# Cloud-Based Data Engineering Pipeline SkylinETL

Submitted in partial fulfillment of the requirements of the degree of

BACHELOR OF COMPUTER ENGINEERING

by

Dnyanesh Bharambe - 23102046

Atharva Bhoir - 23102189

Om Bhoir - 23102010

Vedant Shinde -23107129

Guide :

Prof. Deepali Kayande

Department of Computer Engineering

A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE

2025-26

Parshvanath Charitable Trust's

A. P. SHAH INSTITUTE OF TECHNOLOGY

(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)

(Religious Jain Minority)

# CERTIFICATE

This is to certify that the Mini Project 2A entitled "**Cloud-Based Data Engineering Pipeline SkylinETL** is a bonafide work of **"Dnyanesh Bharambe (23102046), Atharva Bhoir (23102189), Om Bhoir (23102010), Vedant Shinde (23102215)"** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering** in **Computer Engineering.**

| | | |
|---|---|---|
| _____ | _____ | _____ |
| Guide | Project Coordinator | Head of Department |
| Prof. Deepali kayande | Prof. Deepali Kayande | Dr. S.H. Malave |

Parshvanath Charitable Trust's

A. P. SHAH INSTITUTE OF TECHNOLOGY

(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)

(Religious Jain Minority)

# Project Report Approval for Mini Project- 2A

This project report, entitled "**Cloud-Based Data Engineering Pipeline SkylinETL** by **Dnyanesh Bharambe, Atharva Bhoir, Om Bhoir, Vedant Shinde** is approved for the partial fulfilment of the degree of ***Bachelor of Engineering*** in ***Computer Engineering***, ***2025-26***

Examiner Name Signature

1. _____ _____

2. _____ _____

Date:

Place

# Declaration

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be caused for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

_ _ _ _ _ _ _ _ _ _ _ _ _ _

Dnyanesh Bharambe–23102046

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Atharva Bhoir – 23102189

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Om Bhoir – 23102010

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Vedant Shinde – 23102215

Date

# Abstract

The rapid growth of applications that use data has made the need for scalable and automated data engineering solutions even greater. This paper describes the design and implementation of a complete data pipeline that can take in and process data, store it, and show large amounts of data in almost real time. The data comes from APIs and is stored as raw data in MinIO. After that, we perform various operations on it to stage it and sort it. Apache Airflow is used as an orchestration tool that looks over all the tasks and helps in performing extraction, transformation, and loading procedures. We have used libraries in Python like the Pandas library and the PyTorch library, which help us to check data and to change it. Data is saved in the Parquet format so that it can be queried quickly. Then, the data that has been cleaned up is put into Snowflake. This software lets you do fast and scalable analytics. It makes dynamic panels that show data in real time and trends from the past to help people understand and make choices. Putting data pools, storage, and visualization tools together in the cloud can make a strong foundation for analytics that are always running, automatic, and ready for business. The topics include Real-Time Data Pipeline, Data Engineering, ETL, MinIO, Apache Airflow, Snowflake, and Power BI.

**Keywords: Data Pipeline, Apache Airflow, MinIO, Snowflake, Power BI, ETL, Cloud Computing.**

# CONTENTS

# LIST OF FIGURES

# Chapter 1

## Introduction

There are five main sections to this study. In Section I (Introduction), we talked about the goals and gave an outline of how the data flows. In Section II (Methodology), we spoke about how the system was developed step by step, from how data is collected, planned, processed, saved in databases, and shown. We also talked about the research, best practices, and technologies that helped make the project in Section III (Literature Survey). Part IV (Acknowledgment) talks about the help, resources, and advice that were given during the project. Finally, Section V (Conclusion) talks about the results, the contributions, and how to improve things in the future. The References section contains all the technical and academic sources that were used.

A data engineering pipeline is critical to contemporary organizations as it allows the automation of the gathering, processing, and transfer of data from distinct sources into a repository for analysis and decision-making. The key is to convert raw, unstructured, or dissimilar data into clean, structured, and analytics-ready forms with minimal human intervention.

The pipeline is designed to consolidate data from multiple sources (e.g., APIs and databases) into a single location for analytics and business intelligence. It provides real-time and batch insights quickly, allowing organizations to respond rapidly to business changes or events. Additionally, it enhances data quality and reliability through automatic validation, cleaning, and transformation steps that ensure the data is accurate, consistent, and conforms to pre-defined schemas. The pipeline also enables advanced analytics and visualization by supplying clean, transformed data to advanced BI tools and data warehouses while eliminating manual effort, automating repetitive data engineering tasks, and reducing.

# Chapter 2

## Literature Survey

| Author | Title | Findings of the Study |
|---|---|---|
| Chandrapal Singh Dangi ,Sumit Dhariwal | Exploring Cloud Deployment Services through Machine Learning: A Focus on AWS | ML analysis shows AWS services and frameworks enable efficient, scalable, and optimized cloud adoption |
| Ayush Gupta, Namrata Dhanda , Kapil Kumar Gupta | Ingest and Visualize CSV Files using AWS Platform For Transition from Unstructured to Structured Data | AWS services like Glue, Redshift, and QuickSight enable fast, scalable, and automated data processing and visualization. |
| Daliparthi Jeevana Aditya,Sunki Laxmanraj, R. Sathya Bama Krishna | Private Document Vault with Server-Side Encryption in Cloud AWS S3 Bucket | Hybrid encryption ensures secure, reliable, and breach-resistant storage in distributed cloud systems. |
| Vashudhar Sai Thokala, Sandeep Gupta | Integrating Cloud Infrastructure for Scalable Web Applications: Insights from AWS, EC2, and S3 | Dynamic EC2 scaling on AWS improves resource utilization and enhances web application performance |
| Yewon Shin,Jonghyeok Park | Revisiting SQL Statement Logging for SQLite on AWS S3 | SSL-S3 enables faster, more efficient SQL logging and recovery by appending only update statements to S3 |
| Rajeshree Khande, Sheetal Rajapurkar **,**Pratik Barde,Harshi Balsara, Akash Datkhile | Data Security in AWS S3 Cloud Storage | Leveraging AWS cloud features enables secure, efficient, and automated data storage and protection |
| Liang Tian , Rocco Sedona ,Amirpasha Mozaffari , Enxhi Kreshpa , Claudia Paris , Morris Riedel , Martin G. Schultz , Gabriele Cavallaro | End-to-End Process Orchestration of Earth Observation Data Workflows with Apache Airflow on High Performance Computing | Apache Airflow with modular supercomputing enables scalable, parallel, and efficient Earth Observation data processing with ML/DL integration |
| Taylor Hartman, Samir Poudel, Jiblal Upadhya, Md Nahid Hasan , Kritagya Upadhyay , Khem Poudel | A Big Data Optimization Comparison using Apache Spark and Apache Airflow | Apache Spark offers faster processing while Apache Airflow uses less memory, highlighting trade-offs in big data task performance |
| Jirapan Tunpita**,** Khwunta Kirimasthong | Data Integraion and Data Pipeline Model by Using KNIME for Research Data. | Databricks Data Intelligence Platform powers AI-driven analytics to speed up insights and optimize decision-making with a unified lakehouse foundation. |
| Sachin Murarka **,** Anshuj Jain , Laxmi Singh | Advanced Techniques in Data Ingestion and Pipelining for Scalable Big Data Platforms: A Comprehensive Review. | It uses generative AI combined with the lakehouse to automatically optimize data performance and |

