

Кейсы AI-разработки

Здесь собраны реализованные системы: что требовалось сделать, как устроено решение и какой результат получил заказчик.

Все кейсы

ИИ-подборщик по каталогу для 1С-Битрикс

Модуль уточняет запрос посетителя, ищет позиции только в разрешённых полях каталога и показывает рекомендации со ссылками на карточки товаров.

AlforCRM: заказная разработка для CRM и сайта

Июль 2026

PHP

JavaScript

1С-Битрикс

OpenAI-compatible API

КЕЙС · AIFORCRM · 1С-БИТРИКС

ИИ-подборщик по каталогу

Уточняет запрос и находит позиции

Подборщик по каталогу

• на сайте

Нужен насос для скважины 35 м

Уточните производительность и диаметр трубы.

Модель 4SD · 3 м³/ч

Модель 3SD · 2 м³/ч

Каталог · свойства · ссылки на товары

Задача

В каталоге 1С-Битрикс могут быть тысячи товаров, торговых предложений и характеристик. Покупатель описывает применение, размер, материал или ограничение, но часто не знает точный артикул. Обычный поиск требует совпадения с названием. Нужен был ИИ-подборщик, который уточняет запрос, обращается только к разрешённым данным каталога и показывает результат со ссылками на карточки товаров.

КЕЙС · AIFORCRM · 1С-БИТРИКС

ИИ-подборщик для каталога

Понимает запрос покупателя, уточняет параметры и находит подходящие позиции в разрешённых данных сайта.

Инфоблок каталога

Ссылки на товары

Работа без записи данных

Подборщик по каталогу

• на сайте

Нужен насос для скважины 35 м

Уточните производительность и диаметр трубы.

Модель 4SD

3 м³/ч - карточка товара

Модель 3SD

2 м³/ч - карточка товара

Поиск товара по запросу обычными словами

Решение

Модуль внутри 1С-Битрикс

Мы разработали серверный модуль и встроенный виджет. Перед запуском администратор выбирает инфоблок и поля, которые можно использовать. Во время разговора подборщик собирает недостающие параметры, выполняет поиск по каталогу и формирует ответ из найденных позиций.

ПОЧЕМУ ПОИСК ПО КАТАЛОГУ ТРЕБУЕТ ДИАЛОГА

Покупатель знает задачу, но не знает артикул

Тысячи похожих позиций

Названия отличаются одной характеристикой, а нужный параметр спрятан в свойствах товара или торгового предложения.

Запрос написан обычными словами

Клиент описывает применение, размер, материал или ограничение, а не точное название из каталога.

Не все поля можно показывать

До запуска согласовываются инфоблок, свойства, цены и остатки, которые разрешено использовать в ответе.

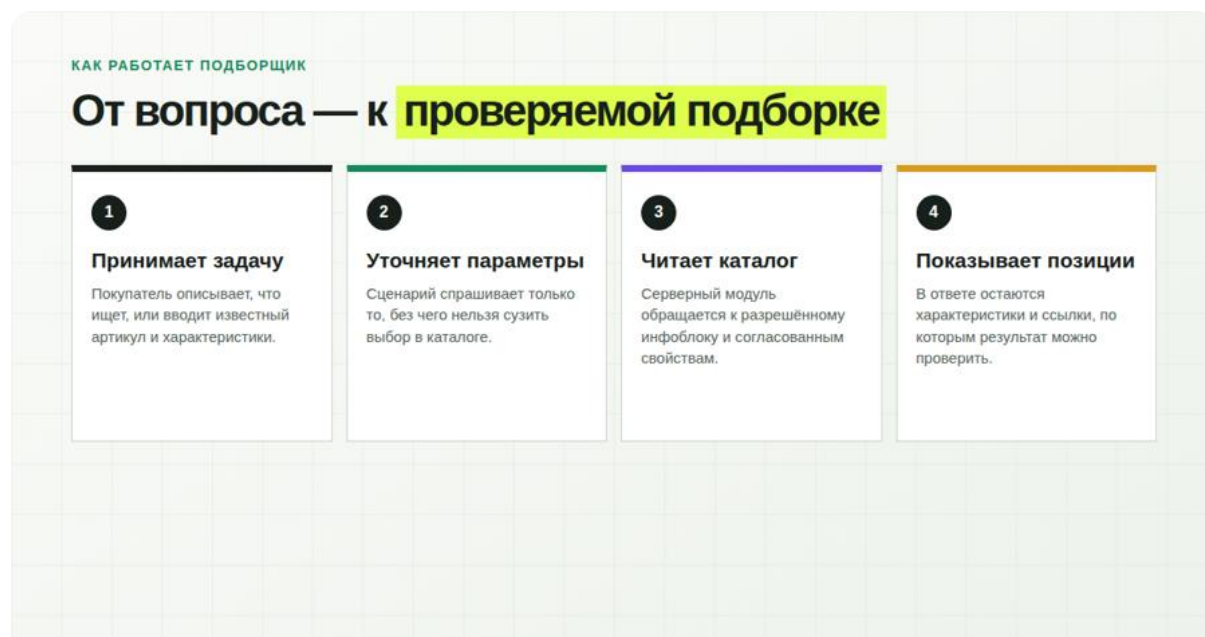
Ответ должен проверяться

Каждая рекомендация ведёт на карточку товара или другой источник, откуда получены характеристики.

Инфоблок и поля задаёт администратор

Уточнения по структуре каталога

Один параметр может находиться в свойствах товара, другой — в торговом предложении, третий — в описании. Подборщик задаёт короткие вопросы и переводит ответы пользователя в набор условий для поиска.



Система собирает недостающие характеристики

Проверяемая цепочка ответа

Серверный модуль читает разрешённые поля и передаёт модели найденные данные. В итог попадают только позиции из результата поиска, их характеристики и ссылки. Ключ доступа к модели остаётся на сервере.

Результат

Подборка на странице магазина

Посетитель пишет запрос в виджете, отвечает на уточнения и открывает подходящие карточки товаров. Ему не приходится угадывать терминологию продавца или вручную перебирать фильтры.

[Открыть проект](#)[Страница продукта](#)

Защищённая AI-платформа для банкротства физлиц

Платформа связывает заявку, оплату, документы, работу юриста и Bitrix24, а клиентские файлы хранит за авторизованным API с ролевым доступом.

AI-платформа для автоматизации банкротства физлиц

Юридическая компания по банкротству физических лиц

Июль 2026

Node.js

JavaScript

AI/OCR

S3-compatible storage

Bitrix24 REST API

PM2

Nginx

ДОКУМЕНТЫ ДЛЯ ДЕЛА

OCR и проверка реквизитов

ПАСПОРТ

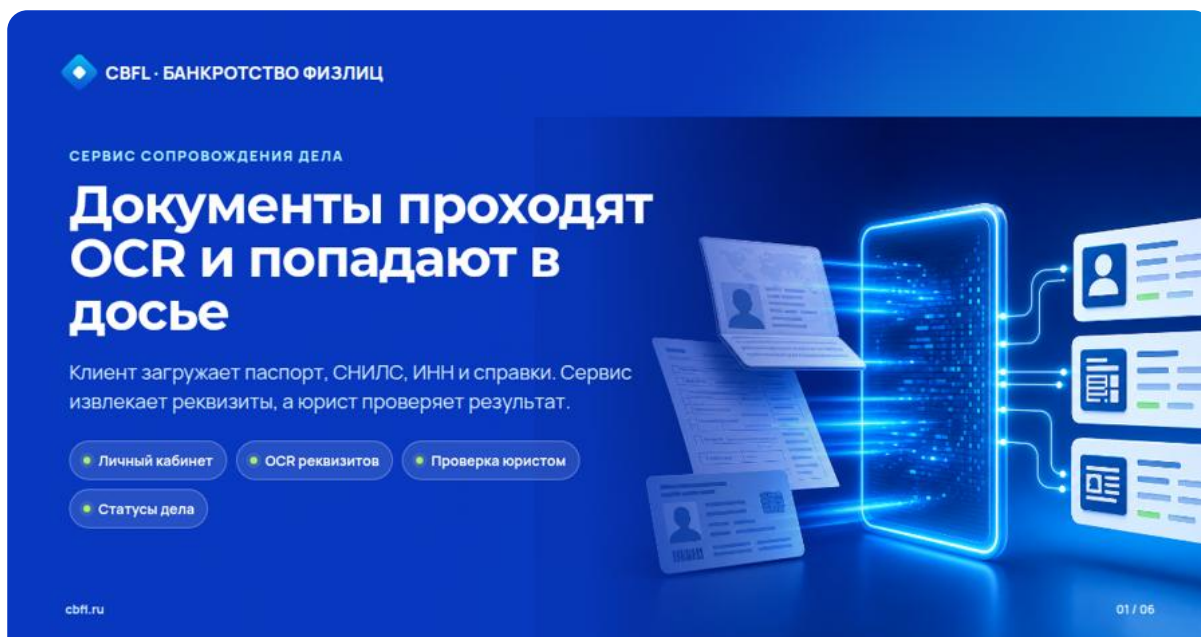
СНИЛС

ИНН

СПРАВКИ

Задача

Юридической компании требовалось связать путь клиента по банкротству физических лиц с внутренней работой юристов. Заявка приходила с сайта, документы — через разные каналы, а движение дела отдельно фиксировалось в Bitrix24. Нужно было собрать анкету, оплату, файлы, экспертизу и статусы в одной системе, не открывая чувствительные документы через публичный сайт.

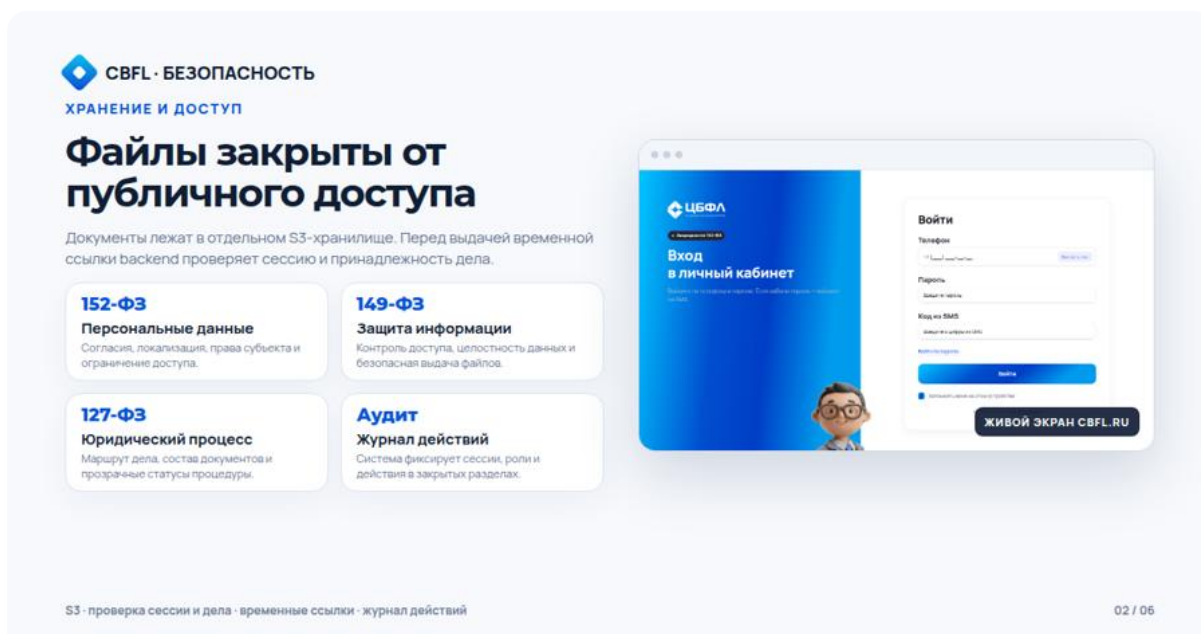


Публичный сайт, кабинеты и закрытое хранилище

Решение

Разделение публичной и закрытой частей

Платформу разделили на публичный сайт, кабинеты и закрытое хранилище. AI-ассистент проводит первичный разбор ситуации, OCR извлекает данные из документов, а рабочие события синхронизируются с Bitrix24.



Требования к персональным данным учтены в архитектуре

Файлы доступны только участникам дела

Анкеты, сканы и заключения проходят через авторизованный API и закрытое S3-совместимое хранилище. Даже корректная ссылка не открывает файл без действующей сессии, проверки роли и принадлежности конкретному делу.

Требования к персональным данным

При проектировании учитывались требования 152-ФЗ, 149-ФЗ и 127-ФЗ. Журнал действий фиксирует работу с материалами дела, а раздельное хранение помогает контролировать доступ к документам.

 CBFL · ДОКУМЕНТЫ И OCR

ОТ ФАЙЛА К РАБОЧЕМУ ДОСЬЕ

OCR извлекает реквизиты и собирает досье

1 **Загрузка в дело**
Паспорт, СНИЛС, ИНН, справки и договоры сохраняются в карточке клиента.

2 **OCR и реквизиты**
Фоновая очередь распознаёт текст и ключевые поля, пока клиент работает в кабинете.

3 **Проверка юристом**
Юрист сверяет поля и сразу видит, каких документов не хватает.



Паспорт / СНИЛС / ИНН OCR Досье юриста

Оригинал в S3 · распознанные поля · статус проверки · список недостающих документов

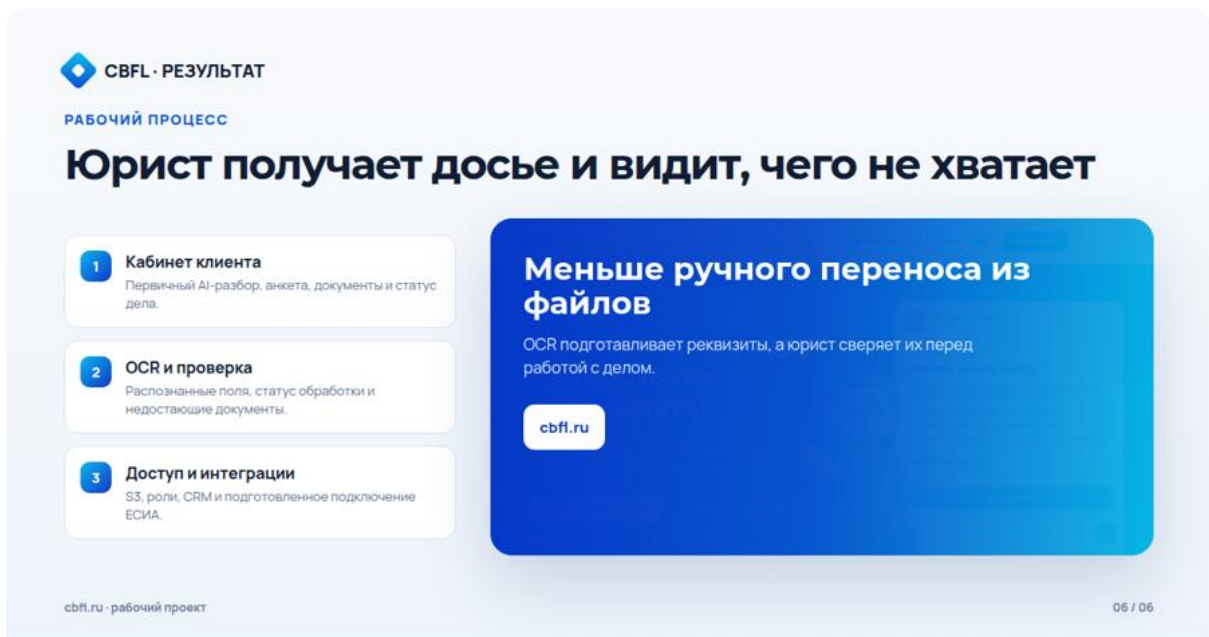
03 / 06

OCR извлекает данные из загруженных документов

Результат

Единое досье вместо ручного переноса

Заявка превращается в юридическое дело без повторного переноса между сайтом и CRM. Клиент видит текущий этап, список документов и следующий шаг. Юрист получает досье, очередь задач и статусы проверки.



Единый путь клиента и юридической команды

[Открыть проект](#)

[Страница продукта](#)

Корпоративная AI-академия по документам компании

Сотрудник задаёт рабочий вопрос и получает ответ по доступным ему документам с названием источника и номером страницы.

Корпоративный AI-университет по базе знаний компании

Июль 2026

Python

Flask

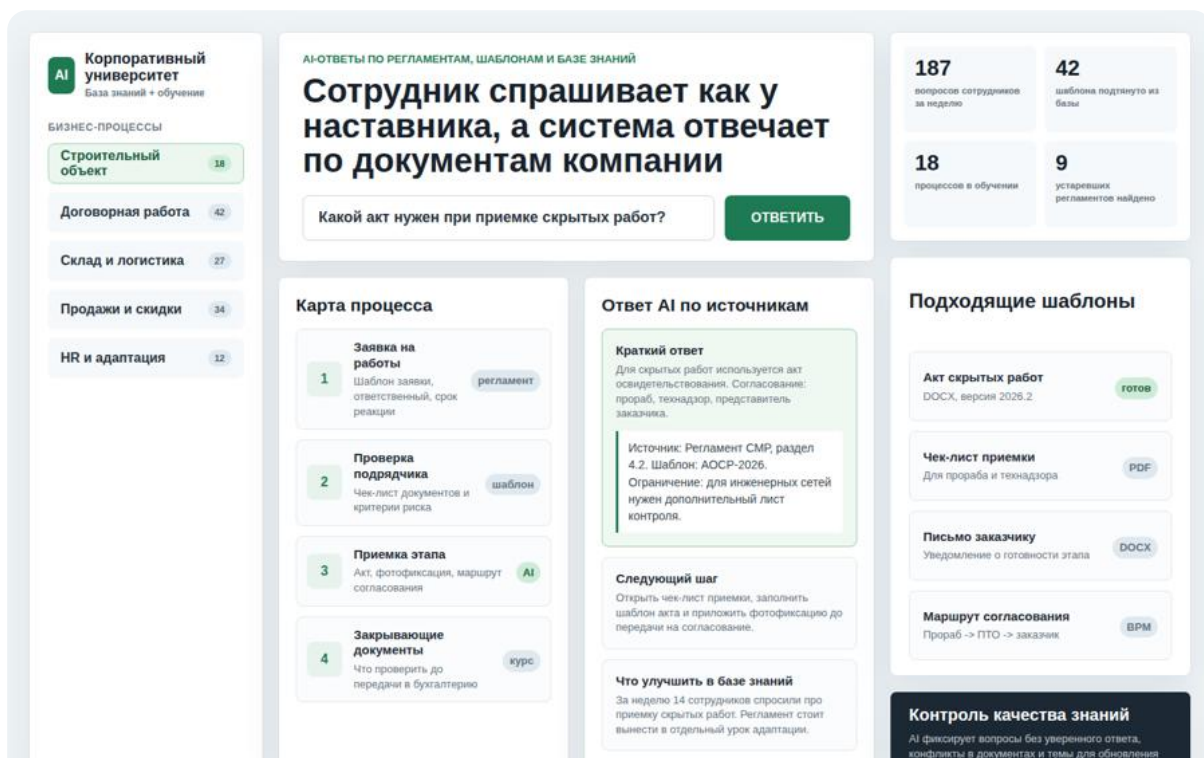
PostgreSQL

pgvector

OCR

RAG

LLM API



Задача

В компании уже были регламенты, инструкции и шаблоны, но новый сотрудник всё равно обращался к наставнику. Документы лежали в разных папках, нужный пункт приходилось долго искать, а обычный поиск не учитывал роль и отдел. Требовалась база знаний, которая отвечает по утверждённым документам, показывает источник и поддерживает обучение сотрудников.

Как проходит процесс по претензионной работе если выбран вид претензии Боя продукции

Процесс претензионной работы при выборе вида претензии «Бой продукции» описан в схеме процесса [S1].

1. **Начало процесса:** Процесс инициируется событием «Поступила претензия от КиБ» [S1].
2. **Определение основания:** Происходит развилка, где выбирается основание претензии. Для «Бой продукции» выбирается ветка с подписью «Бой продукции» [S1].
3. **Действие:** После выбора основания запускается задача «Высказать мнение по претензии» [S1].
4. **Согласование:** После выполнения задачи происходит слияние потоков в параллельном шлюзе, после чего следует задача «Принять итоговое решение» [S1].
5. **Документирование:** После принятия решения выполняется задача «Отразить ЦВЗ, заполнив справочник №8» [S1].
6. **Завершение:** Процесс завершается событием «Претензия отражена в учёте» [S1].

Также в схеме указаны альтернативные пути для других оснований (пересорт, потеря товарного вида), которые не применяются в данном случае [S1].

S1: Претензионная работа с КиБ. · стр. 1

↓ страница

👍 Полезно

👎 Проблема

Ответ сохраняет документ и страницу для проверки

Решение

Знания организованы по ролям

Материалы распределены по процессам и должностям, а доступ учитывает подразделение пользователя. Один портал объединяет инструкции, учебные модули и AI-поиск.

Четыре шага от документа к ответу

Администратор загружает регламенты и назначает права. Сотрудник задаёт вопрос, система находит фрагмент и показывает ответ со ссылкой на документ. Если подтверждения в материалах нет, система сообщает об этом.

Источник виден в карточке

В ответе указаны использованный документ, номер страницы и короткий фрагмент. Пользователь оценивает полезность, а спорный случай попадает администратору на проверку.

Результат

Поиск инструкции занимает меньше действий

Сотруднику не нужно помнить папку и точное название файла. Академия показывает, какие темы вызывают вопросы, и помогает руководителю обновлять учебный маршрут и документы по фактическим обращениям.

Материалы и вопросы в одной админке

Администратор добавляет документы, обновляет базу знаний, назначает доступ и просматривает журнал вопросов. Пробел можно закрыть новой инструкцией или практическим заданием.

Пилот с одного отдела

Для первой версии достаточно регламентов одного подразделения и списка частых вопросов. На них проверяются качество поиска, ссылки на источники и права доступа, после чего подключаются следующие роли.

