**Automated Email Response System**

**A PROJECT REPORT**

***Submitted by***

**Kamal JR (220701117)**

***in partial fulfillment for the course***

**OAI1903 - INTRODUCTION TO ROBOTIC PROCESS AUTOMATION**

***for the degree of***

**BACHELOR OF ENGINEERING**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**RAJALAKSHMI ENGINEERING COLLEGE**

**RAJALAKSHMI NAGAR**

**THANDALAM**

**CHENNAI – 602 105**

**NOVEMBER 2024**

**RAJALAKSHMI ENGINEERING COLLEGE**

**CHENNAI - 602105**

**BONAFIDE CERTIFICATE**

Certified that this project report **“Email Response Sys”** is the bonafide work of **“Kamal JR (220701117)”** who carried out the project work for the subject OAI1903-Introduction to Robotic Process Automation under my supervision.

|  |
| --- |
| Mrs. J. Jinu Sophia  **SUPERVISOR**  Assistant Professor (SG)  Department of  Computer Science and Engineering  Rajalakshmi Engineering College  Rajalakshmi Nagar  Thandalam  Chennai - 602105 |

Submitted to Project and Viva Voce Examination for the subject OAI1903-Introduction to Robotic Process Automation held on \_\_\_\_\_\_\_\_\_\_.

**TABLE OF CONTENTS**

**CHAPTER NO. TITLE PAGE NO.**

**ABSTRACT 5**

**LIST OF TABLE 5**

**LIST OF FIGURES 5**

**LIST OF ABBREVIATIONS 5**

**1. INTRODUCTION 6**

1.1 GENERAL 6

1.2 OBJECTIVE 6

1.3 EXISTING SYSTEM 6

1.4 PROPOSED SYSTEM 7

**2. LITERATURE REVIEW 7**

2.1 GENERAL 7

**3. SYSTEM DESIGN 7**

3.1 GENERAL 7

3.1.1 SYSTEM FLOW DIAGRAM 9

3.1.2 ARCHITECTURE DIAGRAM 10

3.1.3 SEQUENCE DIAGRAM 10

**4. PROJECT DESCRIPTION 11**

4.1 METHODOLOGIE 11

4.2 MODULES 13

**5. CONCLUSIONS 15**

5.1 GENERAL 15

**REFERENCES 17**

**OUTPUT 18**

**APPENDICES 20**

**ACKNOWLEDGEMENT**

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman Mr. S. Meganathan, B.E, F.L.E., our Vice Chairman Mr. Abhay Shankar Meganathan, B.E., M.S., and our respected Chairperson Dr. (Mrs.) Thangam Meganathan, Ph.D., for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution

Our sincere thanks to Dr. S.N. Murugesan, M.E., Ph.D., our beloved Principal for his kind support and facilities provided to complete our work in time We express our sincere thanks to Dr. P. Kumar, M.E., Ph.D., Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guides, Mrs. J. Jinu Sophia, M.E., (Ph.D.), Assistant Professor (SG), Department of Computer Science and Engineering. Rajalakshmi Engineering College for her valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinators, Dr. N. Durai Murugan, M.E., Ph.D., Associate Professor, and Mr. B. Bhuvaneswaran, M.E.. Assistant Professor (SG), Department of Computer Science and Engineering for their useful tips during our review to build our project

**Kamal JR (220701117)**

**ABSTRACT**

This paper presents an Automated Email Response System developed using UiPath to streamline the process of responding to common queries received via email. The system automatically processes unread emails from an inbox, matches them with predefined keywords, sends responses based on those keywords, and logs the details of the processed emails in an Excel sheet. The aim is to improve efficiency in handling repetitive email tasks while ensuring accurate and consistent replies. The system is designed to be scalable, reliable, and easily customizable for different email handling scenarios.

**LIST OF TABLES**

Table 1: List of keywords for email filtering

Table 2: SMTP server configurations

Table 3: Excel sheet logging format

Table 4: System error codes and handling measures

**LIST OF FIGURES**

Figure 1: Architecture Diagram of the Automated Email Response System

Figure 2: System Flow Diagram

Figure 3: Sequence Diagram for Email Processing

Figure 4: Data Flow Diagram of the System

Figure 5: Activity Diagram for Email Processing

**LIST OF ABBREVIATIONS**

IMAP - Internet Message Access Protocol

SMTP - Simple Mail Transfer Protocol

UIPath - A Robotic Process Automation (RPA) tool

DFD - Data Flow Diagram

API - Application Programming Interface

UI - User Interface

Excel - Microsoft Excel for data logging

**1. INTRODUCTION**

**1.1 GENERAL**

Email-based communication is now an important source of contacting people in both personal and professional lives. Since routine operations are associated with e-mail communications, automatic processing and response to e-mails lead to higher productivity. The Automated Email Response System will focus mainly on the processing of common e-mails that do or do not carry keywords and are subject to predefined rules. This project is designed using UiPath, which is an RPA tool that connects to email servers and automatically sends responses.

**1.2 OBJECTIVE**

This project intends to automate the response to frequently asked questions or general queries based on an email. The system, therefore ought to accomplish the following objectives:

Connect to an IMAP-enabled email account.

