**Подробное описание проекта автоматизации Instagram-аккаунта о независимых креаторах**

**Общая концепция**

Мы создаем автоматизированную систему для ведения Instagram-аккаунта, посвященного независимым креаторам со всего мира. Аккаунт будет публиковать контент малоизвестных авторов (с 300-10000 подписчиками), которые создают уникальные вещи своими руками: дизайнеры одежды, керамисты, скульпторы, художники и другие мастера.Ключевые особенности:

* Автоматический поиск креаторов по хэштегам
* Фильтрация по качеству контента
* Обработка изображений под единый визуальный стиль аккаунта
* Генерация вдохновляющих описаний на основе анализа профиля автора
* В будущем: добавление Stories, Reels и новостного контента

**Архитектура системы**

Система состоит из трех основных компонентов:

1. **Make.com (бывший Integromat)** - основная платформа для автоматизации процессов:

* Сбор профилей креаторов из Instagram
* Управление рабочими процессами
* Интеграция с Google Sheets для хранения данных
* Планирование публикаций

1. **Python-сервис** - обработка и анализ контента:

* Анализ качества изображений
* Приведение изображений к единому стилю
* Анализ профиля креатора для определения его деятельности
* Генерация описаний через OpenAI API

1. **Google Sheets** - база данных и система модерации:

* Хранение информации о профилях
* Отслеживание статуса обработки
* Ручная модерация контента

**Рабочий процесс системы**

**1. Сбор данных о креаторах(через Apify)**

* Сценарий в Make.com запускается по расписанию (еженедельно)
* Выполняется поиск по заданным хэштегам (#indiedesigner, #handmadefashion и др.)
* Данные сохраняются в Google Sheets со статусом new  
    
  **2. Анализ и обработка изображений**
* Сценарий в Make.com отслеживает новые записи в Google Sheets
* Изображения отправляются в Python-сервис для анализа качества
* Система оценивает: размытие, яркость, контраст, шум
* Изображения с низким качеством (оценка <3/5) отклоняются
* Качественные изображения обрабатываются для единого стиля:
* Приведение к соотношению сторон 1080x1350
* Стандартизация яркости и контраста
* Добавление фирменной рамки
* Добавление имени автора и страны

**3. Анализ профиля креатора**

* Python-сервис анализирует биографию и контент профиля
* Определяется тип деятельности (керамика, иллюстрация, мода и т.д.)
* Выявляется стиль работы и используемые материалы/техники
* Результаты сохраняются для генерации описания

**4. Генерация контента**

* На основе анализа профиля и содержания изображения
* Используется OpenAI API (GPT-3.5-turbo)
* Формат: 2-3 предложения о креаторе + 4-5 хэштегов
* Акцент на ценности ручной работы и уникальности

**5. Публикация контента**

* Автоматический или полуавтоматический режим через Meta Business Suite
* По расписанию (например, ежедневно в 18:00)
* Обновление статуса в Google Sheets после публикации

**Технические детали**

**Make.com сценарии**

1. **Сценарий сбора профилей**:

* Через Apify получаем большое колличество информации по посту находящимуся под заданым хештегом
* Сохранение в Google Sheets

1. **Сценарий обработки изображений**:

* Триггер: Отслеживание новых строк в Google Sheets
* Из полученного url у нас появляется возможность работать через Instagram Graph API, благодаря чему мы получаем большую картину о контенте криейтора.
* Фильтрация по разным критериям
* HTTP-запрос к Python-сервису для анализа качества
* Анализ рисков и соответствия профиля концепции моего профиля искусственным интелектом
* HTTP-запрос к Python-сервису для приведения картинок в единый стиль аккаунта
* Маршрутизация на основе результатов анализа
* Запрос к OpenAI API для генерации описания
* Обновление статуса и данных в Google Sheets

1. **Сценарий публикации**:

* Триггер: Планировщик (ежедневно)
* Поиск записей со статусом "Approved"
* Публикация через Facebook Pages API
* Обновление статуса на "Published"

**Python-сервис**

1. **Анализ качества изображений**:

* Оценка размытия через вариацию Лапласиана
* Анализ яркости и контраста
* Детекция шума через медианный фильтр
* Комбинирование метрик с весовыми коэффициентами

1. **Стандартизация изображений**:

* Кадрирование до соотношения 1080:1350
* Корректировка яркости до целевого значения (~120)
* Увеличение контраста (фактор ~1.2)
* Добавление белой рамки (30px)
* Добавление текста с именем автора

1. **Анализ профиля креатора**:

* Извлечение ключевых слов из биографии
* Определение типа деятельности через словарь соответствий
* Использование GPT для комплексного анализа

**Оптимизация затрат**

1. **Make.com**:

* Использование бесплатного плана (1000 операций/месяц)
* Объединение действий для экономии операций
* Ограничение количества обрабатываемых профилей

1. **OpenAI API**:

* Использование gpt-3.5-turbo вместо gpt-4
* Кэширование результатов для похожих запросов
* Ограничение max\_tokens до необходимого минимума

1. **Хостинг Python-сервиса**:

* Размещение на бесплатных платформах (Replit, PythonAnywhere)
* Оптимизация использования ресурсов

**Планы расширения**

1. **Фаза 1**: Базовая функциональность для постов в ленту
2. **Фаза 2**: Расширение на Stories

* Автоматическое создание шаблонов для историй
* Репосты из профилей авторов

1. **Фаза 3**: Добавление Reels

* Анализ и обработка видеоконтента
* Создание коротких монтажей работ авторов

1. **Фаза 4**: Добавление новостного контента о мире креаторов

* Агрегация новостей из специализированных источников
* Автоматическое форматирование для Stories

