

//



AI DEVTOOLS

СПИКЕР ОТ КОМАНДЫ ПНВ

АЛИНА
РЫЖЕНКОВА

Backend разработчик



О ЧЕМ БУДЕМ ГОВОРИТЬ



- 01 Задача
- 02 Наша идея
- 03 Описание функционала
- 04 Работа со сценариями
- 05 Наше решение
- 06 Команда ПНВ
- 07 План дальнейшего развития

Текущая ситуация в индустрии QA:

НИЗКАЯ ЭФФЕКТИВНОСТЬ И ПРОДУКТИВНОСТЬ ПРОЦЕССА

- Ручная работа преобладает: 60-70% времени QA-инженеры тратят на написание и поддержку тест-кейсов
- Медленная реакция на изменения: При изменении API или UI тесты устаревают, требуя ручного переписывания

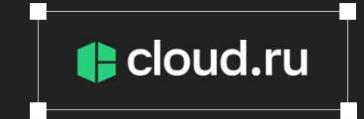
НЕДОСТАТОЧНОЕ КАЧЕСТВО И РИСКИ

- Высокая стоимость ошибок: Пропущенные баги на production стоят компаниям миллионы рублей
- Недостаточное покрытие: Из-за нехватки времени тестируется только "счастливый путь"

ОТСУТСТВИЕ СТАНДАРТИЗАЦИИ И ЕДИНОЙ ЭКОСИСТЕМЫ

- Разрозненные инструменты: Разные команды используют разные форматы, стандарты и инструменты

описание тестовых сценариев для API эндпоинтов



Сценарий 1: GET /vms

Описание: Получение данных из /vms

Шаги теста:

- 1. Подготовить тестовые данные
- 2. Отправить GET запрос на /vms
- 3. Передать параметры: ['limit', 'status']
- 5. Получить и проверить ответ

Сценарий 2: POST /vms

Описание: Создание новой записи через /vms

Шаги теста:

- 1. Подготовить тестовые данные
- 2. Отправить POST запрос на /vms
- 4. Передать тело запроса с минимальными данными
- 5. Получить и проверить ответ

Сценарий 3: DELETE /vms/{vm_id}

Описание: Удаление записи через /vms/{vm_id}

Шаги теста:

- 1. Подготовить тестовые данные
- 2. Отправить DELETE запрос на /vms/{vm_id}
- 5. Получить и проверить ответ

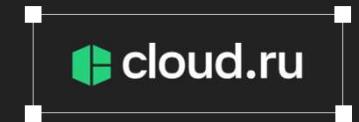
Вывод представляет собой три тестовых сценария для REST API, каждый для своего HTTP метода:

GET /vms - получение списка виртуальных машин

POST /vms - создание новой виртуальной машины

DELETE /vms/{vm_id} - удаление конкретной виртуальной машины

ПРОВЕРКА КЛЮЧЕВЫХ ФУНКЦИЙ



```
# Проверяем, что это OpenAPI спецификация
if not isinstance(spec, dict):
    raise ValueError("OpenAPI файл должен содержать словарь (объект)")

if 'openapi' not in spec and 'swagger' not in spec:
    raise ValueError("Файл не является OpenAPI спецификацией. "
                     "Отсутствует поле 'openapi' или 'swagger'")

# Выводим информацию о версии
if 'openapi' in spec:
    print(f"  Версия OpenAPI: {spec.get('openapi')}")
elif 'swagger' in spec:
    print(f"  Версия Swagger: {spec.get('swagger')}")

endpoints = []

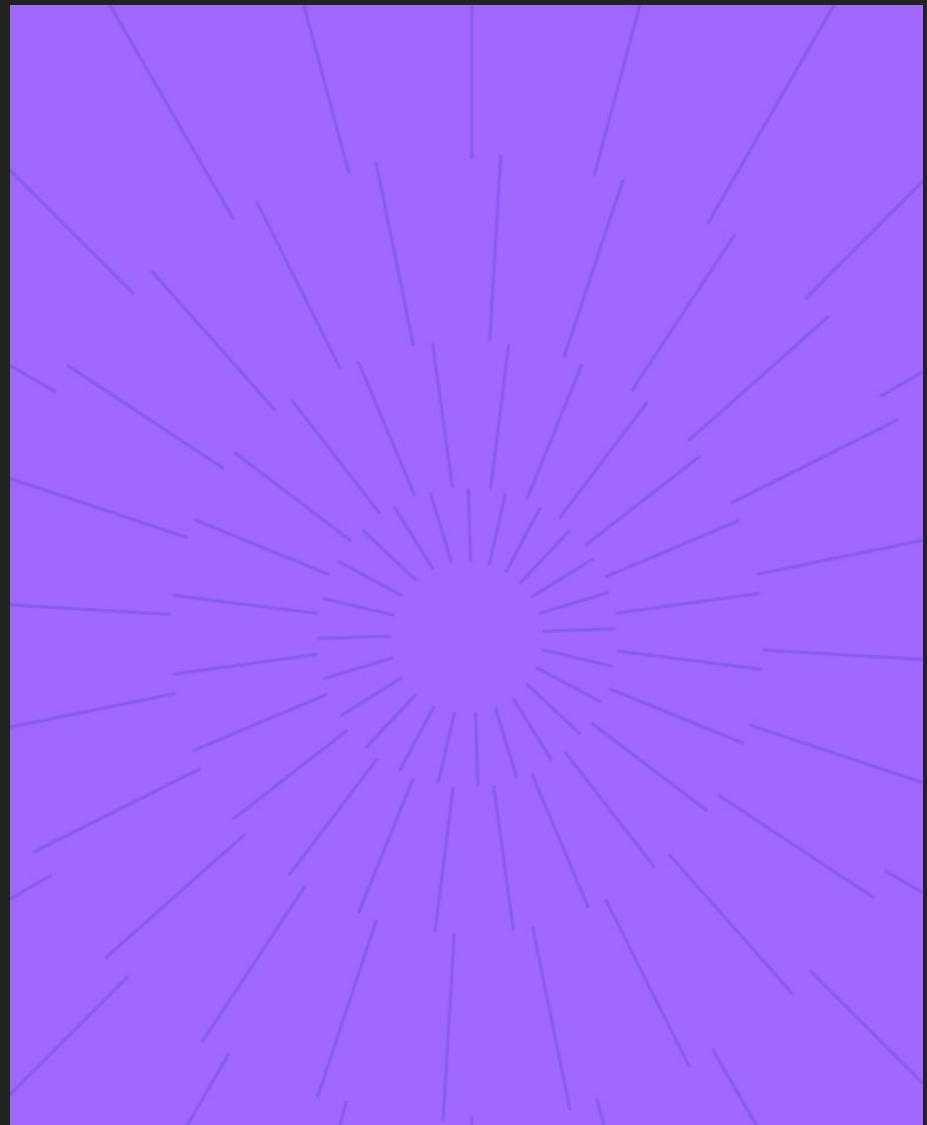
# Извлекаем базовый URL
base_url = ""
if 'servers' in spec and spec['servers']:
    base_url = spec['servers'][0].get('url', '')
    print(f"  Базовый URL: {base_url}")
```

Этот код на Python выполняет валидацию и анализ OpenAPI/Swagger спецификации (обычно файла в формате JSON/YAML).