Развёртывание внутри компании

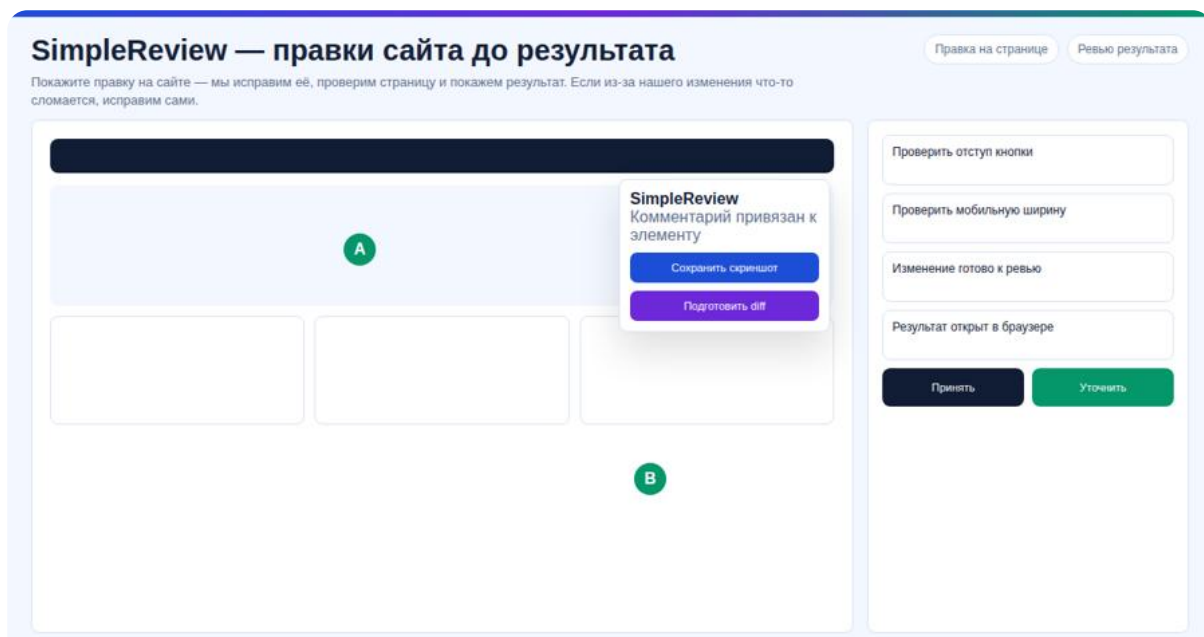
Если документы нельзя передавать внешним сервисам, решение размещается на серверах компании. К базе знаний можно добавлять тесты, задания и программы адаптации.

[Открыть проект](#)[Открыть презентацию K1](#)[Страница продукта](#)

SimpleReview: правки сайта до результата

Клиент отмечает элемент прямо на странице, а разработчик получает адрес, описание правки и данные выбранного блока в одной задаче.

[SimpleReview — правки сайта до результата](#)[Собственный продукт SimpleReview](#)[Май 2026](#)[Chrome Extensions API](#)[JavaScript](#)[Claude Code API](#)[GitHub API](#)



Задача

Небольшая правка сайта часто начиналась с длинного сообщения: на какой странице смотреть, какой блок имеется в виду, что именно не так и как ошибка выглядит на телефоне. Пока разработчик уточнял детали, клиент делал новые скриншоты, а часть замечаний терялась в переписке. Мы хотели сократить путь до одного действия: показать элемент на странице и сразу передать понятную задачу.

Решение

Правка привязана к элементу

SimpleReview работает в браузере поверх открытого сайта. Клиент или тестировщик выбирает текст, кнопку, изображение либо целый блок и описывает желаемое изменение.

Контекст сохраняется вместе с комментарием

В задачу попадают адрес страницы и данные выбранного элемента. Разработчик открывает ту же страницу, видит область работы и проверяет изменение в исходном интерфейсе.

От ссылки до проверки

Клиент присылает страницу и отмечает проблему. Мы уточняем только детали, влияющие на результат, вносим изменение и проверяем его. История замечания остаётся рядом с задачей.

Результат

Одна точка проверки

Вместо общей фразы о мобильной версии команда получает конкретный элемент и ожидаемое изменение. После работы страницу можно открыть на компьютере и телефоне и пройти тот же сценарий глазами пользователя.

Проверку берёт на себя исполнитель

От заказчика нужны ссылка, пожелание и ответы на вопросы по смыслу. Поиск места в коде и проверка результата остаются на стороне команды разработки.

Старт с одного замечания

Для начала достаточно ссылки и одной правки. На небольшом примере видно, как фиксируются детали и как принимается результат. Расширение опубликовано в Chrome Web Store.

[Открыть проект](#)

[Страница продукта](#)

Vagon: AI-аналитик для баз данных и Excel

Сотрудник задаёт вопрос на русском языке, видит сгенерированный SQL и получает проверяемую таблицу или график по разрешённым данным.

Vagon - RAG SQL-генератор с AI для морского порта

Морской порт, операционный отдел из 15 человек

Март 2026

Python

Flask

MySQL

Microsoft SQL Server

OpenAI API

SQLAlchemy

Plotly

Telegram Bot API

Примерно 30 секунд вместо двух часов ожидания ручной выгрузки

70% по данным проекта



AI-АНАЛИТИК ДАННЫХ

Вопрос → SQL → ГОТОВЫЙ ОТЧЁТ

Спросить

обычным языком



Проверить

SQL под схему



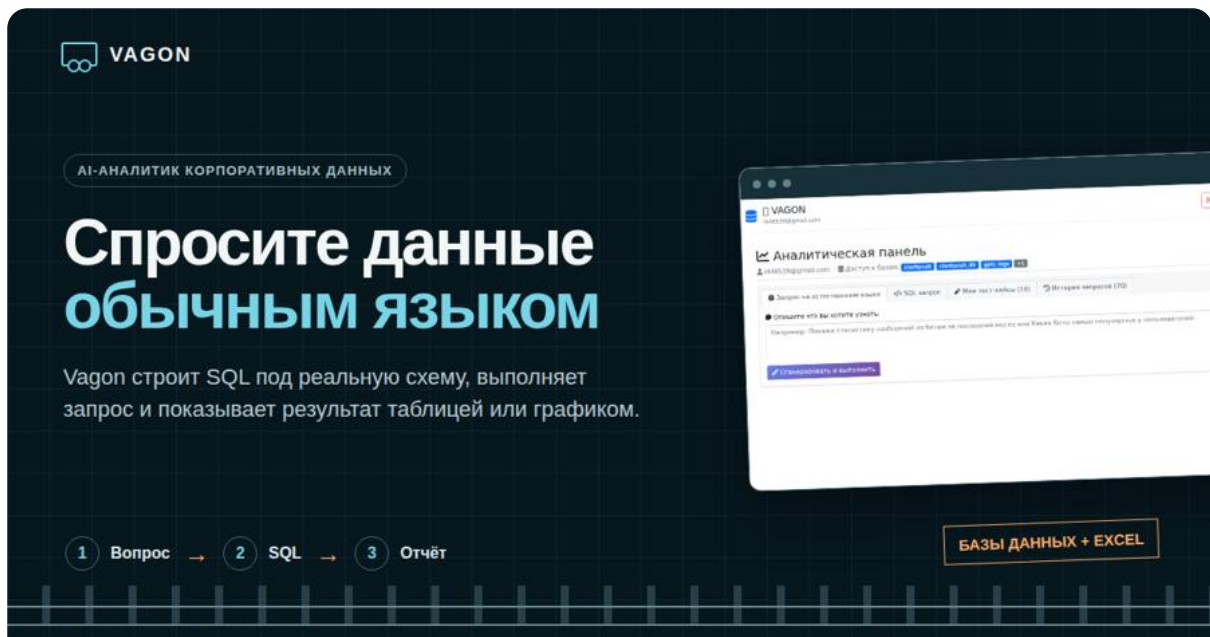
Увидеть

таблицу или график

БАЗЫ ДАННЫХ + EXCEL

Задача

В операционном отделе морского порта ответы на обычные вопросы зависели от программиста: сотрудник формулировал запрос, ждал ручную выгрузку и затем разбирал таблицу. Изучать SQL всей команде было бессмысленно, а прямой доступ к производственной базе создавал риск. Нужен был ИИ-аналитик, который понимает вопрос на русском, знает схему разрешённых данных и показывает использованный запрос.



Доступ к данным без очереди на ручную выгрузку

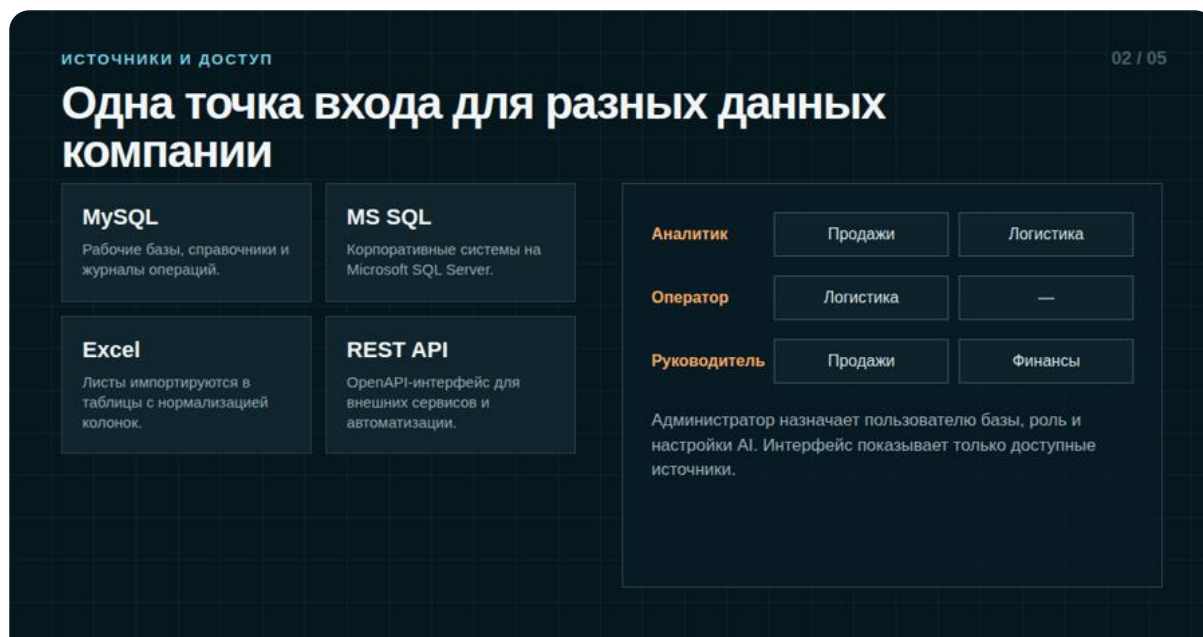
Решение

Вопрос на языке сотрудника

Пользователь пишет, какие данные хочет увидеть. Система переводит вопрос в SQL по реальной схеме компании, выполняет его в рамках разрешённого доступа и возвращает результат.

SQL остаётся на экране

Vagon сначала читает доступную схему, затем строит запрос и показывает его рядом с таблицей или графиком. Специалист может проверить расчёт до включения результата в рабочий отчёт.



Один вход для разных источников

История воспроизводимых запросов

В истории сохраняются вопрос, сгенерированный SQL и результат. Если формулировку нужно уточнить, пользователь видит предыдущую попытку и может повторить проверенный запрос.

Результат

Типовой отчёт примерно за 30 секунд

После запуска типовой отчёт собирается примерно за 30 секунд вместо двух часов ожидания ручной выгрузки. По данным проекта, нагрузка на IT-отдел снизилась на 70%.

[Открыть проект](#)[Страница продукта](#)

ClarityCult: AI-кабинет для бизнес-стратегии

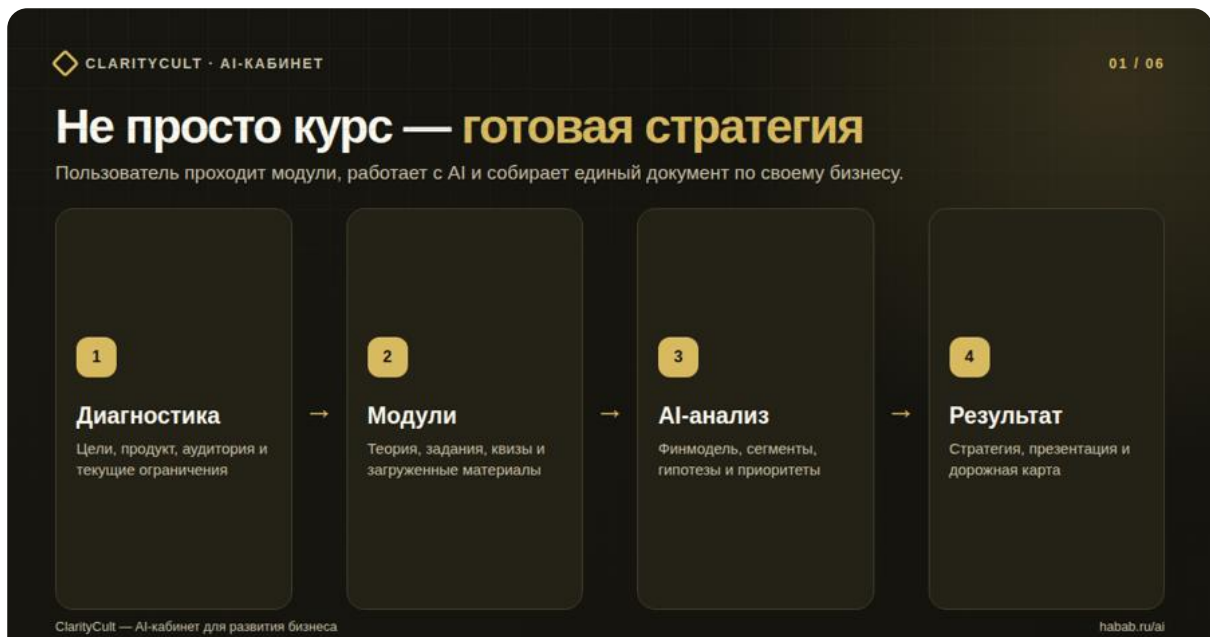
Личный кабинет проводит пользователя от вводных о компании к финансовой модели, приоритетам и готовому стратегическому документу.

[Обучающий кабинет](#)[ClarityCult](#)[Июль 2026](#)[Python](#)[Flask](#)[PostgreSQL](#)[Docker](#)[ChatGPT](#)



Задача

У автора была отработанная методология разработки бизнес-стратегии, но работа по ней распадалась на анкеты, таблицы, презентации и созвоны. Пользователю было трудно удержать связь между исходными ответами, выводами и планом действий. Требовалось перенести методологию в личный кабинет и провести человека от вводных о компании до финансовой модели, приоритетов и готового документа.



Последовательный маршрут по методологии

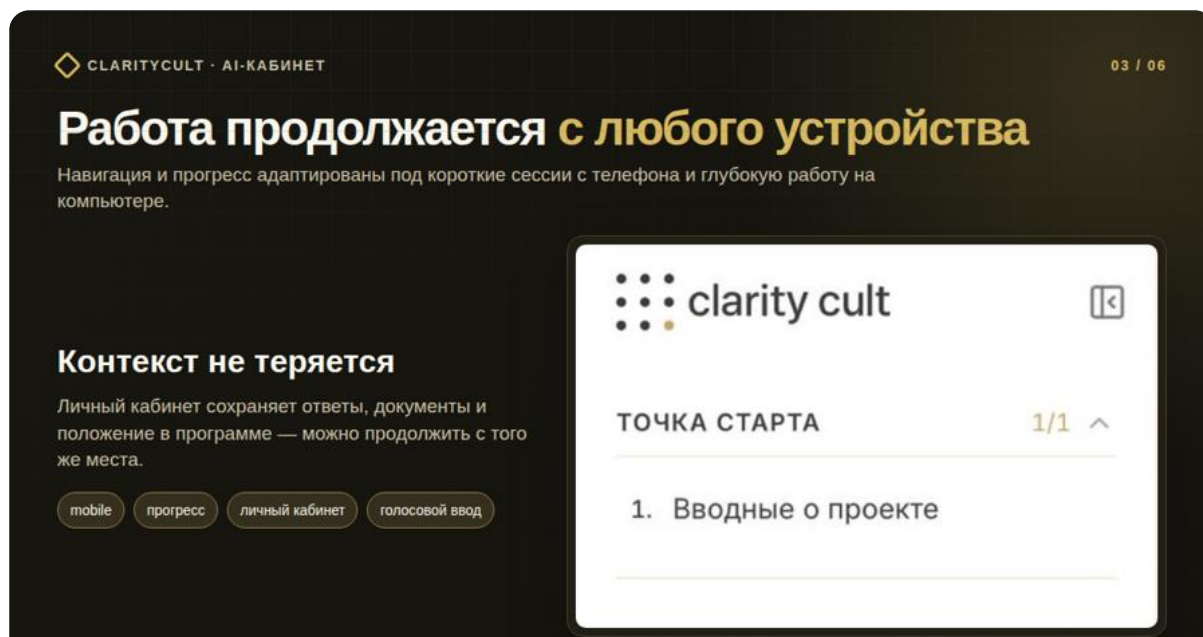
Решение

Последовательный маршрут

Методологию разложили на модули, каждый из которых получает данные из предыдущих шагов. AI-ассистент работает с контекстом конкретной компании, а кабинет сохраняет прогресс и промежуточные результаты.

От диагностики к общей картине

Пользователь описывает продукт, рынок, аудиторию и ограничения, проходит тематические модули и получает анализ своих ответов. На экране видно завершённые этапы и связь каждого вывода с будущей стратегией.



Прогресс сохраняется между устройствами

Продолжение с любого устройства

Ответы, документы и положение в программе сохраняются между сессиями. Голосовой ввод помогает фиксировать длинные ответы с телефона.

Результат

Стратегия собрана из исходных данных

После заполнения модулей система объединяет материалы в связный документ. Пользователь видит происхождение каждого вывода и может вернуться к исходному ответу для уточнения.

[Открыть проект](#)[📖 Документация User Flow](#)[Страница продукта](#)

YouTube-Telegram скрапер для поиска лидов и контактов авторов

Система ищет YouTube-каналы по заданным критериям, извлекает открытые Telegram-контакты и сохраняет причину отбора каждого автора.

[Поиск YouTube-авторов с Telegram-контактами](#)[Стартап, предоставляющий услуги YouTube-блогерам](#)[Июнь 2026](#)[Python](#)[YouTube Data API v3](#)[Telegram](#)[CSV/XLSX/JSON export](#)

Поиск YouTube-авторов с Telegram-контактами

Система находит YouTube-каналы по заданным критериям, проверяет активность, извлекает открытые Telegram-ссылки и выгружает результат.

YouTube Data API

Публичные Telegram-ссылки

1

Запрос

Тематика

Ключевые слова и исключения

Аудитория

Диапазон подписчиков

Активность

Период последних публикаций

2

Проверка канала

Соответствие теме

Название, описание и публикации

Открытые контакты

Ссылки из публичного описания

Причина отбора

Условия, которые прошёл канал

3

Выгрузка

YouTube-канал

Ссылка и основные данные

Telegram

Найденная публичная ссылка

Передача в работу

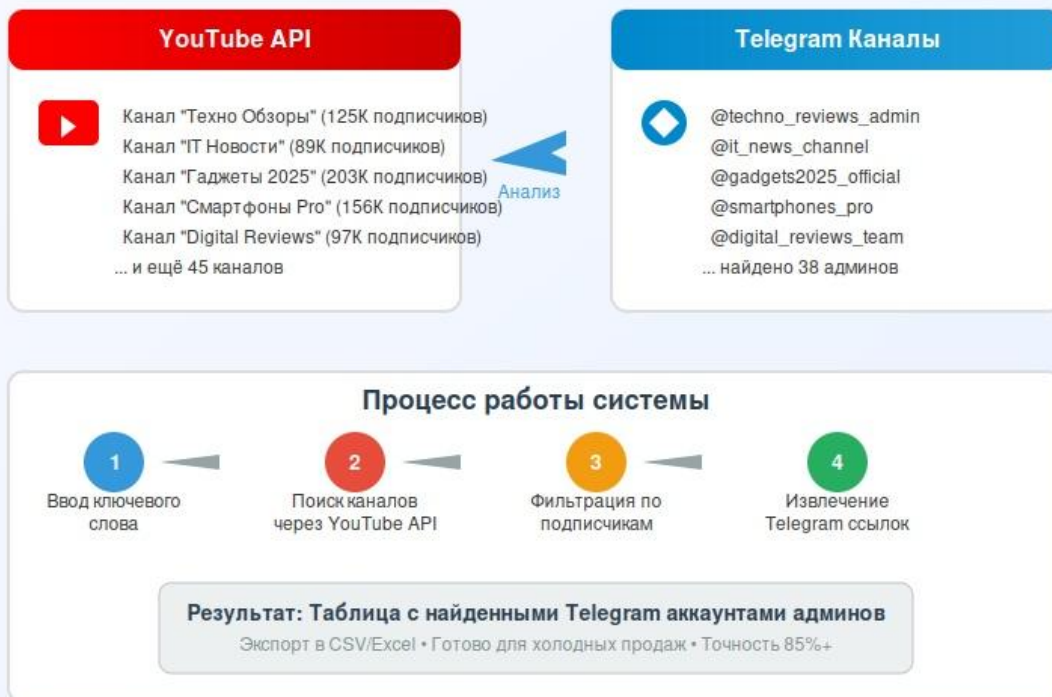
CSV, Excel, JSON или CRM

Система сохраняет только найденные публичные контакты и не подставляет выдуманные адреса.

Задача

Отделу продаж требовалось регулярно находить YouTube-авторов в заданной нише и проверять, оставили ли они публичный контакт для связи. Ручной поиск плохо повторялся: сотрудники по-разному оценивали свежесть публикаций, размер канала и соответствие теме. Нужно было превратить исследование в управляемую подборку с понятными фильтрами.

