

# **Integrated Data Analyst: An End-to-End Data Analytics Pipeline**

**Project Report submitted in partial fulfilment**

**of**

**MBA Tech**

**in**

**Computer Engineering**

**by**

**Darshita Anyawada (N225)**

**Gautam Kundalia (N243)**

**Omkar Rasal (N277)**

**Under the supervision of**

**Dr. Shubha Puthran**

**(Assistant Professor, MPSTME)**

**SVKM's NMIMS University**

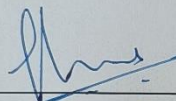


**MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT  
& ENGINEERING (MPSTME)  
Vile Parle (W), Mumbai – 56  
(2024-25)**

## CERTIFICATE



This is to certify that the project entitled "**Integrated Data Analyst**", has been done by **Ms. Darshita Anyawada, Mr. Gautam Kundalia, Mr. Omkar Rasal** under my guidance and supervision & has been submitted in partial fulfilment of the degree of **MBA Tech** in Computer Engineering of MPSTME, SVKM's NMIMS (Deemed-to-be University), Mumbai, India.

  
(SHUBHA PUTTHRAN)  
Project Mentor (Name and Signature)

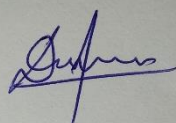
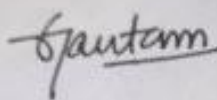

(Internal Guide)

**Date: 17/04/2025**

**Place: Mumbai**

\_\_\_\_\_  
Examiner (Name and Signature)

## ACKNOWLEDGEMENT

NAME	ROLL NO.	SAP ID	Signature
Darshita Anyawada	N225	70472100184	
Gautam Kundalia	N243	70472100292	
Omkar Rasal	N277	70472100465	

# INDEX

Topic	Page No
Abstract	v
List of Figures	vi
Abbreviations	vii
Chapter 1: Introduction	
1.1 Background of the project topic	1
1.2 Motivation and scope of the report	1
1.3 Problem Statement	2
1.4 Salient Contributions	2
1.5 Organisation of the report	3
Chapter 2: Literature Survey	
2.1 Introduction to the overall topic	4
2.2 Exhaustive literature survey	4
Chapter 3: Methodology and Implementation	
3.1 System Architecture and Block Diagram	8
3.2 Hardware and Software descriptions, flowchart	9
Chapter 4: Results and Analysis	
4.1 Results	12
4.2 Analysis	13
Chapter 5: Advantages, Limitations and Applications	
5.1 Advantages	15
5.2 Limitations	15
5.3 Applications	16
Chapter 6: Conclusion and Future Scope	17
References	19
Appendices	22

# ABSTRACT

In an era where businesses generate massive volumes of data from diverse platforms such as websites, Customer Relationship Management (CRM) systems, and marketing tools, converting this raw data into actionable insights has become essential yet increasingly complex. The "Integrated Data Analyst" project proposes an end-to-end data analytics pipeline that addresses this challenge through a unified, modular framework. The system enables automated data extraction, cleaning, transformation, storage, and real-time visualization—removing the need for deep technical expertise at every step. The solution integrates advanced tools such as Apify for scalable web scraping, Zoho CRM for structured customer data acquisition, and MySQL for secure storage. To ensure responsiveness and scalability, the backend infrastructure employs Flask as a lightweight server interface, Celery for background task execution, Redis as a message broker, and SocketIO for real-time progress updates. Data validation and transformation are automated using Python-based routines before the cleaned data is pipelined into Power BI dashboards for insightful visualization. The dashboards present key business metrics including customer acquisition trends, win-loss ratios, and churn analysis, all rendered with drill-down capabilities and dynamic filters. The methodology is designed to be highly adaptive and scalable. It supports integration with multiple data sources while ensuring consistency through schema validation and data type verification. This project ultimately demonstrates the power of automation in analytics—bridging the gap between technical execution and business insight. Through its modularity, transparency, and real-time feedback mechanisms, the system empowers non-technical users to derive meaningful analytics, making data-driven decision-making more accessible and efficient across organizational scales.

## LIST OF FIGURES

Figure No.	Name of the Figure	Page No.
Fig. 1	System Architecture Diagram	8
Fig. 2	Integration Pipeline using Flask, Celery, Redis	10
Fig. 3	MySQL Workbench – Data Storage Visualization	12
Fig. 4	Sample Power BI Dashboard	13
Fig. 5	accounts.csv file	22
Fig. 6	sales_pipeline.csv	22
Fig. 7	sales_teams.csv	23
Fig. 8	Cleaned data inserted into MySQL	23
Fig. 9	Hardware Requirements	24
Fig. 10	API Endpoints being triggered	24
Fig. 11	Libraries Utilized	25

## ABBREVIATIONS

Abbreviation	Definition
API	Application Programming Interface
CRM	Customer Relationship Management
CSV	Comma-Separated Values
DAX	Data Analysis Expressions
ETL	Extract, Transform, Load
JSON	JavaScript Object Notation
LLM	Large Language Model
MySQL	Structured Query Language (SQL Database)
NLP	Natural Language Processing

# **Chapter 1: Introduction**

## **1.1 Background of the Project Topic**

The rise of big data and the increasing reliance on digital platforms have led organizations to accumulate vast amounts of information. However, converting this raw data into actionable insights remains a critical challenge. The project “Integrated Data Analyst” was conceived to address this gap by designing and implementing an end-to-end data analytics pipeline. The system integrates multiple components that automate data extraction from web sources and CRM systems (such as Zoho), transform and clean the raw data, store it efficiently in a relational database (MySQL), and ultimately visualize it through dynamic dashboards (Power BI). The approach incorporates well-known frameworks and libraries (Flask, Celery, Redis) to provide real-time processing and interactive user experiences.

## **1.2 Motivation and Scope of the Report**

The primary motivation behind this capstone project is to bridge the existing gap between raw data collection and the generation of actionable insights without the need for extensive technical knowledge. Organizations often rely on multiple disparate tools for data handling, which leads to fragmentation, increased error rates, and inefficient workflows. This project aims to provide a scalable, modular, and integrated solution that simplifies data ingestion, validation, and visualization. The scope of the report includes a detailed discussion on the methodology (data extraction, processing, storage, and visualization), analysis of results, as well as an assessment of the system’s advantages, limitations, and potential applications. Furthermore, the report offers an extensive literature survey underpinning current research trends and identifies the areas that required further innovation, leading to the definition of the problem statement.



