1. Реализация базы данных;
 - 3.2. Написание «ручек» между бэкендом и ML.
4. Тестирование и публикация:
 - 4.1. Тестирование модуля;
 - 4.2. Развертывание модуля;
5. Защита проектного практикума.

1 Отчет о работе. Тимлид / аналитик – Голубев Илья

В основные задачи тимлида и аналитика в контексте данного проекта входило:

- организация коммуникации между командой и заказчиком (встречи, уточнение требований);
- постановка задач, контроль сроков;
- составление технического задания на основе требований заказчика;
- проектирование архитектуры модуля;
- обзор аналогов, сравнение с нашим модулем;
- составление итогового отчета и презентации для защиты.

В ходе работы была организована эффективная командная координация:

- заведено рабочее пространство в YouGile (рисунок 1) для управления задачами и распределения ответственности;
- заведено облачное хранилище на Гугл Диск;
- регулярные созвоны для ускорения и синхронизация прогресса;
- проведён сравнительный анализ аналогов, выявлены ключевые преимущества решения, проведена аналитическая работа (рисунок 2).

Как итог, все поставленные организационные задачи были выполнены в установленные сроки. Выбранные инструменты управления доказали свою эффективность в ходе работы над проектом.

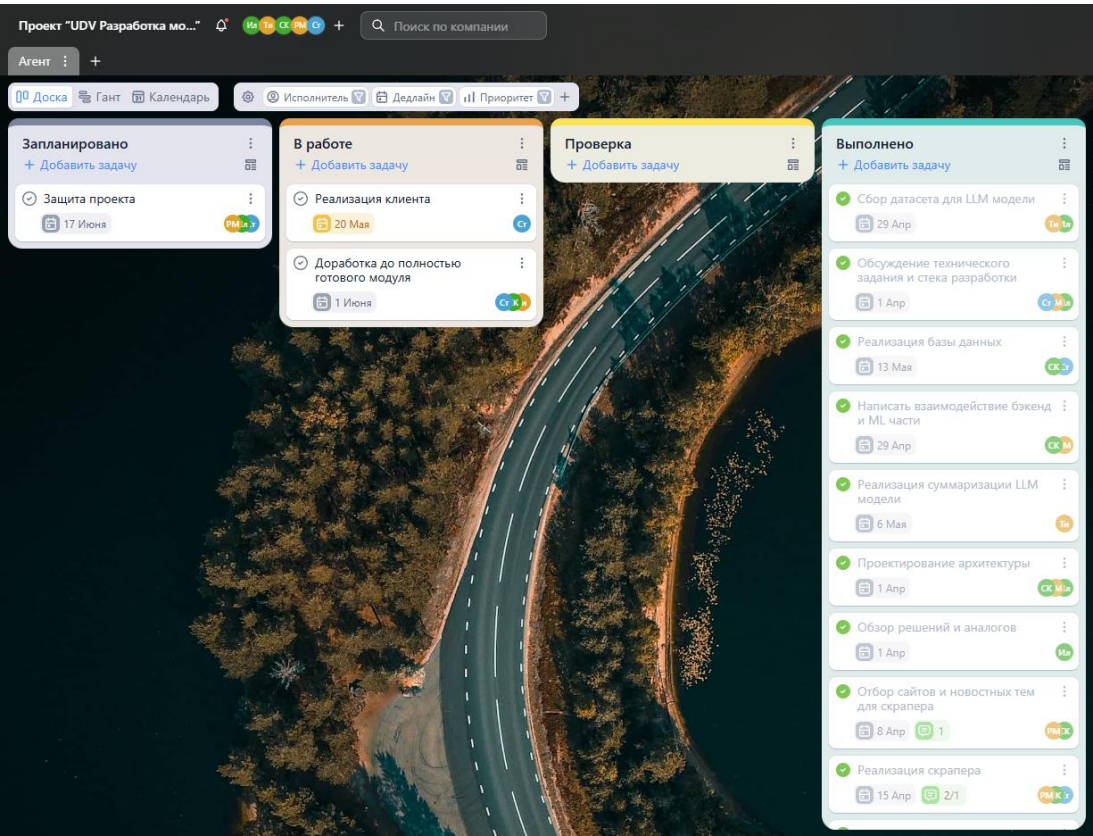


Рисунок 1 – Канбан-доска YouGile

	A	B	C	D
1	Название	Ссылка	Характеристика	Как это выглядит
2	Feedly news reader	feedly.com	<ul style="list-style-type: none">- Автоматический сбор контента- ИИ-фильтрация- Много категорий и большой датасет	
3	Google Alerts	google.com/alerts	<ul style="list-style-type: none">- Мониторинг по ключевым словам- Удобный интерфейс- Система уведомлений- Ограниченные источники	

Рисунок 2 – Обзор аналогов

2 Backend-разработчик – Меслер Роман

В основные задачи backend-разработчика в контексте данного проекта входило:

- проектирование и реализация базы данных PostgreSQL (рисунок 3);
- разработка API для взаимодействия между бэкендом и ML-моделью;
- разработка backend-части модуля (рисунок 5);
- участие в составлении и корректировке технического задания;
- консультирование команды по вопросам взаимодействия компонентов модуля;
- постоянное сотрудничество с ML-разработчиками по интеграции моделей;
- разработка дашбордов для мониторинга системы (ML + backend) на Grafana (рисунок 4).

Все поставленные задачи по разработке бэкенд-части системы были успешно реализованы в установленные сроки. Примененные инструменты разработки и технические решения показали свою надежность и эффективность, а что самое главное: работоспособность.

