Author: Munei Tshamano

# Delivery II: Sanctions Stream Project Overview

## 📁 Project Overview

This project provides a containerized pipeline for ingesting the **United Nations Security Council Sanctions List** from an Excel file and storing it in a local SQLite database. It also includes streaming ingestion script for collecting tweets related to sanctions.

Script and README can be found in the Delivery II folder

Please see **SanctionsStream_Code_Overview.txt** if you have issues opening the code

## 1. 🧶 ingest_sanctions.py

**Purpose**:
Reads the Excel file, removes empty rows, and stores the cleaned data in a SQLite database (sanctions.db).

**Usage**:
1. Place the Excel file named 'Consolidated United Nations Security Council Sanctions List.xlsx' in the project directory.
2. Run the script manually:
   python ingest_sanctions.py

```
python ingest_sanctions.py

```

Or run it inside a container (see Docker instructions below).

## 2. 🐳 Dockerfile

**Purpose**:
Defines a lightweight Python environment using python:3.10-slim, installs required dependencies (pandas, openpyxl), and runs the ingestion script.

**Usage**:
1. Build the Docker image:
   docker build -t sanctions-ingestor .

```
docker build -t sanctions-ingestor .

```

2. **Run the container**:
   docker run --rm -v $(pwd):/app sanctions-ingestor

```
docker run --rm -v $(pwd):/app sanctions-ingestor

```

## 3. 📦 docker-compose.yml

**Purpose**:
Simplifies container management by defining a service named sanctions_ingestor.

**Usage**:
1. Ensure your Excel file is in the root directory.
2. Start the service:
   docker-compose up –build

```
docker-compose up --build

```

3. Stop the service:
   docker-compose down

```
docker-compose down

```

## 4. 🐦 streaming_ingest.py

**Purpose**:
Connects to the Twitter API using tweepy, searches for recent tweets about 'sanctions', and stores them in a SQLite database (tweets.db).

**Usage**:
1. Replace 'YOUR_TWITTER_BEARER_TOKEN' with your actual Twitter API bearer token.
2. Run the script:
   python streaming_ingest.py

```
python streaming_ingest.py

```

Note: You must have a valid Twitter Developer account and bearer token to use this script.

## 5. 🗓️ Scheduling

**To automate the ingestion process**:
- Use cron for simple time-based scheduling.
- Use Apache Airflow for more complex workflows and monitoring.