| | | |
|---|---|---|
| | | infrastructure management unique to each business. |
| Will Girten | Building Modern Data Applications Using Databricks Lakehouse: Develop, optimize, and monitor data pipelines on Databricks | The platform enables simple data discovery through natural language querying and automates code generation and error remediation. |
| F. M. Khalaf and A. M. Sagheer | A Hybrid Encryption Model with Blockchain Integration for Secure Cloud Data Storage and Retrieval | Hybrid encryption with blockchain improves data integrity, confidentiality, and decentralized key management in cloud storage |
| R. Shankar, A. Natarajan, and M. R. Patel | CloudLock: Secure Data Sharing Using a Hybrid Cryptosystem in Multi-Cloud Data Storage | CloudLock's hybrid cryptosystem enhances secure multi-cloud data sharing while reducing latency and unauthorized access |
| S. K. Parisa and S. Banerjee | A Hybrid Encryption Model for Secure Data Storage and Transmission in Cloud Computing | AES–RSA hybrid encryption with blockchain ensures high data security, authentication, and efficient cloud data transmission |
| M. Farina and J. Johnson | MinIO's Object Storage Supports External Tables for Snowflake | MinIO–Snowflake integration enables direct querying of object-stored data, improving accessibility and reducing data transfer costs |

1. C.S. Dangi & S. Dhariwal (2024): Demonstrated AWS-based scalable deployments using ML optimization. Machine learning optimizes AWS deployments using services like EC2 and S3. It ensures scalability, flexibility, and secure cloud performance.

2. A. Gupta, N. Dhanda, and K.K. Gupta (2023), "Ingest and Visualize CSV Files using AWS Platform for Transition from Unstructured to Structured Data." The study explains how AWS tools like Glue, Redshift, and QuickSight automate the process of data ingestion, transformation, and visualization.

3. D.J. Aditya, S. Laxmanraj, and R.S.B. Krishna (2023), "Private Document Vault with Server-Side Encryption in Cloud AWS S3 Bucket." This paper proposes a hybrid encryption technique in AWS S3 that ensures secure cloud storage using layered encryption methods.

4. V.S. Thokala and S. Gupta (2025), "Integrating Cloud Infrastructure for Scalable Web Applications: Insights from AWS, EC2, and S3."The paper focuses on dynamic scaling and optimized performance of web applications using AWS EC2, S3, and CloudWatch for monitoring.

5. Y. Shin and J. Park (2025), "Revisiting SQL Statement Logging for SQLite on AWS S3."The authors propose SSL-S3, which logs only SQL update statements to improve database performance and recovery in cloud environments.

6. R. Khande, S. Rajapurkar, P. Barde, H. Balsara, and A. Datkhile (2023), "Data Security in AWS S3 Cloud Storage." This paper discusses secure data storage using encryption and role-based access control in AWS

environments.

7. L. Tian, R. Sedona, A. Mozaffari, and C. Paris (2023), “End-to-End Process Orchestration of Earth Observation Data Workflows with Apache Airflow on High-Performance Computing.” The study demonstrates how Apache Airflow can manage large-scale, parallel workflows effectively using modular orchestration.

8. T. Hartman, S. Poudel, and K. Poudel (2025), “A Big Data Optimization Comparison using Apache Spark and Apache Airflow.” This research compares the efficiency of Spark and Airflow, showing trade-offs between processing speed and memory usage.

9. J. Tunpita and K. Kirimasthong (2024), “Data Integration and Data Pipeline Model by Using KNIME for Research Data.” This paper focuses on automating research data integration and improving data quality through workflow tools like KNIME.

10. M. Farina and J. Johnson (2023), “MinIO’s Object Storage Supports External Tables for Snowflake.”
This technical paper explores MinIO and Snowflake integration for direct data querying and reduced data movement costs.

# Chapter 3

# Problem Statement, Objective & Scope

## Problem Statement: -

To design and implement an end-to-end automated data pipeline for real-time data ingestion, processing, and visualization using Apache Airflow, Snowflake, MinIO (or AWS S3), and Power BI.

Enterprises today face significant challenges in automating, scaling, and maintaining large data pipelines. Manual ETL (Extract, Transform, Load) processes are time-consuming, prone to human errors, and difficult to manage as data volume and complexity increase. Managing and analyzing vast amounts of real-time data from multiple sources has become a critical requirement for modern organizations seeking timely insights.

Traditional ETL systems often lack automation, flexibility, and scalability, resulting in delayed and inconsistent analytics. Hence, there is a need for a robust, cloud-native, and automated data pipeline that can efficiently handle data ingestion, transformation, storage, and visualization in real time.