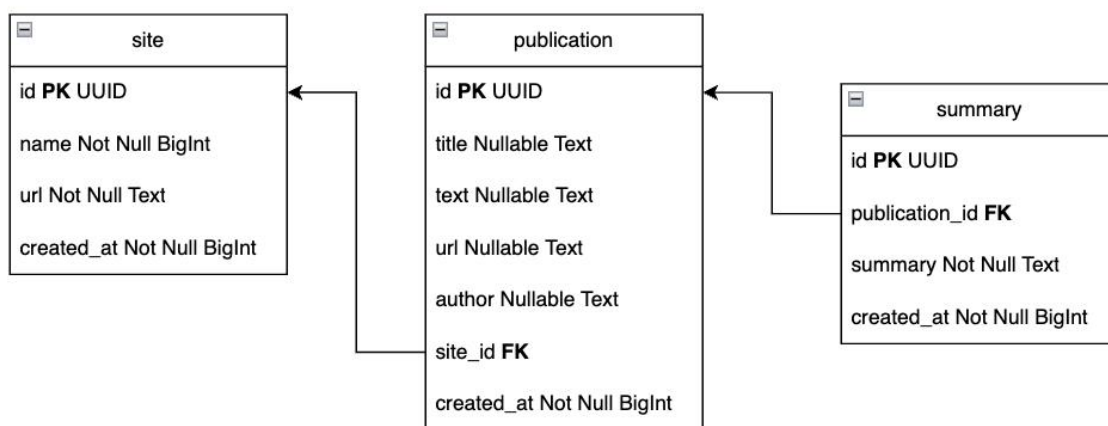


Рисунок 3 – Схема базы данных

3 Backend-разработчик – Алексеев Никита

В рамках проекта backend-разработчик выполнял аналогичный комплекс задач, сильно взаимодействуя с коллегой (Меслер Р.) для обеспечения согласованной работы системы. Основные задачи:

- совместное проектирование и реализация базы данных PostgreSQL (рисунок 3);
- параллельная разработка API для интеграции бэкенда с ML-компонентами;
- создание и оптимизация backend-части модуля (рисунок 5);
- консультирование команды по вопросам системного взаимодействия
- постоянная координация с ML-разработчиками при интеграции моделей;
- разработка дашбордов для мониторинга системы (ML + backend) на Grafana (рисунок 4).

Все поставленные задачи по разработке бэкенд-части системы были успешно реализованы в установленные сроки. Примененные инструменты разработки и технические решения показали свою надежность и эффективность, а что самое главное: работоспособность.

Особенностью работы являлось четкое разделение зон ответственности при постоянной синхронизации между разработчиками, что обеспечило согласованность всех компонентов системы.