Filter emails based on keywords previously defined

Send an appropriate automated response through SMTP.

Record the email details in an Excel file for record-keeping and tracking purposes.

The system decreases the manual effort, elimination of error in response, and accelerates the turnaround time in communication by way of an email.

**1.3 EXISTING SYSTEM**

These responses to customer inquiries are manually drafted and mailed out by support teams. Such processes take a long time to be accomplished because of human error. Automation tools available for email currently, mostly need a complex manual setup and require high technical expertise, limiting their adoption to small and medium enterprises. Currently, systems may not be flexible enough to change or update response templates to accommodate the dynamic evolution of businesses.

**1.4 PROPOSED SYSTEM**

The Automated Email Response System will make email communications easy by: automatically responding emails using keyword filtering within the email subject or body logging details about emails (name of recipient, timestamp, etc.) into an Excel sheet for auditing purposes utilizing the intuitive automation tool UiPath in handling all emails, assuring the non-technical user will not find a problem with usability. Such a system can easily be highly configurable to accommodate many keywords, SMTP servers, and configurations of email accounts.

**2. LITERATURE REVIEW**

**2.1 General**

Although there have been many suggestions for automating email responses in the past-from simple keyword detection to full-nature natural language processing to produce dynamic replies-there are many more user-friendly tools available today, offering users opportunities to automate and create tasks concerning email flows without requiring complex code. However, these tools are limited, for example, by low customizability or scalability.

Most of the automation solutions focus on predefined response templates, with little focus on real-time customization and easy scalability. It has become pretty apparent that an organization like UiPath is much needed for applications in which businesses or individuals receive similar inquiries in high volumes daily in the environments.  
  
**3. SYSTEM DESIGN**

**3.1 GENERAL**

The system design describes the architecture and workflow of the proposed solution, showing how varied components interact to automate the process of email handling.

**3.1.1 SYSTEM FLOW DIAGRAM**

The system flow diagram depicts the step-by-step process of an email automation system, from retrieval of an email, logging the response. Let's draw it out in this manner:

Email Arrival: Email arrived at the server.

Email Fetch: The system retrieves the email using IMAP .

Keyword Scan: The system scans the email for keywords.

Response Generation: An appropriate response is generated based on the identified keywords.

Response Sending: The system sends the email through SMTP .

Logging: All interaction details are logged in an Excel file.

Attach flowchart representing the above process here.

**3.1.2 ARCHITECTURE DIAGRAM**

Architecture Diagram As shown above, the architecture diagram of the whole system depicting the major components within the system:

IMAP Server: To receive emails.

Processing Engine: Where keyword analysis and response generation occur.

SMTP Server: To send automated responses.

Logging Module: Where all interactions are logged in an Excel file.

**3.1.3 SEQUENCE DIAGRAM**

Sequence diagram shows a detailed view of the interaction sequences of the system components. Here is how to organize it:

User Sends Email: A user sends an email to the server.

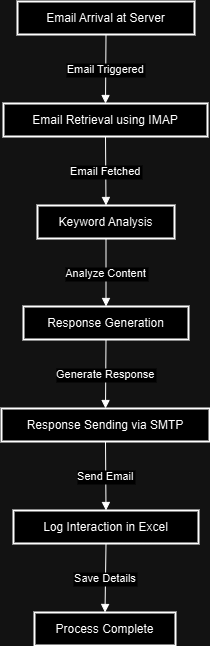
Email Retrieved: The system retrieves the mail through an IMAP protocol.