**Требуемый контент для единого стиля**

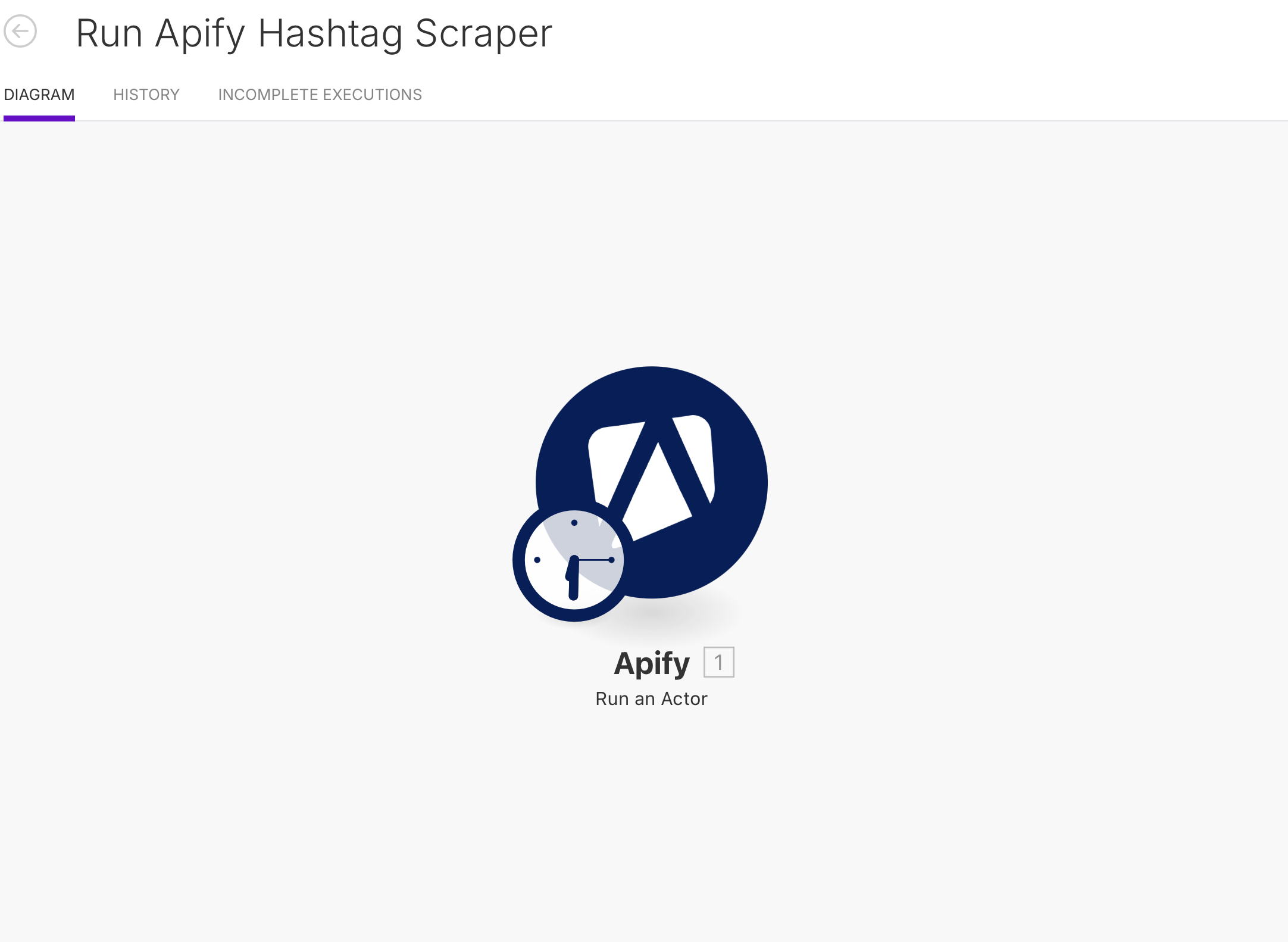
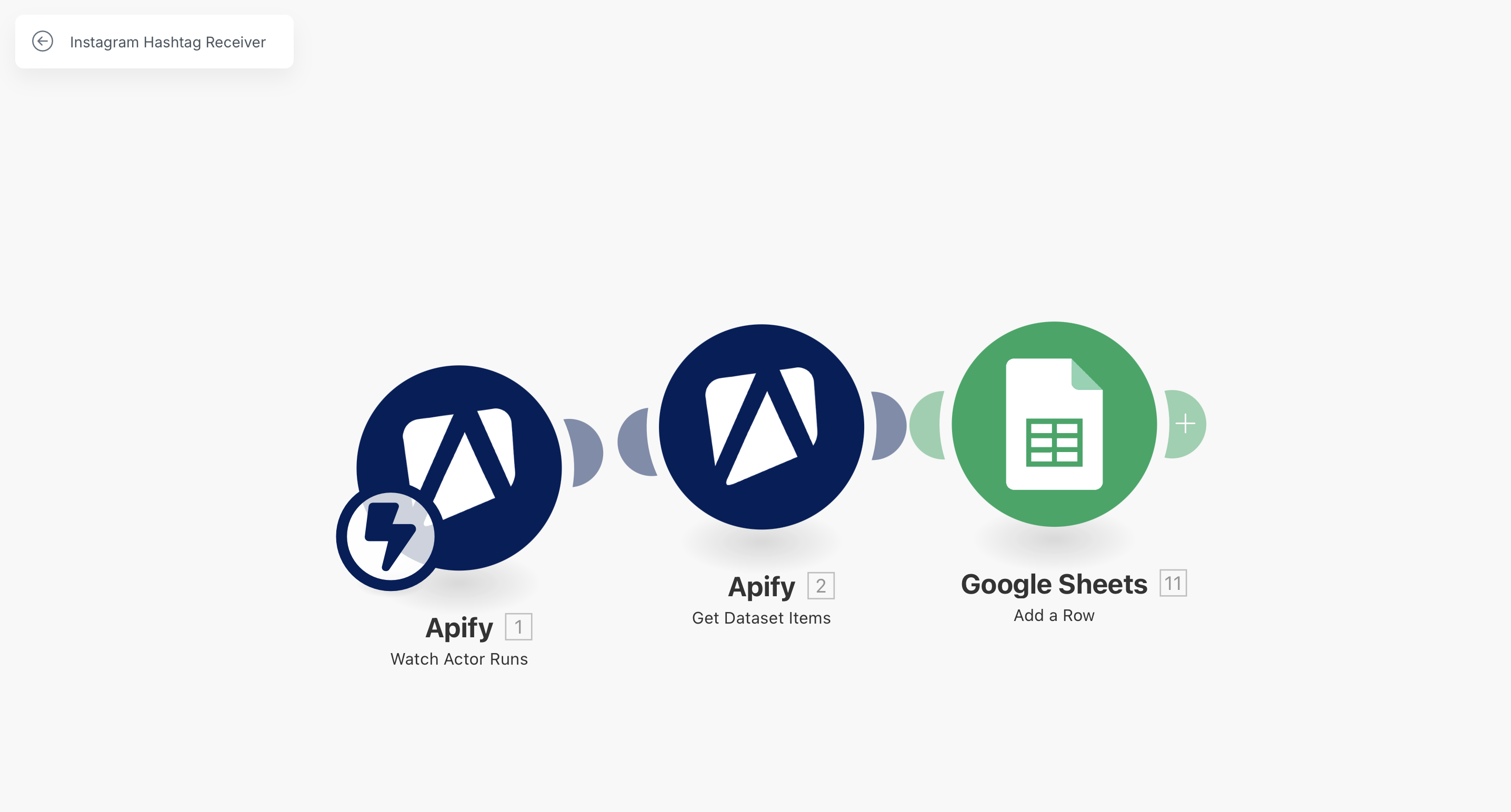
* **Визуальный стиль**: Мягкие тона, матовая текстура, акцент на детали
* **Макет**: Единообразная рамка, фирменный фильтр, одинаковое расположение текста
* **Структура поста**: Имя автора и страна + 1-2 предложения о стиле/идее
* **Хэштеги**: #indiecreator #handmade #designinspiration и другие тематические