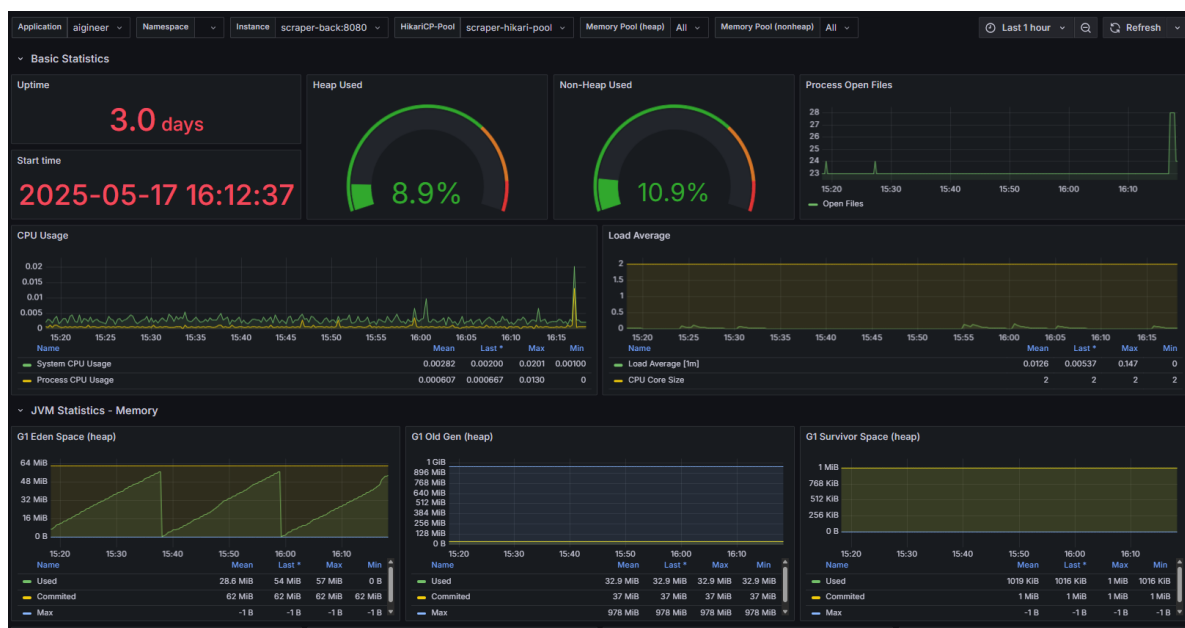


Рисунок 4 – Dashboard метрики на Grafana

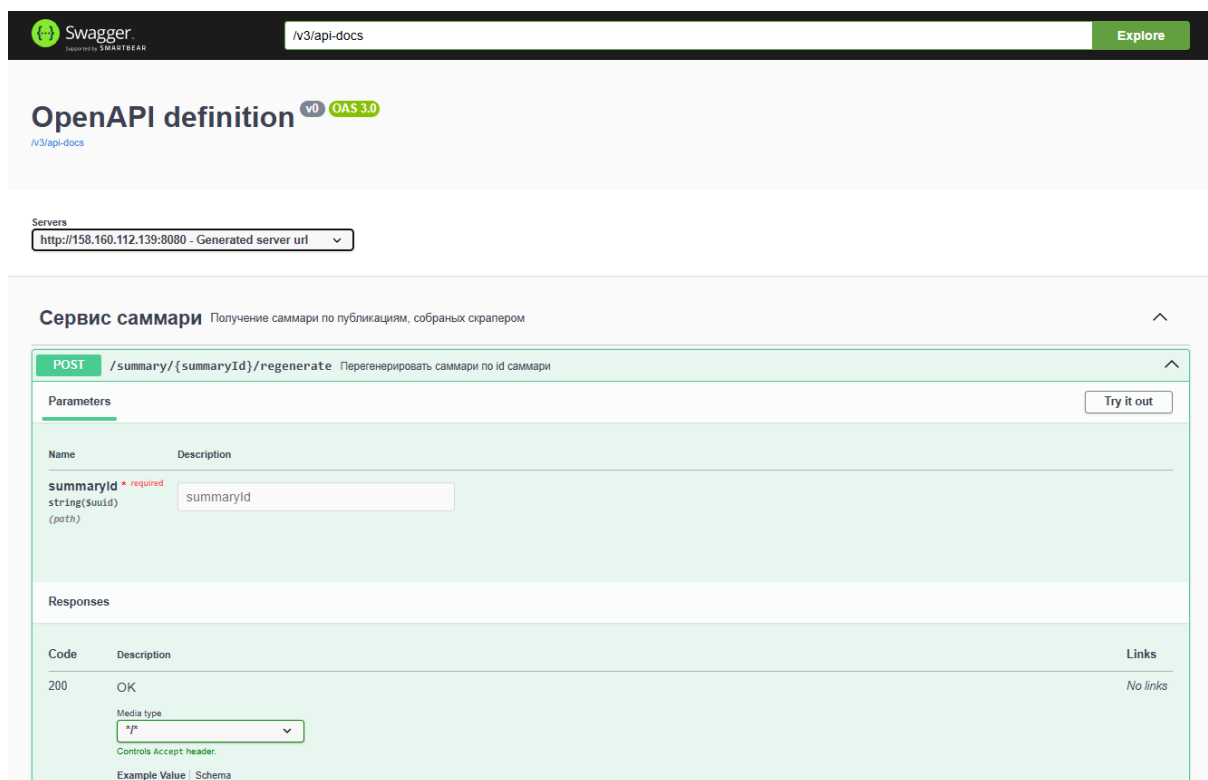


Рисунок 5 – Backend часть на Swagger

4 ML-разработчик – Хабибулин Тимур

В рамках проекта ML-разработчик выполнил следующие ключевые задачи:

- выбор и адаптация LLM для анализа новостей;
- реализация суммаризации текстов новостных источников;
- оптимизация модели для работы с ограниченными ресурсами;
- сбор и предобработка датасета новостей;
- обеспечение совместимости данных с backend (рисунок 6);
- участие в проектировании и согласовании технического задания.

Результат работы: готовая ML-модель, настроенная интеграция с Backend.

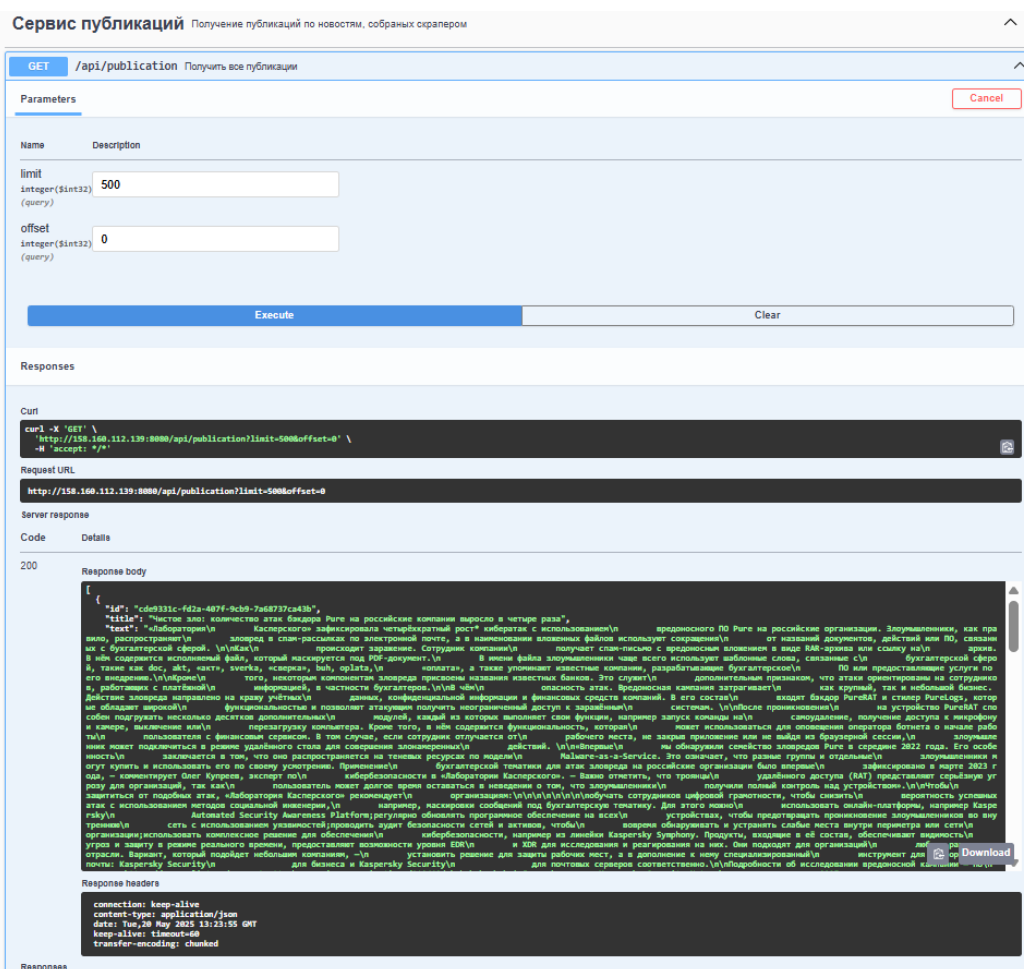


Рисунок 6 – Получение всех публикаций в Swagger

5 ML-разработчик – Ибатулов Степан

В рамках проекта ML-разработчик выполнил следующие ключевые задачи:

- разработка системы сбора данных – написание парсера для сбора данных с новостных сайтов (рисунок 7);
- выбор новостных ресурсов для мониторинга, определение ключевых полей для сбора;
- сотрудничество в реализации структуры базы данных;
- участие в проектировании и согласовании технического задания;
- консультация по вопросам хранения и обработки данных;
- настройка передачи данных в backend-систему;
- реализация планировщика задач cron.