This project aims to address these challenges by building an automated data engineering solution where:

- **Apache Airflow** is used for workflow orchestration and scheduling,
- **MinIO** (or AWS S3) serves as a scalable cloud storage system,
- **Snowflake** acts as a cloud data warehouse for structured analytics, and
- **Power BI** provides real-time visualization and business insights.

This integrated solution ensures efficient data flow from source to dashboard with minimal human intervention, improved scalability, and better decision-making capabilities for enterprises.

## Objectives:

- To build an end-to-end data pipeline for automating ETL processes.

- To integrate Apache Airflow, MinIO (or AWS S3), and Snowflake for scalable and cloud-native data workflows.
- To perform real-time data extraction, transformation, and loading using Python and SQL.
- To visualize the final processed data in Power BI for analytical insights.
- To schedule, monitor, and manage workflows automatically using Apache Airflow.
- To learn and implement cloud storage, workflow orchestration, and data warehousing concepts.

## Scopes :

- Data is collected from a public API or dataset and stored in MinIO (or AWS S3) for staging and archival.
- Apache Airflow automates the entire ETL process — extracting data, transforming it (via Python or SQL scripts), and loading it into Snowflake.
- Processed and cleaned data is stored in Snowflake, enabling advanced querying and analytics.
- The final dataset is connected to Power BI or Tableau for visualization and dashboard creation.
- Airflow handles task scheduling, dependency management, and error monitoring. GitHub is used for version control, collaboration, and CI/CD workflow management.

# Chapter 4

# Proposed System Architecture

## Description

The proposed system follows a structured multi-stage pipeline architecture, ensuring automation, scalability, and reliability. The methodology is divided into six major phases:

**A. Data Ingestion**
Data is collected from public APIs through both real-time streaming and scheduled batch extractions. Ingested raw data is stored in AWS S3, which serves as the central data lake. This layer ensures a persistent copy of incoming data and separates raw data from transformed datasets.

**B. Workflow Orchestration**
Apache Airflow is employed as the orchestration engine.
Data workflows are designed as Directed Acyclic Graphs (DAGs), which define task dependencies, scheduling intervals, and retries. Airflow ensures fault tolerance and automation by handling task failures, logging, and monitoring.

**C. Data Transformation and Validation**
Python libraries (Pandas, PyArrow) are used to perform cleaning and transformation.
Key transformation steps include:

- Removing duplicates and invalid records.
- Handling missing values.
- Validating schema consistency.
- Filtering negative or out-of-range values (e.g., negative fares).
- Transformed data is stored in Parquet format, which provides compression and efficient columnar querying.

**D. Data Storage**
Both raw data and processed data are maintained in AWS S3 using a structured folder hierarchy.S3 acts as the persistent storage layer, supporting reproducibility and scalability of the pipeline.

**E. Data Warehousing**
Processed datasets are loaded into Snowflake for scalable query execution. COPY INTO commands are used for batch loading, while it supports near real-time ingestion.
Snowflake's separation of storage and compute allows independent scaling of analytical queries.

**F. Data Visualization and Reporting**
Power BI dashboards are developed on top of Snowflake tables.
Dashboards provide:
Real-time monitoring metrics.

