





Хакатон «Транснефть — Технологии» по искусственному интеллекту

Кейс: **ASR**

Проект: Voice Transcriber Pro

Команда:

Федоров Максим

Федоров Дмитрий

Полубарьев Михаил

Самсоненко Станислав

ПОСТАНОВКА ЦЕЛИ И ЗАДАЧ

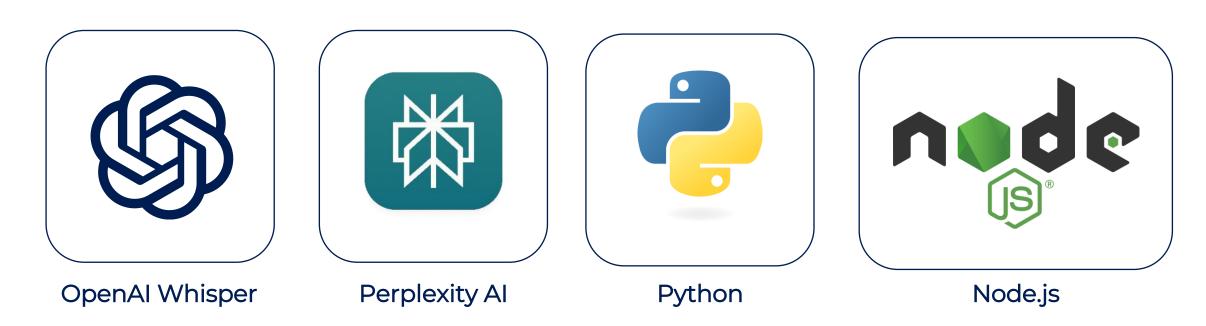
Цель проекта:

 Создать веб-приложение для автоматического распознавания речи и интеллектуальный тайм-трекер, которые преобразуют аудиофайлы в текст и фиксируют трудозатраты по голосовым командам.

Задачи проекта:

- Разработать веб-приложение для преобразования аудиофайлов в текст с поддержкой диаризации и транскрипции в реальном времени.
- Создать интеллектуальный тайм-трекер для фиксации трудозатрат сотрудников на основе голосовых команд, с выводом данных в структурированном формате.

ВЫБОР ТЕХНОЛОГИЙ РАЗРАБОТКИ

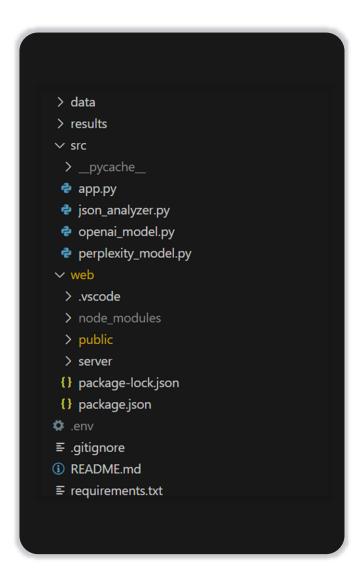


Whisper используется для автоматического распознавания речи и преобразования аудио в текст. Python обеспечивает реализацию серверной логики, включая обработку данных и интеграцию моделей. Node.js используется для создания веб-сервера и управления пользовательским интерфейсом. Perplexity применяется для анализа транскрипции и структурирования данных в удобном формате.

АРХИТЕКТУРА ПРИЛОЖЕНИЯ

Архитектура приложения состоит из следующих компонентов:

- **Веб-интерфейс** реализован на HTML, CSS и JavaScript, обеспечивает загрузку аудиофайлов, отображение результатов транскрипции и формализованных данных.
- Серверная часть Flask (Python) обрабатывает запросы от вебинтерфейса, управляет транскрипцией аудио с использованием модели Whisper и анализирует данные через Perplexity API.
- Модели и обработка данных: Whisper используется для преобразования аудио в текст, Perplexity API — для структурирования текста в формате JSON.
- Результаты транскрипции и анализа сохраняются в **JSON-файлах** на сервере для дальнейшего использования.



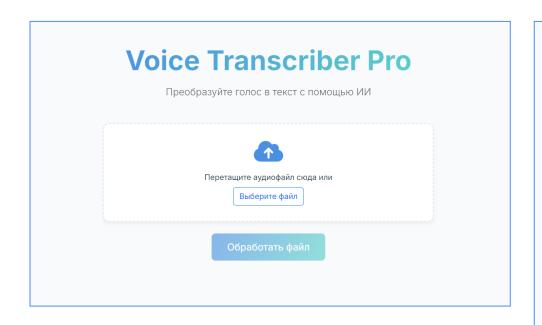
СЕРВЕРНАЯ ЧАСТЬ ПРИЛОЖЕНИЯ

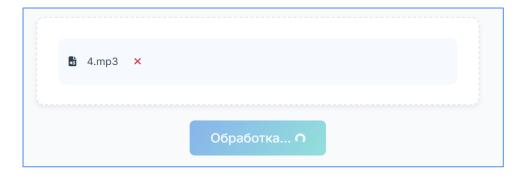
```
127.0.0.1 - - [13/Dec/2024 15:24:46] "OPTIONS /transcribe-audio HTTP/1.1" 20 Используемая модель:
Filename: ..\data\2.mp3
Model_name: Whisper large
Company: OpenAI
Architecture: Transformer кодировщик-декодер
Готово!
Результат транскрипции сохранен в файл: ..\results\transcription.json
```