В результате: создан стабильно работающий скрапер, определены предпочтительные для заказчика источники новостей и реализована интеграция со всей системой.

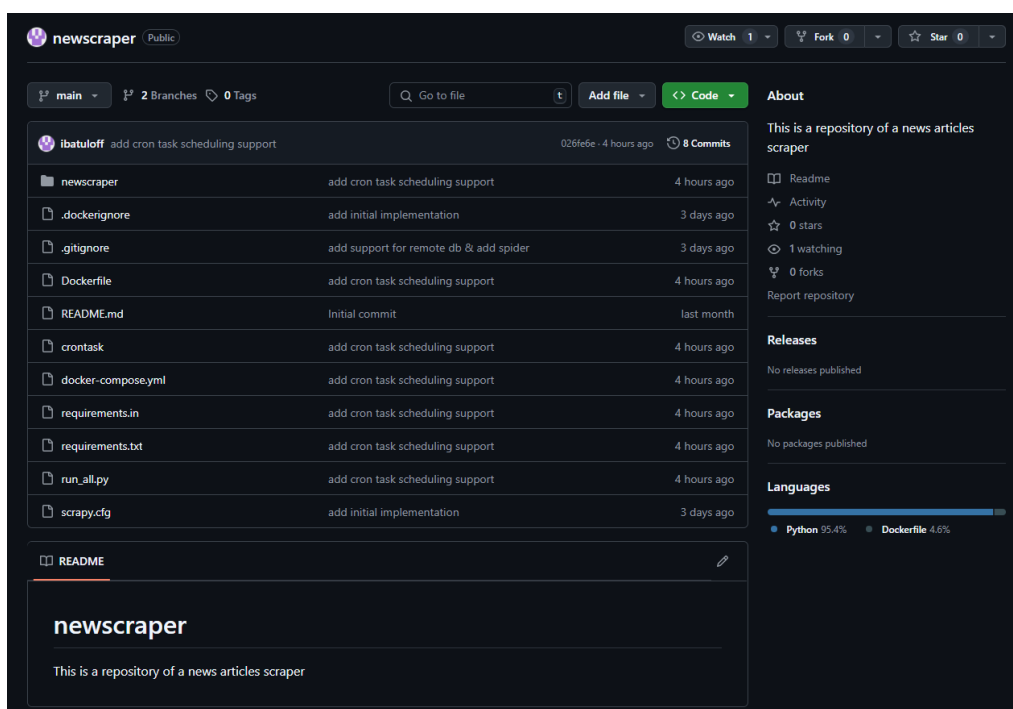


Рисунок 7 – Репозиторий скрапера

6 Требования заказчика

Основные потребности:

- агент для сбора новостей, т. е. автономный краулер для открытых предпочитаемых заказчиком новостных ресурсов (Kaspersky, ptsecurity, angarasecurity и т. д.);
- обработка на основе LLM – классификация новостей по темам, ранжирование по важности, генерация саммари;
- цифровой ассистент (RAG) – автоматические дайджесты (топ-5 новостей дня, тематические еженедельные подборки);
- в перспективе: интеграция в NEO Lab Manager.

Полное цитирование заказчика: «Надо создать агрегатор новостей на основе LLM и агента, который ходит по новостным ресурсам в открытом Интернет, достает новости, складывает наиболее подходящие в векторной базе. Цифровой ассистент с помощью RAG помогает пользователям находить актуальные новости, делает обзоры наиболее популярных новостей (например, 5 наиболее важных новостей дня), делает тематические подборки и саммари по ним. Все это в перспективе надо интегрировать в виде микросервиса в NEO Lab Manager – программное обеспечение для использования в образовательном процессе. Neo Lab Manager другая команда разрабатывает, от вашей нужна только часть с LLM и агентом для цифрового ассистента по новостям.»

Критерии приёмки и план действий для достижения цели (backlog) с перечнем ключевых задач и этапов разработки представлен в разделе «Введение» отчета. Он включает в себя: функции системы, уровни результата, формулировку основной цели, требуемый результат, критерии оценки, стек разработки и описание решения.

7 Анализ и сопоставление аналогов разрабатываемого продукта

Для сравнения с нашим продуктом на обзор были выбраны четыре аналога: Feedly news reader, Google Alerts, Talkwalker, Dzen.

Разберем характеристику каждого.

7.1 Feedly news reader

Feedly (рисунок 8) – облачный сервис для мониторинга новостей, который автоматически собирает обновления из выбранных источников – блоги, новостные сайты, соц.сети и т.д.).

Ключевые характеристики:

- автоматический сбор контента;
- ИИ-фильтрация;
- большое количество категорий новостей и большой датасет.

Feedly самый подходящий аналог для нашего проекта. Как и в нашем проекте, здесь агент так же автоматически находит новости и генерирует качественные саммари и дайджесты через LLM.