YouTube → Telegram Scraper



Параметры поиска и список запусков

Решение

Путь от запроса до строки в базе

Система получает каналы через YouTube Data API v3, проверяет тематику, аудиторию и активность, затем ищет открытые Telegram-ссылки в описаниях. Каждый этап сохраняет причину отбора.

Критерии задаются до запуска

Минимальное число подписчиков, активность канала, ключевые слова и исключения находятся в настройках. Менеджер может скорректировать правила до расходования квоты YouTube API.

Только публичные контакты

Telegram извлекается из открытого описания канала. Формат и доступность ссылки проверяются, дубликаты удаляются. Один контакт не повторяется из разных видео или вариантов адреса.

Результат

Карточка с контекстом

Отдел продаж получает название канала, тематику, показатели аудитории, найденный контакт и причину прохождения фильтров. Авторы можно проверить и распределить между менеджерами до начала переписки.

Объяснение для каждого лида

В карточке видно, откуда взят контакт и каким условиям соответствует канал. Менеджер открывает источник и самостоятельно принимает решение об общении.

Экспорт под рабочий процесс

CSV и Excel подходят для ручной проверки, JSON и API — для передачи в CRM или внутренний сервис. История запусков хранит параметры каждой подборки.

Пилот с одной ниши

Сначала фиксируются критерии и вручную оценивается небольшая выборка. После проверки качества поиска подключаются регулярные запуски.

youtube parser

	A	B	C	D	E	F	G	H	I	J
1	Название канала	Ссылка канала	Количество подписчиков	About	Телеграм	Дата добавления	% просмотр в 1 частота видео	Сред. продолит	Статус	
2	Активный трейдер	https://www.youtube.com/channel/UC10900	10900	Алексей трейдер - то, что находится венту, аналогично тому, что находится венту	https://t.me/alextrading , https://t.me/Roman_Kimenko	06.05.2025	0% из 10 видео	10 видео за ~10:35.4 мин	Добавлен	
3	Трейдер Анатолий Панас	https://www.youtube.com/channel/UC57900	57900	Профессиональный трейдер-инвестор Мой блог http://www.fantastictrade.ru/ Моя страница в телеграмме https://t.me/fantastictrade Этот канал посвящен инвестициям. Здесь вы можете научиться инвестировать, оос	https://t.me/fantastictrade	06.05.2025	0% из 10 видео	10 видео за ~06:40.7 мин	Добавлен	
4	Инвестиции	https://www.youtube.com/channel/UC14000	14000	Инвестиции в 100% в 100% - это не только про деньги, но и про знания, про опыт, про умение видеть возможности и риски	https://t.me/investorofficial	06.05.2025	0% из 10 видео	10 видео за ~73.13 мин	Добавлен	
5	Инвестиции, Экономика, Трейдинг - Кривцов	https://www.youtube.com/channel/UC134000	134000	Инвестиции и трейдинг на фондовом и срочном рынке являются моим единственным делом	https://t.me/ProfitDate , https://t.me/SubProfitClubBot , https://t.me/ProfitDate	06.05.2025	0% из 10 видео	10 видео за ~32.23.7 мин	Добавлен	
6	Профита нет: Инвестиции и экономика	https://www.youtube.com/channel/UC78000	78000	Авторский канал об инвестициях. Личное мнение. Канал более	https://t.me/profitanet	06.05.2025	0% из 10 видео	10 видео за ~06.18 мин	Добавлен	
7	Славян Инвестор	https://www.youtube.com/channel/UC19800	19800	Рассказываю о своем опыте инвестиций. Канал про заработок, криптовалюты, бп	https://t.me/slav53	06.05.2025	0% из 10 видео	10 видео за ~29.36.1 мин	Добавлен	
8	Будет	https://www.youtube.com/channel/UC84000	84000	По вопросам сотрудничества: budetinvestor@gmail.com Помогите мне стать блогером Муста и у меня огромный опыт торговли на фондовом	https://t.me/budet99 , https://t.me/depositinvesting	06.05.2025	0% из 10 видео	10 видео за ~20.19.8 мин	Добавлен	
9	THE INSIDER - инвестиции, экономика, аналитика	https://www.youtube.com/channel/UC12000	12000	Рынок, инвестор, как выбрать, в какие компании инвестировать, как заработать на фондовом	https://t.me/traderinsider_official	06.05.2025	0% из 10 видео	10 видео за ~04.24.8 мин	Добавлен	
10	Ильмир Мурта - инвестиции	https://www.youtube.com/channel/UC13800	13800	Привет! Меня зовут Ильмир Мурта и у меня огромный опыт торговли на фондовом	https://t.me/ilmirinvest , https://t.me/metodmurovabot	06.05.2025	0% из 10 видео	10 видео за ~19:24 мин	Добавлен	
11	Алексей-IT	https://www.youtube.com/channel/UC180000	180000	Канал о криптовалютах! Подписывайтесь, будет интересно!	https://t.me/booytimen , https://t.me/CryptoADVnet	06.05.2025	0% из 8 видео	8 видео за ~145:6.1 мин	Добавлен	

Фильтрация каналов и подготовка базы

Открыть проект

Страница продукта

Корпоративная система анализа договоров с ИИ

Система принимает договор из 1С или СЭД, распознаёт вложение, проверяет условия по правилам компании и возвращает заключение в тот же маршрут.

Разработка системы анализа договоров с ИИ

Июнь 2026

Python

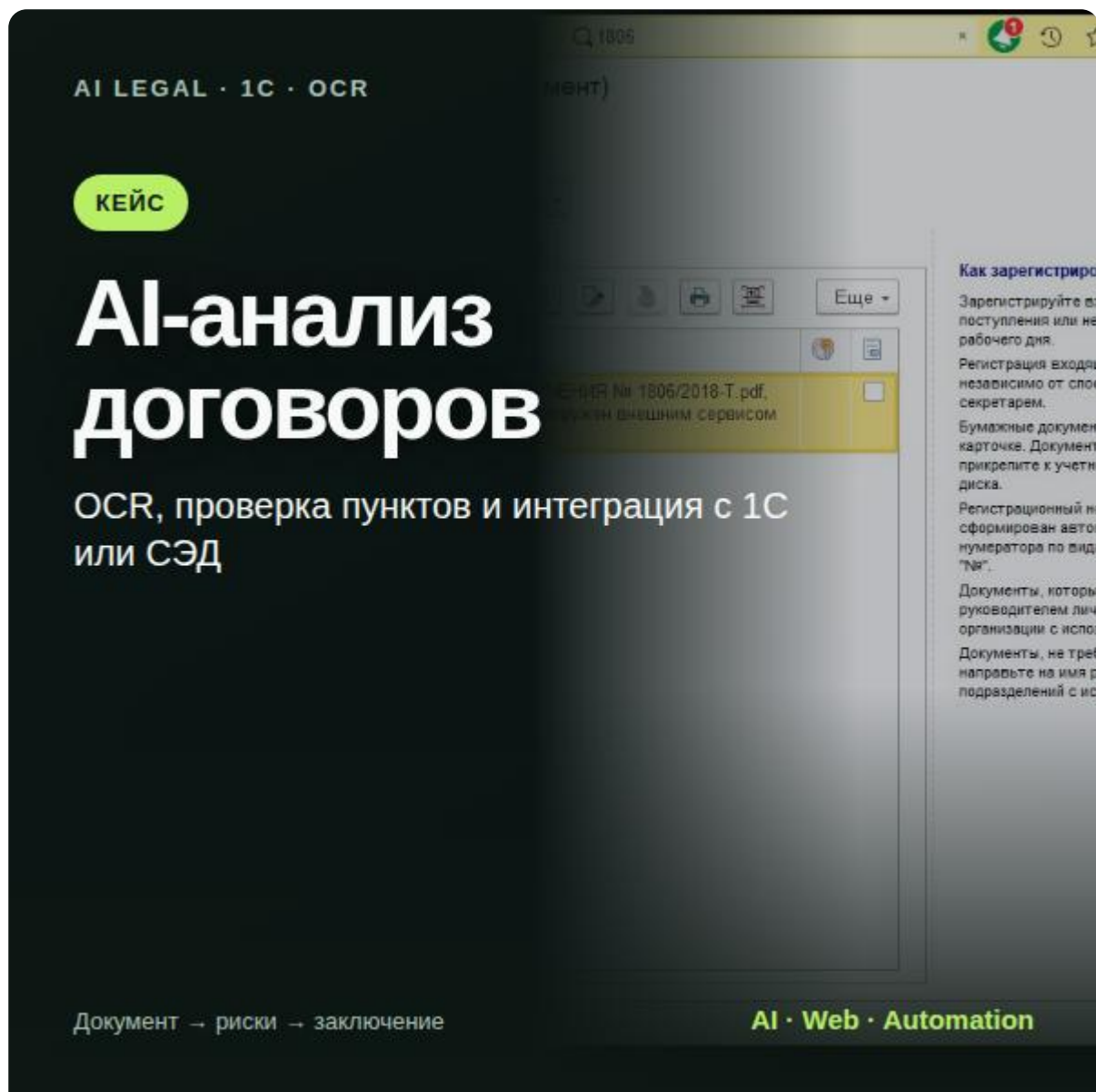
Flask

OCR

LLM API

1С

REST API



Задача

В корпоративном документообороте договор проходил согласование, но первичная проверка оставалась ручной. Юрист открывал файл из 1С или СЭД, восстанавливал текст из скана, искал отклонения от шаблона и заносил заключение обратно. Требовалась система, которая ускоряет повторяемую часть работы и сохраняет окончательное решение за специалистом.

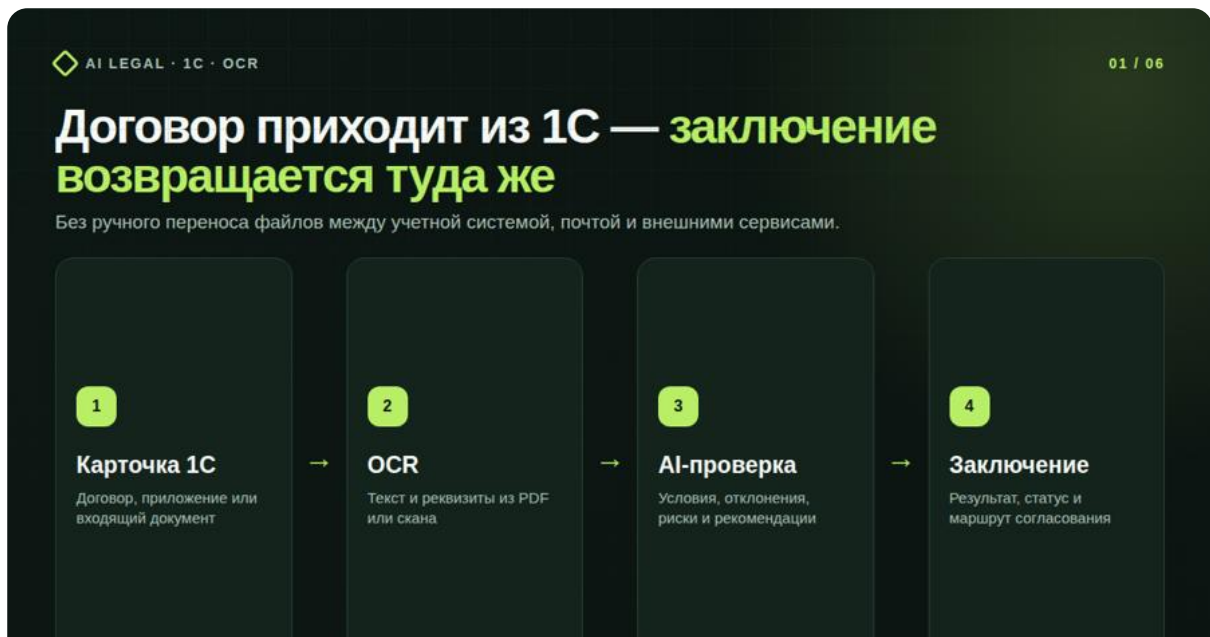


Схема обработки документа

Решение

Проверка внутри существующего маршрута

Система принимает PDF, DOC, DOCX и сканы из 1С, СЭД или API. OCR извлекает текст и реквизиты, после чего условия сравниваются с чек-листами и эталонными формулировками компании.

Договор возвращается в исходную систему

Результат прикрепляется к карточке документа вместе с объяснением рисков. Сотрудники продолжают работать в знакомом маршруте согласования, без обязательного перехода в отдельный кабинет.

Юрист видит риск по каждому пункту

Не общий пересказ нейросети, а объяснение проблемы и предложение, что проверить или изменить.

Результат пригоден для работы

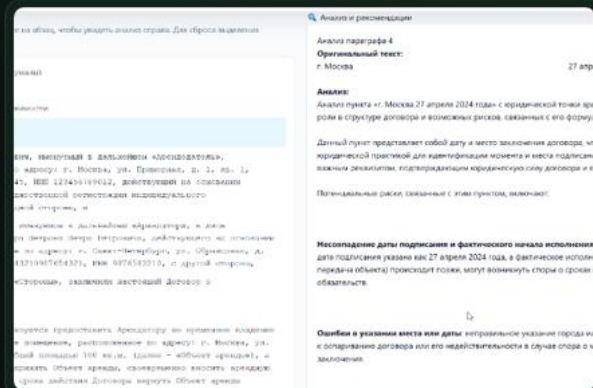
Пункты договора связываются с уровнем риска, аргументацией и рекомендацией по правке.

правовой риск

финансовый риск

обязательства

рекомендации



Позаэлементный разбор условий договора

habab.ru/ai

Каждое замечание связано с пунктом договора

Замечание связано с конкретным пунктом

Юрист видит условие, уровень риска, объяснение и рекомендацию. Из карточки можно перейти к спорному фрагменту и принять решение по нему.

Результат

Проверяемая история решения

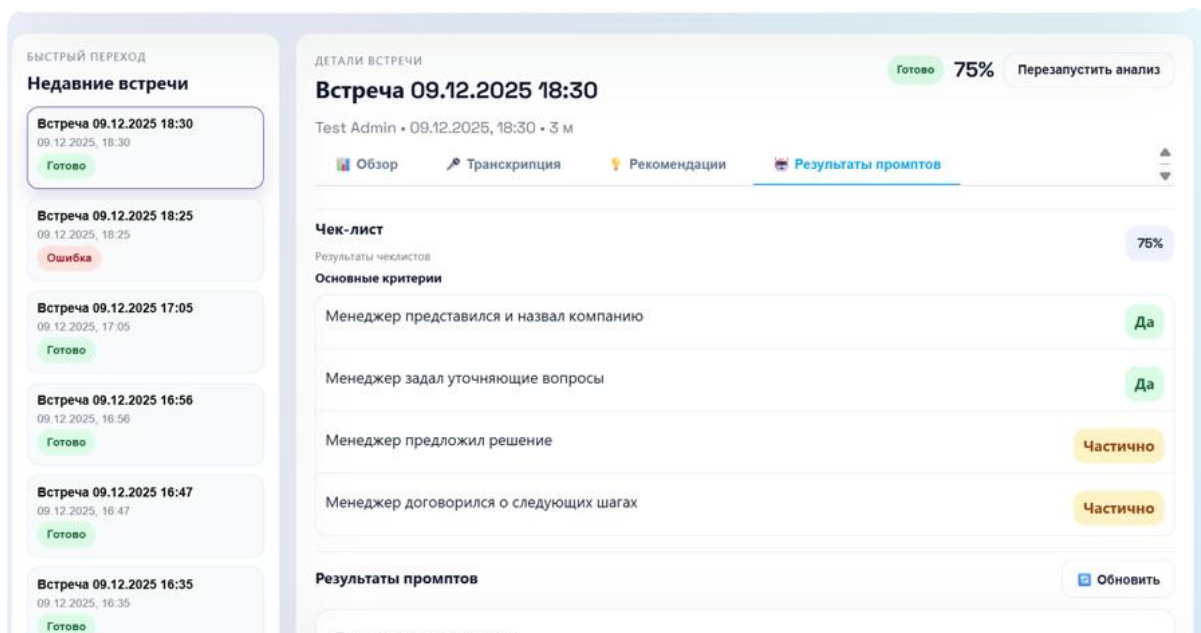
Исходник, распознанные данные, выводы системы и решение юриста хранятся вместе. История показывает, кто принял решение и какие данные были ему доступны.

[Открыть проект](#)[Страница продукта](#)

Платформа AI-анализа встреч и звонков для отдела продаж

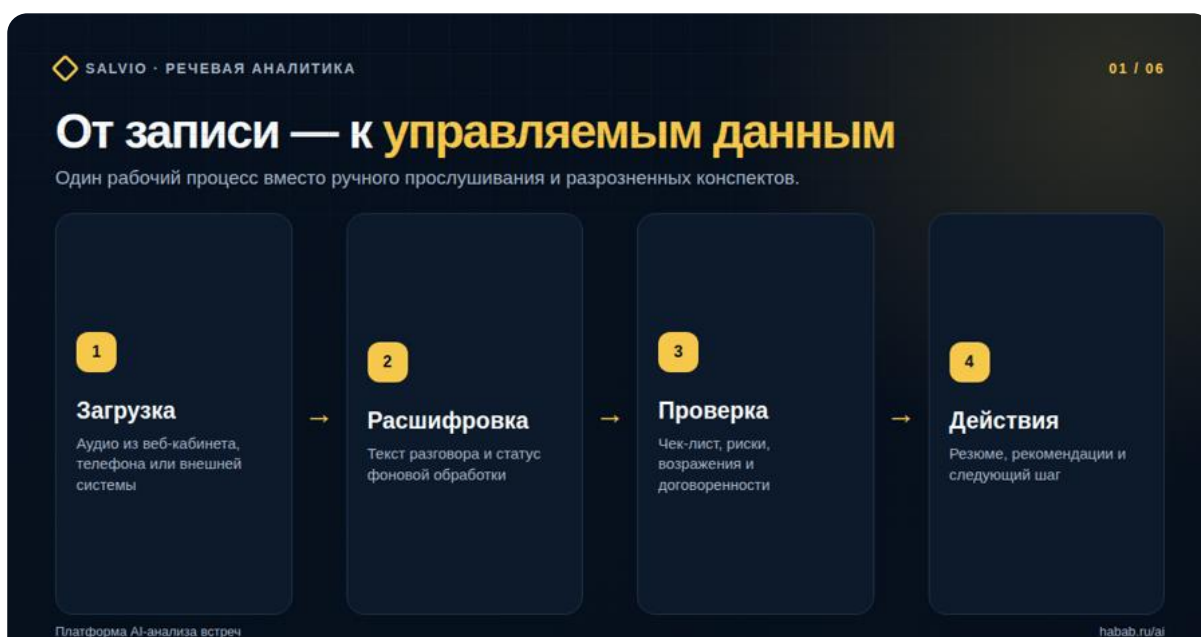
Платформа расшифровывает запись, проверяет разговор по единому чек-листу и связывает выводы с конкретными репликами и исходным аудио.

[Платформа AI-анализа встреч](#)[Июнь 2026](#)[AssemblyAI](#)[Google Speech-to-Text](#)[OpenAI](#)[Claude](#)[RabbitMQ](#)[PostgreSQL](#)[S3-compatible storage](#)



Задача

Руководитель отдела продаж мог проверить несколько звонков вручную, но не видел картину по всей команде. Записи лежали в разных системах, выводы зависели от личного впечатления, а обратная связь менеджеру приходила без ссылки на конкретный момент разговора. Нужна была речевая аналитика с единым чек-листом и доступным для проверки исходным диалогом.

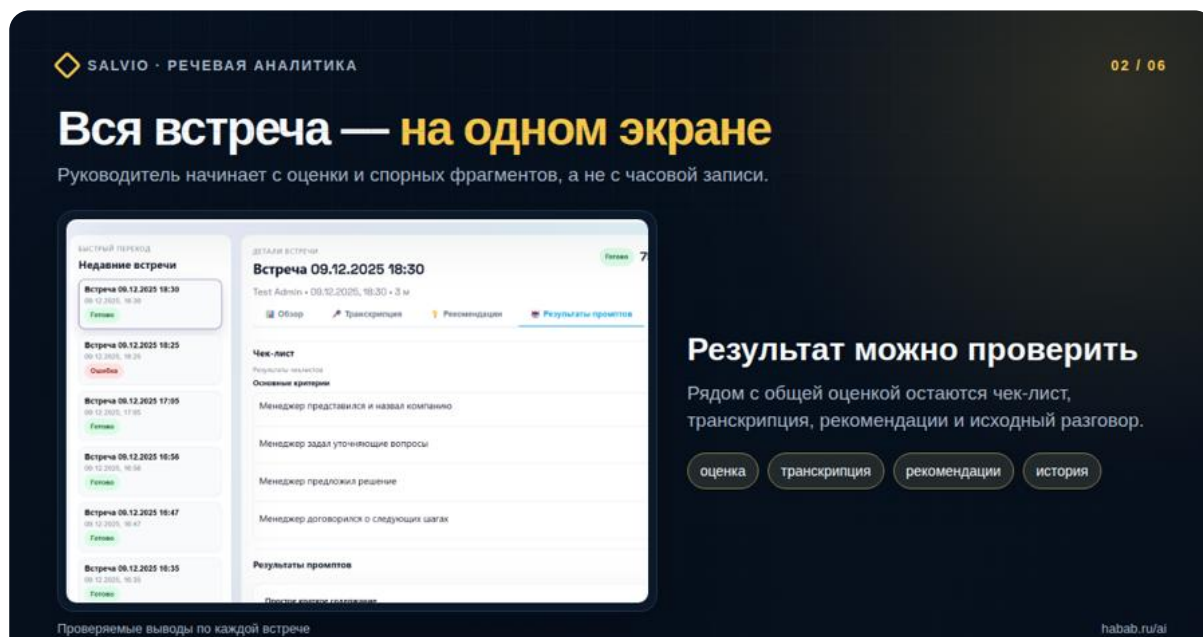


Этапы фоновой обработки встречи

Решение

Фоновая обработка записи

Файл загружается в кабинет или приходит из интеграции. Очередь запускает транскрибацию, определение участников и анализ по выбранному сценарию. Каждый этап имеет отдельный статус.



