FINAL PROJECT REPORT
NEWS DATA PIPELINE

Team Members
Omar Alua - 22B030417
Sagatkyzy Firuza - 22B030425
Yesserkey Dana - 23B030349

API Selection and Justification

We use the NewsData.io High-frequency News API (https://newsdata.io/). This API provides real-time news articles categorized by keywords, categories, languages, and countries, which makes the data suitable for analytical processing.

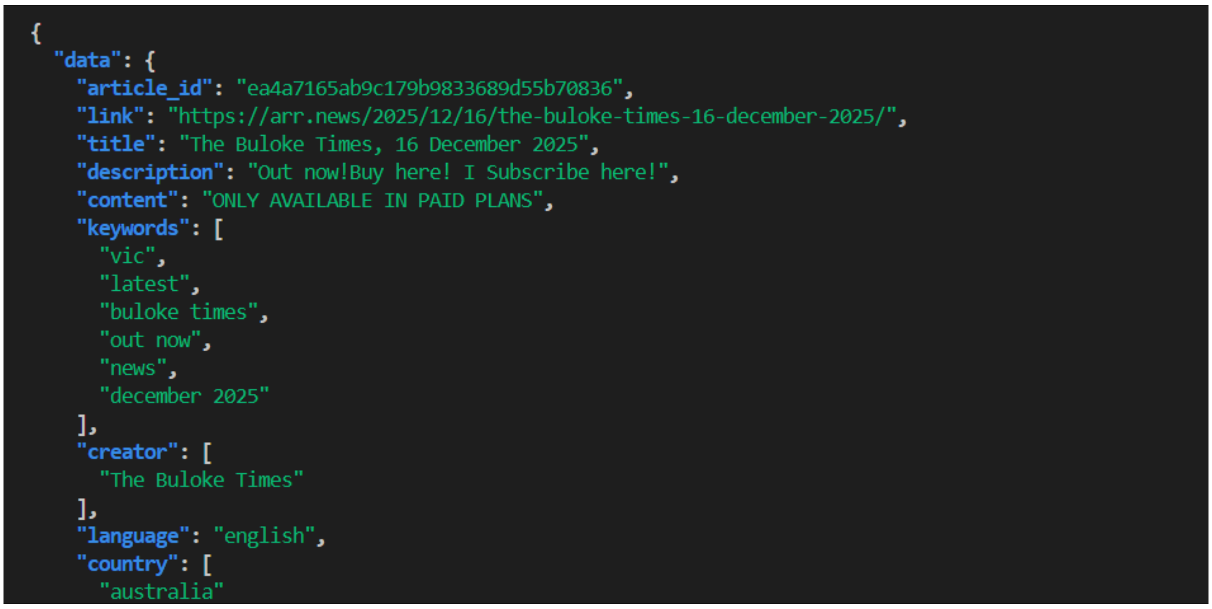
Justification

– The API fully satisfies the assignment requirements
– It is stable and well-documented
– It is frequently updated, with new news appearing every few minutes
– It provides real-world, meaningful data from trusted sources
– It returns structured data in JSON format

Kafka topic schema

In the first stage of the pipeline, a producer is implemented, which is responsible for receiving news data from an external API. NewsData.io and sending them to Kafka topic raw_events

Topic name: **raw_events**
message format: **JSON**
One message corresponds to one news article



```
{
  "data": {
    "article_id": "ea4a7165ab9c179b9833689d55b70836",
    "link": "https://arr.news/2025/12/16/the-buloke-times-16-december-2025/",
    "title": "The Buloke Times, 16 December 2025",
    "description": "Out now!Buy here! I Subscribe here!",
    "content": "ONLY AVAILABLE IN PAID PLANS",
    "keywords": [
      "vic",
      "latest",
      "buloke times",
      "out now",
      "news",
      "december 2025"
    ],
    "creator": [
      "The Buloke Times"
    ],
    "language": "english",
    "country": [
      "australia"
```

Message schema:

```
{
  "article_id": "string",
  "title": "string",
  "description": "string | null",
  "content": "string | null",
  "link": "string",
  "source_id": "string",
  "source_name": "string",
  "category": "string",
  "country": "string",
  "language": "string",
  "pubDate": "string",
  "ingested_at": "timestamp"
}
```

Each message is posted to Kafka by the developer in DAG 1 (job1_news_ingestion) and later processed by the DAG cleanup team.