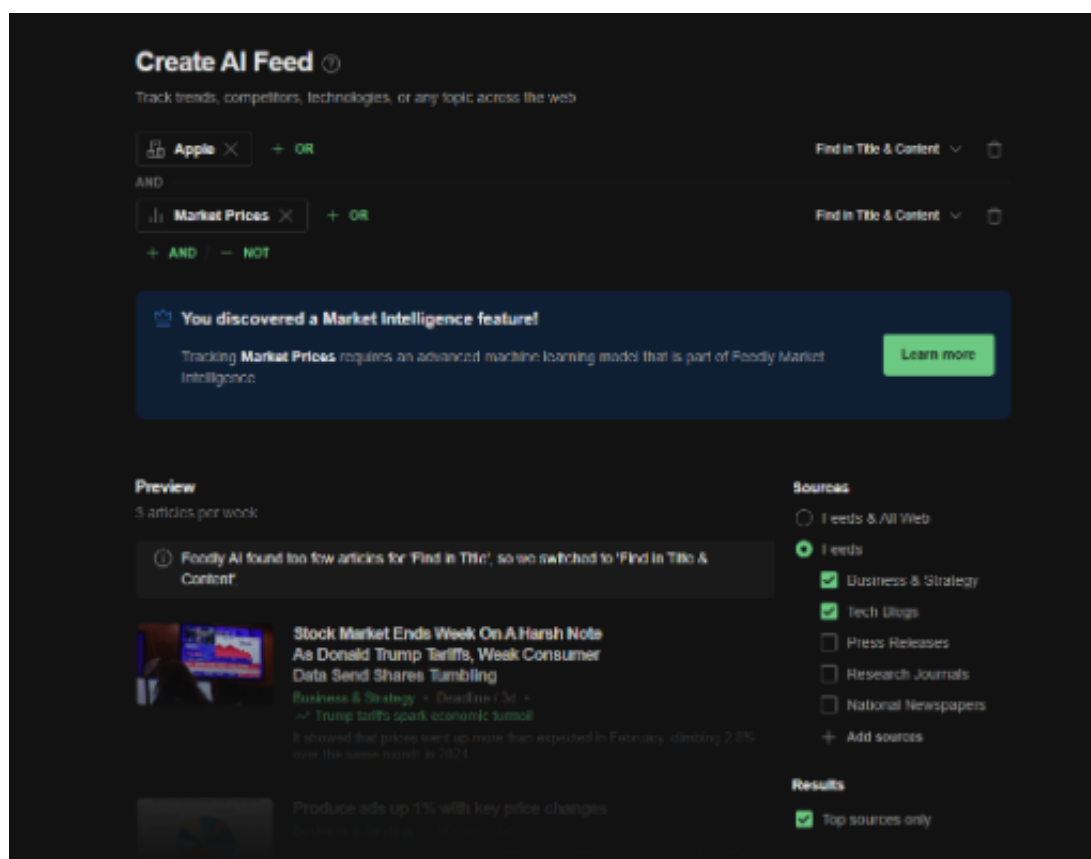


Рисунок 8 – Выбор категории для создания персонального дайджеста в Feedly

7.2 Google Alerts

Google Alerts (рисунок 9) – бесплатный сервис мониторинга контента от Google, который отправляет уведомления на email при появлении новых материалов по заданным ключевым словам.

Ключевые характеристики:

- автоматический мониторинг;
- гибкие настройки;
- простота использования;
- ограниченные источники;
- система уведомлений.

Google Alerts полезен для базового мониторинга, но уступает нашему решению в гибкости и глубине анализа. Наш проект предполагает

полноценный LLM-ассистент с автономным сбором данных и интеграцией в образовательную систему.

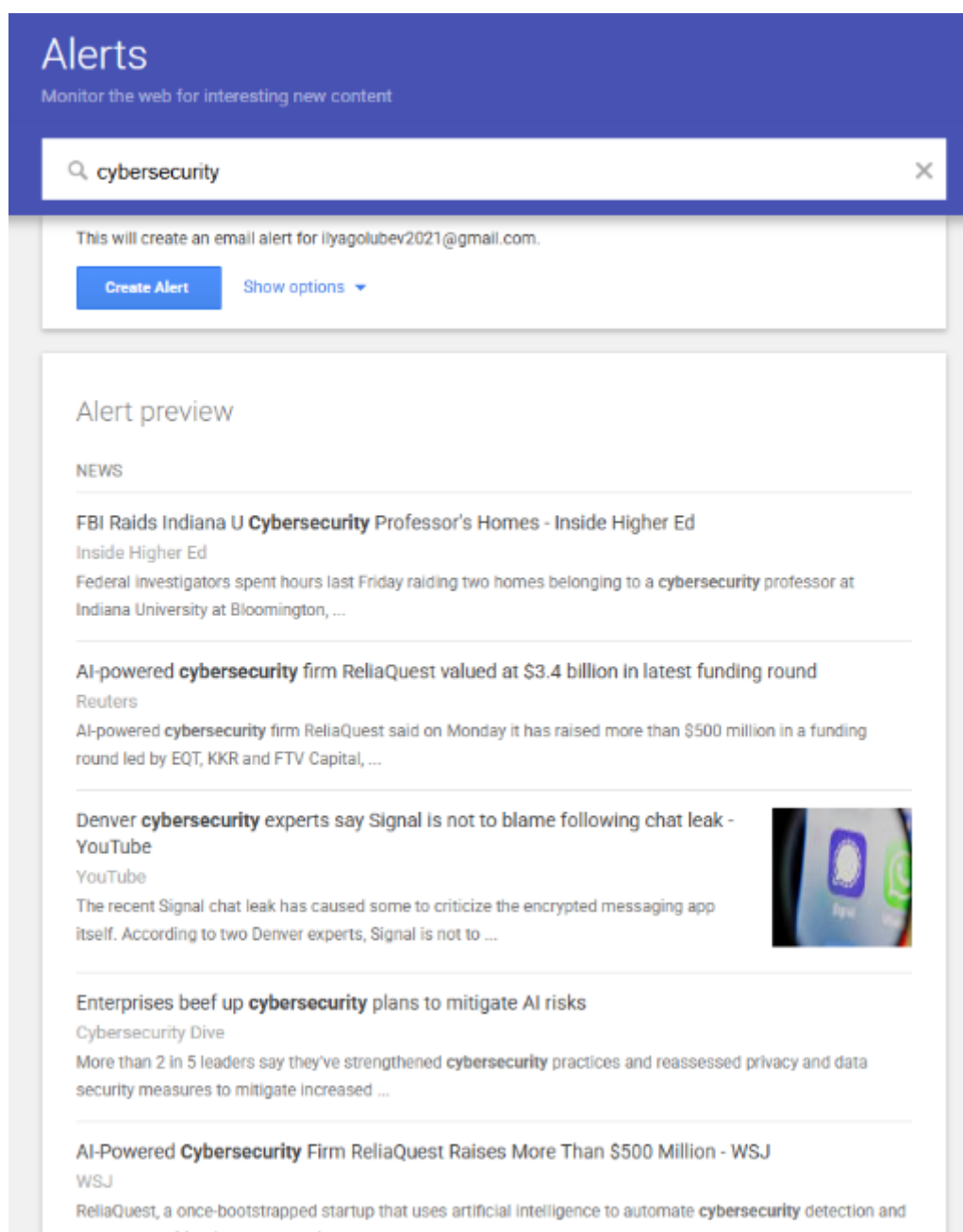


Рисунок 9 – Интерфейс Google Alerts

7.3 Talkwalker

Talkwalker (рисунок 10) – профессиональная платформа для мониторинга медиа и социальных сетей с расширенной ИИ-аналитикой, ориентированная на корпоративных клиентов.

