ETL Pipeline Project Report - Stock Market Data

1. Project Overview

This ETL pipeline project focuses on Stock Market data. The aim is to ingest, clean, enrich, and load stock-related data into a MongoDB database. Data is collected from a CSV file, real-time APIs (NewsAPI and Finnhub), and Google Sheets.

2. Data Sources Used

- CSV File: Stock historical data ('historical_data_large.csv') from Google Drive.
- NewsAPI: For fetching real-time financial news related to the stock market.
- Finnhub API: For stock prices and financial metrics.
- Google Sheets: Connected via exportable CSV link.

3. ETL Process (Extract, Transform, Load)

EXTRACT:

- Loaded CSV data from Google Drive.
- Pulled data from APIs using requests.
- Fetched Google Sheets as CSV.

TRANSFORM:

- Cleaned missing values.
- Removed duplicates.
- Formatted timestamps to ISO 8601.
- Performed unit conversion and added calculated features.

LOAD:

- Loaded all cleaned and transformed data into MongoDB using PyMongo.

4. Data Cleaning & Feature Engineering

- Filled missing values with forward fill or default values.
- Converted temperatures to Celsius where applicable.
- Engineered a 'news_sentiment_score' and normalized timestamps.

- Unified date columns for easier querying.

5. Automation

Used Python's schedule module to run the ETL job daily. The code includes scheduling logic to automate the pipeline execution.

6. CI/CD Plan

Though not implemented yet, GitHub Actions will be used to:

- Run automated unit tests
- Validate schema consistency
- Deploy updates to the pipeline
- Improve reliability and feedback loops during development.

7. Tools & Technologies Used

- Python (pandas, schedule, requests, pymongo)
- Google Colab
- MongoDB Atlas
- Git & GitHub
- APIs: NewsAPI, Finnhub

8. Project Folder Structure

```
ETL_Pipeline_AbdulSami_DS055/
 etl_pipeline.py
 config/db_config.json
 data/
    historical_data_large.csv
    sample_weather.json
    google_sheet_sample.csv
 scheduler.py
 requirements.txt
```

README.md

output/final_cleaned_data.csv

load_to_db.py

.github/workflows/ci_cd.yml

9. Conclusion

The ETL pipeline is functional and meets most of the exam requirements. Data sources are integrated, data is cleaned and enriched, and automation has been implemented. CI/CD and documentation are prepared for final packaging and submission.