## Architecture Diagram

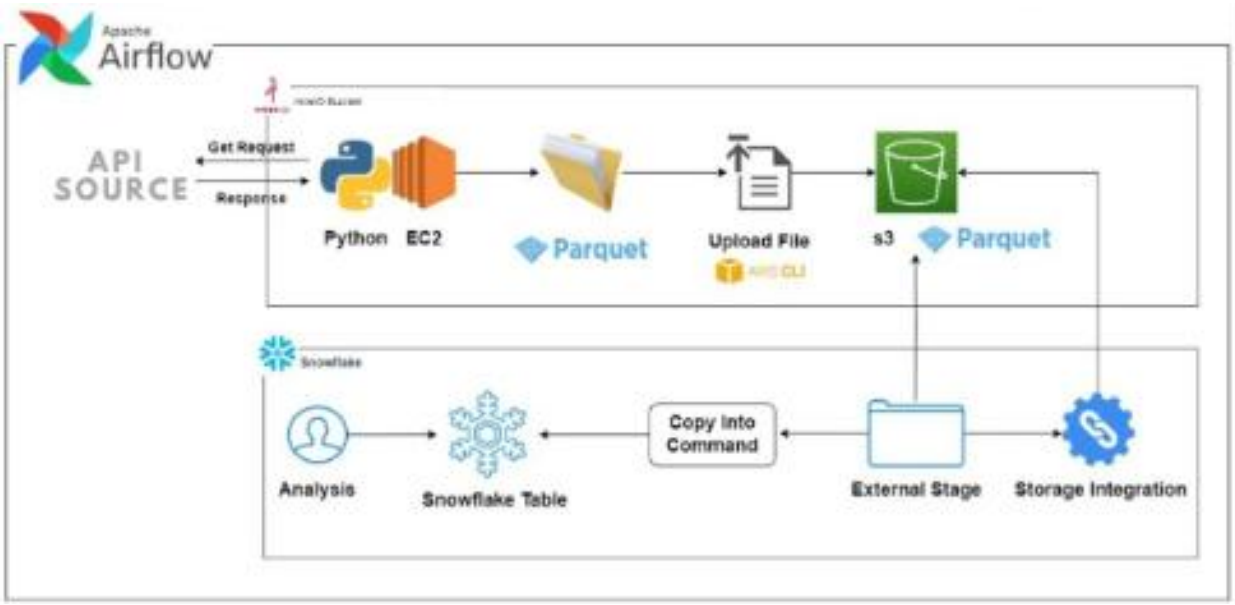



Fig 4.1: Architecture Diagram

## UML Dataflow Diagrams

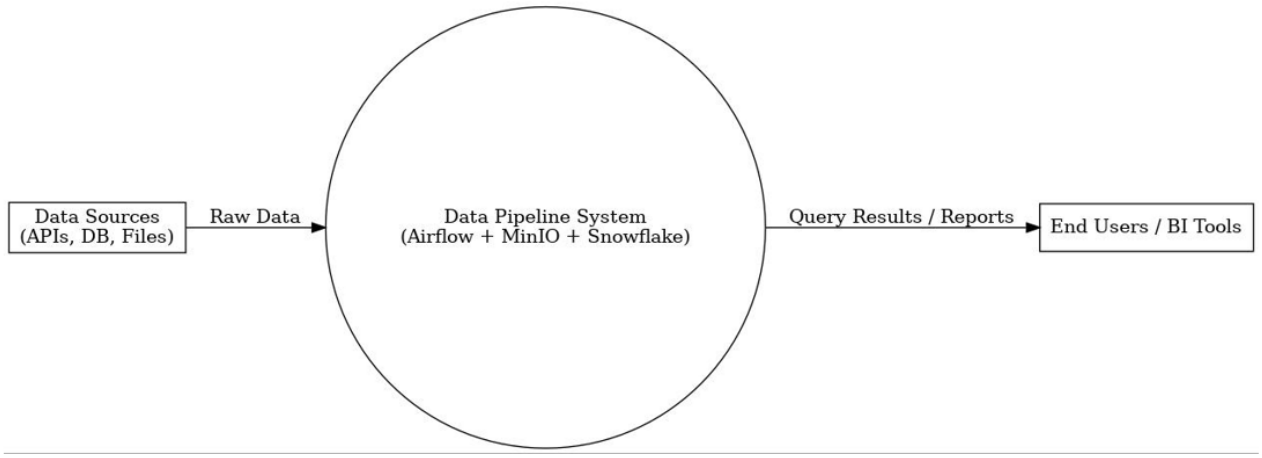



Fig 4.2: Level 0

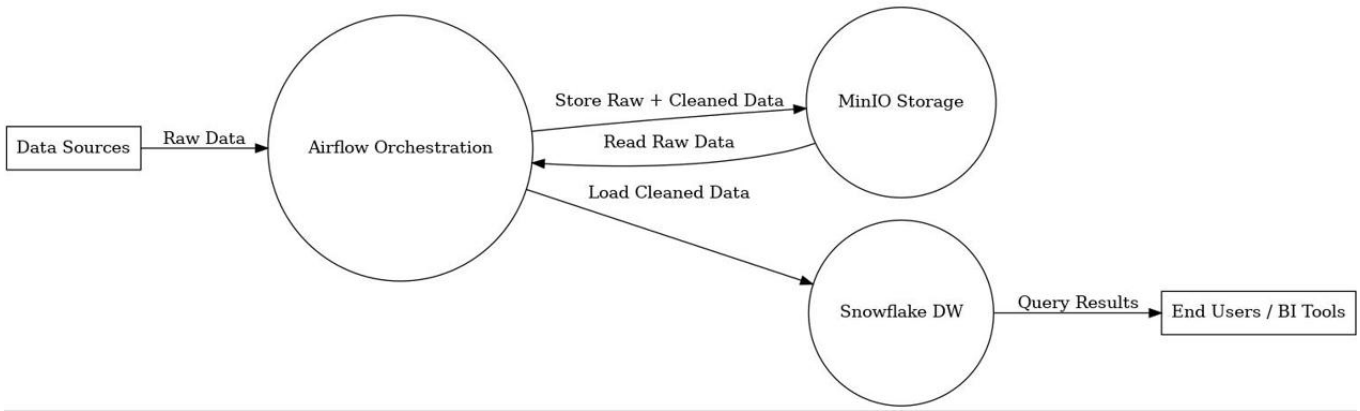



Fig 4.3: Level 1

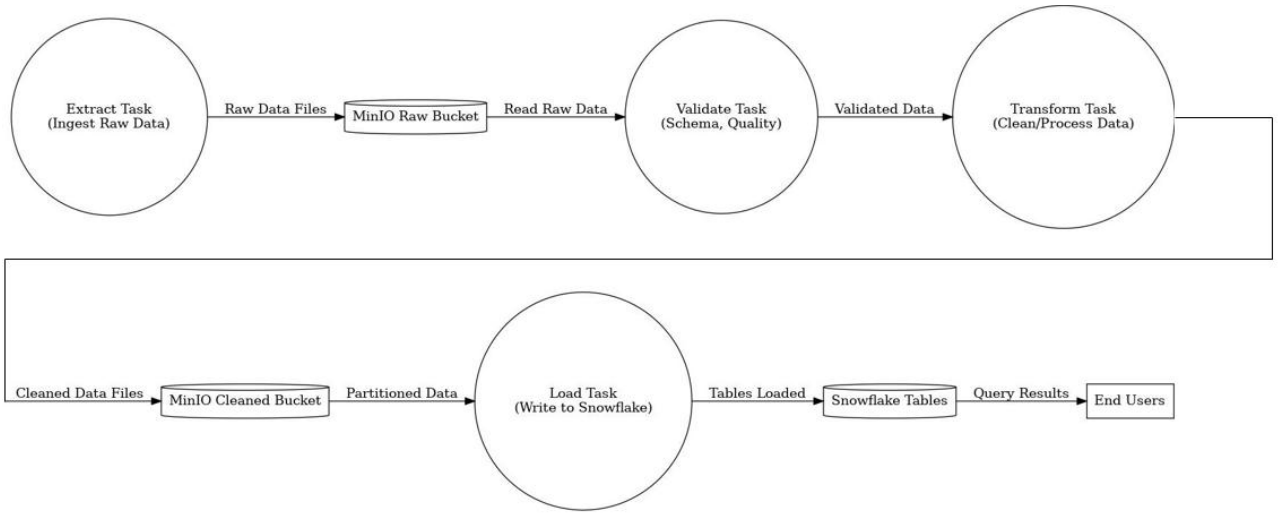



Fig 4.4: Level 2

## Use Case Diagram

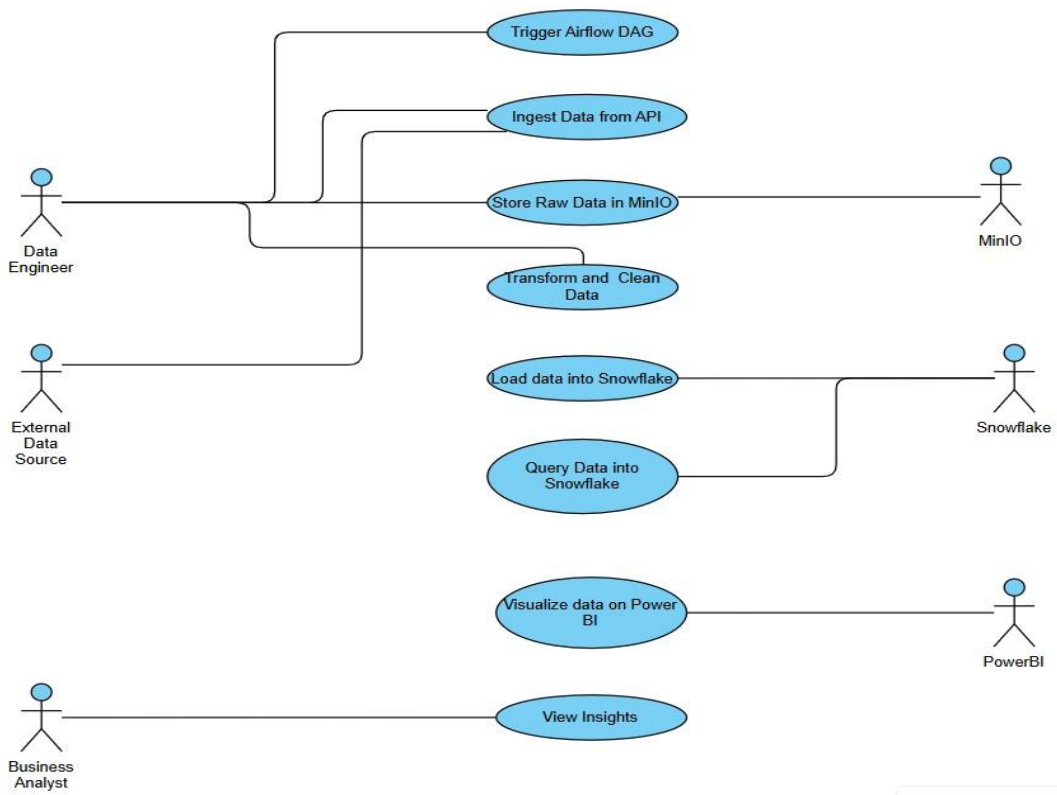



Fig 4.5: Use Case Diagram

## Sequence Diagram

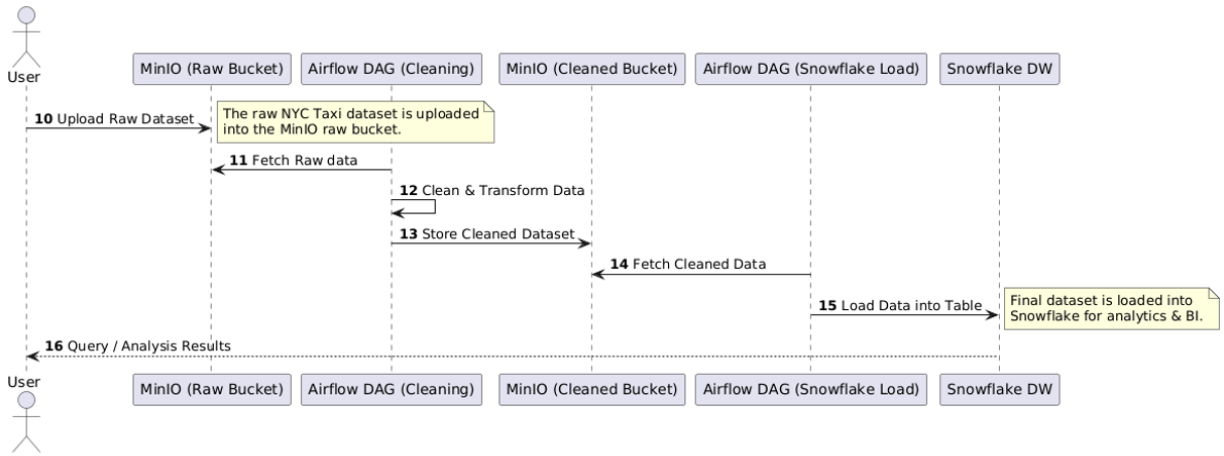



Fig 4.6: Sequence Diagram

## Activity Diagram

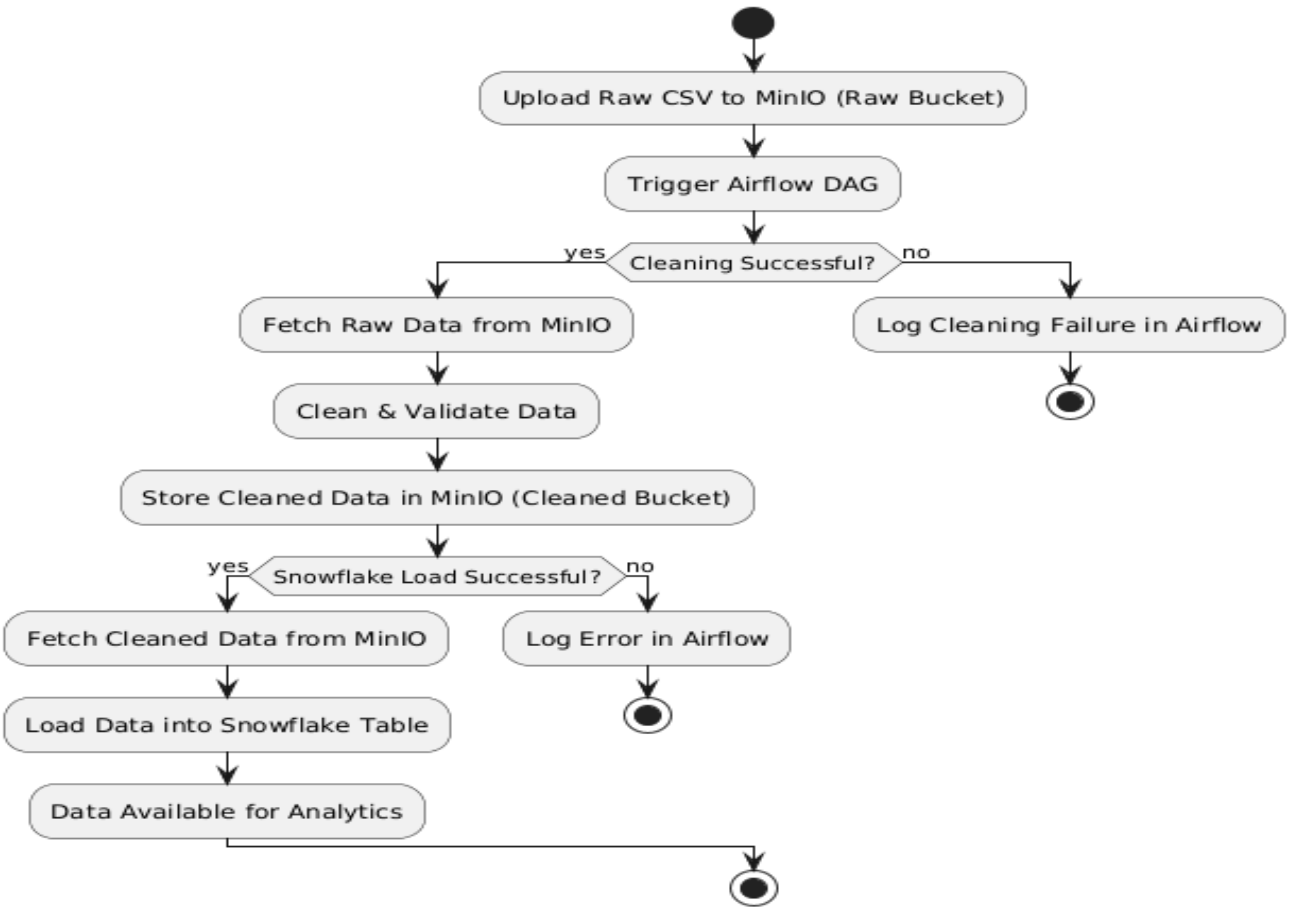



Fig 4.7: Activity Diagram

# Chapter 5

# Project Planning

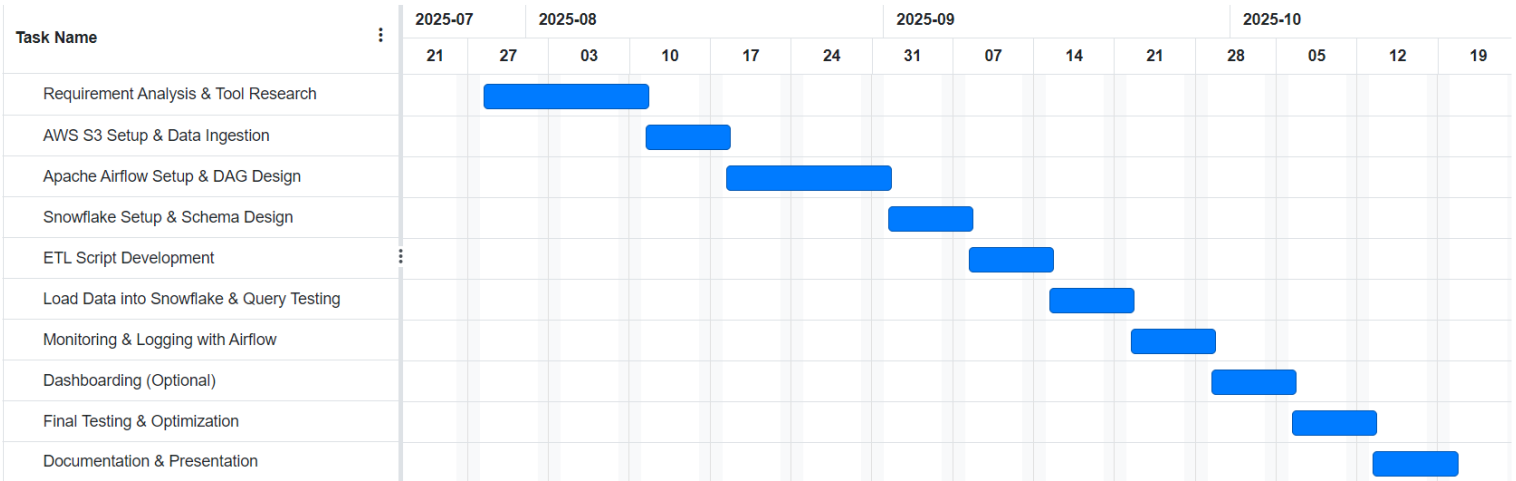



Fig 5.1: Gantt Chart

# Chapter 6

# Experimental Setup

## Software Requirements:

- OS: Windows
- Language: Python 3.x
- Tools: Apache Airflow, MinIO, Snowflake, Power BI
- Libraries: Pandas, PyArrow, SQLAlchemy
- IDE: VS Code / PyCharm

## Hardware Requirements:

- CPU: Intel i5 or higher
- RAM: Minimum 8 GB
- Storage: 100 GB or more
- GPU: Not required

# Chapter 7

# Implementation Details

## Modules:

1. **Data Ingestion Module:** Data Ingestion Module **f**etches data from public APIs and stores it in MinIO, supporting both real-time and batch processing. It performs initial validation to ensure data quality, organizes ingested data in structured buckets, and tags records with metadata for traceability. Integrated with tools like Apache Airflow, it enables automated scheduling, logging, and error handling, providing a reliable foundation for downstream data processing and analytics.

2. **Transformation Module:** Transformation Module uses Pandas and PyArrow to clean and validate ingested data. It handles missing values, corrects inconsistencies, enforces schema rules, and filters out invalid records to ensure high data quality. Optimized data is then converted to Parquet format for efficient storage and analytics.