Ключевые характеристики:

- мониторинг в реальном времени;
- мощная ИИ-аналитика
- визуализация данных
- высокая стоимость

Talkwalker не подходит для образовательных целей из-за высокой цены, избыточности и закрытости, т.е. невозможности дообучить модель под специфические задачи.

Несмотря на мощную аналитику, Talkwalker – это «тяжелое» коммерческое решение, тогда как наш проект предлагает гибкость, открытость и экономичность.

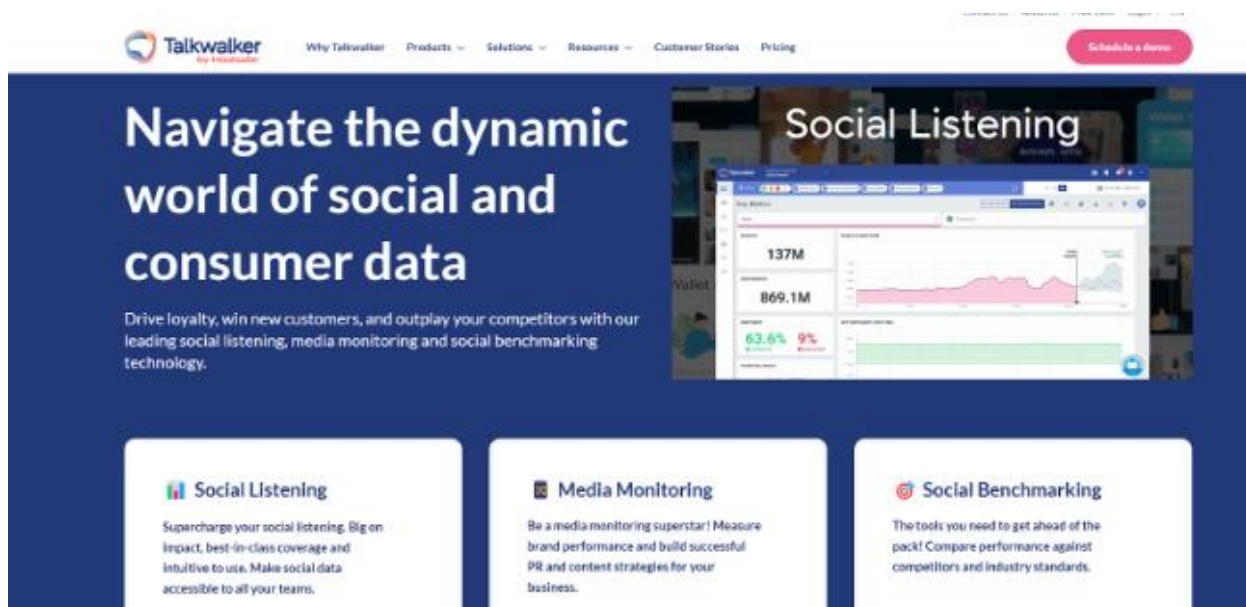


Рисунок 10 – Интерфейс Talkwalker

7.4 Dzen

Dzen (рисунок 11) – русифицированная платформа для мониторинга соцсетей и медиа, разработанная Яндексом.

Ключевые характеристики:

- мониторинг соцсетей;
- русифицированный интерфейс;
- слабая аналитика
- бесплатный

Dzen подходит для базового мониторинга русскоязычного контента, но слишком сильная локализация и примитивная аналитика. Уступает нашему решению в глубине анализа, гибкости и масштабируемости

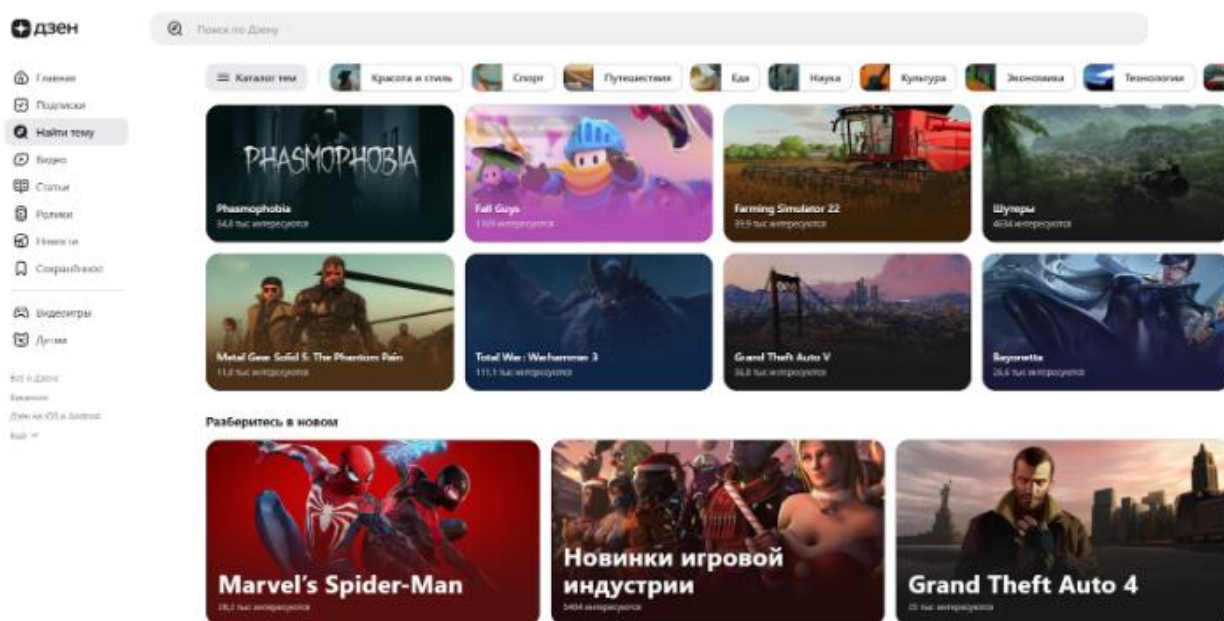


Рисунок 11 – Интерфейс Dzen

Проведенный обзор аналогов для мониторинга новостей позволил выявить их ключевые ограничения и определить конкурентные преимущества нашего решения на базе LLM + автономного агента. Наш проект должен устранять ключевые недостатки аналогов, а само решение должно быть специализированным для образовательной платформы.