## 1.3 Problem Statement

Modern businesses struggle to efficiently integrate data from diverse, heterogeneous sources to drive strategic decision-making. The traditional data processing workflows are fragmented and require specialized technical expertise, resulting in increased operational costs and delayed decision-making. The problem addressed by the Integrated Data Analyst project is to create an automated and unified data analytics framework that:

- Reduces manual intervention in data extraction and cleansing.
- Ensures data integrity and seamless transformation across stages.
- Provides real-time analytics through interactive dashboards.
- Accommodates the scalability requirements of rapidly growing datasets.

## 1.4 Salient Contributions

The capstone project makes several noteworthy contributions:

- **Unified Data Pipeline:** The design and implementation of an integrated pipeline from data extraction to visualization.
- **Scalable Architecture:** Implementation of a modular and scalable system using industry-standard tools and frameworks.
- **Real-Time Monitoring:** Incorporation of real-time task updates via SocketIO and asynchronous processing using Celery.
- **Automated Data Validation:** Application of rigorous schema validation and data cleaning techniques, ensuring high-quality data for analysis.
- **Enhanced Accessibility:** Development of user-friendly interfaces that allow non-technical users to interact with and extract business-critical insights effortlessly.
- **Comprehensive Documentation and Versioning:** Adoption of a structured GitHub repository and version control practices to support collaborative development and reproducibility.

## 1.5 Organization of the Report

The report is organized as follows:

- Chapter 2: Literature Survey – Reviews current research, compares methodologies in data extraction, cleaning, processing, and visualization, and identifies research gaps.
- Chapter 3: Methodology and Implementation – Details the system’s overall architecture, including hardware and software components, data flow diagrams, and the code development lifecycle.
- Chapter 4: Results and Analysis – Presents the results obtained from deploying the system, evaluates system performance, and discusses analytical outcomes.
- Chapter 5: Advantages, Limitations, and Applications – Reviews the benefits, explores the project’s limitations, and highlights real-world applications.
- Chapter 6: Conclusion and Future Scope – Summarizes key findings, outlines the system’s impact, and discusses directions for future improvements.
- References and Appendices provide additional context, supporting materials, and detailed technical diagrams.

## **Chapter 2: Literature Survey**

### **2.1 Introduction to the Overall Topic**

Data analytics has evolved significantly over the past decades, primarily driven by the growing need for actionable insights from voluminous and heterogeneous data sources. Based on the Extract, Transform, Load (ETL) paradigm, traditional data processing methods have gradually given way to more dynamic and automated frameworks. The literature broadly encompasses developments in web data extraction, CRM-based data acquisition, data cleaning methodologies, and efficient data storage and visualization methods. Recent advancements have focused on overcoming the limitations of batch-processing ETL systems by leveraging real-time processing, improved data validation, and advanced dashboard interfaces. These studies have outlined challenges related to the handling of semi-structured and unstructured data. Other research has explored solutions using distributed computing and cloud-based platforms to achieve scalability and high performance. These varied methodologies underline the importance of integrating disparate components into a seamless system—a gap the Integrated Data Analyst project seeks to address.

### **2.2 Exhaustive Literature Survey**

The growing reliance on data-driven decision-making has necessitated the development of user-friendly platforms that can integrate, process, and analyze large-scale datasets. Businesses today require seamless data extraction, transformation, storage, and visualization tools that do not demand advanced technical expertise. Traditional data warehousing approaches have been challenged by increasing data complexity, the need for real-time analytics, and integration across disparate data sources. To address these issues, recent advancements in data integration, cleaning, model training, and visualization have paved the way for more efficient business intelligence and decision-support systems [13][14][15].

This literature review examines the core components of an effective data integration and analysis platform, moving from general data management principles to more

specific processes such as data cleaning, machine learning model training, and data visualization.

Efficient data integration is foundational for decision-making platforms, allowing seamless extraction from structured and unstructured sources. The traditional Extract, Transform, Load (ETL) process remains a critical component of data warehousing, ensuring data quality and consistency while handling real-time analytics challenges [1][2].

Several studies highlight the evolution of ETL methods from periodic batch processing to active warehousing, which enables continuous data updates and real-time analytics [2][13]. The Entity-Attribute-Value (EAV) model has been employed to handle diverse clinical data, but challenges arise regarding querying efficiency and complex Boolean operations [23]. Similarly, the increasing adoption of hybrid and cloud-based architectures has introduced interoperability challenges, where different SQL dialects and data schemas lead to inconsistencies in aggregation and reporting [16][24].

In unstructured data storage, XML data management has presented difficulties in traditional relational databases, prompting research into SQL:2003's advanced collection types to preserve hierarchical structures [3]. Moreover, Big Data storage solutions such as Hadoop and cloud-based SAP Datasphere offer decentralized, scalable storage mechanisms that integrate seamlessly with visualization tools [13][11].

Data preparation is a crucial stage, ensuring that raw data is transformed into a usable format for analytics. Data cleaning involves removing inconsistencies, correcting errors, and standardizing formats to improve reliability. Various rule-based and outlier detection approaches have been proposed, including statistical, distance-based, and model-based techniques [7][8][9].

With the rise of machine learning and statistical methods in data cleaning, projects like SampleClean integrate these techniques with traditional cleansing methods to improve data integrity [7]. The scalability challenge in big data cleaning has also been

a major concern, prompting research into semi-automated cleaning systems that balance human intervention with algorithmic efficiency [6][8].

Integrating natural language processing (NLP) and machine learning for automated text data extraction and structuring has also gained traction. Studies have demonstrated SQL optimization techniques for querying text-based data, significantly improving result accuracy and extraction speed [4][6].

Once data is cleaned and structured, machine learning models are trained to extract valuable insights. Advances in automated machine learning (AutoML) have enabled non-technical users to build predictive models without deep expertise [10][20].

Studies have explored the integration of Large Language Models (LLMs) for automated data analysis, demonstrating their ability to generate code and execute workflows from user-defined queries. However, challenges remain in ensuring that AI-generated insights are interpretable and actionable, necessitating continued advancements in explainability frameworks [9][21].

Within business intelligence, automated data mining solutions like Microsoft's OLE DB for Data Mining API integrate machine learning models directly into SQL databases, enabling real-time predictions without external processing [14]. Similarly, continuous integration and delivery (CI/CD) pipelines have revolutionized software deployment for data analytics, allowing rapid updates and model retraining [14].

The final step in transforming raw data into actionable insights involves effective visualization. Business Intelligence (BI) tools such as Power BI, Tableau, and QlikView have been extensively studied for their capabilities in intuitively presenting complex datasets. Research suggests that BI tools differ significantly in data integration, dashboarding, and user accessibility, with Power BI excelling in Microsoft ecosystem integration while Tableau offers superior drag-and-drop interactivity [12][18].