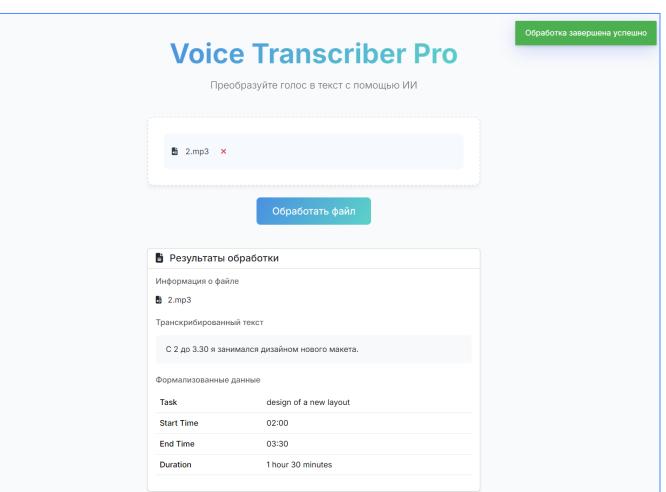
```
"text": " Я проводил обучение сотрудников с часа дня до половины третьего.",
"segments": [
        "id": 0.
        "seek": 0.
       "start": 0.0,
       "end": 3.66,
       "text": " Я проводил обучение сотрудников с часа дня до половины третьего.",
       "tokens": [ ···
        "temperature": 0.0,
        "avg_logprob": -0.03095232021240961,
        "compression ratio": 1.2688172043010753,
        "no_speech_prob": 0.05505261942744255
"language": "ru",
"model info": {
    "filename": "..\\data\\5.mp3",
    "model name": "Whisper large",
    "company": "OpenAI",
    "architecture": "Transformer кодировщик-декодер"
"audio_file_name": "5.mp3"
```

```
POST
                localhost:5000/transcribe-audio
       Authorization Headers (8)
                                   Body • Scripts
○ none ○ form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL JSON ∨
         "filename": "1.mp3"
Body Cookies Headers (6) Test Results
                                        JSON V
                 Preview Visualize
 Pretty
          "formalized_data": {
              "duration": "2 hours 30 minutes",
              "end time": "12:30",
              "start time": "10:00",
              "task": "worked on a programming project"
           "transcription": {
              "audio_file_name": "1.mp3",
              "language": "ru",
              "model info": {
                  "architecture": "Transformer кодировщик-декодер",
                  "company": "OpenAI",
                  "filename": "..\\data\\1.mp3",
                  "model_name": "Whisper large"
               "segments": [
                       "avg lognrob": -0 09201018583206903
```

ВЕБ-ИНТЕРФЕЙС ПРИЛОЖЕНИЯ



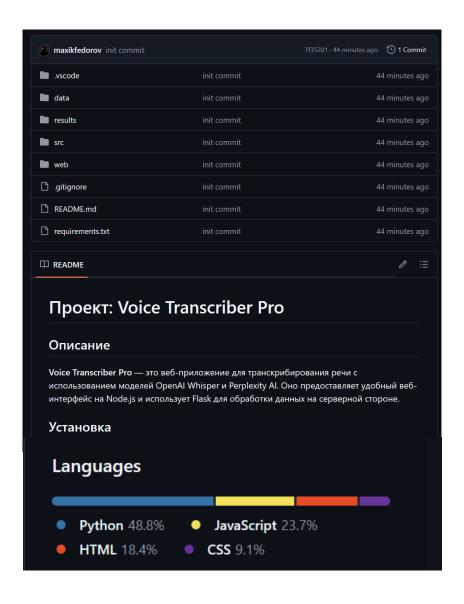




ИТОГИ РАЗРАБОТКИ

- Создано веб-приложение для автоматического распознавания речи с использованием модели Whisper от OpenAI, реализованное на Flask и Node.js.
- Реализован интерфейс для загрузки аудиофайлов и отображения результатов транскрипции и анализа.
- Интегрирован анализатор для структурирования данных транскрипции через Perplexity API.
- Организовано хранение данных в формате JSON, включая результаты транскрипции и формализованные данные.
- Оформлено решение в виде GitHub-репозитория с подробной документацией и инструкциями по использованию.

https://github.com/maxikfedorov/HACK-TRANSNEFT-ASR



СПАСИБО ЗА ВНИМАНИЕ!