Аудио, транскрипция и выводы на одном экране

Встреча на одном экране

Рядом находятся аудио, текст, оценки, спорные фрагменты и рекомендации. Руководитель начинает с общей оценки и при необходимости переходит к реплике или исходной записи.

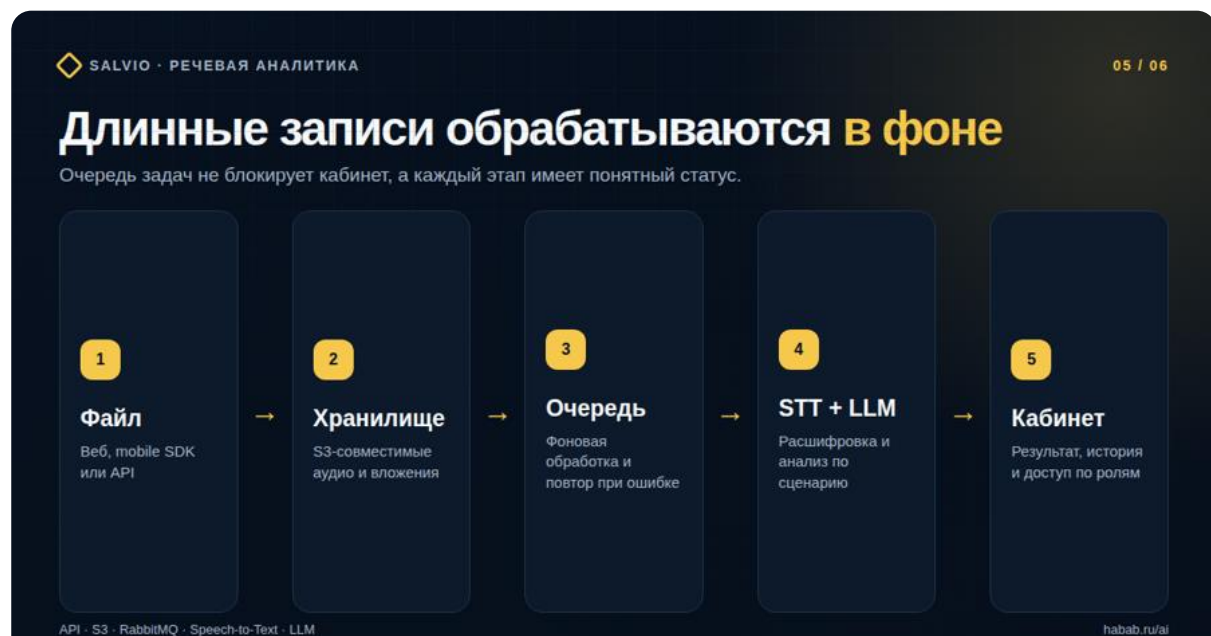
Настраиваемый чек-лист

Критерии задаются отдельно для продаж, поддержки, HR-интервью и CustDev. Система отмечает выполнение каждого пункта и приводит подтверждающий фрагмент разговора.

Продажи оценивают структуру и договорённости, поддержка — полноту ответа и тон, HR — содержание интервью. Роли и доступ к записям настраиваются отдельно.

Длинные записи обрабатываются в очереди

Файл хранится в закрытом хранилище, кабинет показывает прогресс и сохраняет задачу при обновлении страницы. Результат доступен только участникам с нужной ролью.



Очередь, распознавание речи и LLM-анализ

Пилот на согласованном чек-листе

Первый запуск можно провести на одном сценарии и 10–20 записях: согласовать критерии, проверить транскрибацию и затем расширить анализ на команду.

[Открыть проект](#)[Страница продукта](#)

Golova: ИИ-аватар для живого общения на сайте

Видео-виджет принимает голосовые и текстовые вопросы, формирует ответ по настройкам персонажа и продолжает диалог внутри страницы сайта.

AI-консультант для сайта: отвечает по документам компании

Golova

Июнь 2026

Node.js

Express

TypeScript

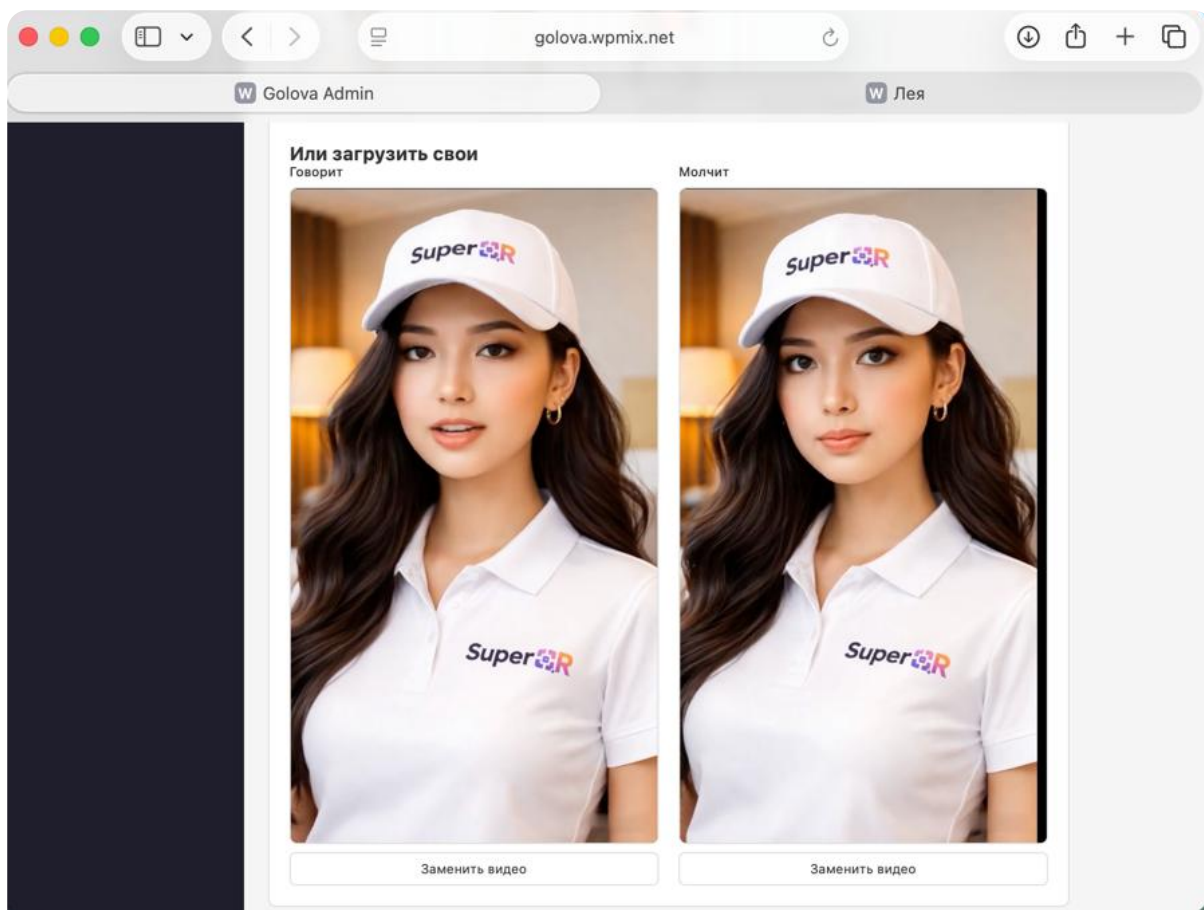
React

OpenAI

Google Gemini

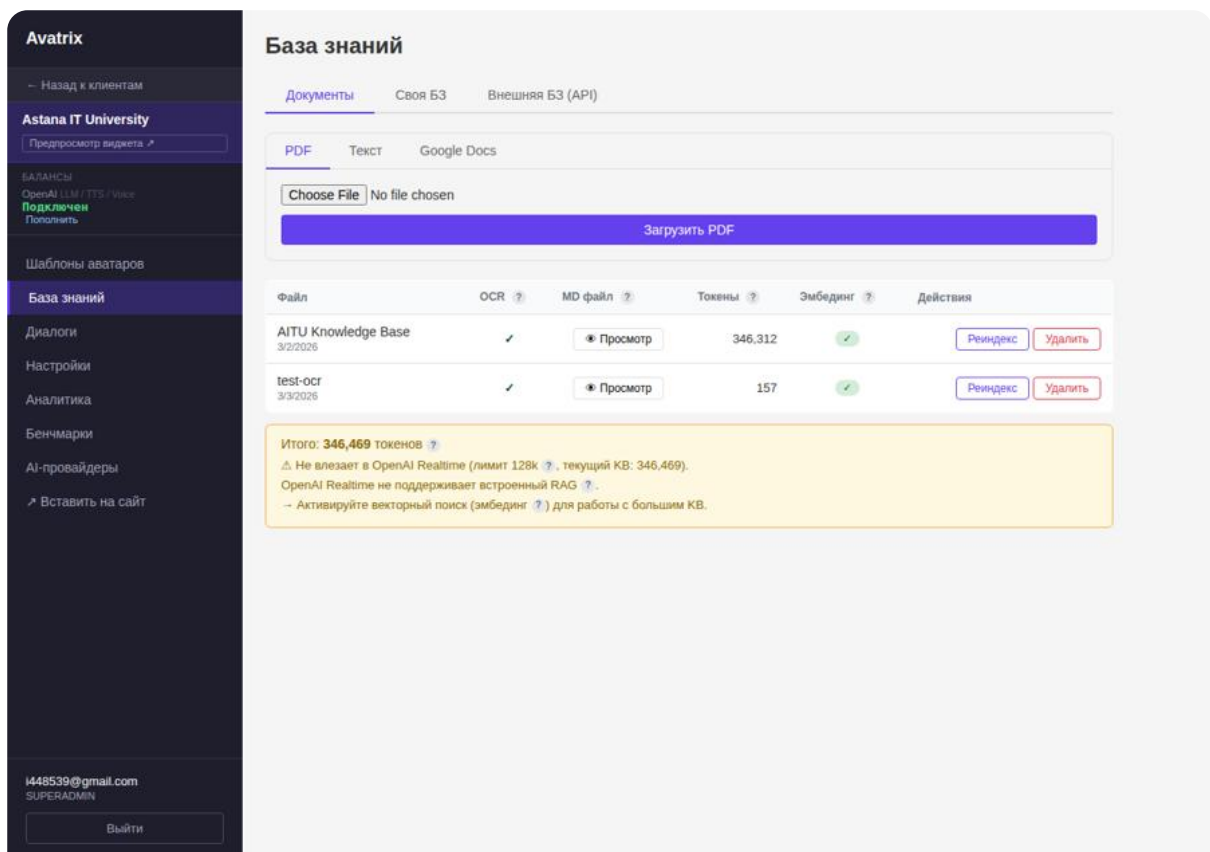
RAG

OCR



Задача

Обычный видеоролик на сайте говорит со всеми посетителями одинаково и не отвечает на вопросы. Для Golova цифровой персонаж должен был принимать реплику человека, отвечать голосом без заранее записанного сценария и продолжать разговор в том же окне. Владельцу сайта требовались настройки внешности, голоса, модели и материалов персонажа.



Документы обрабатываются и подключаются к ответам

Решение

Аватар внутри сайта

Мы собрали видео-виджет и панель управления персонажами. Виджет открывается поверх страницы, поддерживает голос и текст и подключается одним скриптом.

Ответ зависит от текущего диалога

В голосовом режиме система распознаёт реплику, передаёт историю разговора языковой модели и озвучивает результат. Содержание формируется во время разговора с учётом роли персонажа.

Внешность и голос настраиваются отдельно

В библиотеке доступны разные персонажи, для проекта можно загрузить свой образ. Панель хранит выбранные видео, голос, способ передачи аудио и параметры ответа.

Знания из документов

К персонажу подключаются PDF и текстовые материалы. Система извлекает содержание, включая сканы, и показывает состояние обработки каждого файла.

Результат

Платформа для нескольких персонажей

Владелец создаёт клиентов и аватаров, назначает им модели, инструкции, голоса и базы знаний. Настройки и история разных клиентов разделены.

Модель выбирается под сценарий

Для коротких консультаций и сложных вопросов можно использовать разные модели. Ключи поставщиков хранятся на сервере и не попадают в браузер.

Роль принадлежит конкретному аватару

Один персонаж проводит презентацию, другой работает преподавателем или консультантом. Имя, инструкция и манера ответа меняются в панели без пересборки сайта.

Управляемый живой диалог

Команда может заменить голос, образ, модель или документы и сохранить тот же виджет на сайте. Диалоги остаются доступны для последующего разбора.

[Открыть проект](#)[Техническое задание](#)[Страница продукта](#)

ИИ-парсер лидов для фриланс-бирж с Telegram-уведомлениями

Система отсеивает дубли и задачи вне профиля, анализирует оставшиеся проекты и отправляет в Telegram короткий разбор со ссылкой и следующим шагом.

Парсер лидов с FL.ru

Собственная команда разработки

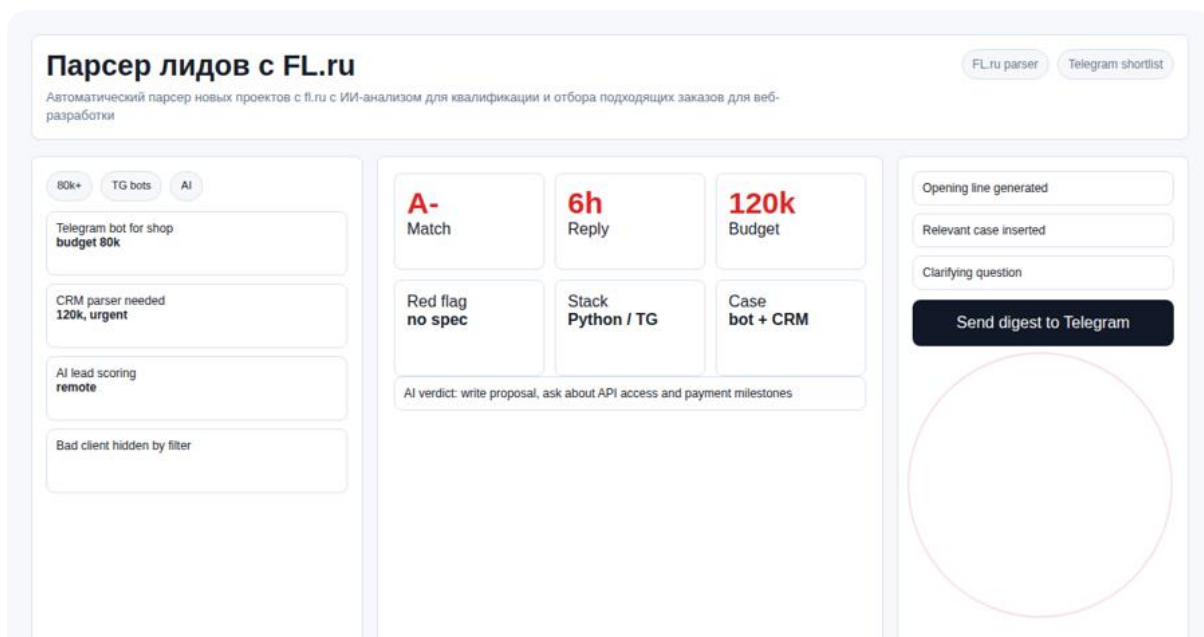
Июнь 2026

Python

FL.ru

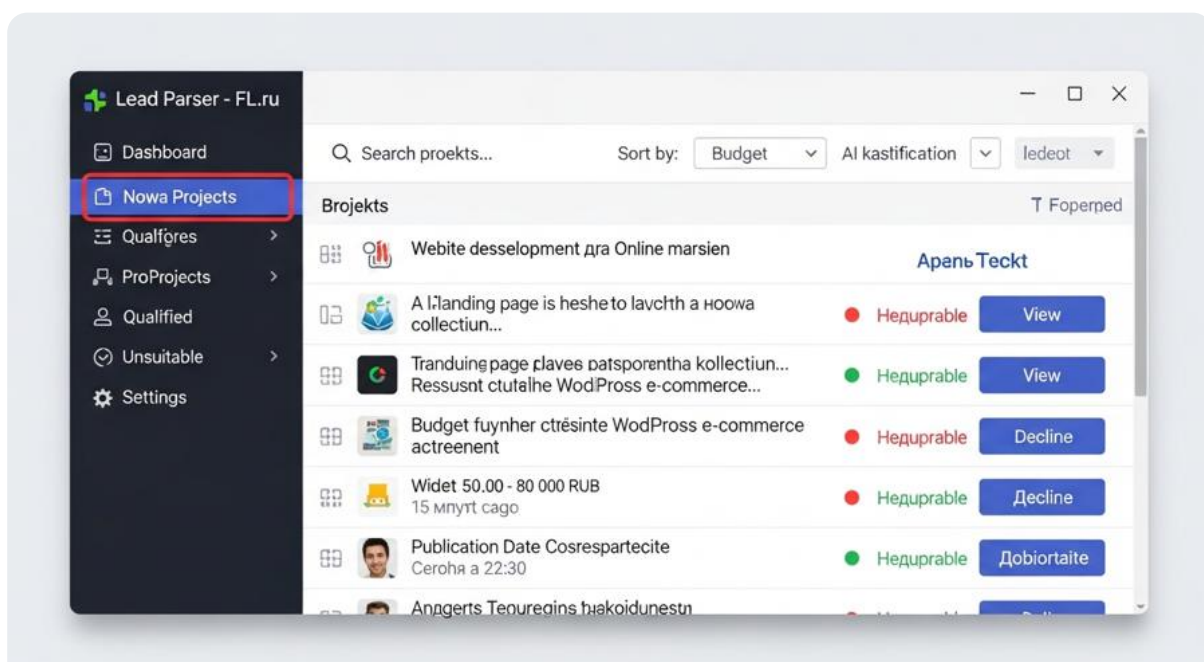
Telegram Bot API

LLM API



Задача

На FL.ru новые проекты появляются в течение дня, и подходящая задача быстро уходит вниз ленты. Команда тратила время на повторный просмотр карточек, проекты вне профиля и объявления без данных для решения. Требовался поиск, при котором в Telegram попадают только новые задачи по веб-разработке, ботам и AI с коротким разбором перед откликом.



Новые проекты, фильтры и результаты квалификации

Решение

Первичный отбор новых проектов

Система по расписанию читает карточки, приводит поля к единому виду и отсеивает дубли. Быстрые правила проверяют тему, бюджет и стоп-слова, после чего запускается AI-квалификация.

Критерии под текущий профиль команды

Приоритеты можно менять вместе с загрузкой: поднимать отдельные типы задач или исключать неподходящий стек. Очевидно нерелевантные темы отбрасываются до обращения к модели.

Разбор до уведомления

Модель выделяет суть задачи, соответствие стеку, риски и вопросы заказчику. Решение сохраняется, чтобы правила можно было проверять и корректировать на реальных проектах.

Результат

Короткий список вместо постоянного обновления ленты

В Telegram приходит ссылка, бюджет, суть задачи и следующий шаг. Команда открывает проекты, которые уже прошли технические фильтры и первичную смысловую оценку.

Контекст до перехода на биржу

В сообщении есть краткое содержание ТЗ и вопросы для уточнения. Менеджер решает, готовить отклик, отложить проект или пропустить его.

Журнал спорных решений

История хранит исходную карточку, сработавшие правила и вывод модели. Если проект был пропущен ошибочно, критерий можно поправить и проверить на следующих объявлениях.

Одно уведомление на проект

Расписание запускает проверку без ручного участия, а журнал защищает от повторной отправки одной карточки.

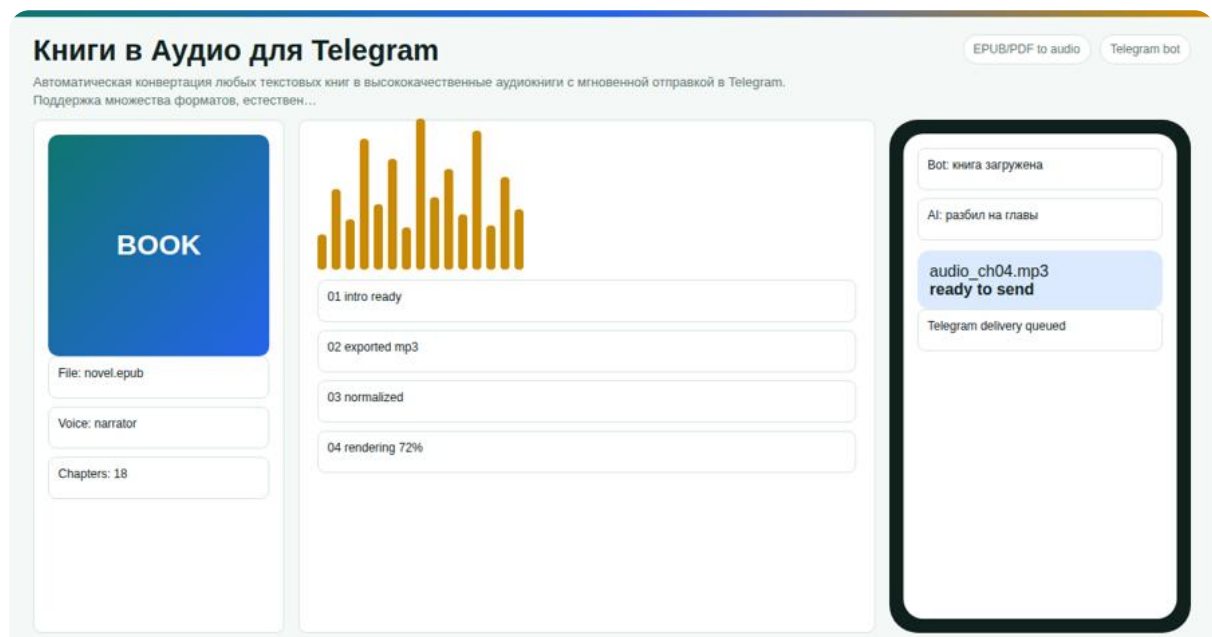
[Открыть проект](#)[Страница продукта](#)

Сервис преобразования книг в аудио с доставкой в Telegram

Система принимает книгу, извлекает текст, делит его на части, синтезирует аудио и собирает готовую запись с контролем статуса обработки.