Cleaning Rules

Data cleaning is performed in DAG 2, which consumes raw news messages from the Kafka topic raw_events and prepares them for storage in SQLite

We applied the following rules:

1. Mandatory fields validation
   Records must contain non-empty article_id and title.
   Titles shorter than 10 characters are discarded.

2. Text normalization
   Extra whitespace is removed and text fields (title, description, content) are normalized.

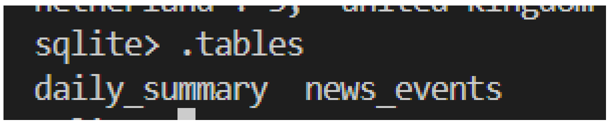
3. Handling optional fields
   Optional fields (description, content) may be null and are stored as NULL in SQLite.

4. Category, country, and language normalization
   List values are joined into comma-separated strings, default values are applied when missing, and all values are converted to lowercase.

5. Date and metadata handling
   The publication date (pubDate) is stored as received, and a cleaned_at timestamp is added.

6. Duplicate prevention and error handling
   Duplicate records are ignored during insertion, and malformed messages are logged and skipped without stopping the batch job.

SQLite schema (for both tables)



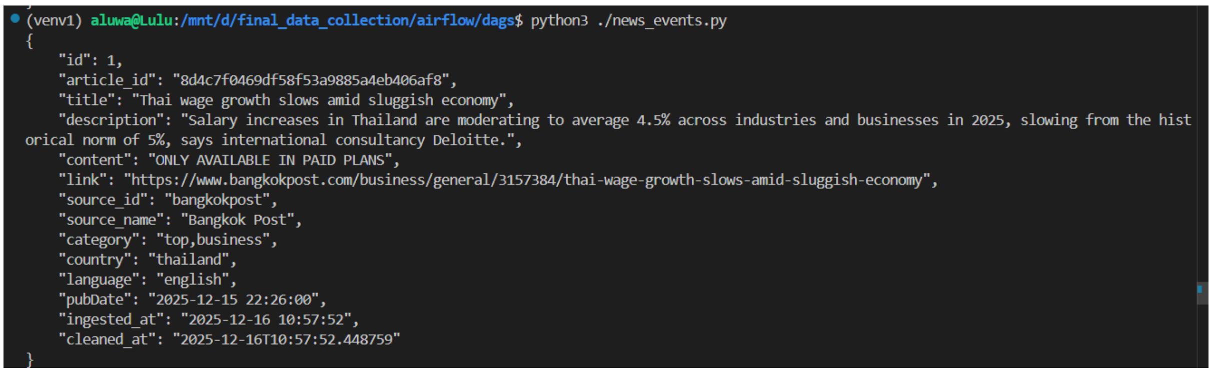
```
sqlite> .tables
daily_summary  news_events
```

Table: news_events

This table stores cleaned and normalized news articles after processing in DAG 2.

```
news_events (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  article_id TEXT UNIQUE,
  title TEXT NOT NULL,
  description TEXT,
  content TEXT,
  link TEXT,
  source_id TEXT,
  source_name TEXT,
  category TEXT,
  country TEXT,
  language TEXT,
  pubDate TEXT,
  ingested_at TEXT,
  cleaned_at TEXT
)
```

The news_events table stores cleaned and normalized news articles received from Kafka after processing in DAG 2



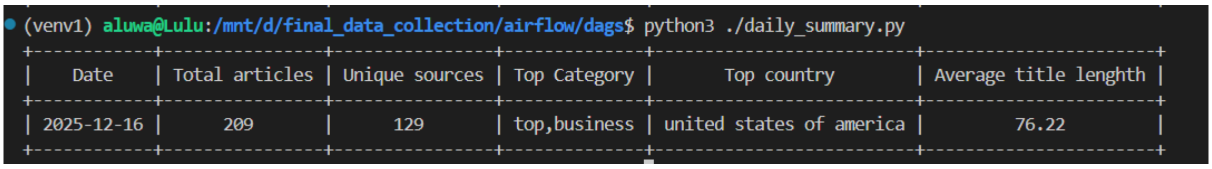
```
(venv1) aluwa@Lulu:/mnt/d/final_data_collection/airflow/dags$ python3 ./news_events.py
{
    "id": 1,
    "article_id": "8d4c7f0469df58f53a9885a4eb406af8",
    "title": "Thai wage growth slows amid sluggish economy",
    "description": "Salary increases in Thailand are moderating to average 4.5% across industries and businesses in 2025, slowing from the hist
orical norm of 5%, says international consultancy Deloitte.",
    "content": "ONLY AVAILABLE IN PAID PLANS",
    "link": "https://www.bangkokpost.com/business/general/3157384/thai-wage-growth-slows-amid-sluggish-economy",
    "source_id": "bangkokpost",
    "source_name": "Bangkok Post",
    "category": "top,business",
    "country": "thailand",
    "language": "english",
    "pubDate": "2025-12-15 22:26:00",
    "ingested_at": "2025-12-16 10:57:52",
    "cleaned_at": "2025-12-16T10:57:52.448759"
}
```