8 Обзор архитектуры программного продукта. Описание основных компонентов и связей между ними.

Основные компоненты системы:

8.1 Скрапер

Автономный модуль для сбора новостей с веб-источников. Поддерживает парсинг HTML и обработку динамического контента. Отправляет данные LLM-обработчику

8.2 LLM-обработчик

Принимает текст от скрапера, выполняет суммаризацию, проверяет новость на фейковость. Преобразует данные в базу данных (PostgreSQL) с полями

8.3 База данных

База данных (PostgreSQL) (Рисунок 3). Причины выбора данного кластера заключается в поддержке JSON для хранения неструктурированных данных новостей, интеграция с pgvector для векторного поиска и надежные механизмы репликации и бэкапов.

8.4 News client

Интерфейс для NEO Lab Manager. Чтение новостей и фильтрация по тематикам и дате (рисунок 12).

Dockerfile	README.md	crontask	docker-compose.yml	logs	newscrafer	requirements.in	requirements.txt	run_all.py	scrapy.cfg
scrape@scrapersbackend:~/newscrafer\$ docker ps									
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES			
ebbf1c98ad8	newscrafer-scraper	"/bin/sh -c 'cron -f'"	3 minutes ago	Up About a minute		newscrafer-scraper-1			
ee266c0ed6b9	grafana/grafana:latest	"/run.sh"	5 days ago	Up 5 days	0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp	scrapersapi-grafana-1			
a1cfa42839fd	prom/prometheus:latest	"/bin/prometheus --c=..."	5 days ago	Up 32 hours	0.0.0.0:9090->9090/tcp, [::]:9090->9090/tcp	scrapersapi-prometheus-1			
8f2ece9559a6	prom/node-exporter:latest	"/bin/node_exporter"	5 days ago	Up 5 days	0.0.0.0:9100->9100/tcp, [::]:9100->9100/tcp	scrapersapi-node_exporter-1			
2f40009cccf1	scrapersapi-scraper-back	"java -jar /backend..."	5 days ago	Up 5 days	0.0.0.0:8080->8080/tcp, [::]:8080->8080/tcp	scrapersapi-scraper-back-1			
scrape@scrapersbackend:~/newscrafer\$									

Рисунок 12 – Деплой компонентов системы в Docker-контейнерах

8.5 Пошаговый процесс

8.5.1 Сбор информации

Скрапер парсит целевые ресурсы (Kaspersky, Хабр) с периодичностью. Для каждой новости извлекаются полный текст статьи и метаданные (дата публикации, автор, источник).

8.5.2 Обработка в LLM

Скрапер отправляет текст на API-сервис модели. Модель обрабатывает запрос и возвращает структурированный ответ с полями («summary», «type», «isFake» и другие). Сохраняет их в базу данных

8.5.3 База данных PostgreSQL

Данные записываются в таблицу PostgreSQL и сохраняет их

8.5.4 Формирование дайджеста

Ежедневно в определенное время система автоматически выбирает самые важные новости за последние 24 часа и генерирует сводку через LLM

8.5.5 Доставка клиенту

Дайджест отправляется на веб-интерфейс NEO Lab Manager, где пользователи могут взаимодействовать с нашей системой и получать подборки новостей.

8.6 Итог

Такое описание сохраняет простоту, но добавляет техническую строгость. Четко отделяет компоненты системы (скрапер/LLM/БД/клиент)

9 Методология разработки и процесс тестирования. Мониторинг качества. Распределение задач между участниками команды разработчиков.

9.1 Методология разработки

Принципы Agile. Индивидуальные рабочие среды:

- каждый разработчик работал в изолированной среде;
- спринты по две недели;
- планирование в YouGile – разбивка на задачи по компонентам системы;
- критерии завершения: еженедельные созвоны, демонстрация результата;
- контроль версий в Git.

Распределение технологий по разработчикам:

- ML-разработка: Python;
- Backend-разработка: Java (Kotlin), Grafana;
- инструменты аналитики и управления: YouGile, Google Drive, Excalidraw, Git.

Для мониторинга качества использовались дашборды Grafana, где указана нагрузка на систему и её работоспособность.

По ходу проекта команда работала в рамках трех итераций. Так же команда была на постоянном взаимодействии с заказчиком. Оперативная связь проводилась в чате Телеграма, где уточнялись требования и давались подтверждения.

Итог: гибкая методология с акцентом на автоматизацию и изолированность задач позволила:

- минимизировать зависимость и конфликты между разработчиками;
- обеспечить стабильность процесса разработки;

- быстро реагировать на изменения требований;
- быть на постоянной обратной связи с заказчиком.

10 Результаты проекта

Целевая платформа: развертывание системы на этапе третьей итерации проектного практикума выполнено на Яндекс Облаке.

Конфигурация инфраструктуры:

- скрапер;
- LLM-сервис;
- база данных PostgreSQL;
- клиентский API.

Команда успешно реализовала все ключевые требования, обозначенные заказчиком:

- агрегатор новостей на базе LLM. Реализован автономный агент для сбора данных с новостных ресурсов. Настроена интеграция с базой данных и бэкендом;
- цифровой ассистент. Внедрена генерация ежедневных дайджестов, тематических подборок новостей;
- микросервис для Neo Lab Manager. Обеспечена совместимость с API образовательной платформы;
- соблюдены все этапы разработки согласно дорожной карте. MVP за 8 недель и подготовка финальной версии к защите.