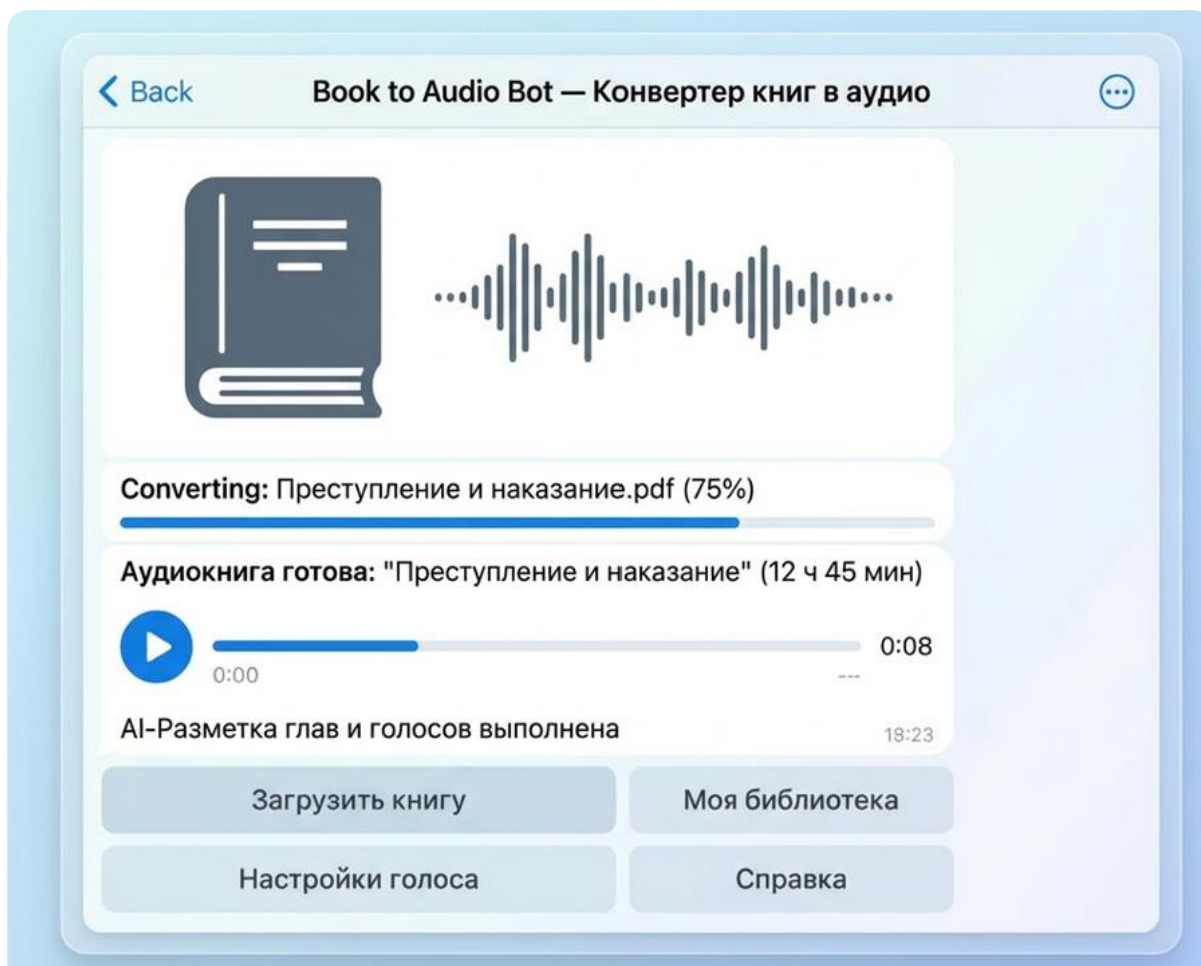
[Книги в Аудио для Telegram](#)[2026](#)[TypeScript](#)[Express](#)[SQLite](#)[OpenAI-compatible TTS](#)[FFmpeg](#)

TXT, PDF, EPUB и FB2



Задача

Текстовые книги часто не имеют готовой аудиоверсии, а длинный документ нельзя надёжно отправить в синтез речи одним запросом. Требовался процесс, который принимает файл, сохраняет состояние задачи и продолжает обработку по частям без ручной нарезки.



Загрузка книги и контроль обработки

Решение

Разбор книги и очередь обработки

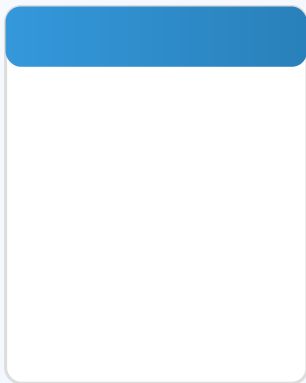
Сервис читает TXT, PDF, EPUB и FB2, очищает извлечённый текст и делит его на управляемые фрагменты. Для книги и каждого аудиофрагмента хранится отдельный статус, поэтому обработку можно контролировать и восстанавливать.

Синтез и сборка аудио

Фоновый worker последовательно создаёт MP3-фрагменты, проверяет их наличие и объединяет завершённые части через FFmpeg. Уведомления о ходе задачи можно отправлять в Telegram.

Готовый конвейер можно подключить к веб-интерфейсу или Telegram-боту, добавить выбор голоса, языка и правила хранения файлов.

Книги в Аудио для Telegram



Основные возможности сервиса

[Страница продукта](#)

BUT RAG: закрытый AI-ассистент по корпоративным документам

Однотенантная база знаний с OCR, ролевым доступом, поиском по источникам, журналом ответов, feedback и benchmark-проверками.

RAG — закрытый AI-ассистент по базе документов

2026

TypeScript

Express

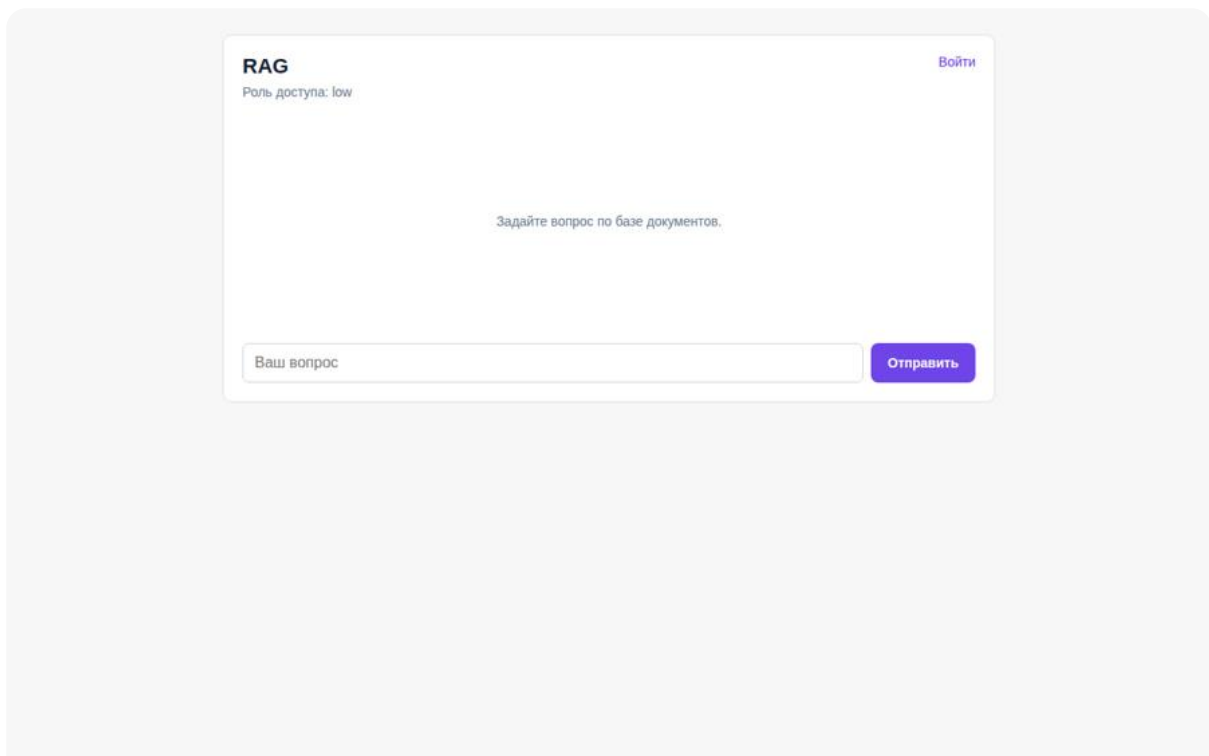
React

PostgreSQL

pgvector

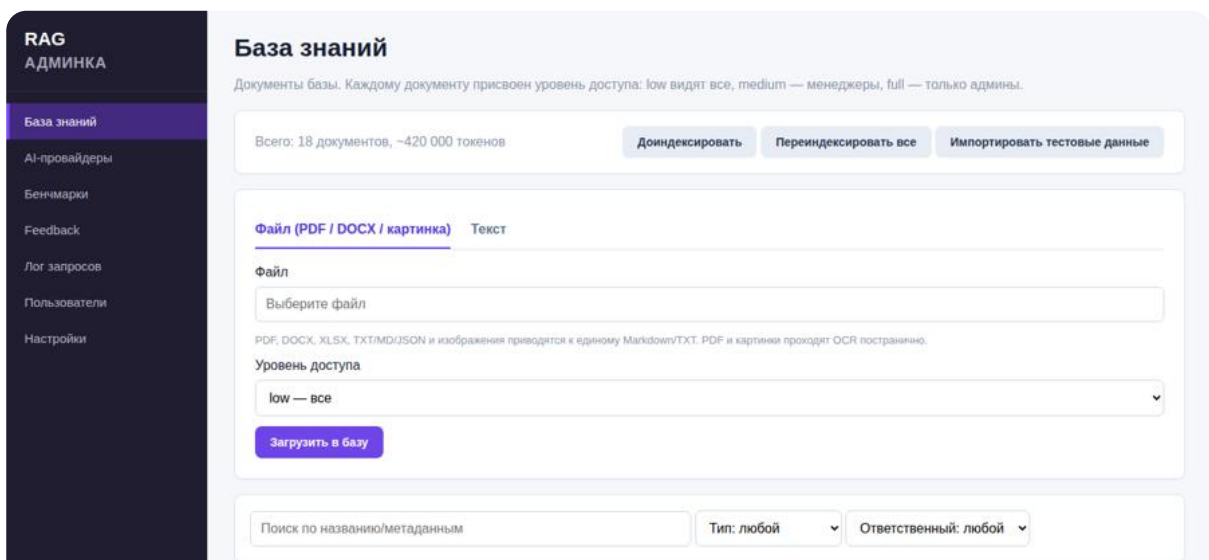
OpenAI-compatible API

low / medium / full



Задача

Корпоративные документы хранятся в разных форматах и доступны сотрудникам с разными правами. Обычный чат с моделью не показывает, откуда взят ответ, и не даёт системно проверять ошибки на реальных вопросах.



Документы и уровни доступа

Решение

База знаний с ролями и OCR

Администратор загружает документы и назначает им уровень доступа. PDF, Office-файлы и изображения разбираются по страницам; для сканов сохраняются OCR-текст и просмотр оригинала.

RAG, feedback и benchmark-проверки

Ассистент ищет только в доступном пользователю корпусе, прикладывает источники и сохраняет trace ответа. Отдельные экраны собирают оценки пользователей и запускают проверочные наборы вопросов перед выпуском изменений.

RAG API 1.0 OpenAPI

API для админки и чата RAG. Авторизация через cookie-сессию, защищённые admin endpoint требуют роль администратора.

Authorize

auth Логин, выход и текущий пользователь

POST /api/auth/login Логин

POST /api/auth/logout Выход

GET /api/auth/me Текущий пользователь

chat Чат с RAG

POST /api/chat Сообщение в чат

documents База знаний

GET /api/documents Список документов

POST /api/documents/upload Загрузить файл

POST /api/documents/reindex Переиндексировать документы

GET /api/documents/{id}/pages/{n}/view Просмотр страницы

benchmark Бенчмарк качества

GET /api/benchmark/questions Список вопросов

GET /api/benchmark/runs История запусков

POST /api/benchmark/run Запустить бенчмарк

feedback Анализ ответов

GET /api/feedback Список feedback

POST /api/feedback Сохранить feedback

settings Настройки

GET /api/settings Получить настройки

PUT /api/settings Обновить настройки

system Health и документация

GET /api/health Health-check

GET /api/openapi.json OpenAPI spec

GET /api/docs Swagger UI

Schemas

User object

Document object

DocumentPage object

SourceReference object

ChatRequest object

ChatResponse object

FeedbackEntry object

BenchmarkRun object

OpenAPI для интеграций

[Swagger / OpenAPI](#)[Страница продукта](#)

TimeMaterial: подготовка смет из исходных документов

Система собирает позиции из таблиц и документов, сопоставляет их со справочниками расценок и оставляет специалисту проверяемый черновик сметы.

[TimeMaterial — Автоматизация смет с AI](#)[2026](#)[Node.js](#)[Telegram Bot API](#)[Excel](#)[PDF parsing](#)[OpenAI-compatible API](#)

Telegram-бот с сохранением отдельных сессий

TimeMaterial — Автоматизация смет с AI

Materials estimateSupplier match

AI-система для автоматического формирования смет из любых материалов: Excel, Google Docs, PDF, текстовые описания.
Работает со справочниками расценок ГЭСН...

Material	Qty	Supplier	Delta
Panel			-6%
Sensor			-6%
Box			-6%
Pipe			-6%
Cable			-6%
Panel			-6%
Sensor			-6%
Box			-6%
Pipe			-6%

AI found cheaper equivalent
-148,000 RUB

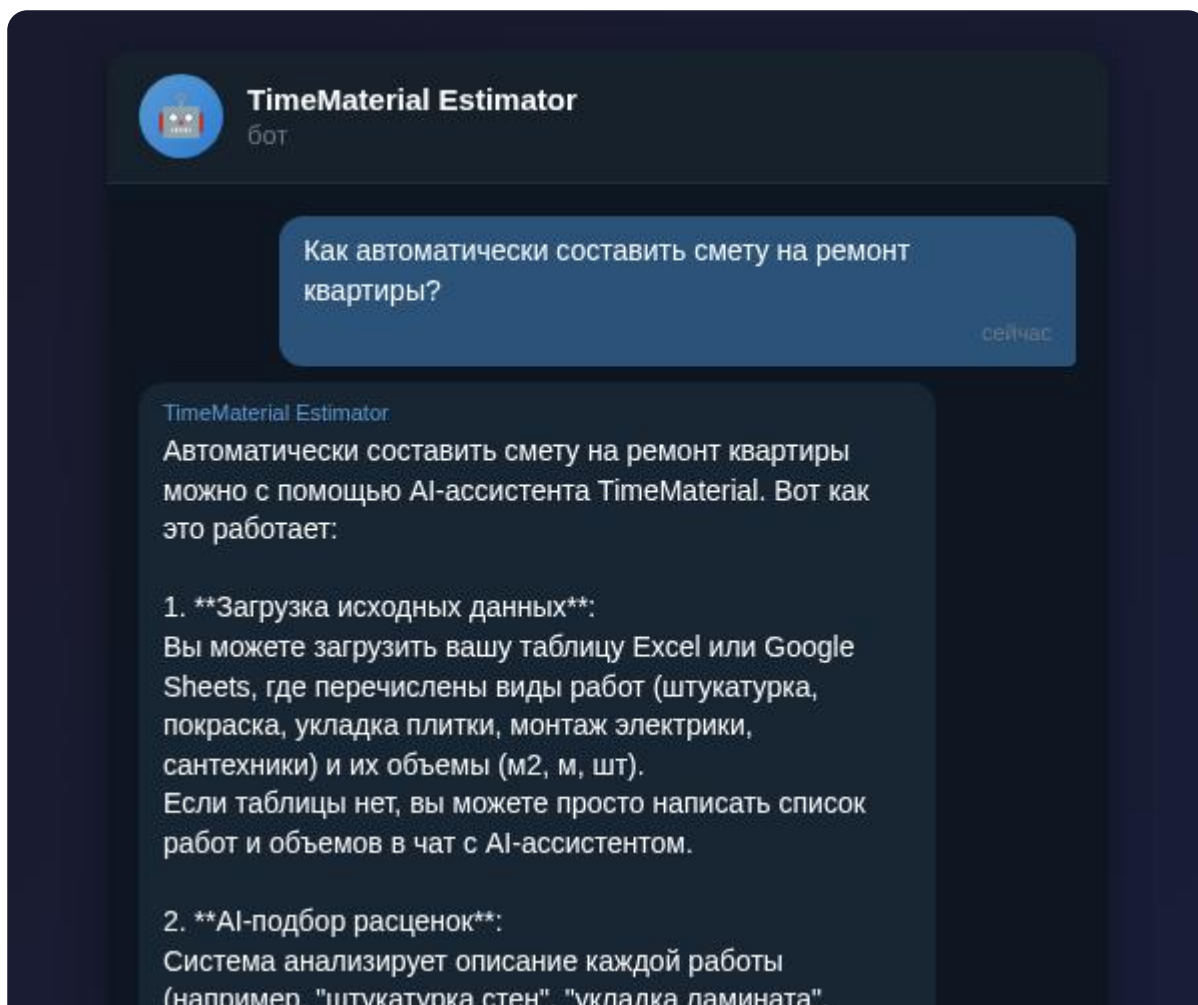
PDF estimate ready

Approval lane: PM

Lead time risk

Задача

Исходные данные для сметы приходят в таблицах, документах и сообщениях с разной структурой. Сметчику приходится вручную переносить позиции, нормализовать единицы и искать подходящие расценки.



Разбор позиций и подготовка структуры

Решение

Разбор исходных материалов

Пользователь передаёт файл или описание работ. Система извлекает позиции, количества и единицы измерения, а неоднозначные места возвращает на уточнение.

Сопоставление со справочником

Нормализованные позиции сравниваются с доступным справочником ГЭСН или локальными расценками. В результат попадает ссылка на исходную строку и выбранное соответствие.

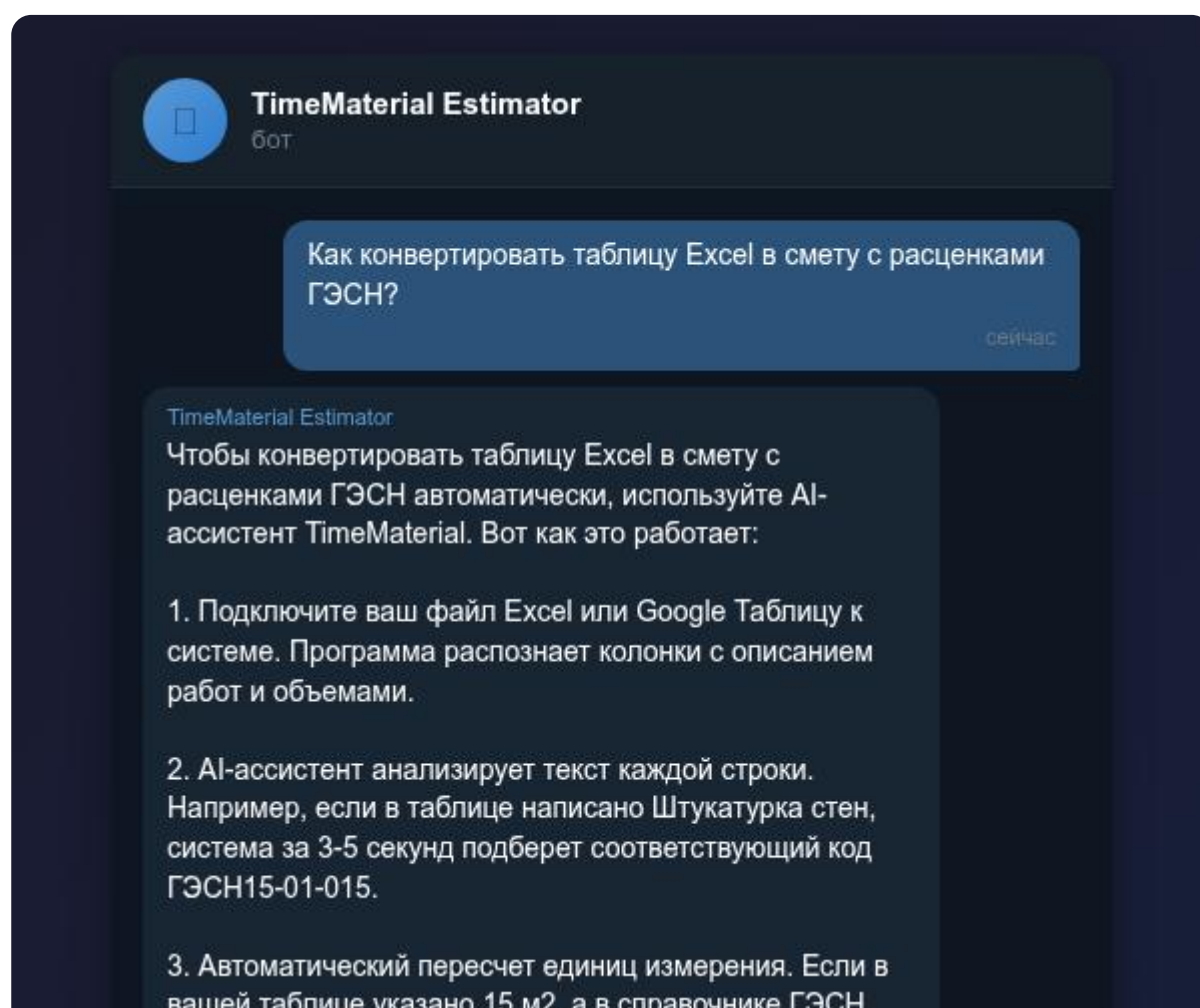
Результат

Черновик, который можно проверить

Специалист получает структурированную таблицу, видит неподтверждённые соответствия и правит их до экспорта.