The daily_summary table contains the results of daily analytics calculated in DAG 3 based on data from the news_events table.It is used to store aggregated metrics on news for the day

Table: daily_summary

This table stores aggregated daily analytics calculated in DAG 3.

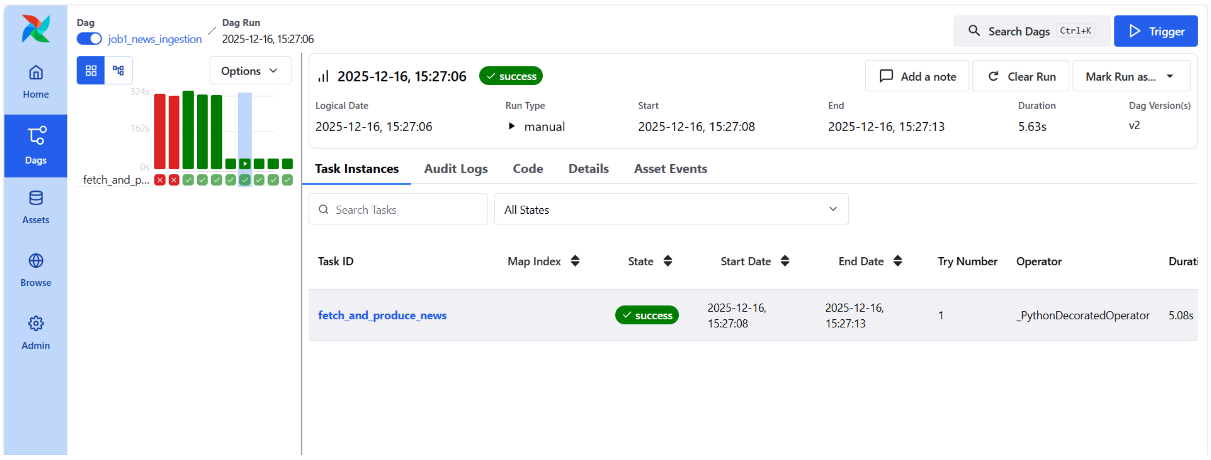
```
daily_summary (
  date TEXT,
  total_articles INTEGER,
  unique_sources INTEGER,
  top_category TEXT,
  top_country TEXT,
  average_title_length REAL
)
```



```
(venv1) aluwa@Lulu:/mnt/d/final_data_collection/airflow/dags$ python3 ./daily_summary.py
+------------+----------------+----------------+--------------+--------------------------+-----------------------+
|    Date    | Total articles | Unique sources | Top Category |       Top country        | Average title lenghth |
+------------+----------------+----------------+--------------+--------------------------+-----------------------+
| 2025-12-16 |      209       |      129       | top,business | united states of america |         76.22         |
+------------+----------------+----------------+--------------+--------------------------+-----------------------+
```

DAG1

Successful execution of the news ingestion DAG (job1_news_ingestion) that fetches data from the NewsData.io API and publishes raw news messages to the Kafka topic raw_events.



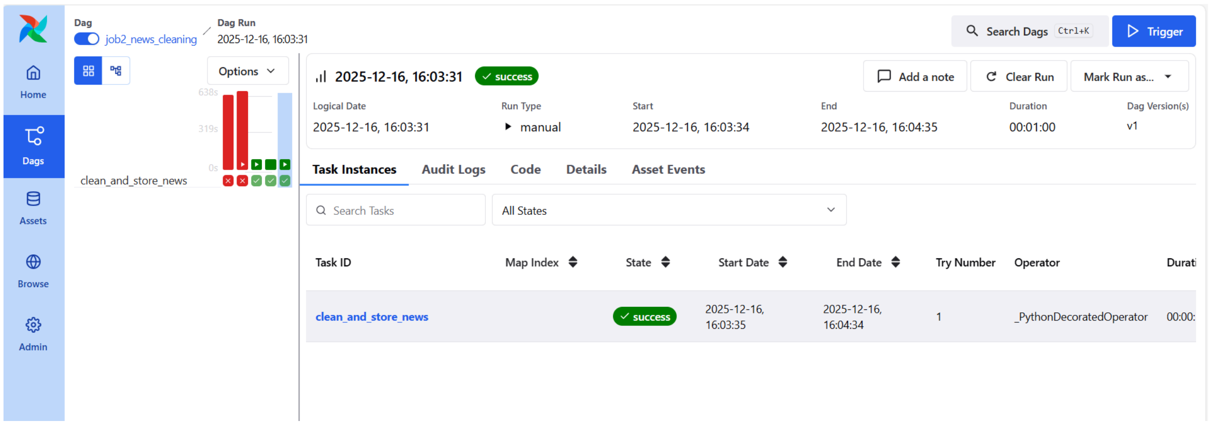
Dag: job1_news_ingestion / Dag Run: 2025-12-16, 15:27:06