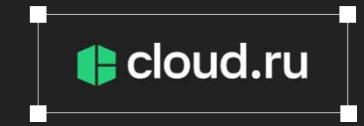
НАШЕ РЕШЕНИЕ

Ассистент будущего

АДАПТИВНЫЙ ТЕСТ-ГЕНЕРАТОР С
МАШИННЫМ ОБУЧЕНИЕМ



КОМАНДА ПНВ



ДЖАСУР
ХОЛДЖИГИТОВ



Backend-
разработчик

АРИНА
ЦЫГУЛИНА



Тимлидер,
аналитик

АЛИНА
РЫЖЕНКОВА



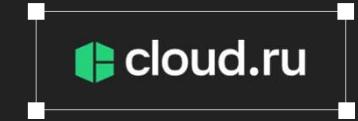
Backend-
разработчик

ЕВГЕНИЯ
ГОГУНСКАЯ



Frontend-
аналитик

ПЛАН ДАЛЬНЕЙШЕГО РАЗВИТИЯ



01

Реализация backend

Реализация ключевых функций, создание рабочего генератора тестов, настройка LLM

02

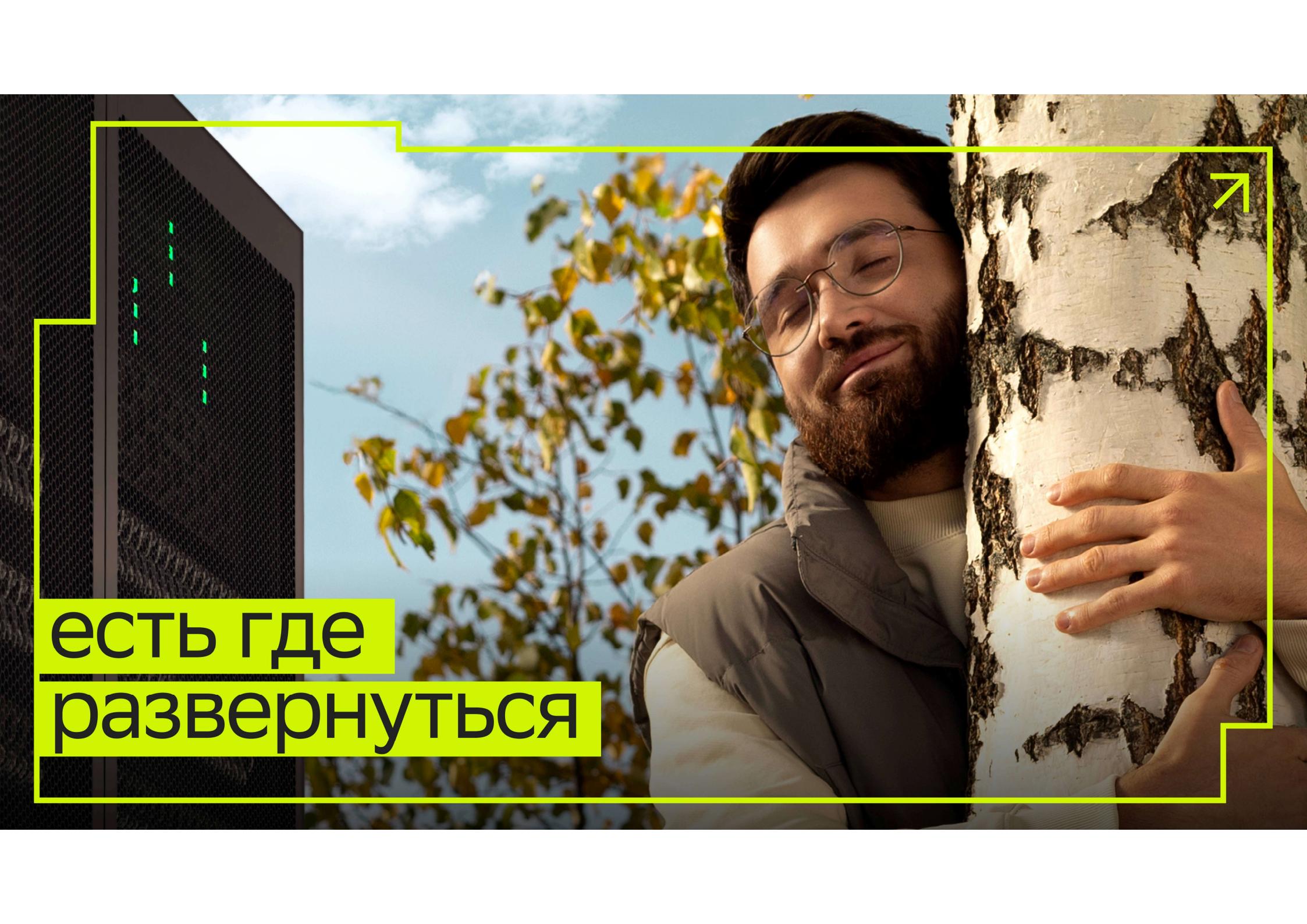
Реализация frontend

Создание проекта React (Vite + TypeScript), создание главной страницы, функции просмотра тестов

03

Совмещение front и back

Объединение предыдущих модулей, создание работоспособной системы, способной к реализации адаптивной генерации тестов



**есть где
развернуться**

//



AI DEVTOOLS

EVOLUTION FREE TIER



СОЗДАВАЙТЕ,
ТЕСТИРУЙТЕ
И РАЗВЕРТЫВАЙТЕ
ПРИЛОЖЕНИЯ
БЕСПЛАТНО



//



AI DEVTOOLS

GitHub:

[https://github.com/ShattIgori/
ai_devtools_hack_pnv](https://github.com/ShattIgori/ai_devtools_hack_pnv)