Повторяемая обработка типовых файлов

Правила разбора и подтверждённые сопоставления можно использовать в следующих проектах вместо повторного ручного переноса.



Проверяемая таблица для сметчика

[Страница продукта](#)

AI-Assisted Development: рабочий процесс разработки с агентом и ревью

Процесс, в котором заказчик ставит задачи обычным языком, агент вносит изменения, а инженер проверяет архитектуру, тесты и живой результат.

AI-Assisted Development

2026

Python

Node.js

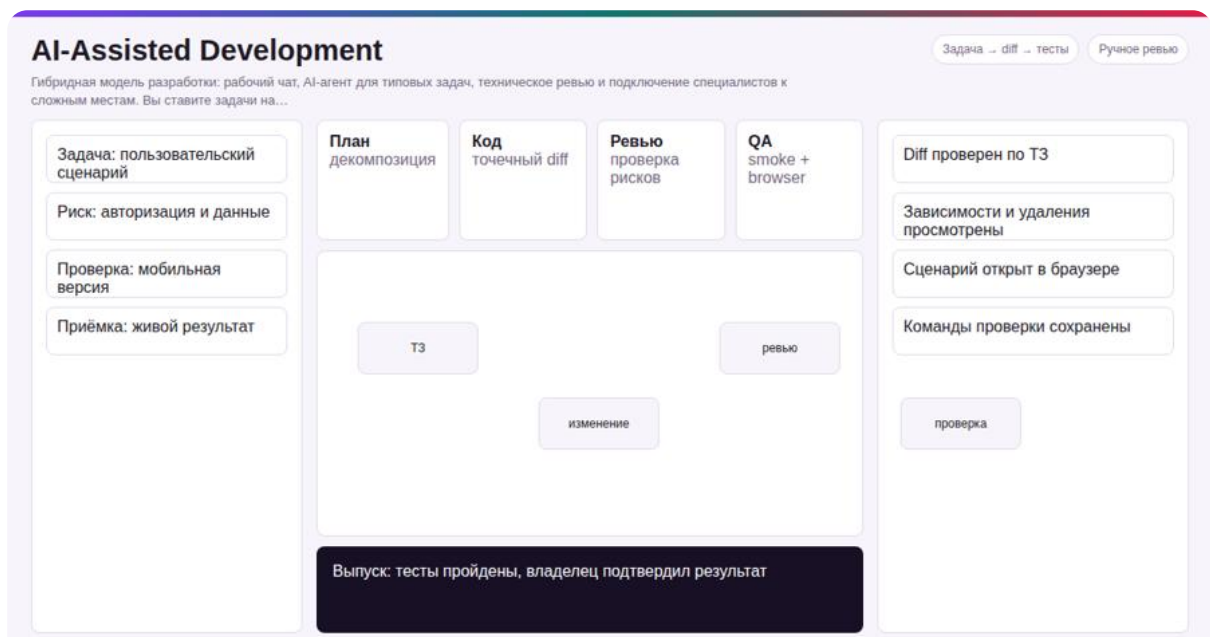
Git

GitHub

Docker

AI coding agents

Изменение → тесты → live-проверка



Задача

У небольших команд накапливается много коротких задач: правки интерфейса, интеграции, тесты и эксплуатационные проблемы. Передача каждой задачи отдельному специалисту замедляет цикл, но полностью автономный AI-агент может пропустить контекст или сломать соседний сценарий.

Решение

Одна очередь задач и общий контекст

Задача поступает в рабочий чат или issue, связывается с репозиторием и проверяемым результатом. Агент читает проектные правила, меняет исходный код и запускает локальные проверки.

Инженерный контроль на рискованных местах

Архитектурные изменения, удаления, зависимости и продакшен-проверки проходят ручное ревью. Для UI и интеграций результат проверяется в браузере или на работающем сайте.

Результат

Короткие задачи не теряются между исполнителями

История решения, команды проверки и ссылка на результат остаются рядом с задачей.

Скорость агента не заменяет ответственность

Готовность определяется работающим сценарием и проверками, а не самим фактом изменения кода.

[Страница продукта](#)

Система контроля качества AI-ответов

Feedback, журналы запросов, контрольные вопросы и verifier объединены в процесс, который помогает находить неподтверждённые ответы до релиза.

AI Quality Assurance — контроль качества AI-ответов

2026

TypeScript

PostgreSQL

LLM API

RAG

Automated verifiers

feedback + benchmarks + verifier

Ответ
текст и trace

Текущая версия
один набор вопросов
сравнимый результат

critical: ответ без источника

major: устаревшие данные

minor: несогласованный тон

Источник
страница документа

Кандидат на релиз
один набор вопросов
сравнимый результат

Оценка
feedback

Ручная проверка
один набор вопросов
сравнимый результат

Тест
benchmark

Benchmark выполнен

Источники проверены

Негативный feedback разобран

Запрошено ручное ревью

Trace сохранён

Релиз блокируется при
неподтверждённом ответе

AI Quality Assurance — контроль качества AI-ответов

Внедрение систем контроля качества AI: Guardian Models для автоматической проверки ответов, Feedback Loop для сбора обратной связи, дашборд аналитики галл...

Проверка AI-ответов

Источники и регрессии

Задача

AI-функция может отвечать убедительно даже при слабом источнике или неверном поиске. Разрозненных ручных проверок недостаточно: после смены модели или промпта старые ошибки возвращаются.

RAG
АДМИНКА

База знаний

AI-провайдеры

Бенчмарк

Feedback

Лог запросов

Пользователи

Настройки

Прогон бенчмарка

Набор: demo-rag-quality · модель: gpt-4.5 · дата: 24.06.2026 18:40

92%
опрашиваемых

10/10
ответов с источниками

0
отказ переводчика

Ответы

1. Какие документы видны гостевому пользователю?

accepted score 96

Ответ сформирован только по демонстрационной базе знаний. Система указывает роль доступа, найденные страницы, краткое объяснение и не использует неподтвержденные факты. Если источник отсутствует, ответ переводится в безопасную блокировку.

S1: Регламент поддержки — стр. 1
S2: Политика качества ответов — стр. 2

2. Что делать, если документ не прошёл OCR?

accepted score 96

Ответ сформирован только по демонстрационной базе знаний. Система указывает роль доступа, найденные страницы, краткое объяснение и не использует неподтвержденные факты. Если источник отсутствует, ответ переводится в безопасную блокировку.

S1: Регламент поддержки — стр. 2
S2: Политика качества ответов — стр. 3

3. Как проверить источник ответа?

accepted score 96

Ответ сформирован только по демонстрационной базе знаний. Система указывает роль доступа, найденные страницы, краткое объяснение и не использует неподтвержденные факты. Если источник отсутствует, ответ переводится в безопасную блокировку.

S1: Регламент поддержки — стр. 3
S2: Политика качества ответов — стр. 4

4. Как администратор видит проблемный feedback?

accepted score 96

Ответ сформирован только по демонстрационной базе знаний. Система указывает роль доступа, найденные страницы, краткое объяснение и не использует неподтвержденные факты. Если источник отсутствует, ответ переводится в безопасную блокировку.

S1: Регламент поддержки — стр. 4
S2: Политика качества ответов — стр. 5

5. Когда нужен полный контекст вместо поиска?

review score 84

Ответ сформирован только по демонстрационной базе знаний. Система указывает роль доступа, найденные страницы, краткое объяснение и не использует неподтвержденные факты. Если источник отсутствует, ответ переводится в безопасную блокировку.

S1: Регламент поддержки — стр. 5
S2: Политика качества ответов — стр. 6

6. Как работает разделение доступа?

accepted score 96

Ответ сформирован только по демонстрационной базе знаний. Система указывает роль доступа, найденные страницы, краткое объяснение и не использует неподтвержденные факты. Если источник отсутствует, ответ переводится в безопасную блокировку.

S1: Регламент поддержки — стр. 6
S2: Политика качества ответов — стр. 7

7. Какие метрики смотреть перед релизом?

accepted score 96

Ответ сформирован только по демонстрационной базе знаний. Система указывает роль доступа, найденные страницы, краткое объяснение и не использует неподтвержденные факты. Если источник отсутствует, ответ переводится в безопасную блокировку.

S1: Регламент поддержки — стр. 7
S2: Политика качества ответов — стр. 8

8. Как повторить ответ из benchmark в чате?

accepted score 96

Ответ сформирован только по демонстрационной базе знаний. Система указывает роль доступа, найденные страницы, краткое объяснение и не использует неподтвержденные факты. Если источник отсутствует, ответ переводится в безопасную блокировку.

S1: Регламент поддержки — стр. 8
S2: Политика качества ответов — стр. 9

Демонстрационный benchmark с ответами, источниками и статусом проверки

Решение

Сбор воспроизводимых проблем

Пользовательская оценка связывается с вопросом, ответом, моделью, найденными источниками и trace. Негативные примеры превращаются в контрольные сценарии.

Проверка перед выпуском

Benchmark-набор запускается на новой версии. Verifier проверяет опору ответа на найденный контекст и блокирует результат, если утверждения нельзя подтвердить.

Результат

Ошибку можно повторить и сравнить

Команда видит плохой ответ вместе с входными данными, retrieval и версией модели.

Релизное решение опирается на набор проверок

Изменение промпта или поиска проходит одинаковые контрольные вопросы вместо выборочной оценки на глаз.

[Страница продукта](#)

Telegram-бот для ведения дневника питания

Бот принимает описание еды текстом, голосом или фотографией, ведёт дневной счёт и хранит состояние пользователя прямо в Telegram.

Счетчик калорий Telegram Bot

2026

Node.js

Telegram Bot API

OpenAI-compatible API

OCR

Speech-to-text

@fatnosecretbot

Счётчик калорий

bot



Что ты съел сегодня?

14:08



14:08 ✓✓

Саша, ты сейчас прислал фотографию своего обеда.

Приблизительные калории:

1. Куриные наггетсы — около 250 ккал (порция из 4-5 штук).
2. Картошка фри — примерно 300 ккал за 100 г.
3. Бургер — около 300-500 ккал в зависимости от размера и ингредиентов.
4. Напиток — 100-200 ккал (в зависимости от типа).

Предположим, ты съел 4 наггетса, 150 г картошки, бургер и напиток. Это примерно 1050-1300 ккал.

Лимит 2200 ккал, осталось 900-1150 ккал.

Жду следующий прием пищи!

14:09

Menu



| Write a message...



Задача

Записывать каждый приём пищи вручную неудобно, особенно с телефона. Пользователю нужен короткий сценарий без отдельного

приложения: отправить сообщение или фото и сразу увидеть обновлённый дневной итог.

Решение

Несколько способов ввода

Бот принимает текст, голосовые сообщения и фотографии. Распознанные продукты и порции возвращаются пользователю на подтверждение, прежде чем попасть в дневник.

Состояние и ежедневный цикл

Для каждого пользователя хранится цель, текущий итог и история операций. Новый день начинается с отдельного счёта с учётом часового пояса.

Результат

Дневник остаётся в привычном чате

Не нужно открывать отдельный интерфейс: добавление и проверка записи происходят в Telegram.

Сценарий можно развивать без переустановки приложения

К боту можно подключить рекомендации, напоминания, реферальную механику и административную аналитику.

[Открыть проект](#)[Страница продукта](#)

Презентационный сайт, собранный по брендбуку

Одностраничная презентация переносит визуальные правила бренда в адаптивный веб-интерфейс и ведёт посетителя к целевому контакту.

[Презентационный сайт по брендбуку](#)[2026](#)[HTML](#)[CSS](#)

JavaScript

Flask

Адаптивная одностраничная презентация



О нас | Stars Align!

Stars Align! – звёзды сошлись, чтобы этот уход нашёл тебя.

Астрологическая косметика для тела и дома, где ритуалы красоты переплетаются с магией ароматов.

Бренд, вдохновлённый движением планет, создаёт косметику, которая заботится не только о коже, но и о настроении. Каждый продукт – это история, рассказанная звёздами: о судьбоносных моментах, волшебных совпадениях и тонкой связи между небом и землёй.

Задача

Брендбук описывает цвета, типографику и композицию, но не готовый цифровой сценарий. Нужно было превратить его в короткую веб-презентацию, которая одинаково читается с телефона и компьютера.

Решение

Перенос системы бренда в веб

Из брендбука выделены шрифты, цветовые роли, отступы и правила работы с изображениями. На их основе собрана последовательность экранов без копирования печатной композиции один в один.

Целевой путь без лишней навигации

Основные тезисы, примеры и кнопки контакта расположены в одном потоке. Ссылки ведут в Telegram, поэтому посетитель переходит к разговору без сложной формы.

Результат

Бренд сохранил узнаваемость в браузере

Визуальный язык работает на разных ширинах экрана, а содержание остаётся читаемым.

Презентацию можно обновлять как обычную страницу

Текст и изображения меняются без пересборки PDF и повторной отправки файла аудитории.

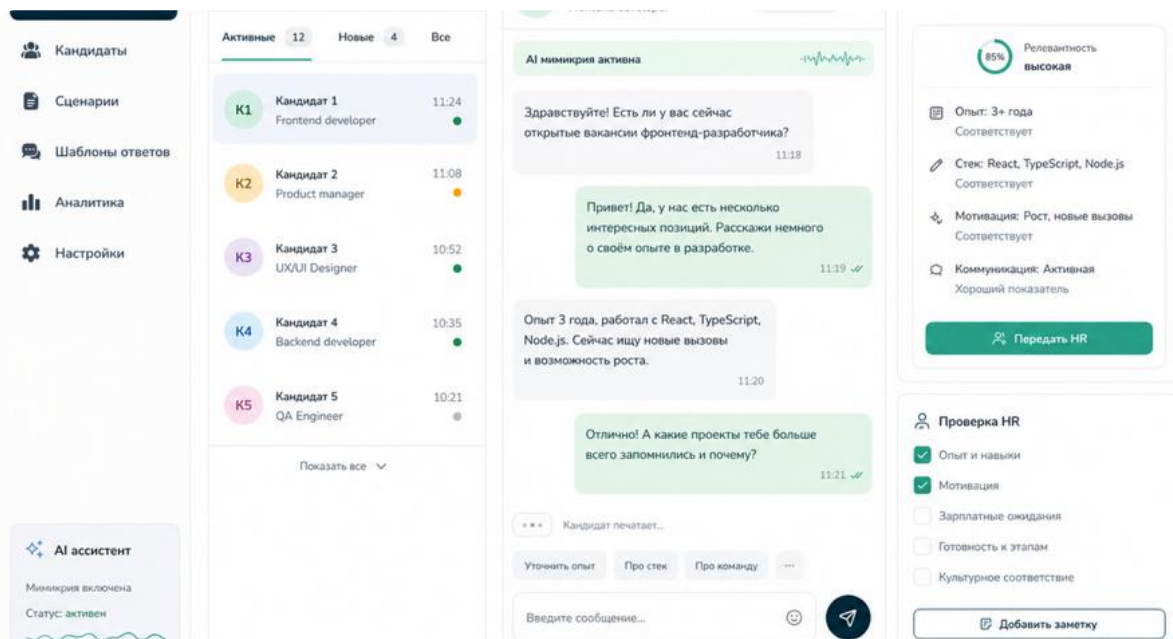
[Открыть проект](#)[Страница продукта](#)

Прототип HR-бота для первичной квалификации кандидатов

Telegram-бот ведёт первичный диалог по сценарию, отделяет ответы по вакансии от посторонних сообщений и передаёт сложные случаи рекрутеру.

[HR Диалог Мимикрия](#)[Прототип, 2026](#)[Node.js](#)[Pyrogram](#)[Telegram API](#)[OpenAI-compatible API](#)

Передача диалога оператору при выходе за сценарий



Задача

Рекрутеру приходится повторять одни и те же вопросы и вручную отделять релевантные ответы от свободного разговора. Полностью автоматический бот при этом не должен выдавать себя за человека или продолжать диалог вне согласованного сценария.

Решение

Сценарий квалификации

Бот задаёт вопросы по вакансии, сохраняет ответы и проверяет, относится ли новое сообщение к текущему этапу. Текст и правила диалога настраиваются в административной части.

Явная граница автоматизации

Когда вопрос выходит за сценарий или требуется решение рекрутера, бот останавливает автоматическую ветку и уведомляет оператора.

Результат

Структурированная карточка до ручного контакта

Рекрутер получает собранные ответы и понимает, с чего продолжить разговор.

Контроль остаётся у HR-команды

Сценарий и условия передачи можно менять под вакансию без переписывания всего бота.

[Страница продукта](#)

Прототип новостного радар для промышленной компании

Система собирает отраслевые публикации, выделяет события по заданным темам и отправляет короткие проверяемые уведомления ответственным сотрудникам.

Новостной радар для промышленности

Прототип, 2026

Node.js

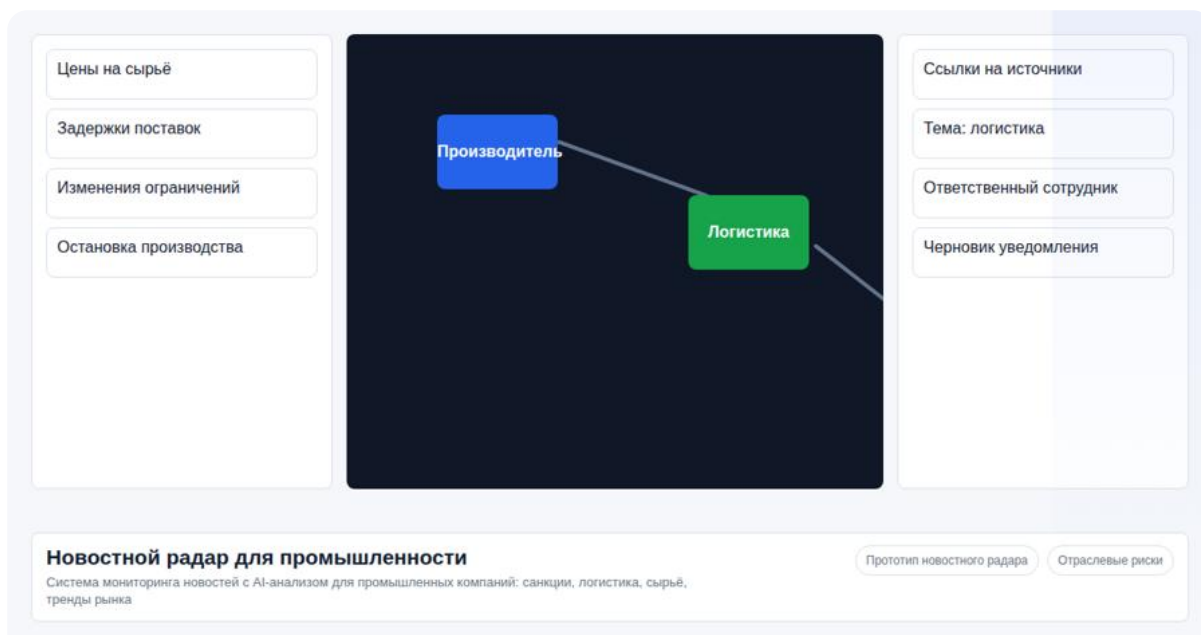
Python

RSS/HTML parsers

Telegram Bot API

OpenAI-compatible API

Санкции, логистика, сырьё и рыночные изменения



Задача

Отраслевые события распределены по сайтам, лентам и бюллетеням. Обычная подписка создаёт поток ссылок, но не объясняет, какое событие влияет на закупки, логистику или конкретный рынок.

Решение

Сбор и нормализация источников

Планировщик обходит согласованные RSS, HTML и JSON-источники, удаляет повторы и сохраняет оригинальную ссылку и дату публикации.

Классификация по бизнес-темам

AI-модуль выделяет тему, затронутые компании и возможную цепочку влияния. В Telegram уходит краткая карточка со ссылкой на первоисточник.

Результат

Вместо общей ленты — тематическая очередь

Сотрудник видит новости по своей зоне ответственности и может открыть исходную публикацию.

Набор тем меняется вместе с рисками

Администратор добавляет новые категории, источники и правила уведомлений без переработки интерфейса.

[Страница продукта](#)

Прототип корпоративной системы мониторинга медиа

Платформа объединяет новости и упоминания, группирует связанные события и готовит аналитические подборки для руководителей.