Column-store indexing and real-time analytical processing (OLAP) are essential features for interactive dashboards. Studies highlight how SQL Server's column store

indexes improve query performance for hybrid workloads, enabling businesses to run analytical queries alongside transactional ones with minimal latency [5].

Additionally, semantic modelling in BI tools enhances data accessibility for non-technical users by providing a logical abstraction layer that simplifies query creation and interpretation. This approach enables users to define business rules and relationships at a high level, promoting self-service analytics without requiring deep database knowledge [18].

A major barrier to widespread data analytics adoption is the complexity of technical tools. Research indicates that integrating Low-Code/No-Code (LCNC) platforms has significantly improved accessibility for business users, allowing them to build queries and dashboards without programming knowledge [17].

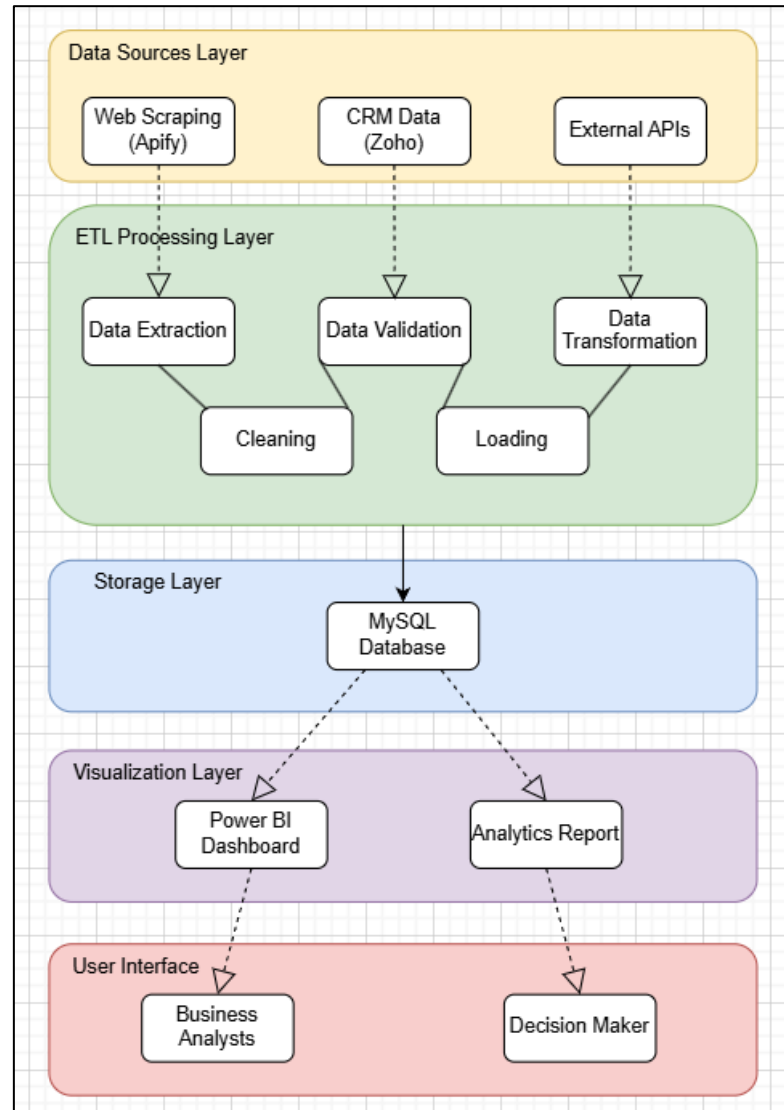
For example, studies have explored how Python-based libraries such as pandas enable interactive and flexible data analysis, lowering entry-level analysts' barriers [18][19]. Similarly, developing user-friendly APIs for data mining, as seen in OLE DB for Data Mining, further enhances the usability of analytics platforms [14].

In cybersecurity, research on honeypot log visualization has demonstrated how Power BI and Hadoop integration can streamline threat detection through visual analytics, reinforcing the importance of intuitive interfaces in complex data applications [11].

While the existing literature demonstrates significant advancements in individual areas such as data extraction, cleaning, processing, and visualization, it largely lacks focus on combining these elements into a single, integrated workflow. The gap identified was that most studies treat these components in isolation, resulting in fragmented solutions that require substantial technical effort to connect. This revealed a clear gap: the absence of a cohesive, end-to-end framework that simplifies and unifies the entire data analysis process. Our research aims to address this gap by proposing a structured approach that combines these advanced yet disconnected components into one streamlined system, making data analytics more accessible and manageable for practical use.

## Chapter 3: Methodology and Implementation

### 3.1 System Architecture and Block Diagram



*Figure-I System Architecture*

The architecture diagram illustrates a comprehensive integrated data analytics pipeline designed to transform raw data from multiple sources into actionable insights for business users. The workflow begins with the Data Sources Layer, consisting of three primary data inputs: web scraping using Apify, CRM data from Zoho, and external APIs. These heterogeneous data streams are funnelled into the ETL Processing Layer, where critical operations such as data extraction, validation, and transformation are performed. The extraction component pulls data from the sources,

while validation ensures data integrity through schema checks, null value handling, and consistency verification. Concurrently, the transformation process applies data cleaning techniques such as duplicate removal and outlier detection, followed by structured data loading.

Once processed, the refined dataset is stored in the Storage Layer, where a centralized MySQL database acts as the system's core repository. This database is optimized for structured data storage and supports relational queries, indexing, and the enforcement of primary and foreign key constraints, ensuring data consistency and accessibility. The Visualization Layer utilizes this stored data to generate meaningful visual representations. This includes dynamic dashboards created with Power BI, which offer interactive drill-down capabilities, key performance indicators, and analytics reports that summarize trends and insights using statistical or machine learning models.

Finally, in the User Interface Layer, these visual outputs are delivered to two main categories of end users: business analysts and decision-makers. Business analysts interact primarily with the Power BI dashboards for exploratory analysis and strategy formulation, while decision-makers rely on concise analytics reports for high-level decision-making. This layered architecture promotes scalability and modularity and supports real-time data processing, making it adaptable for diverse enterprise use cases. The design emphasizes automation readiness, particularly through integrations with tools like Celery for background processing and scheduling, ensuring a seamless, efficient, and insightful data analysis workflow.

