DATA INTERN ASSIGNMENT

Objective: Build an application that collects news articles from various RSS feeds (e.g: listed below), stores them in a database, and categorizes them into predefined categories.

Categories the news item should fall under are:

- Terrorism / protest / political unrest / riot
- Positive/Uplifting
- Natural Disasters
- Others

List of RSS Feeds:

- http://rss.cnn.com/rss/cnn_topstories.rss
- http://qz.com/feed
- http://feeds.foxnews.com/foxnews/politics
- http://feeds.reuters.com/reuters/businessNews
- http://feeds.feedburner.com/NewshourWorld
- https://feeds.bbci.co.uk/news/world/asia/india/rss.xml

Requirements:

- Programming language: Python, NodeJS(Javascript/Typescript)
- Libraries(example for python):
    - Feedparser: For parsing RSS feeds
    - SQLAlchemy: For database interaction (e.g., PostgreSQL)
    - Celery: For managing the task queue
    - Natural Language Processing (NLTK or spaCy) for text classification
- Database: Any relational database (e.g., PostgreSQL, MySQL)

Note: These are examples, please feel free to use any relevant libraries and frameworks.

Additional detail for the Assignment task**:**

1. Feed Parser and Data Extraction:
    - Create a script that reads the provided list of RSS feeds.
    - Parse each feed and extract relevant information from each news article, including title, content, publication date, and source URL.
    - Ensure handling of duplicate articles from the same feed.
2. Database Storage:
    - Design a database schema to store the extracted news article data.
    - Implement logic to store new articles in the database without duplicates.
3. Task Queue and News Processing:
    - Set up a Celery queue to manage asynchronous processing of new articles.

   - Configure the parser script to send extracted articles to the queue upon arrival.
   - Create a Celery worker that consumes articles from the queue and performs further processing:
     - Category classification: Utilize NLTK or spaCy to classify each article into the provided categories.
     - Update the database with the assigned category for each article.

4. Logging and Error Handling:
   - Implement proper logging throughout the application to track events and potential errors.
   - Handle parsing errors and network connectivity issues gracefully.

Deliverables:

- Python code for the application
- Documentation explaining the implemented logic and design choices.
- The resulting data as sqldump, csv or json.

This enhanced test task will comprehensively assess your abilities in:

- Building and managing complex ETL pipelines.
- Building other scheduling and distributed task management infrastructure
- Working with data sources(RSS, APIs,etc)
- Implementing existing machine learning models

Note: This is a general outline of the test task. Feel free to add your own creative flair and showcase your problem-solving and technical skills through the implementation. This should not take you more than a single day.