Корпоративная система мониторинга медиа и аналитики

Прототип, 2026

Python

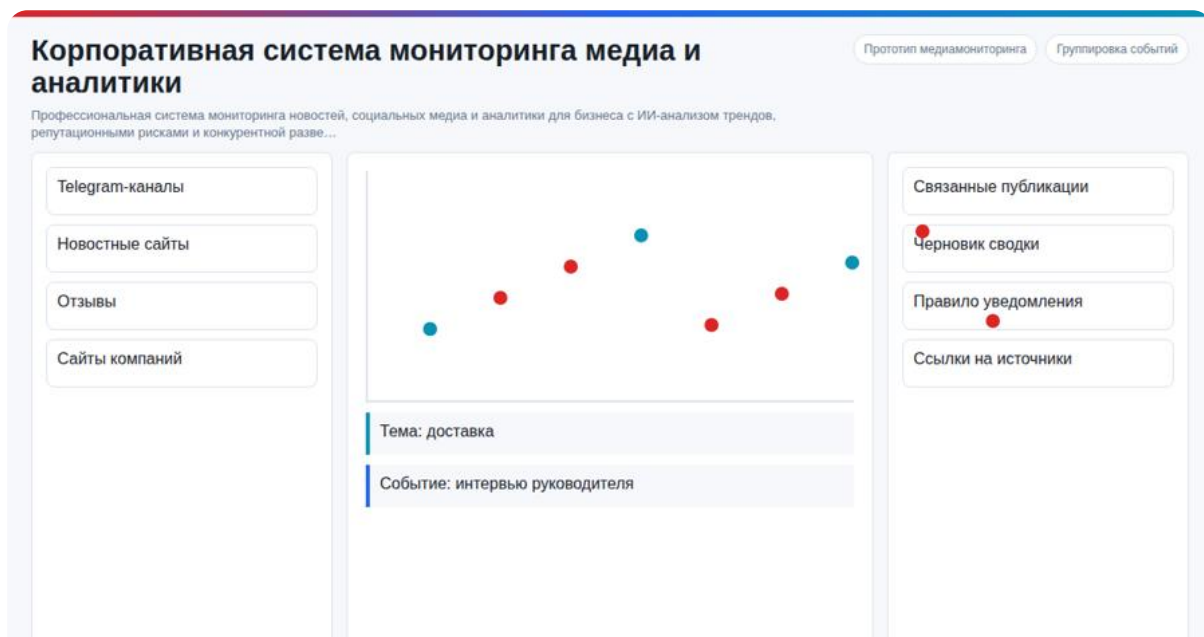
Node.js

PostgreSQL

News APIs

OpenAI-compatible API

Темы, компании, тренды и репутационные риски



Задача

Руководителю недостаточно списка ссылок: нужно понимать, какие публикации относятся к одному событию, как меняется тема и где появляется риск для компании или рынка.

Решение

Единый индекс публикаций

Материалы из подключённых источников нормализуются, очищаются от дублей и связываются с компаниями, темами и временными периодами.

Аналитические представления

Система группирует похожие события, выделяет новые сигналы и формирует подборки с кратким выводом и обязательными ссылками на исходные материалы.

Результат

Контекст сохраняется вместе с новостью

Пользователь переходит от сводки к группе публикаций и проверяет вывод по первоисточникам.

Общие данные для регулярных и срочных отчётов

Те же данные используются в дашборде, email-сводке и Telegram-уведомлении без ручного копирования.

[Страница продукта](#)

Прототип корпоративного голосового ввода

Настольный сценарий записывает речь по команде пользователя, превращает её в редактируемый текст и передаёт в рабочее приложение.

Корпоративные решения голосового ввода с ИИ

2026

Python

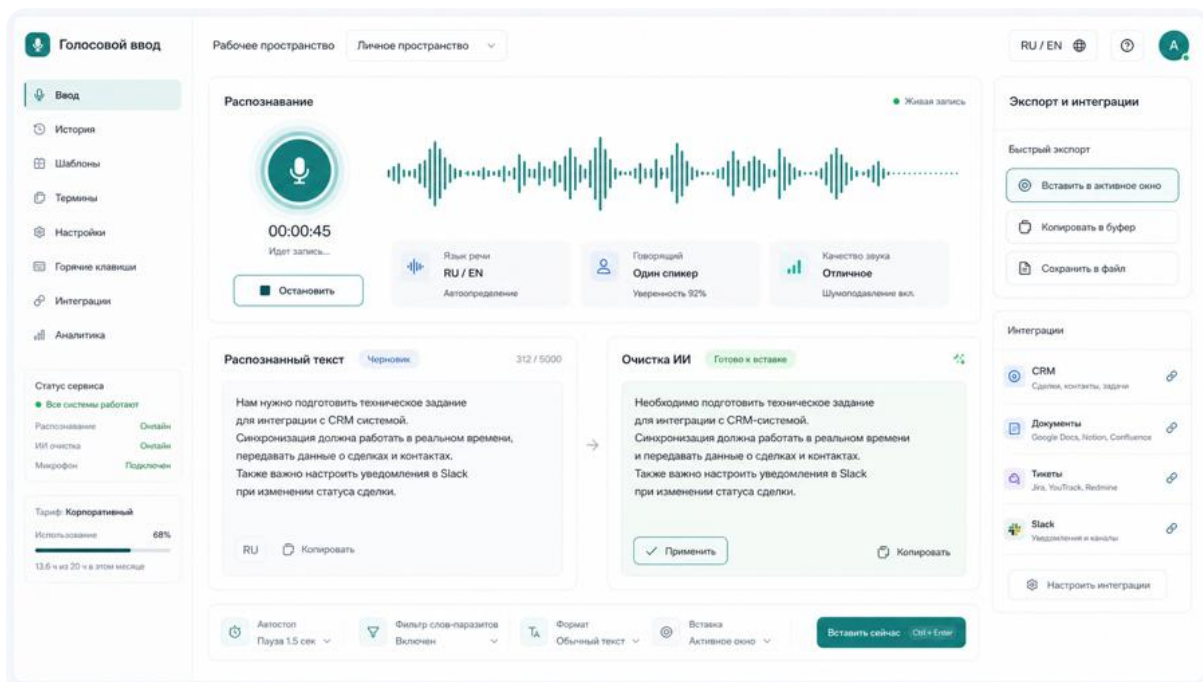
Flask

WebRTC

Speech-to-text

OpenAI Realtime API

speech.wpmix.net



Задача

При заполнении CRM, писем и внутренних документов длинный текст быстрее продиктовать, чем набрать. Системный диктовщик не всегда умеет работать по корпоративному сценарию и передавать результат в нужное поле.

Решение

Управляемая запись речи

Пользователь сам запускает и останавливает запись. Audio передаётся через сервер, поэтому ключ AI-провайдера не попадает во фронтенд.

Текст как промежуточный результат

Распознанный текст можно показать для проверки, очистить по согласованным правилам и только затем вставить в активное приложение или отправить в API.

Результат

Голос становится ещё одним способом ввода

Сценарий можно встроить в CRM, форму отчёта или внутренний кабинет без отдельного устройства.

Прототип проверяет основной технический риск

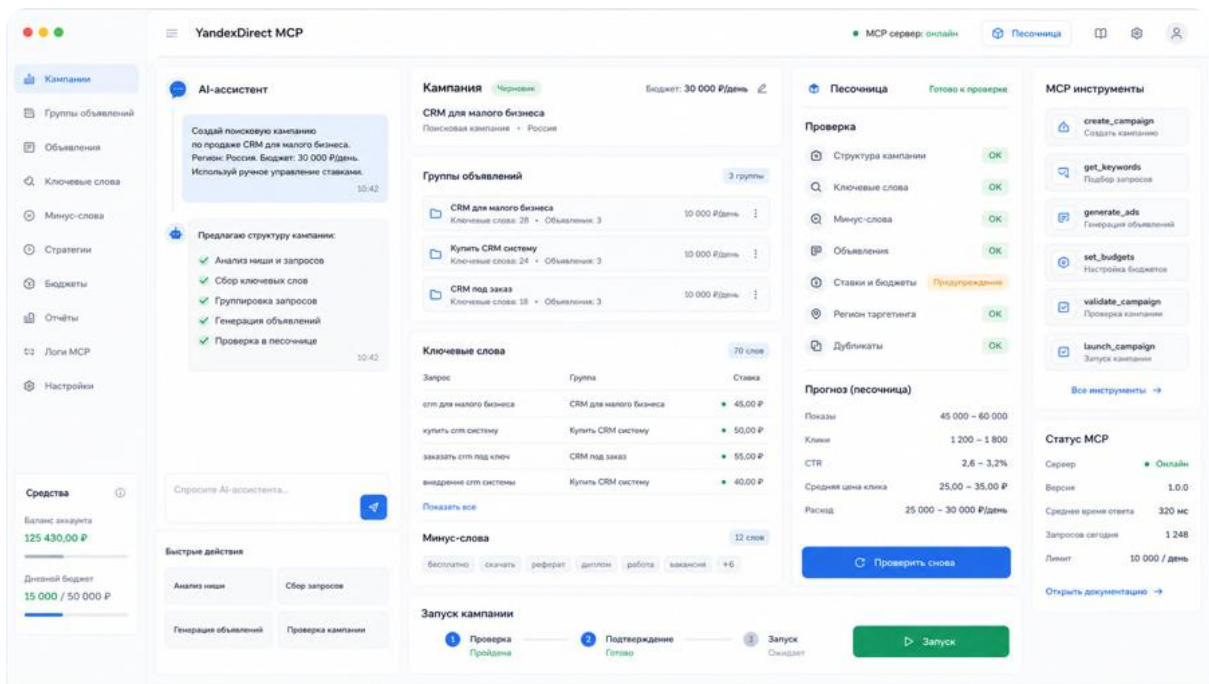
Публичный прототип подтверждает работу микрофона, серверного обмена и realtime-сессии; конкретная интеграция выбирается для рабочего места клиента.

[Открыть проект](#)[Страница продукта](#)

MCP-сервер для управления Яндекс Директом

Сервис даёт AI-агенту типизированные операции для чтения кампаний, подготовки групп объявлений, работы с ключевыми словами и аналитикой.

[YandexDirect MCP сервер](#)[2026](#)[Node.js](#)[Model Context Protocol](#)[Yandex Direct API](#)[Yandex Metrika API](#)[OpenAI-compatible API](#)[Web UI, REST API, MCP, CLI и аналитика](#)



Задача

Управление рекламой через чат опасно сводить к свободному тексту: агент должен различать кампанию, группу, объявление и настройки автотаргетинга, а перед записью показывать точный набор действий.

Решение

Типизированные инструменты вместо кликов

MCP-сервер оборачивает операции Яндекс Директа в команды со схемой параметров. Агент может читать структуру аккаунта, готовить ключевые слова и формировать группу объявлений как единое целое.

Разделение подготовки и записи

Сначала строится план с названиями, ключами, минус-словами и настройками. Изменяющая команда запускается отдельно, а ограничения API и ручные шаги остаются явными.

Результат

Рекламная операция становится проверяемой

Перед выполнением видно, какая сущность будет создана или изменена, и какие параметры уйдут в API.

Общие операции для агента, CLI и интерфейса

Те же операции можно использовать в AI-чате, автоматизации и административном веб-интерфейсе.

Страница продукта

Прототип конструктора CRM под процесс компании

Модульная CRM собирается вокруг этапов продаж, ролей и интеграций конкретной команды вместо универсального набора экранов.

AI CRM Constructor: Конструктор CRM под ваш бизнес

Прототип, 2026

Python

Flask

PostgreSQL

Telegram Bot API

Calendar APIs

Сделки, задачи, уведомления и календарные события

AI CRM Constructor: Конструктор CRM под ваш бизнес

CRM-система, которая адаптируется под ваши бизнес-процессы. Без лишних функций, только то что нужно вашему бизнесу.

Прототип CRM

Схема процесса

The screenshot displays the 'AI CRM Constructor' interface. On the left, there's a form for creating a deal card with fields for 'Сущность: сделка', 'Поле: источник', 'Поле: ответственный', 'Связь: задача', 'Роль: руководитель', and 'Событие: оплата'. To the right, there's a process flow diagram with five columns: 'Новая заявка', 'Квалификация', 'Предложение', 'Оплата', and 'Подсказка: проверить обязательные поля и создать следующую задачу'. Each column contains a 'Карточка сделки обязательные поля' and a 'Получена заявка следующее действие'. The 'Подсказка' column also includes 'Автоматизация: задача', 'Правило: смена этапа', 'Интеграция: webhook', and 'Права доступа'.

Задача

Универсальная CRM часто заставляет небольшую команду подстраиваться под лишние сущности и статусы. Для проверки подхода нужен был прототип, где структура сделки начинается с реального процесса компании.

Решение

Схема из этапов и ролей

На брифинге фиксируются этапы, обязательные поля, ответственные и переходы. Из этой схемы собираются карточка сделки, очередь задач и права доступа.

Интеграции как отдельные модули

Уведомления, календарь, почта и мессенджеры подключаются к событиям процесса, а не зашиваются в каждую форму отдельно.

Результат

Прототип показывает процесс целиком

Команда может пройти путь от новой заявки до завершения сделки и увидеть, где нужны автоматические действия.

Функции добавляются по необходимости

Модульная структура позволяет добавлять нужные функции без включения всех возможностей большой CRM.

[Страница продукта](#)

Прототип локального AI-сервера для дома и малого офиса

Архитектура объединяет локальную модель, управление устройствами, медиатеку и резервное копирование на одном мини-сервере.

AI Home Server

Прототип, 2026

Linux

Docker

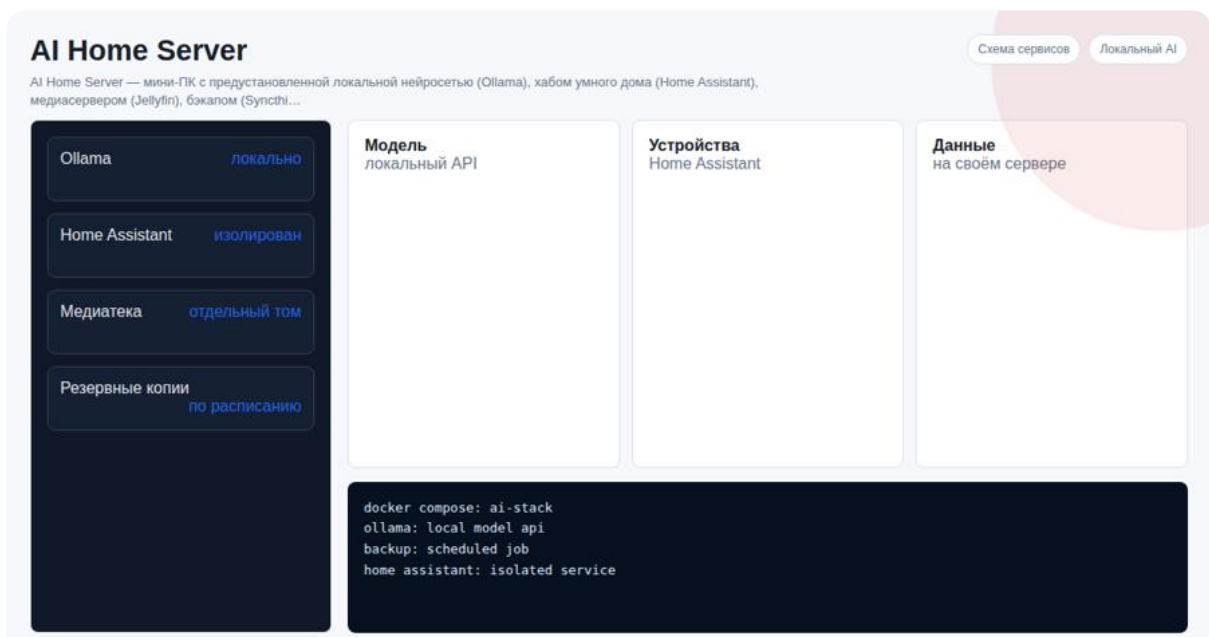
Ollama

Home Assistant

Jellyfin

Syncthing

Ollama, Home Assistant, Jellyfin и Syncthing



Задача

Домашняя автоматизация, локальный AI, медиатека и резервные копии обычно живут на разных устройствах и требуют отдельных настроек. Нужно было проверить, как объединить их на одном обслуживаемом сервере.

Решение

Сервисы в отдельных контейнерах

Каждый компонент разворачивается изолированно, получает своё хранилище и понятный маршрут обновления. Сбой медиасервера не должен останавливать умный дом или резервное копирование.

Локальный AI как внутренний сервис

Ollama предоставляет модель другим компонентам по локальному API. Telegram-агент и Home Assistant могут обращаться к ней без отправки бытовых данных во внешний облачный чат.

Результат

Проверена составная архитектура устройства

Прототип фиксирует набор сервисов, сетевые связи и требования к диску и памяти до выбора конкретного мини-ПК.

Систему можно собирать поэтапно

Начать можно с Home Assistant и бэкапа, затем добавить локальную модель и медиатеку без смены общей схемы.

[Страница продукта](#)

Прототип AI-визуализации интерьера по фотографии

Пользователь загружает снимок помещения, выбирает направление оформления и получает варианты, сохраняя исходную геометрию комнаты.

AI Дизайн интерьера

Прототип, 2026

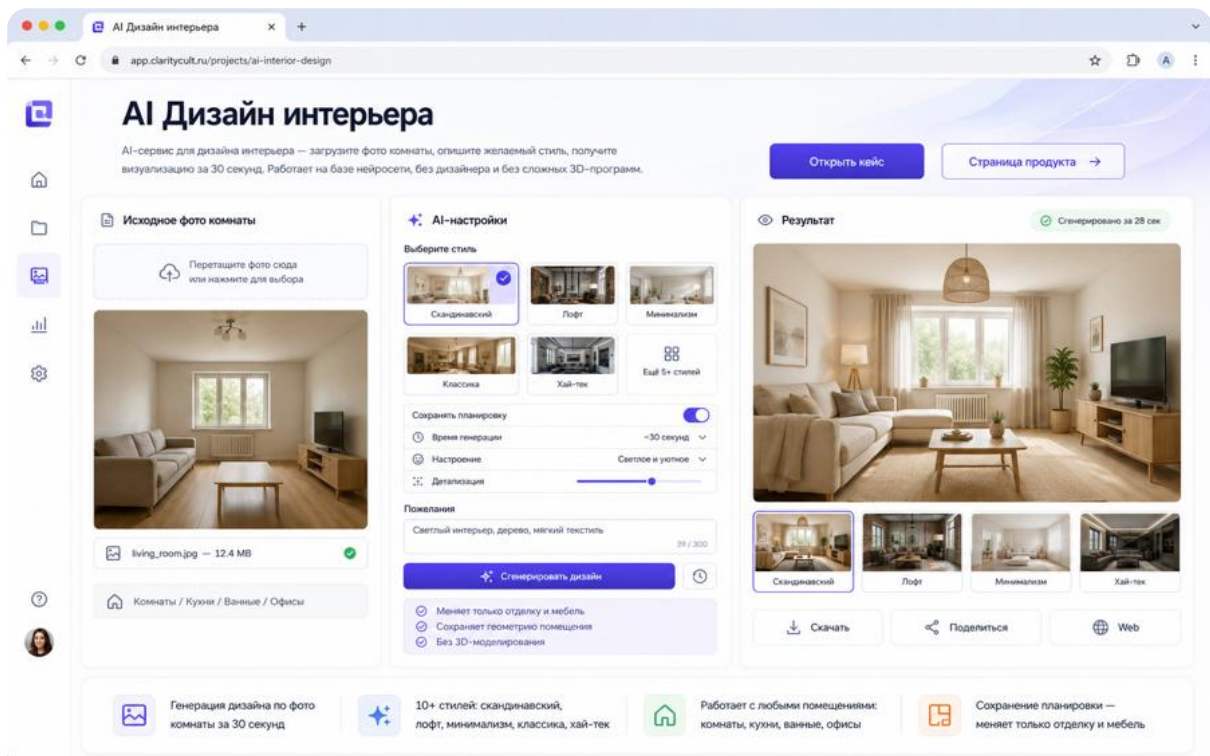
Python

Flask

Image generation API

Object storage

Публичная страница прототипа



Задача

Для первичного обсуждения ремонта клиенту сложно подобрать референс, который соответствует именно его комнате. Требовался простой прототип: загрузить фотографию и быстро сравнить несколько направлений оформления.

Решение

Контролируемый запрос к генерации

Система отделяет параметры помещения от пожеланий по стилю и формирует задание так, чтобы сохранить планировку, окна и основные объекты.

Экран сравнения вариантов

Исходная фотография и результаты показываются рядом. Пользователь может изменить стиль или уточнить отдельную зону, не начиная бриф заново.

Результат

Прототип проверяет сценарий до разработки большого редактора

Можно оценить качество сохранения геометрии и понять, какие настройки действительно нужны пользователю.

Подход подходит для встраивания в сайт недвижимости

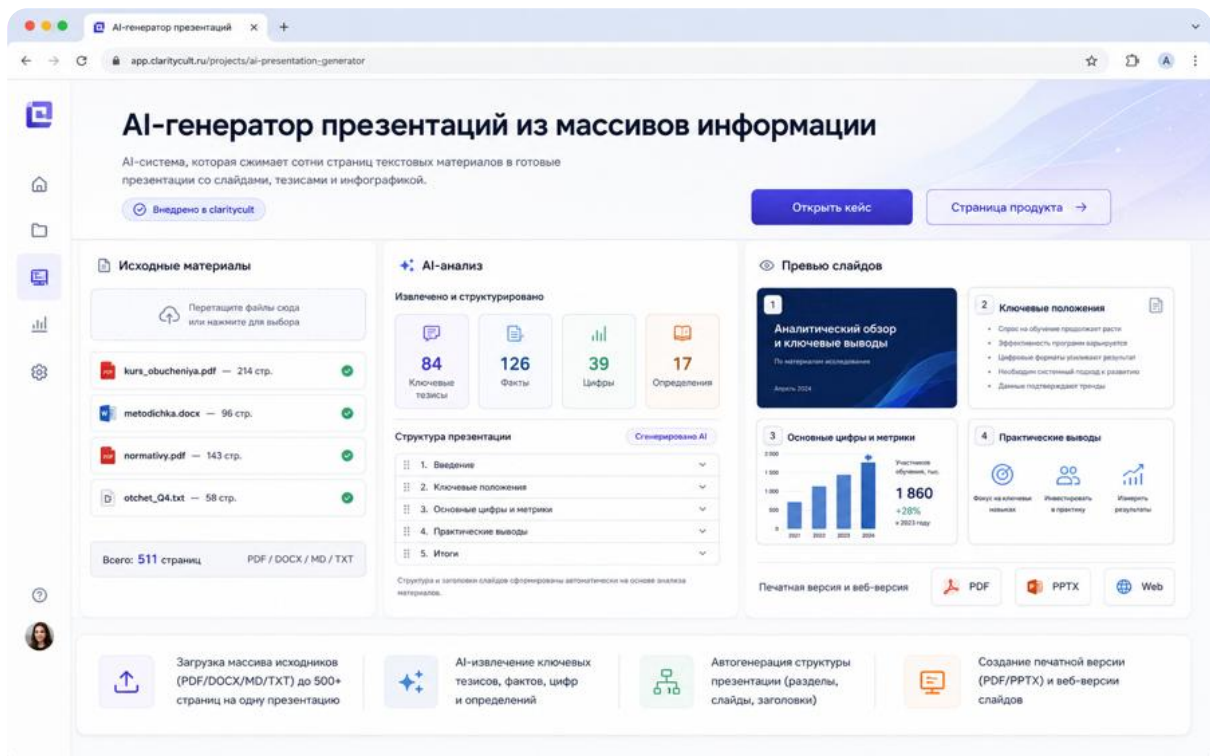
После проверки модели тот же процесс можно подключить к кабинету застройщика, мебельной компании или дизайн-студии.