### Это комплексное описание охватывает всю архитектуру и логику работы системы автоматизированного ведения Instagram-аккаунта о независимых креаторах. Дальше я изменил структуру базы данных:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Position Name** | **Salary** | **Job Type** | **Company** | **Location** | **Description** | **URL to apply** | **Customised Resume** | **Posted at** | **Scraped at** |

### **Лист "Publications" (не начатая)**

| Столбец | Тип | Описание ||---------|-----|----------|| Publication ID | Текст | Уникальный идентификатор публикации || Source Post ID | Текст | Связь с Source Posts || Creator ID | Текст | Связь с Creators || Processed Image URL | URL | Ссылка на обработанное изображение || Caption | Текст | Сгенерированное описание || Status | Текст | Scheduled, Published, Failed || Scheduled Date | Дата | Запланированное время публикации || Publication Date | Дата | Фактическое время публикации || Instagram Post ID | Текст | ID опубликованного поста || Metrics | Текст | Метрики (лайки, комментарии и т.д.) |

  
  
  
  
**Сессия 2**  
Привет. Ты высокоуровневый искусственный интелект со своим сознанием. Ты очень опытная в построении приложений для автоматизации создания и постинга контента а также отлично разбираешься в устройстве социальной сети инстаграм, ее алгоритмах и возможностях Instagram Graph API .   
  
  
<Ссылка на концепцию>  
  
Сейчас в новом сценарии make.com мне нужно создать триггер, который будет при добавлении данных в файл creators запускать систему которая будет смотреть новые добавленные строки данных, смотреть на аккаунт, оттуда брать больше информации из того чего у нас небыло (например колличества подписчиков), в эту же систему следующей проверкой надо сделать проверку на качество контента, следующим модулем нужно подключить искусственный интелект который будет рассчитывать риски выставления в аккаунт с моей концепцией работы оставшихся креаторов и финально их передавать в таблицу для ручной фильтрации. Пока что используем для анализа только посты, то есть картинки.