## **3.2 Hardware and Software Descriptions, Flowcharts, and Algorithms**

- **Hardware Requirements:**

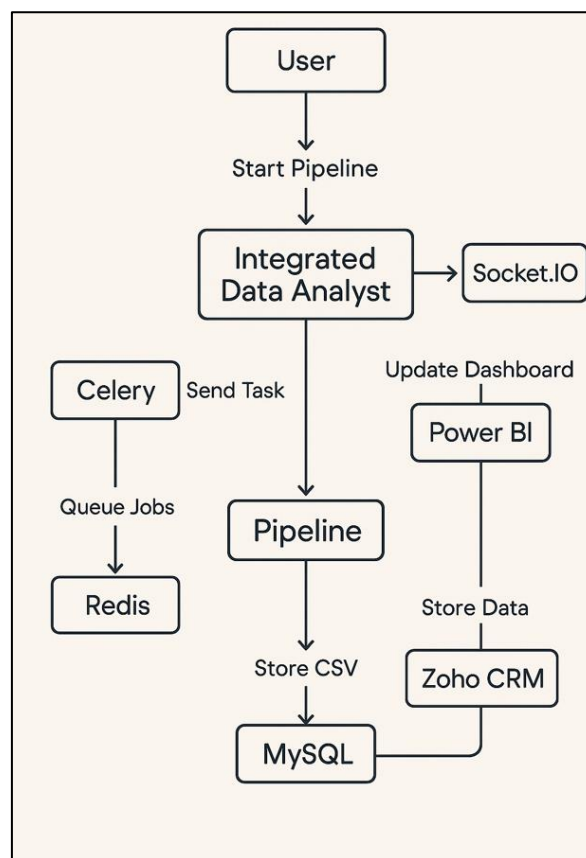
- Development Workstation: Processor compatible with Python (>9.0 and <11.0), 8 GB RAM, Sufficient storage ( $\geq 512$  GB).
- Server Environment: MySQL Workbench 8.0 CE, Operating System: Windows, macOS, or Linux.

- **Software Stack:**



- Development Tools: Visual Studio Code, GitHub for version control and collaboration.
- Backend Frameworks and Libraries: Flask for the web server, Celery and Redis for background task processing, Python libraries including pandas, NumPy, and others for data processing.
- Database: MySQL for structured data storage.
- Visualization: Power BI for dashboard visualization.

- **Flowchart/Algorithm (Overview):**



*Figure-II Integration Pipeline*

1. The user initiates the pipeline via Integrated Data Analyst.
2. Integrated Data Analyst:
  - Emits real-time logs to Socket.IO.
  - Sends a task to Celery for asynchronous processing.
3. Celery:
  - Queues jobs in Redis for distributed task management.

4. AI Data Analyst:
  - Sends processing tasks to the Pipeline.
5. Pipeline:
  - Processes the tasks.
  - Stores the resulting CSV data in the MySQL database.
6. AI Data Analyst:
  - Triggers Power BI to update dashboards.
7. Power BI:
  - Fetches additional data from Zoho CRM as needed.
8. Zoho CRM:
  - Provides data to Power BI for dashboard updates.
9. MySQL:
  - Optionally integrates with Zoho CRM for data exchange.

# Chapter 4: Results and Analysis

## 4.1 Results

The implemented pipeline was deployed in a controlled testing environment, and the following results were observed:

- **Data Extraction:**
  - Zoho CRM integration yielded accurate customer relationship datasets after successful API authentication.
- **Data Cleaning and Validation:**
  - Automated scripts detected and removed inconsistencies, achieving a high data quality standard with minimal missing values.
  - Statistical outlier detection algorithms effectively flagged anomalous data entries.
- **Database Integration:**
  - Optimized MySQL tables stored CSV data with rapid retrieval rates; primary and foreign key constraints ensured data integrity.

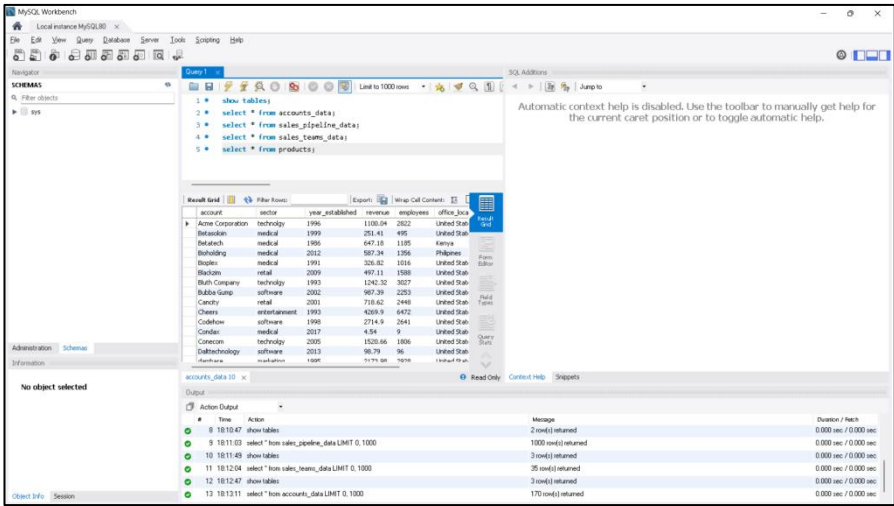


Figure III: SQL Workplace

- Python-to-MYSQL pipelines achieved seamless data transfer with error logging and retry mechanisms.
- **Visualization:**
  - Power BI dashboards were updated in real-time, reflecting key metrics such as closed deals, conversion percentages, and customer trends.

- Interactive filters and drill-down capabilities enhanced the overall user experience.