2025-12-16, 15:27:06 success

Logical Date	Run Type	Start	End	Duration	Dag Version(s)
2025-12-16, 15:27:06	manual	2025-12-16, 15:27:08	2025-12-16, 15:27:13	5.63s	v2

Task ID	Map Index	State	Start Date	End Date	Try Number	Operator	Durati
fetch_and_produce_news		success	2025-12-16, 15:27:08	2025-12-16, 15:27:13	1	_PythonDecoratedOperator	5.08s

DAG2



Dag: job2_news_cleaning / Dag Run: 2025-12-16, 16:03:31

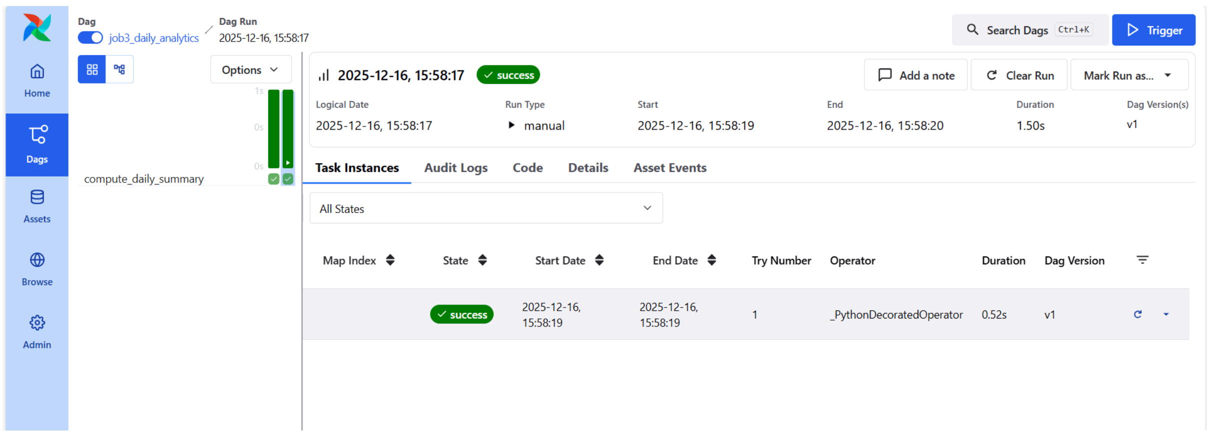
2025-12-16, 16:03:31 success

Logical Date	Run Type	Start	End	Duration	Dag Version(s)
2025-12-16, 16:03:31	manual	2025-12-16, 16:03:34	2025-12-16, 16:04:35	00:01:00	v1

Task ID	Map Index	State	Start Date	End Date	Try Number	Operator	Durati
clean_and_store_news		success	2025-12-16, 16:03:35	2025-12-16, 16:04:34	1	_PythonDecoratedOperator	00:00:

Successful execution of the data cleaning DAG (job2_news_cleaning) that consumes messages from Kafka, applies cleaning rules, and stores cleaned data in SQLite.

DAG3



2025-12-16, 15:58:17 success

Logical Date	Run Type	Start	End	Duration	Dag Version(s)
2025-12-16, 15:58:17	manual	2025-12-16, 15:58:19	2025-12-16, 15:58:20	1.50s	v1

Map Index	State	Start Date	End Date	Try Number	Operator	Duration	Dag Version
	success	2025-12-16, 15:58:19	2025-12-16, 15:58:19	1	_PythonDecoratedOperator	0.52s	v1

Successful execution of the daily analytics DAG (job3_daily_analytics) that computes aggregated metrics from SQLite and writes results to the daily_summary table.