**Техническое задание для Python-разработчика: Веб-скрапинг данных участников Eurobike Frankfurt**

Мы ищем опытного Python-разработчика для выполнения тестового задания по веб-скрапингу. Ваша задача — разработать скрипт для извлечения информации об участниках предстоящего мероприятия Eurobike Frankfurt с официального веб-сайта.

**Сайт-источник:** https://eurobike.com/frankfurt/de.html

**Цель задания:**

Разработать решение на Python, которое автоматизирует сбор ключевой информации об участниках выставки Eurobike Frankfurt (проходящей с 24 по 29 июня).

**Требуемые данные для извлечения:**

Для каждого участника необходимо получить следующие поля:

1. **Название компании:** Полное наименование компании-участника.
2. **Краткое описание:** Краткое текстовое описание деятельности компании или ее специализации (если доступно).
3. **Страна:** Страна регистрации или происхождения компании.
4. **Сайт:** Официальный веб-сайт компании.
5. **Email:** Контактный адрес электронной почты компании (если доступен).
6. **Телефон:** Контактный телефон компании (если доступен).

**Технические требования:**

* **Язык программирования:** Python 3.x.
* **Библиотеки:** Используйте соответствующие библиотеки для веб-скрапинга (например, requests для HTTP-запросов, BeautifulSoup или lxml для парсинга HTML). Допускается использование Scrapy или других фреймворков для скрапинга, если разработчик считает это оптимальным.
* **Сохранение данных:** Извлеченные данные должны быть сохранены в реляционную базу данных. Предпочтительно использовать **SQLite**, но допустимо использование PostgreSQL или другой СУБД, если это обосновано.
  + Создание таблицы с соответствующими полями для хранения полученных данных.
  + Реализация механизма сохранения данных в эту таблицу.
* **Обработка ошибок:** Скрипт должен быть устойчивым к возможным ошибкам (например, отсутствию данных для некоторых полей, ошибкам сети).
* **Идентифицируемость:** Код должен быть чистым, хорошо структурированным и содержать необходимые комментарии.
* **Запуск:** Предоставьте четкие инструкции по установке зависимостей и запуску вашего скрипта.
* **Уважение к сайту-источнику:** Решение должно учитывать базовые правила этикета веб-скрапинга (например, использование time.sleep() для задержек между запросами, анализ robots.txt сайта, если он доступен).

**Ограничения и условия:**

* **Время выполнения:** На выполнение тестового задания отводится **16 рабочих часов**.
* **Отсутствие Hardcode:** Решение должно быть гибким и не содержать "жестко закодированных" значений, которые могут измениться (например, прямые ссылки на конкретные страницы, если данные распределены по пагинации).
* **Масштабируемость (опционально, но приветствуется):** Если это не займет много времени, подумайте о том, как ваше решение можно было бы масштабировать для сбора данных с большего числа страниц или аналогичных событий в будущем.

**Что мы ожидаем от вас:**

Помимо работающего кода, мы ожидаем:

* **Репозиторий:** Ссылку на публичный Git-репозиторий (GitHub, GitLab и т.п.) с вашим решением.
* **README.md:** Файл README.md в репозитории с:
  + Краткое описание вашего решения.
  + Инструкции по установке зависимостей и запуску скрипта.
  + Предположения, сделанные вами в процессе разработки (например, если какая-то информация не была доступна или требовала особых условий).
  + Краткий отчет о том, сколько времени вы потратили на выполнение задания.

Мы ищем не только рабочее решение, но и понимание вашего подхода к написанию чистого, эффективного и поддерживаемого кода.

Удачи!