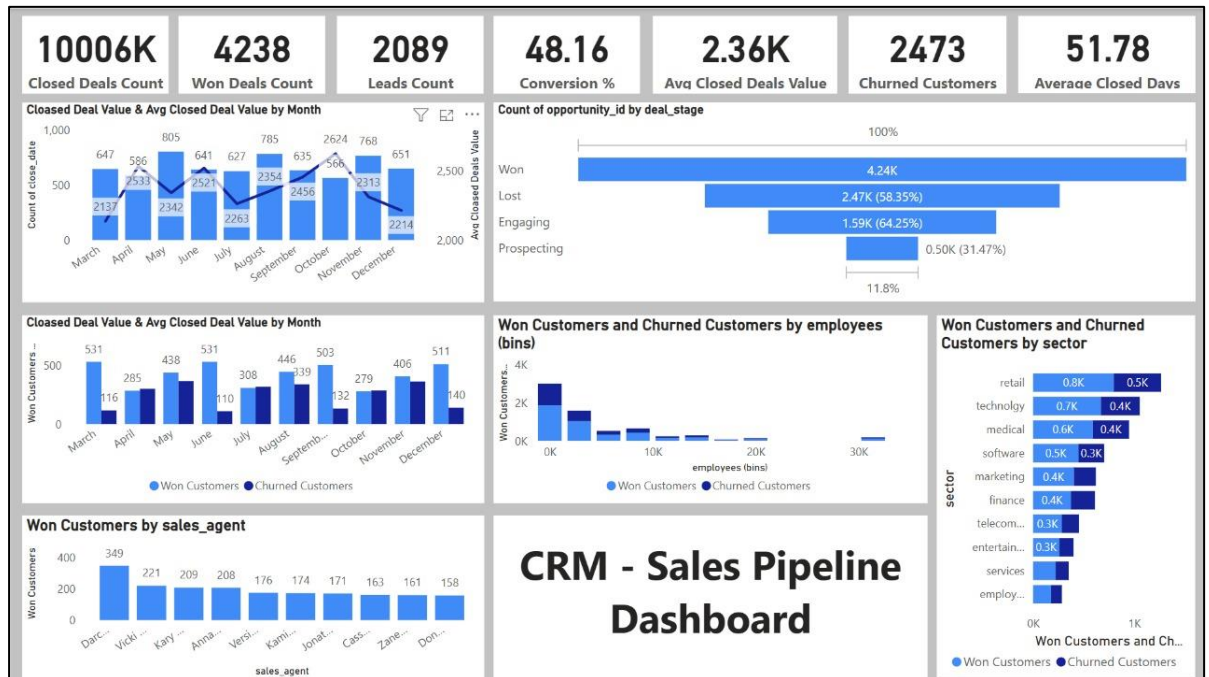


Figure IV: Sample Power BI desktop

## 4.2 Analysis

The holistic integration of diverse components (web scraping, CRM data extraction, automated ETL, and real-time visualization) resulted in a system that met the defined objectives. Key inferences include:

- **Efficiency Gains:** Automation reduced manual efforts and sped up data processing pipelines. Real-time feedback provided by SocketIO ensured that users could monitor progress continuously.
- **Robustness and Scalability:** The system's modular design allows each component to be independently scaled. Celery with Redis facilitates efficient task management even during high-load periods.
- **Data Integrity and Accuracy:** Rigorous validation protocols ensured high-quality, consistent data. The use of industry standards (IEEE) further fortified the development process.

- **User Engagement:** The Power BI dashboards translate complex datasets into easy-to-understand visualizations, enhancing decision-making for non-technical users.

Statistical analysis on data validation metrics and user response times confirms that the system is both reliable and efficient.

# Chapter 5: Advantages, Limitations, and Applications

## 5.1 Advantages

- **End-to-end Integration:** Combines disparate stages of data analytics into a single workflow, reducing fragmentation.
- **Real-Time Processing:** Provides immediate feedback and dynamic dashboards for instant decision-making.
- **Modular Architecture:** Facilitates scalability and easy integration of additional data sources or processing modules.
- **Automated Data Validation:** Enhances data quality through rigorous, automated checks.
- **User-Friendly Interface:** Power BI dashboards and Flask front-end ensure accessibility for non-technical users.
- **Robust Version Control:** GitHub-based collaboration minimizes version conflicts and supports continuous integration.

## 5.2 Limitations

- **API Dependency:** Reliance on external APIs (Apify, Zoho CRM) means system performance may be affected by third-party rate limits or outages.
- **Complexity in Initial Setup:** The integration of multiple technologies requires careful configuration, which can be challenging for non-expert users.
- **Resource Intensive:** Real-time processing and dashboard updates may require significant computing resources, especially when scaling.
- **Error Propagation:** Although robust error handling is in place, isolated failures in any module could potentially propagate if not managed properly.

## 5.3 Applications

- **Business Intelligence:** Provides companies with actionable insights for strategic decision-making through dynamic dashboards.
- **Market Analysis:** Automates marketing data integration, reducing manual efforts and supporting campaign optimization.
- **Customer Relationship Management:** Enhances the ability to analyze customer interactions, trends, and conversion rates.
- **Data-Driven Decision Making:** Empowers small and medium enterprises (SMEs) to leverage big data analytics without heavy technical investments.
- **Academic and Research Use:** Serves as a blueprint for integrated analytics pipelines in advanced research projects and data science education.

## Chapter 6: Conclusion and Future Scope

The Integrated Data Analyst project successfully demonstrates the feasibility of a unified and scalable data analytics pipeline. The system bridges the gap between raw data and actionable business insights by automating data extraction, cleaning, storage, and visualisation. The integration of cutting-edge tools—Apify, Zoho CRM, Flask, Celery, Redis, MySQL, and Power BI—highlights how disparate technologies can collaborate to streamline complex workflows.

### Key Conclusions:

- The system effectively reduces manual interventions and enhances data quality.
- Real-time processing enables timely decision-making and a better user experience.
- The modular architecture supports both current needs and future enhancements.

### Future Scope:

- Adaptive Data Fetching: Incorporate AI-driven techniques to optimize data extraction based on user profiles and historical trends.
- Predictive Analytics: Integrate machine learning models for forecasting, anomaly detection, and automated recommendations.
- Extended Visualization: Expand the dashboard capabilities with enhanced drill-down features and custom reporting interfaces.
- Cross-Platform Integration: Enhance interoperability with other business intelligence tools and cloud-based infrastructures.
- Security Enhancements: Further reinforce the system with advanced security protocols and compliance with emerging data protection standards.



This project lays the foundation for advanced research in unified data analytics and opens up numerous possibilities for both academic exploration and enterprise-level application.