На сегодняшний день написания отчета команда планирует продемонстрировать свой результат заказчику, получить обратную связь и готовиться к финальной защите проектного практикума.

Все ссылки проекта предоставлены в Приложении А.

ЗАКЛЮЧЕНИЕ

Наше решение проекта, т. е. разработанный программный продукт полностью соответствует заявленным требованиям, что подтверждается тем, что для заказчика – реализован функционал агрегации новостей с анализом (LLM + RAG), включая автоматические дайджесты и в перспективе интеграцию с NEO Lab Manager. Обеспечена гибкость системы: поддержка новостных источников добавление конфигурационных файлов. Для пользователей – создан удобный сервис для поиска релевантных новостей с их классификацией.

Разработанный модуль был специально спроектирован для интеграции с исключительно с образовательной платформой NEO Lab Manager. Но несмотря на узкую специальность, архитектура системы позволяет адаптацию для других проектов, благодаря модулю сбора данных и LLM-модели. В перспективе и некоторых доработок проект может стать отдельным open-source модулем для любых пользователей сети Интернет.

Разработка модуля не обошлась без сложностей – были и технические проблемы, и неочевидные решения, которые требовали пересмотра подходов (например отказа от Kafka). Иногда казалось, что задача нерешаема: то модель не выдавала нужный результат, то решение отдельных задач просрочивали дедлайн. Но каждая проблема находила свое решение, как только мы сосредотачивались на ней все командой. Главное, что мы вынесли из этого проекта – опыт командной работы, умение распределить задачи, слушать друг друга и находить нужные решения. Научились лучше взаимодействовать с заказчиком – как четко формулировать вопрос, оперативно вносить правки и подтверждать требования. Кроме того, строгая отчётность и системное документирование позволили команде глубже проработать архитектуру проекта и выстроить чёткую структуру работы.

И хотя технический результат важен – работающий агрегатор новостей с ИИ-аналитикой – куда ценнее оказались навыки, которые мы приобрели в процессе. Уверены, что этот опыт поможет нам в будущих проектах, а собранные наработки станут хорошей базой для новых идей. Мы не просто сделали продукт, а улучшили свои навыки работы в команде.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Веб-скрапинг: как обстоят дела с законностью подобных действий [Электронный ресурс] / Мобильные прокси. - 2025. - URL: <https://mobileproxy.space/pages/veb-skraping-kak-obstoyat-dela-s-zakonnostu-podobnyh-deistvii.html> (дата обращения: 31.03.2025).
2. Подключение к базе данных в кластере PostgreSQL [Электронный ресурс] / Yandex Cloud. - 24.04.2025. - URL: yandex.cloud/ru/docs/managed-postgresql/operations/connect?utm_referrer=https%3A%2F%2Fyandex.ru%2F&utm_referrer=about%3Ablank (дата обращения: 14.05.2025).
3. Zanini, A. Как парсить новости с помощью Python и ИИ [Электронный ресурс] / Bright Data. — 2025. - URL: ru-brightdata.com/blog/web-data-ru/how-to-scrape-news-articles (дата обращения: 14.04.2025).
4. Grafana OSS and Enterprise [Электронный ресурс] / Grafana Labs. — URL: <https://grafana.com/docs/grafana/latest/> (дата обращения: 17.05.2025).
5. Курс по документированию API [Электронный ресурс] / Learnapidoc.ru 2.0. - URL: <https://starkovden.github.io/swagger-ui-tutorial.html> (дата обращения: 16.05.2025).
6. Митчелл Р. Парсинг с помощью Python. Веб-скрапинг в действии : 3-е международное издание / Р. Митчелл. — Астана : Спринт Бук, 2025. — 352 с. — ISBN 978-601-08-4570-1.
7. Парсинг с помощью LLM: зачем, как и сколько стоит? [Электронный ресурс] / Хабр. - 2025. - URL: <https://habr.com/ru/articles/892954/> (дата обращения: 07.04.2025).
8. 10 библиотек Python для машинного обучения - подборка для начинающих [Электронный ресурс] / К. Симонов / Хабр. - 10.07.2024. - URL: habr.com/ru/companies/skillfactory/articles/828002/ (дата обращения: 04.04.2025).
9. Настройка PostgreSQL для LLM [Электронный ресурс] / А. Пан / Хабр. - 09.03.2025. - URL: <https://habr.com/ru/articles/889324/> (дата обращения: 09.03.2025).

05.04.2025).

10. Настройки Apache Kafka в кластере Managed Service for Apache Kafka [Электронный ресурс] / Yandex Cloud. – 22.04.2025 – URL: <https://yandex.cloud/ru/docs/managed-kafka/concepts/settings-list> (дата обращения 07.05.2025).

ПРИЛОЖЕНИЕ А

Ссылки проекта

1. Организационные документы:

- Хранилище проекта:
drive.google.com/drive/folders/1euiqGYv10TctHXEwgiXeyBH3qBDXNbsd?usp=drive_link
- Техническое задание: docs.google.com/document/d/15K2WWnfnEaw-wfJYn5JtzdaZcDn19X_kw3pwm10s9dM/edit?usp=drive_link

- Обзор аналогов:
docs.google.com/spreadsheets/d/1oPthclbYh8sCiWMJSHNqzo_6Fht57sfJe2MpMR1LnPo/edit?usp=drive_link

2. Управление проектом:

- YouGile (трекер задач): ru.yougile.com/board/kwbb38ens4xj
- MindMap (архитектура):
excalidraw.com/#room=ad7d291322af899031d2,vTaLzm-Y05-Tv2POPnWqDQ

3. Исходный код:

- Репозиторий скрапера: github.com/ibatuloff/newscrapper
- Репозиторий бэкенда: github.com/mesler1x/ScraperApi
- Репозиторий ML-модели: github.com/habeeep/summary_model

4. Демо-стенды:

- API (Swagger): 158.160.112.139:8080/swagger-ui/index.html#/
- Grafana (дашборды): 158.160.112.139:3000
- Клиент: 158.160.112.139/

Примечание: в случае ограниченного доступа источников, запросите доступ по контакту: t.me/ilyaglbv