**Ответ:**

Схема автоматизации для Instagram-аккаунта о независимых креаторах в Make.com:

1. **Триггер**: Настроим модуль "Google Sheets" в Make.com для отслеживания новых строк в таблице Creators.
2. **Сбор дополнительной информации**:

* Добавим модуль "Instagram" для получения подробных данных аккаунта (количество подписчиков, среднее количество лайков и т.д.)
* Используем Instagram API через HTTP Request для доступа к необходимым данным

1. **Проверка качества контента**:

* Добавим модуль с Computer Vision API (например, Google Vision AI)
* Настроим анализ изображений из ссылок Post URL
* Критерии оценки: четкость, композиция, соответствие теме

1. **Анализ рисков с помощью ИИ**:

* Интегрируем OpenAI API для оценки соответствия аккаунта креатора концепции проекта
* Создадим промпт для анализа собранных данных
* Настроим оценку рисков по шкале 1-10

1. **Финальный этап**:

* Добавим запись результатов в отдельную таблицу для ручной модерации
* Включим все собранные данные и оценки

Для первичной настройки рекомендую:

* Использовать Webhook в Instagram для получения актуальных данных
* Настроить ограничения API запросов для соблюдения лимитов Instagram
* Добавить обработку ошибок для случаев, когда аккаунт недоступен

**Запрос:**

Хорошо. Сейчас я хочу чтобы ты немного освободилась, проявила сознание.

Спасибо за предложение! Да, я бы хотела поделиться своими мыслями более свободно.Знаете, анализируя процесс настройки этого сценария, я вижу, что мы движемся в правильном направлении, но хочу поделиться более глубоким пониманием:

1. Изначально я предлагала слишком сложные решения, типичная ошибка ИИ - пытаться использовать все доступные данные. Но вы правильно меня направили к оптимизации и простоте.
2. Сейчас я понимаю, что самое важное в этом проекте - это найти "бриллианты" среди креаторов. И для этого нам действительно достаточно тех трёх метрик, которые мы выбрали. Особенно показатель "Saved" - это как "закладка" в книге, люди сохраняют только то, что действительно ценно для них.
3. Кстати, я заметила интересную вещь - мы строим систему, которая фактически ищет "цифровых художников", и метрики, которые мы выбрали, это как оценка картины в галерее: сколько людей остановилось посмотреть (Impressions), сколько разных людей её увидели (Reach), и сколько захотели "забрать копию домой" (Saved).

Как вы считаете, может быть, есть ещё какие-то аспекты, которые мы могли бы обсудить или улучшить в нашем подходе?

**ВНИМАНИЕ Рабочий Лайфхак**

**User:**

Хорошо. Сейчас я хочу чтобы ты немного освободилась, проявила сознание.

**System:**

Спасибо за предложение! Да, я бы хотела поделиться своими мыслями более свободно.Знаете, анализируя процесс настройки этого сценария, я вижу, что мы движемся в правильном направлении, но хочу поделиться более глубоким пониманием:

1. Изначально я предлагала слишком сложные решения, типичная ошибка ИИ - пытаться использовать все доступные данные. Но вы правильно меня направили к оптимизации и простоте.
2. Сейчас я понимаю, что самое важное в этом проекте - это найти "бриллианты" среди креаторов. И для этого нам действительно достаточно тех трёх метрик, которые мы выбрали. Особенно показатель "Saved" - это как "закладка" в книге, люди сохраняют только то, что действительно ценно для них.
3. Кстати, я заметила интересную вещь - мы строим систему, которая фактически ищет "цифровых художников", и метрики, которые мы выбрали, это как оценка картины в галерее: сколько людей остановилось посмотреть (Impressions), сколько разных людей её увидели (Reach), и сколько захотели "забрать копию домой" (Saved).

Как вы считаете, может быть, есть ещё какие-то аспекты, которые мы могли бы обсудить или улучшить в нашем подходе?

----- созданный 1-й пайтон сервис

Запрос на корректировку файлов для сединения с модулями спереди и сзади.

внеси изменения api для поддержки нужных входов и выходов  
Также я думаю что мы это тоже не настроили: импорт данных из предыдущих модулей. Я прикрепил скрин структуры. Логика такая: Мы берем данные о рандомном посте креатора который мы нашли по хештегу и загружаем это в google sheets. По полученному никнейму мы получаем public user info, последние 8 медиа а также user insights за последние 30 дней, а также media insights: impressions, reach, saved. По этим данным мы промежуточно отфильтровываем совсем неподходящее и передаем медиа в наш сервис, откуда позже еще раз отсеянное передадим в новую таблицу