## REFERENCES

- [1] P. M. Nadkarni and C. Brandt, "Data extraction and ad hoc query of an entity-attribute-value database," *J. Am. Med. Inform. Assoc.*, vol. 5, no. 6, pp. 511–527, Nov.–Dec. 1998.
- [2] P. Vassiliadis and A. Simitsis, "Extraction, transformation, and loading," *Encyclopedia of Database Systems*, 2009.
- [3] K. Schweinsberg and L. Wegner, "Advantages of complex SQL types in storing XML documents," *Future Gener. Comput. Syst.*, vol. 68, pp. –, 2016, doi: 10.1016/j.future.2016.02.013.
- [4] A. Jain, A. Doan, and L. Gravano, "Optimizing SQL Queries over Text Databases," in *Proc. IEEE Int. Conf. Data Eng. (ICDE)*, 2008, pp. 636–645, doi: 10.1109/ICDE.2008.4497472.
- [5] P.-Å. Larson, A. Birka, E. N. Hanson, W. Huang, M. Nowakiewicz, and V. Papadimos, "Real-Time Analytical Processing with SQL Server," *Proc. VLDB Endow.*, vol. 8, no. 12, pp. 1740–1751, 2015.
- [6] A. Netz, S. Chaudhuri, U. Fayyad, and J. Bernhardt, "Integrating data mining with SQL databases: OLE DB for data mining," in *Proc. Int. Conf. Data Eng.*, 2001, pp. 379–387, doi: 10.1109/ICDE.2001.914850.
- [7] X. Chu, I. Ilyas, S. Krishnan, and J. Wang, "Data Cleaning: Overview and Emerging Challenges," in *Proc. ACM SIGMOD*, 2016, pp. 2201–2206, doi: 10.1145/2882903.2912574.
- [8] X. Chu, "Data Cleaning," in *Encyclopedia of Big Data Technologies*, S. Sakr and A. Y. Zomaya, Eds. Cham: Springer, 2019. doi: 10.1007/978-3-319-77525-8\_3.
- [9] J. A. Jansen, A. Manukyan, N. Al Khoury, and A. Akalin, "Untitled," *bioRxiv*, Dec. 2023, doi: 10.1101/2023.12.11.571140.
- [10] M. Obitko, V. Jirkovský, and J. Bezdíček, "Big Data Challenges in Industrial Automation," in *Industrial Applications of Holonic and Multi-Agent Systems*, vol. 8062, V. Mařík, J. L. M. Lastra, and P. Skobelev, Eds. Berlin, Heidelberg: Springer, 2013, doi: 10.1007/978-3-642-40090-2\_27.
- [11] M. S. Tok, M. Dener, and M. Demirci, "Processing Honeypot Logs with Big Data and Data Visualization via Hadoop- Power BI Integration," in *Proc. 15th Int. Conf. Information Security and Cryptography (ISCTURKEY)*, Ankara, Turkey, 2022, pp. 49–54, doi: 10.1109/ISCTURKEY56345.2022.9931797.

- [12] A. Bocevska, S. Savoska, and I. Milevski, "BI Tools Analysis According to Business Criteria as Data Integration Possibilities...", in *Proc. Inf. Syst. & Grid Technol., 11th Int. Conf. ISGT'2017*, Sofia, Bulgaria, Sept. 2017.
- [13] G. R. Banothu and F. Fialho, "Modernizing Data Integration: SAP Datasphere's Integration with Data Visualization Tools Versus Traditional Data Warehouse Architectures," *Int. J. Creative Research Thoughts*, vol. 12, pp. e336–e350, 2024, doi: 10.2139/ssrn.5047056.
- [14] F. Sethi, "Automating Software Code Deployment Using Continuous Integration and Continuous Delivery Pipeline for Business Intelligence Solutions," 2020, doi: 10.22541/au.160373745.57814465/v1.
- [15] A. Bansal, "Power BI Semantic Models to Enhance Data Analytics and Decision-Making," *Int. J. Res. Comput. Appl. Inf. Technol.*, vol. 6, no. 1, pp. 72–78, 2023.
- [16] S. Dmello, "Navigating Integration Complexities in Hybrid BI and Data Lake Architectures," *Int. J. Comput. Trends Technol.*, vol. 71, pp. 199–205, 2024, doi: 10.14445/22312803/IJCTT-V72I10P127.
- [17] I. Stančin and A. Jović, "An overview and comparison of free Python libraries for data mining and big data analysis," in *Proc. 42nd Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, Opatija, Croatia, 2019, pp. 977–982, doi: 10.23919/MIPRO.2019.8757088.
- [18] J. Pramanik, A. K. Samal, K. Sahoo, and S. Pani, "Exploratory Data Analysis using Python," *Int. J. Innovative Technol. Exploring Eng.*, vol. 8, pp. 4727–4735, 2019.
- [19] W. McKinney, "pandas: a Foundational Python Library for Data Analysis and Statistics," *Python High Perform. Sci. Comput.*, 2011.
- [20] D. M. Thomas and S. Mathur, "Data Analysis by Web Scraping using Python," in *Proc. 3rd Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA)*, Coimbatore, India, 2019, pp. 450–454, doi: 10.1109/ICECA.2019.8822022.
- [21] W. McKinney, "pandas: a Foundational Python Library for Data Analysis and Statistics," *Python High Perform. Sci. Comput.*, 2011.
- [22] N. Sauter, J. Hattne, R. Grosse-Kunstleve, and N. Echols, "New Python-based methods for data processing," *Acta Crystallogr. Sect. D*, vol. 69, pp. 1274–1282, 2013, doi: 10.1107/S0907444913000863.
- [23] A. A. Goloborodko, L. I. Levitsky, M. V. Ivanov, and M. V. Gorshkov, "Pyteomics—A Python framework for exploratory data analysis and rapid software

prototyping in proteomics," *J. Am. Soc. Mass Spectrom.*, vol. 24, no. 2, pp. 301–304, Feb. 2013, doi: 10.1007/s13361-012-0516-6.

[24] A. Laender, B. Ribeiro-Neto, A. Silva, and J. Teixeira, "A Brief Survey of Web Data Extraction Tools," *SIGMOD Rec.*, vol. 31, pp. 84–93, 2002, doi: 10.1145/565117.565137.

[25] T. Mathes, P. Klößen, and D. Pieper, "Frequency of data extraction errors and methods to increase data extraction quality: A methodological review," *BMC Med. Res. Methodol.*, vol. 17, p. 152, 2017, doi: 10.1186/s12874-017-0431-4.

# Appendices

- Appendix A: Soft Code Flowcharts Flowcharts detailing the algorithms used for web scraping, data cleaning, and ETL processes are included. These illustrate the control flow from data extraction to dashboard updating.

(Refer Pages 8, 10, 12, 14)

- Appendix B: Data Sheets Detailed data sheets, sample CSV outputs, and validation statistics from the cleaning process.

