# ETL Pipeline Project Report

## 1. Project Overview

This ETL (Extract, Transform, Load) project is designed to consolidate and process data from five distinct sources: CSV, JSON/API, Google Sheets, a SQL database, and a REST API. The processed data is cleaned, transformed, and loaded into a unified SQL database for analysis.

## 2. Data Sources

- CSV Files: Local CSV files containing raw data.
- JSON/API: Remote APIs providing JSON data.
- Google Sheets: Data fetched using Google Sheets API.
- SQL Database: Extracted via SQL queries.
- REST API: Accessed through requests for dynamic data extraction.

## 3. ETL Pipeline Architecture

The pipeline is structured into modular scripts for each stage:
- Extract: Retrieves data from all defined sources.
- Transform: Handles data cleaning, formatting timestamps, unit conversions, and feature engineering.
- Load: Inserts the cleaned and processed data into a centralized SQL database.
- Automation: Pipeline is integrated with GitHub Actions for scheduled runs and CI/CD workflows.

## 4. Data Transformation Details

- Timestamp Formatting: Converts all datetime formats to a standard ISO format.
- Unit Conversion: Converts measurement units to maintain consistency.
- Feature Engineering: Adds new calculated columns to enhance the dataset.
- Data Cleaning: Handles null values, invalid entries, and data type normalization.

## 5. CI/CD Automation

GitHub Actions is used for continuous integration and deployment:
- Automated testing of scripts.
- Scheduled execution of the pipeline.
- Linting and code quality checks.

## 6. Technologies Used

- Python (pandas, requests, SQLAlchemy, etc.)
- SQL (SQLite/PostgreSQL)
- Google Sheets API
- GitHub Actions for CI/CD
- Docker (if used)

## 7. Conclusion

This ETL pipeline efficiently automates the collection, processing, and storage of data from diverse sources. It ensures data consistency and quality while enabling easy scalability and maintainability.