3. **Airflow Orchestration:** The Airflow Orchestration component uses Directed Acyclic Graphs (DAGs) to automate task scheduling and monitor pipeline execution. DAGs define the order, dependencies, and timing of ETL steps, enabling reliable automation, retry logic, and real-time monitoring of data workflows.

4. **Data Loading Module:** The Data Loading Module transfers transformed and validated data into Snowflake, ensuring it is optimized for scalable querying and advanced analytics. This module manages the data transfer process securely and efficiently, preparing information for business intelligence and reporting needs.

5. **Visualization Module:** The Visualization Module builds Power BI dashboards connected to Snowflake, displaying key performance indicators, trends, and real-time metrics. This enables stakeholders to monitor data, gain insights, and make informed decisions through interactive visual reports.

**Tech Stack:** Tech Stack: Python, Apache Airflow, MinIO, Snowflake, and Power BI. This combination powers

automated data pipelines, cloud-native storage, scalable warehousing, and interactive business .

# Chapter 8

# Result

The paper demonstrates a robust, scalable, and secure data processing and visualization system using MinIO, Apache Airflow, Snowflake, API, and Power BI. Key outcomes include:

**MinIO Storage:**

MinIO serves as the centralized storage layer in the proposed data pipeline, designed to store and manage both structured and unstructured data efficiently. It offers compatibility with Amazon S3 APIs, allowing seamless integration with other tools such as Apache Airflow and Snowflake.

The storage system ensures data security through advanced encryption mechanisms and role-based access control, which help protect sensitive information and maintain user-specific permissions. In addition, MinIO's high-performance architecture supports rapid data access and transfer, enabling faster execution of downstream processes such as transformation, loading, and visualization.

Its scalability and lightweight design make it ideal for handling large datasets in local or cloud-based environments, providing reliability and speed across the entire data lifecycle.

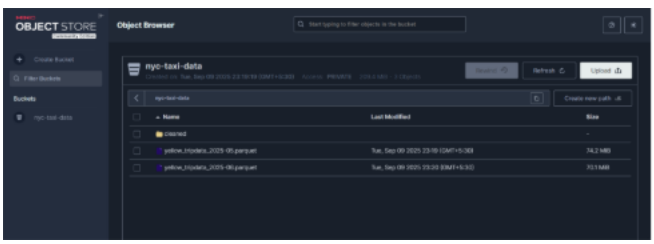


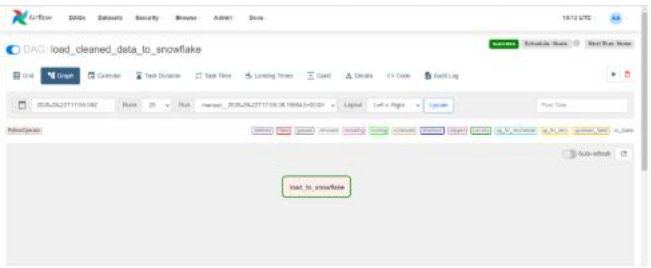
Figure 8.1: MinIO Object Browser interface.

**Workflow Orchestration with Apache Airflow:**

Apache Airflow functions as the **core orchestration tool** in the proposed data pipeline, managing the automation and scheduling of ETL (Extract, Transform, Load) processes. It allows for **concurrent execution of multiple pipelines**, ensuring efficient utilization of system resources with **no net time lost** between tasks.

Its **flexible scheduling mechanism** enables the smooth movement of both large datasets and real-time streaming data, accommodating varying data loads and processing frequencies.

Airflow is also designed for **lightweight execution**, requiring limited memory to launch and manage multiple workflows simultaneously. This makes it highly suitable for scalable and distributed data environments, ensuring continuous and reliable pipeline execution.




e 8.2: Airflow UI showing DAG status.

**Data Warehousing with Snowflake:**

Snowflake serves as the cloud-based data warehousing platform in the proposed pipeline, providing fast querying and high-performance processing of large datasets. Its architecture separates compute and storage layers, enabling independent scaling and optimized resource utilization for different workloads.

The platform supports automatic scaling, allowing it to handle growing volumes of data and user queries without requiring manual configuration or downtime. This ensures continuous availability and efficiency during both batch and real-time analytics operations.

Furthermore, Snowflake integrates seamlessly with business intelligence (BI) tools such as Power BI, facilitating smooth data visualization, reporting, and advanced analytical insights. Its built-in support for secure data sharing and structured query optimization enhances the overall speed and reliability of the analytics process.

Figure 8.3: Snowflake chart view.

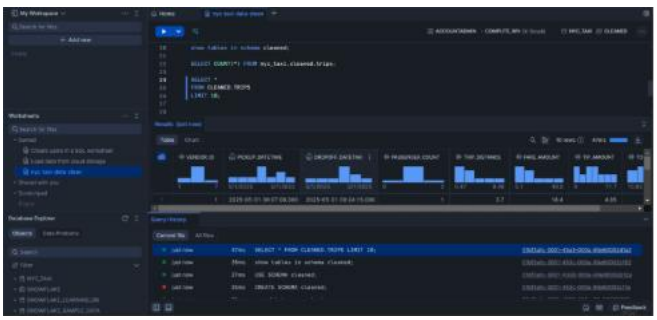


Figure 8.4: Snowflake SQL worksheet with query and results.

**API Integration:**
The API integration layer enables the connection to multiple external data sources, allowing seamless collection of data from various APIs. It can fetch live or real-time data feeds, ensuring that the information ingested into the pipeline remains accurate and up to date at all times.

**Power BI Visualization:**
Power BI enables users to access interactive dashboards and derive meaningful insights from the processed data stored in Snowflake. It allows stakeholders to analyze trends, track operational performance, and make informed, data-driven decisions based on real-time visual analytics.