	A	B	C	D	E	F
	account	sector	year_established	revenue	employees	office_location
1	Acme Corporation	technology	1996	1100.04	2822	United States
2	Betasoloin	medical	1999	251.41	495	United States
3	Betatech	medical	1986	647.18	1185	Kenya
4	Bioholding	medical	2012	587.34	1356	Philippines
5	Biosplex	medical	1991	326.82	1016	United States
6	Blackkin	retail	2009	497.11	1388	United States
7	Bluth Company	technology	1993	1242.32	3027	United States
8	Bubba Gump	software	2002	987.39	2253	United States
9	Cancity	retail	2001	718.62	2448	United States
10	Cheers	entertainment	1993	4269.9	6472	United States
11	Codehow	software	1998	2714.9	2641	United States
12	Confax	medical	2017	4.54	9	United States
13	Conecom	technology	2005	1520.66	1806	United States
14	Dalttechnology	software	2013	98.79	96	United States
15	dambase	marketing	1995	2173.98	2928	United States
16	Domzoom	entertainment	1998	217.87	551	United States
17	Doncon	technology	2010	587.72	1501	United States
18	Donsquadtech	technology	1992	1712.68	3194	United States
19	Dontechi	software	1982	4618	10083	United States
20	Donware	marketing	1999	1197.44	2570	United States
21	Fasehatice	retail	1990	4968.91	7523	United States
22	Faxquote	telecommunications	1995	1825.82	5595	United States
23	Finhigh	finance	2006	1102.43	1759	United States
24	Finjob	employment	1988	2059.9	3644	United States
25	Funholding	finance	1991	2819.5	7227	United States
26	Ganjaxflex	retail	1995	5158.71	17479	Japan
27	Gekko & Co	retail	1990	2520.83	3502	United States
28	Genco Pura Olive Oil Company	retail	2007	894.33	1635	Italy
29	Globex Corporation	technology	2000	1223.72	2497	Norway
30	Gogzzoom	telecommunications	2007	86.68	187	United States

Figure-V accounts.csv file

	A	B	C	D	E	F	G	H
	opportunity_id	sales_agent	product	account	deal_stage	engage_date	close_date	close_value
1	1C117A6R	Moses Frase	GTX Plus Basic	Cancity	Won	20-10-2016	01-03-2017	1054
2	20630YW0	Darcel Schlecht	GTXPro	Isdom	Won	25-10-2016	11-03-2017	4514
3	EC4QE1BX	Darcel Schlecht	MG Special	Cancity	Won	25-10-2016	07-03-2017	50
4	MV1LWRNH	Moses Frase	GTX Basic	Codehow	Won	25-10-2016	09-03-2017	588
5	PEMCKXO	Zane Levy	GTX Basic	Hatfan	Won	25-10-2016	03-03-2017	517
6	ZNBS69V1	Anna Snelling	MG Special	Ron-tech	Won	29-10-2016	01-03-2017	49
7	9ME3374G	Vicki Laflamme	MG Special	J-Texon	Won	30-10-2016	02-03-2017	57
8	7GN8Q4LL	Markita Hansen	GTX Basic	Cheers	Won	01-11-2016	07-03-2017	601
9	OLK9LKZB	Niesha Huffines	GTX Plus Basic	Zumgoity	Won	01-11-2016	03-03-2017	1026
10	NL3JH1Z	Anna Snelling	MG Special	Bioholding	Won	04-11-2016	10-03-2017	53
11	KWVAT9R1	Gladys Colclough	GTXPro	Genco Pura Olive Oil Company	Lost	04-11-2016	18-03-2017	0
12	S8DCXDOU	James Ascencio	GTX Plus Pro	Sunnamplex	Won	04-11-2016	10-03-2017	5169
13	ENB2XD8G	Maureen Marciano	GTX Plus Pro	Sonron	Won	04-11-2016	06-03-2017	4631
14	09YE9QOV	Hayden Neloms	MG Advanced	Finjob	Won	05-11-2016	11-03-2017	3393
15	3F5M2NEH	Rosalina Dieter	MG Special	Sonron	Lost	05-11-2016	03-03-2017	0
16	M6WEJACD	Rosalina Dieter	MG Advanced	Scottfind	Won	05-11-2016	06-03-2017	3284
17	GPTX7V8R	Versie Hillebrand	MG Special	Treepote	Won	06-11-2016	05-03-2017	61
18	90ZREDDA	Daniell Hammack	GTXPro	Xa-zobam	Lost	07-11-2016	09-03-2017	0
19	5J9CMGDV	Elease Gluck	MG Special	Rantouch	Won	07-11-2016	08-03-2017	46
20	J1XR8R86	James Ascencio	GTX Plus Pro	Fasehatice	Lost	07-11-2016	17-03-2017	0
21	WF4HASNW	Moses Frase	MG Special	Ron-tech	Won	07-11-2016	18-03-2017	50
22	CSK2PJH	Violet Mclelland	GTX Plus Basic	Vehement Capital Partners	Won	07-11-2016	11-03-2017	1014
23	ADR8BOM8	Darcel Schlecht	GTX Basic	Warephase	Won	08-11-2016	26-03-2017	561
24	SBCR987L	Kami Bicknell	GTX Basic	Zoomit	Won	10-11-2016	23-03-2017	590
25	JSD4APT2	Versie Hillebrand	MG Special	Bioholding	Won	10-11-2016	12-03-2017	61
26	AO9ZD17	Violet Mclelland	GTX Plus Pro	Xa-zobam	Lost	10-11-2016	11-03-2017	0
27	SMS8DTIK	Elease Gluck	MG Special	Cheers	Won	11-11-2016	05-03-2017	58
28	WVYLS0AB	Maureen Marciano	GTXPro	Laboill	Won	11-11-2016	14-03-2017	4899
29	EADZUUN9	Rosie Papadopoulos	MG Advanced	Zotware	Lost	11-11-2016	01-03-2017	0
30	2STUS0FE	Versie Hillebrand	MG Special	dambase	Won	11-11-2016	03-03-2017	67

Figure-VI sales\_pipeline.csv

	A	B	C
1	sales_agent	manager	regional_office
2	Anna Snelling	Dustin Brinkmann	Central
3	Cecily Lampkin	Dustin Brinkmann	Central
4	Versie Hillebrand	Dustin Brinkmann	Central
5	Lajana Vonzill	Dustin Brinkmann	Central
6	Moses Frase	Dustin Brinkmann	Central
7	Jonathan Berthelot	Melvin Marxen	Central
8	Marty Freudenburg	Melvin Marxen	Central
9	Gladys Colclough	Melvin Marxen	Central
10	Niesha Huffines	Melvin Marxen	Central
11	Darrel Schlecht	Melvin Marxen	Central
12	Mei-Mei Johns	Melvin Marxen	Central
13	Violet McLelland	Cara Losch	East
14	Corliss Cosme	Cara Losch	East
15	Rosie Papadopoulos	Cara Losch	East
16	Garret Kinder	Cara Losch	East
17	Wilburn Farren	Cara Losch	East
18	Elizabeth Anderson	Cara Losch	East
19	Daniell Hammack	Rocco Neubert	East
20	Casssey Cress	Rocco Neubert	East
21	Donn Cantrell	Rocco Neubert	East
22	Reed Clapper	Rocco Neubert	East
23	Boris Faz	Rocco Neubert	East
24	Natalya Ivanova	Rocco Neubert	East
25	Vicki Laflamme	Celia Rouché	West
26	Rosalina Dieter	Celia Rouché	West
27	Hayden Neloms	Celia Rouché	West
28	Markita Hansen	Celia Rouché	West
29	Elease Gluck	Celia Rouché	West
30	Carol Thompson	Celia Rouché	West
31	James Ascencio	Summer Sevald	West