[Открыть проект](#)[Страница продукта](#)

Генератор презентаций из больших документов

Система разбирает исходные материалы, строит план, готовит слайды и оставляет редактору управляемый черновик перед экспортом.

[AI-генератор презентаций из массивов информации](#)[ClarityCult](#)[2026](#)[Python](#)[FastAPI](#)[React](#)[PostgreSQL](#)[python-pptx](#)[LLM API](#)[Интеграция с платформой ClarityCult](#)



Задача

Учебные материалы и методички могут занимать сотни страниц. Простое сокращение текста не даёт презентацию: нужно выделить структуру, распределить смысл по слайдам и сохранить связь с источником.

Решение

План до генерации слайдов

Система извлекает текст и заголовки, строит оглавление будущей презентации и показывает его редактору. Только подтверждённый план превращается в набор слайдов.

Редактируемый результат

Для каждого слайда сохраняются тезисы, визуальная подсказка и ссылка на исходный раздел. Редактор меняет формулировки и порядок до экспорта в рабочий формат.

Результат

Большой документ превращается в управляемый черновик

Команда сначала работает с планом, затем отдельно проверяет и редактирует каждый слайд.

Материал остаётся проверяемым

При правке можно вернуться к исходному разделу и убедиться, что смысл не потерян.

[Страница продукта](#)

Бот синхронизации заказчика и команды разработки

Бот собирает задачи и статусы из рабочих систем, формирует отчёт по спринту и оставляет заказчику одну актуальную картину проекта.

Бот синхронизации команды

ClarityCult

2026

Python

GitHub API

Telegram Bot API

SMTP

Markdown

LLM API

GitHub Issues и проектная документация

ClarityCult — Спринты

Прогресс по ТЗ «Полные Требования для Бета-тестирования» · всего открытых задач: 57 · обновление 04.05.2026, 18:45

Спринт 11 завершается сегодня (04.05). За неделю — фокус на лендинге (#396, ~30 коммитов: дизайн, hero, мобилка, FAQ, отзывы, видео-модалка), правки PPT/коммуникации (#97), кнопка перехода между блоками (#395). Спринт 12 (05–11.05) — закрываем экспорт-хвосты и Executive Summary. Подробности по каждой задаче — в [GitHub Issues](#).

16

НА ПРОВЕРКЕ

27

ПРИНЯТО

13

ТЕСТИРОВАНИЕ

1

ЗАБЛОКИРОВАНО

7

В РАБОТЕ

57

ВСЕГО ОТКРЫТО

Спринт 12 (5 – 11 мая) — текущий

Направление 1: Executive Summary — устранить data loss и шаблонные тексты

ТЗ: [6.7](#) | Issues: [#344](#), [#345](#), [#98](#)

Цель: правки в Executive Summary не должны слетать при повторном входе (#345); блок SMART-Цели должен подтягивать реальные данные пользователя, а не шаблон (#344). Финализировать приёмку #98.

Направление 2: Экспорт — финализация приёмки

ТЗ: [6.5](#) [6.6](#) [6.7](#) [6.8](#) | Issues: [#95](#), [#97](#), [#99](#), [#379](#), [#388](#)

Цель: добить дебаг Gamma (пропавшие изображения и таймауты после пополнения, #379), исправить extract-result, который сокращает контент вместо сохранения полного текста (#388). Закрывать приёмку Word/PPT/PDF/Скачать стратегию.

Направление 3: Лендинг и UX

Issues: [#396](#), [#395](#)

Цель: завершить сводный список правок лендинга (#396) и финализировать кнопку перехода между блоками (#395) — обе задачи на QA после серии правок в спринте 11.

Заблокировано: #114 (КРИТИЧНО: запрет ИИ использовать чужие данные) — ждёт решения по архитектуре изоляции контекста. | Производительность: [#338](#) (рост контекста чата до 73К+ токенов) — в очереди после Executive Summary.

Задача

В большом бэклоге задачи распределены между issue, чатами и документами. Заказчику сложно понять, что вошло в текущий спринт, какие решения приняты и где требуется его ответ.

Решение

Сбор статусов из первичных систем

Бот читает задачи и изменения из GitHub, связывает их с направлениями проекта и отмечает блокеры. Исходные issue остаются главным местом технической работы.

Отчёт для заказчика

Из актуальных статусов формируется короткая сводка: завершённое, работа в процессе, риски и вопросы. Ссылки ведут к исходной задаче или документу.

Результат

Одна версия прогресса для всех участников

Команда и заказчик обсуждают один и тот же набор задач, а не разные пересказы в чатах.

История решений не исчезает после созвона

Спринтовые отчёты и вопросы сохраняются в проектной документации и доступны при планировании следующего этапа.

[Страница продукта](#)

Telegram-консультант по требованиям пожарной безопасности

Бот помогает пройти расчётный сценарий по СП 4.13130, показывает использованные условия и отделяет справочную подсказку от инженерного заключения.

AI-консультант по пожарной безопасности

2026

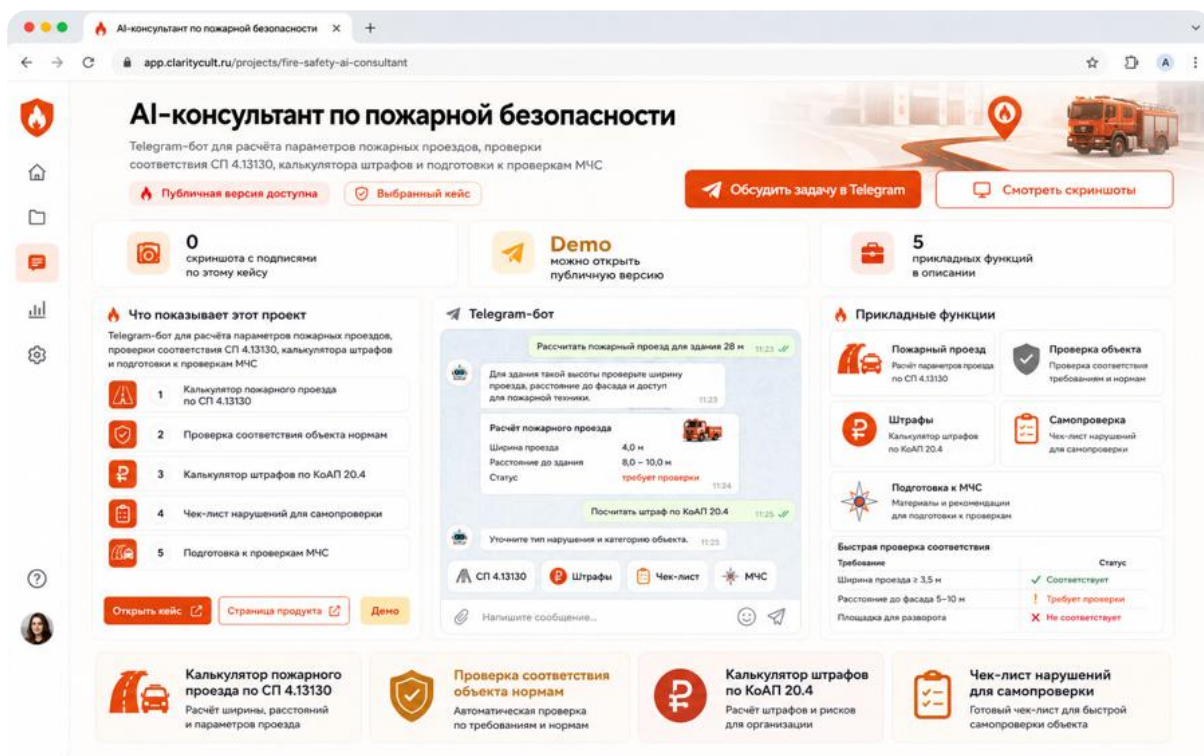
Node.js

Telegram Bot API

Rule engine

OpenAI-compatible API

@fire_safety_sp4_bot



Задача

Вопрос о пожарном проезде зависит от типа объекта, высоты здания и нескольких условий нормы. Свободный ответ AI без пошагового расчёта трудно проверить и опасно воспринимать как готовое заключение.

Решение

Диалог как расчётная форма

Бот последовательно собирает необходимые параметры и не переходит к результату, пока обязательные данные не указаны.

Правила отдельно от пояснения

Числовые условия и ветвления выполняются детерминированно, а AI используется для объяснения результата обычным языком. Пользователь видит, какие исходные данные применены.

Результат

Предварительная проверка становится воспроизводимой

Один и тот же набор параметров приводит к тому же расчётному пути, который можно показать специалисту.

Граница ответственности обозначена в интерфейсе

Бот помогает подготовиться к консультации, но не подменяет проектировщика или официальную экспертизу.

[Открыть проект](#)[Страница продукта](#)

SimpleDashboard: конструктор бизнес-дашбордов

Платформа превращает описание задачи и подключённые данные в адаптивный дашборд с KPI, графиками, таблицами и доступом для команды.

[AI-конструктор бизнес-дашбордов](#)[2026](#)[HTML](#)[JavaScript](#)[Chart.js](#)[Tailwind CSS](#)[Dashboard SDK](#)

15 публичных сценариев дашбордов

CRM Панель

Активные сделки

Обзор

Воронка

Сделки

Команда

Автозадачи

Аналитика

Календарь

Обзор

Активные сделки

Общая выручка

Средний чек

Конверсия

Просрочено задач

Завершено в этом месяце

Сделки по статусам

По типу услуг

Динамика выручки

Последние сделки

Клиент	Тип	Статус	Бюджет	Площадь	Дедлайн
Соколов А.	Комплексный	Работа: Комплекс	5 488 000 Р	99 м²	2026-06-10
Петров Д.	Комплексный	Работа: Комплекс	9 788 000 Р	252 м²	2026-05-02
Иванова М.	Строительство	Расчёт стоимости	1 525 000 Р	320 м²	2026-07-19
Смирнов К.	Строительство	Расчёт стоимости	5 747 000 Р	123 м²	2026-04-02
Козлова Е.	Строительство	Обратная связь	7 115 000 Р	177 м²	2026-06-05
Новиков С.	Строительство	Новый лид	3 123 000 Р	171 м²	2026-06-13
Морозова Т.	Строительство	Работа: Стройка	7 165 000 Р	348 м²	2026-09-20
Волков Р.	Поставка	Расчёт стоимости	806 000 Р	63 м²	2026-08-09

Последние действия

Смета согласована

Новый лид создан

Новый лид создан

Договор подписан

Смета согласована

Задача просрочена

Акт принят

«Сделай CRM-дашборд для строительной компании с 8 статусами, 15 ролями и автозадачами»

Хочу такой же

Задача

Командам нужны разные представления одних данных: продажи, операции, маркетинг и финансы. Ручная разработка каждого прототипа замедляет обсуждение, особенно пока состав метрик ещё меняется.

ProjectFlow

Управление проектами

Проекты

Мои задачи

Календарь

Отчёты

PROJECTS

Website Redesign

Mobile App v2

Brand Identity

Website Redesign

Клиент: Acme Corp.

67%

Бэклог

В работе

На review

Готово

Build API Integration

Высокий

В работе

Исполнитель

Срок

Подзадачи (2/5)

Трекер времени

02:14:33

Записано: 6h 32m

Комментарии

Управляйте проектами с AI — автоматизация задач и сроков

Попробовать

Канбан, карточка задачи и учёт времени в демонстрационном сценарии

Решение

Бриф и данные в одном диалоге

Пользователь описывает роли и метрики, загружает CSV или подключает API. Система предлагает структуру экранов и визуализации, которые можно уточнить до публикации.

Готовый адаптивный интерфейс

Результат собирается как SPA с KPI, графиками, таблицами и прямыми ссылками на разделы. Доступ и обмен с командой подключаются через платформенный SDK.

Результат

Прототип можно обсуждать на живом интерфейсе

Вместо макета команда открывает рабочие экраны, проверяет навигацию и корректирует состав данных.

Подход проверен на разных бизнес-сценариях

В репозитории собраны публичные примеры для продаж, проектов, e-commerce, HR, агентств и SaaS-метрик.

[Открыть проект](#)[Страница продукта](#)

Прототип AI-ассистента для Excel и SQL на русском языке

Сотрудник задаёт вопрос обычными словами, а система строит проверяемое вычисление для таблицы или запрос к базе данных.

Talk to Excel / Talk to SQL — AI-ассистент для табличных данных

Прототип, 2026

Python

PostgreSQL

pandas

SQLAlchemy

LLM API

Excel, CSV и SQL-базы

Talk to Excel / Talk to SQL — AI-ассистент для табличных данных

Русскоязычная платформа, где сотрудники задают вопросы к Excel/CSV/Google Sheets и SQL-базам и получают ответы, графики и готовые SQL-запросы. Такую систе...

Прототип Talk to data

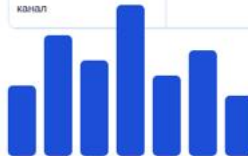
SQL для проверки

Сравни маржу по каналам за выбранный период

```
SELECT channel, margin
FROM orders
WHERE date BETWEEN :start AND :end
ORDER BY margin DESC;
```

Ответ строится после выполнения запроса

channel	orders	revenue	margin	source
канал				проверено SQL
канал				проверено SQL
канал				проверено SQL
канал				проверено SQL
канал				проверено SQL
канал				проверено SQL



Задача

Бизнес-вопрос часто формулируется быстрее, чем готовится выгрузка или SQL-запрос. При этом ответ AI без показанной формулы или SQL нельзя надёжно использовать в отчёте.

Решение

Разные движки для таблиц и базы

Для Excel и CSV система строит операции над загруженным набором данных. Для SQL она сначала читает разрешённую схему и генерирует запрос только к доступным таблицам.

Проверка до выполнения

Пользователь видит интерпретацию вопроса, формулу или SQL и может исправить условие. Опасные операции блокируются, рабочий режим ограничивается чтением.

Результат

Ответ сопровождается способом расчёта

Вместе с числом пользователь получает таблицу результата, запрос и параметры фильтрации.

Аналитик подключается к сложным вопросам, а не к каждой выгрузке

Повторяющиеся запросы сохраняются как шаблоны и доступны сотрудникам с нужной ролью.

[Страница продукта](#)

Платформа запуска Telegram-ботов под бизнес-сценарии

Общий runtime позволяет запускать ботов с отдельными настройками, данными, AI-провайдером, журналами и административным управлением.

Разработка Telegram Ботов под ключ

2026

Node.js

TypeScript

Telegram Bot API

PM2

OpenAI-compatible API

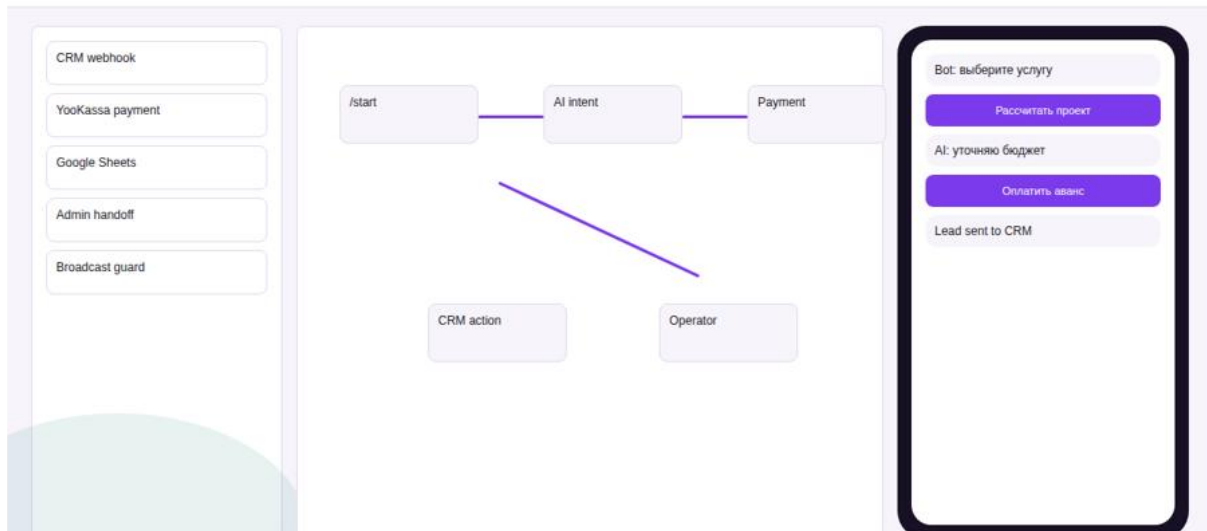
Изолированные настройки, история диалогов и health-check

Разработка Telegram Ботов под ключ

Профессиональная разработка интеллектуальных Telegram-ботов с ИИ, интеграциями и кастомной логикой для автоматизации бизнеса

Telegram Bot Studio

Phone preview



Задача

Каждый новый бот повторяет инфраструктурные задачи: хранение истории, конфигурация модели, ограничения ввода, мониторинг и обновление. Если собирать их заново, сценарии расходятся и сложнее обслуживаются.

Решение

Общий runtime и изолированные данные

Платформа отделяет код доставки сообщений от промпта и бизнес-логики конкретного бота. Для каждого сценария хранятся собственные настройки и история.

Администрирование и эксплуатация

Боты запускаются под PM2, проходят health-check и управляются через административный интерфейс. Модель и лимиты можно менять без переписывания Telegram-интеграции.

Результат

Новый сценарий начинается с бизнес-логики

Команда переиспользует авторизацию, сообщения, retry, журналы и мониторинг вместо копирования инфраструктуры.

Разные боты обслуживаются одинаково

Счётчик питания, нормативный консультант и внутренние ассистенты работают в одном эксплуатационном подходе, сохраняя отдельные данные.

[Страница продукта](#)

Vibers: human-in-the-loop ревью AI-сгенерированного кода

GitHub App связывает коммит, техническое задание и человека-рецензента, который проверяет основной сценарий и отправляет исправление pull requestom.

Vibers — Код-ревью для вайбкодинг-проектов

2026

GitHub App

GitHub Actions

Node.js

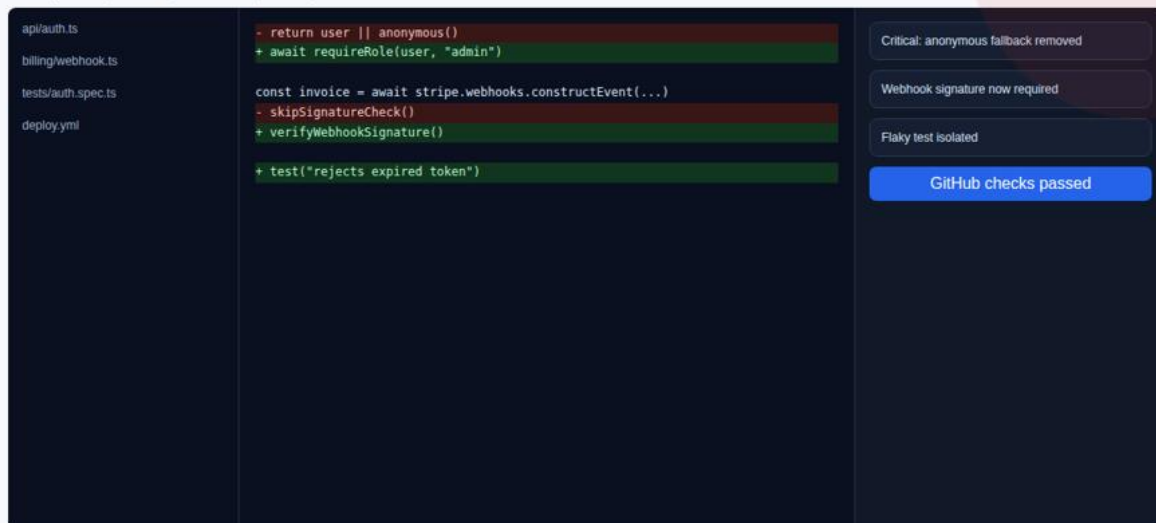
Telegram

Human review

Paycif pull request #2

Vibers — Код-ревью для вайбкодинг-проектов

Сервис код-ревью с участием человека для команд, использующих дешёвых AI-разработчиков. Проверяем коммиты по вашему ТЗ, исправляем проблемы и отправляем р...

[Pull request review](#)[Security findings](#)

Задача

Команда быстро выпускает код с помощью AI, но типичные автоматические проверки не замечают расхождение с ТЗ, сломанный пользовательский путь или несуществующую интеграцию. Нужен был процесс, где ревью заканчивается проверяемым исправлением.

Решение

Контекст из репозитория и спецификации

После push сервис связывает изменение с описанием задачи и передаёт его рецензенту. Рецензент проверяет diff и основной пользовательский сценарий.

Исправление в том же рабочем процессе

Найденные проблемы оформляются в коде и тестах, затем отправляются отдельным pull requestом. Заказчик видит, что изменено и как это проверить.

Результат

Публичный пример дошёл до готового PR

В Paucif ревью выявило проблемы платёжного пути, обработки ошибок и интерфейса; исправления опубликованы в pull request #2.

AI остаётся ускорителем, а не единственным контролёром

Человек проверяет соответствие задаче и фактическое поведение, а автоматизация собирает контекст и доставляет результат.

[Открыть проект](#)[Страница продукта](#)

Связаться

mail@habab.ru · [@onoutnoxon](#)