**Overall Performance:**
The automation workflow ensures that the entire data pipeline operates without manual intervention, maintaining smooth and consistent execution across all stages. It offers high scalability, allowing the system to efficiently handle increasing data volumes and evolving processing requirements.
Overall, the pipeline provides a secure, reliable, and fully automated end-to-end data solution, ensuring accuracy, speed, and operational efficiency throughout the data lifecycle.

# Chapter 9

# Conclusion

This project demonstrates the necessity of constructing robust, scalable, and automated data engineering pipelines to meet the rising demands of real-time data processing. The system consists of multiple technologies which are working in unison, and it also demonstrates the complete life cycle of data - from ingestion to visualization. The data ingestion is depended on APIs which helps in enabling a continuous data flow and near real-time data availability. AWS S3 is an appropriate data lake for storing both raw data and intermediate data for cost effectiveness and reliability. The overall framework benefits from Apache Airflow's ability to manage workflows, schedule automated tasks, and manage errors. Essentially, Apache Airflow reduces the risk of human error and reliance on manual work due to pipeline failures.

Python with Pandas and PyArrow makes it quick and simple to clean, change, and check data. we use Parquet as the main storage format which improves query performance, and it lowers storage costs. By using Snowflake as the data warehouse, it makes the system extremely flexible, safe for storing data, and make it capable to run complicated analytical queries with ease. Finally, by using Power BI we transform raw and processed data into useful information through interactive dashboards. This closes the gap between technical processing and decision-making.

This project demonstrates contemporary practices in information engineering that can produce a complete system that copes with both real-time and batch data while generating insightful metrics for stakeholders. The pipeline architecture is adaptable, modular and expandable when new features come into play, such as creating a hyperlink to real-time streaming data sources like Apache Kafka, applying cutting-edge machine learning models for predictive analytics, or possibly monitoring tools for proactive system health monitoring. Thus, the project functions both as a scalable solution and a tangible example of how data engineering can offer firms timely insights derived from data for decision-making.

# Chapter 10

# References


[1] C. S. Dangi and S. Dhariwal, "Exploring Cloud Deployment Services through Machine Learning: A Focus on AWS," 2024 International Conference on Augmented Reality, Intelligent Systems, and Industrial Automation (ARIIA), Manipal, India, 2024, pp. 1-4, Doi: 10.1109/ARIIA63345.2024.11051682.

[2] A. Gupta, N. Dhanda and K. K. Gupta, "Ingest and Visualize CSV Files using AWS Platform For Transition from Unstructured to Structured Data," 2023 11th International Conference on Emerging Trends in Engineering & Technology - Signal and Information Processing (ICETET - SIP), Nagpur, India, 2023, pp. 1-6, Doi: 10.1109/ICETET-SIP58143.2023.10151634.

[3] D. J. Aditya, S. Laxmanraj and R. S. B. Krishna, "Private Document Vault with Server-Side Encryption in Cloud AWS S3 Bucket," 2023 8th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2023, pp. 1774-1778, Doi: 10.1109/ICCES57224.2023.10192782.

[4] V. S. Thokala and S. Gupta, "Integrating Cloud Infrastructure for Scalable Web Applications: Insights from AWS, EC2, and S3," 2025 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC), Bhubaneswar, India, 2025, pp. 1-6, Doi: 10.1109/ASSIC64892.2025.11158377.

[5] Y. Shin and J. Park, "Revisiting SQL Statement Logging for SQLite on AWS S3," 2025 IEEE 18th International Conference on Cloud Computing (CLOUD), Helsinki, Finland, 2025, pp. 457-459, Doi: 10.1109/CLOUD67622.2025.00056.

[6] R. Khande, S. Rajapurkar, P. Barde, H. Balsara and A. Datkhile, "Data Security in AWS S3 Cloud Storage," 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 2023, pp. 1-6, doi: 10.1109/ICCCNT56998.2023.10306922

[7] L. Tian et al., "End-to-End Process Orchestration of Earth Observation Data Workflows with Apache Airflow on High Performance Computing," IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium, Pasadena, CA, USA, 2023, pp. 711-714, doi: 10.1109/IGARSS52108.2023.10283416.

[8] T. Hartman, S. Poudel, J. Upadhya, M. N. Hasan, K. Upadhyay and K. Poudel, "A Big Data Optimization Comparison using Apache Spark and Apache Airflow," 2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2025, pp. 00732-00737, doi: 10.1109/CCWC62904. 2025.10903735.

[9] J. Tunpita and K. Kirimasthong, "Data Integration and Data Pipeline Model by Using KNIME for Research Data," 2024 8th International Conference on Information Technology (InCIT), Chonburi, Thailand, 2024, pp. 423-426, doi: 10.1109/InCIT63192.2024.10810545.

[10] S. Murarka, A. Jain and L. Singh, "Advanced Techniques in Data Ingestion and Pipelining for Scalable Big Data Platforms: A Comprehensive Review," 2024 IEEE 4th International Conference on ICT in Business Industry & Government (ICTBIG), Indore, India, 2024, pp. 1-6, doi: 10.1109/ICTBIG64922.2024.10911053.

[11] F. M. Khalaf and A. M. Sagheer, "A Hybrid Encryption Model with Blockchain Integration for Secure Cloud Data Storage and Retrieval," Journal of Information Systems Engineering & Management (JISEM), 2025, pp. 1–6.

[12] R. Shankar, A. Natarajan and M. R. Patel, "CloudLock: Secure Data Sharing Using a Hybrid Cryptosystem in Multi-Cloud Data Storage," Cluster Computing, Springer, 2025, doi: 10.1007/s10586-025-05433-7.

[13] S. K. Parisa and S. Banerjee, "A Hybrid Encryption Model for Secure Data Storage and Transmission in Cloud Computing," Transactions on Recent Developments in Artificial Intelligence and Machine Learning (TRDAIML), 2025, pp. 1–5.

[14] Will Girten, “Building Modern Data Applications Using Databricks Lakehouse: Develop, optimize, and monitor data pipelines on Databricks”, Packt Publishing, 2024.

[15] M. Farina and J. Johnson, "MinIO's Object Storage Supports External Tables for Snowflake," The New Stack, 2023, pp. 1–3.