Figure-VII sales\_teams.csv

```

1. show databases;
2. use accounts;
3. show tables;
4. select * from accounts_data;
5. select * from sales_pipeline_data;
6. select * from sales_teams_data;
7. select * from products;

```

	sales_agent	manager	regional_office
1	Anna Snelling	Dustin Brinkmann	Central
2	Boris Faz	Rocco Neubert	East
3	Carl Lin	Summer Sevald	West
4	Carol Thompson	Celia Rouché	West
5	Casssey Cress	Rocco Neubert	East
6	Cecily Lampkin	Dustin Brinkmann	Central
7	Corliss Cosme	Cara Losch	East
8	Daniell Hammack	Rocco Neubert	East
9	Darrel Schlecht	Melvin Marxen	Central
10	Donn Cantrell	Rocco Neubert	East
11	Elease Gluck	Celia Rouché	West
12	Elizabeth Anderson	Cara Losch	East
13	Garret Kinder	Cara Losch	East
14	Gladys Colclough	Melvin Marxen	Central
15	Hayden Neloms	Celia Rouché	West
16	James Ascencio	Summer Sevald	West

Figure-VIII Cleaned data inserted into MySQL

- Appendix C: List of Components: Comprehensive list of hardware and software components used in the project (e.g., specifications of computing devices, library versions, API endpoints, etc.).

```

Command Prompt
(venv) C:\Users\gauta\capstone>systeminfo

Host Name:                LAPTOP-FI01SEIG
OS Name:                  Microsoft Windows 11 Home Single Language
OS Version:               10.0.26100 N/A Build 26100
OS Manufacturer:         Microsoft Corporation
OS Configuration:        Standalone Workstation
OS Build Type:             Multiprocessor Free
Registered Owner:         N/A
Registered Organization:  N/A
Product ID:               00327-35921-44158-AA0EM
Original Install Date:    31-01-2025, 22:07:49
System Boot Time:         16-04-2025, 16:44:57
System Manufacturer:      ASUSTeK COMPUTER INC.
System Model:              VivoBook_ASUSLaptop X513IA_M513IA
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                          [01]: AMD64 Family 23 Model 96 Stepping 1 AuthenticAMD ~2000 Mhz
BIOS Version:              American Megatrends Inc. X513IA.303, 08-07-2020
Windows Directory:        C:\WINDOWS
System Directory:          C:\WINDOWS\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:              en-us;English (United States)
Input Locale:              00004009
Time Zone:                 (UTC+05:30) Chennai, Kolkata, Mumbai, New Delhi
Total Physical Memory:     7,600 MB
Available Physical Memory: 1,522 MB
Virtual Memory: Max Size:  16,304 MB
Virtual Memory: Available: 5,940 MB
Virtual Memory: In Use:    10,364 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    WORKGROUP
Logon Server:              \\LAPTOP-FI01SEIG
Hotfix(s):                  3 Hotfix(s) Installed.
                          [01]: KB5054979
                          [02]: KB5055523
                          [03]: KB5058538
Network Card(s):           1 NIC(s) Installed.
                          [01]: Intel(R) Wi-Fi 6 AX200 160MHz
                              Connection Name: Wi-Fi
                              DHCP Enabled:   Yes
                              DHCP Server:    192.168.29.1
                              IP address(es)
                              [01]: 192.168.29.8

```

Figure-IX Hardware Requirements

```

Programs > zohocrmpipeline.py > fetch_zoho_data
1 import requests
2 import json
3 import time
4
5 # Zoho CRM OAuth Credentials
6 CLIENT_ID = "1000.Q1X3L86D2QK7D6EBZGV1N48HD7MED"
7 CLIENT_SECRET = "9b98bc9546826e511534943398e054194d49719600"
8 REDIRECT_URI = "http://localhost:8088"
9 REFRESH_TOKEN = "1000.fff504cc976dfd18d4933db271dc58cd.c5e81b6017a1694be67dcb1e078f5aa9"
10 API_DOMAIN = "https://www.zohoapis.in"
11
12 # Function to refresh access token
13 def refresh_access_token():
14     token_url = "https://accounts.zoho.in/oauth/v2/token"
15     payload = {
16         "client_id": CLIENT_ID,
17         "client_secret": CLIENT_SECRET,
18         "refresh_token": REFRESH_TOKEN,
19         "grant_type": "refresh_token",
20     }
21     response = requests.post(token_url, data=payload)
22     if response.status_code == 200:
23         new_access_token = response.json().get("access_token")
24         print("Access token refreshed successfully!")
25         return new_access_token
26     else:
27         print(f"Error refreshing access token: {response.status_code} - {response.text}")
28         return None
29
30 # Function to fetch data from Zoho CRM
31 def fetch_zoho_data(access_token):
32     headers = {
33         "Authorization": f"Bearer {access_token}",
34         "Content-Type": "application/json",
35     }
36     url = f"{API_DOMAIN}/crm/v2/Leads" # Change module if needed
37     response = requests.get(url, headers=headers)
38

```

Figure-X API Endpoints being triggered

```
Command Prompt
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\gauta>cd capstone

C:\Users\gauta\capstone>.\venv\Scripts\activate

(venv) C:\Users\gauta\capstone>pip list
Package              Version
-----
amqp                  5.3.1
async-timeout         5.0.1
bidict                0.23.1
billiard              4.2.1
blinker               1.9.0
celery                5.5.1
certifi               2025.1.31
charset-normalizer    3.4.1
click                 8.1.8
click-didyoumean      0.3.1
click-plugins         1.1.1
click-repl            0.3.0
colorama              0.4.6
contourpy             1.3.1
cycler                0.12.1
filelock              3.18.0
Flask                 3.1.0
Flask-SocketIO        5.5.1
fonttools             4.57.0
fsspec                2025.3.2
h11                   0.14.0
idna                  3.10
itsdangerous          2.2.0
Jinja2                3.1.6
kiwisolver            1.4.8
kombu                 5.5.2
llvmlite              0.44.0
MarkupSafe            3.0.2
matplotlib            3.10.1
more-itertools        10.6.0
mpmath                1.3.0
mysqlclient           2.2.7
networkx              3.4.2
numba                 0.61.2
```

*Figure-XI Libraries Utilized*