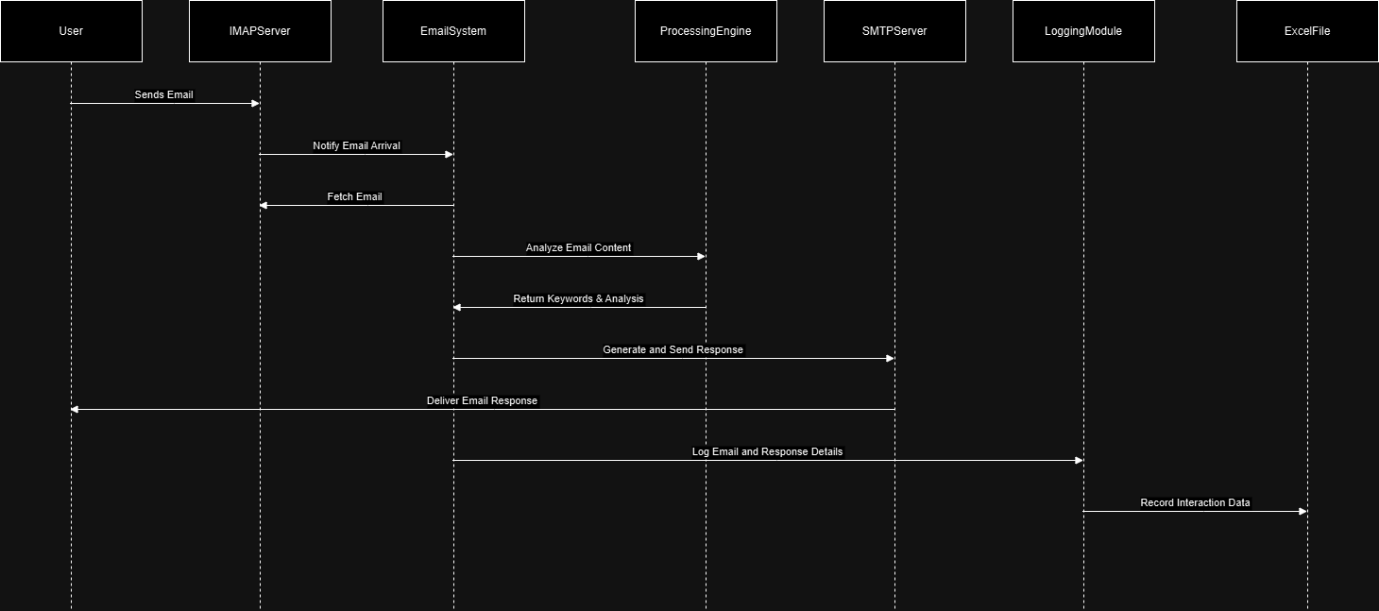
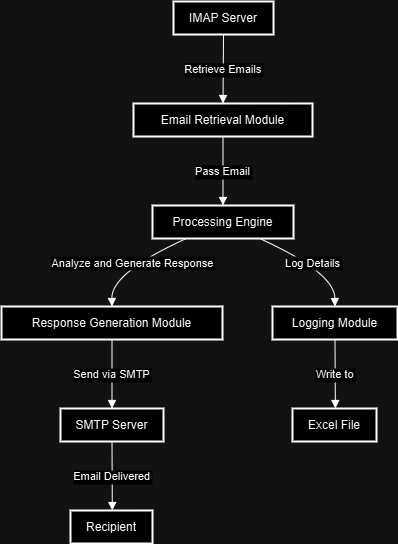
Analysis Performed: The text is parsed for keywords.

Response Formulated: The response is crafted appropriately.

Email Sent: The response is sent through the SMTP protocol.

Details Logged: The details are logged in the logging module.

**  
  
System Flow Diagram**

****  
  
**Architecture Diagram**

**Sequence Diagram**

**4. Project Description:**

The Automated Email Management System is an integrated solution designed with the objective to streamline and automate incoming as well as outgoing communications through the means of email. Advanced email protocols, machine learning techniques, and efficient logging mechanisms allow for the development of a robust and reliable automated system. Now, we move on to the methodologies and modules that constitute the base of this system.

**4.1 Methodologies**

This section explains the various techniques and technologies utilized in the project to achieve email automation. These methodologies ensure efficiency, accuracy, and scalability of the system.

**4.1.1 IMAP (Internet Message Access Protocol)**

Purpose: IMAP is used to retrieve and manage emails stored on a mail server. Unlike the older POP3 protocol, IMAP allows for two-way communication with the mail server, keeping emails in sync across multiple devices.

Implementation: The system uses IMAP to securely connect into the email server, check for new messages, and download emails for processing. IMAP protocol enables selective retrieval; that is, it allows the system to retrieve only the necessary information, for example, subject, sender, and content, without downloading all email attachments, which can consume bandwidth and process time.

Advantages of IMAP: IMAP enables the retrieval of messages efficiently, supports real-time updates of the repository, and the system always has the latest information about emails.

**4.1.2 Natural Language Processing**

Objective: NLP is actually a subarea of AI where these machines can understand and interpret human language. Here, in this project, NLP finds out which keywords or phrases from the content of the incoming emails apply to a particular criterion.

Implementation: This NLP engine processes the email content by decompounding sentences into words, removing stop words such as "the," "and," "of," and then applying algorithms to identify important keywords. These keywords are matched against a predefined dictionary of terms that have corresponding automated responses.

Use Case: If the word "account locked" appears in an email, the NLP engine would know to provide a response about unlocking accounts.

Benefits: NLP enables searching for content that is semantically much more intelligent than keyword matching, thereby reducing false positives and making it easier for the system to manage very complex queries about emails.

**4.1.3 SMTP (Simple Mail Transfer Protocol)**

Purpose: SMTP is used to send emails from the system to recipient mail boxes. It is the standard protocol for sending emails over the internet.

Implementation: The system makes use of SMTP to send out automatically generated responses. The system supports authentication mechanisms, so that emails can be sent from only authenticated accounts - reducing spam and abuse.

Advantages: SMTP sends e-mail messages reliably and very quickly, with built-in error handling and retry mechanisms in the event that the recipient's server is temporarily down.

**4.1.4 Logging**

Purpose : Logging is important in the recordation of every interaction that the system performs. An audit trail exists to be used in debugging, analysis, and compliance.

Implementation: Each interaction via email, including the communication being analyzed, the response that was created for it, along with the timestamp, is recorded to an Excel file or database. The structured logging allows easy search capability and review of previous interactions.

Benefits: The logging mechanism ensures transparency error tracking, and valuable data retrieval for performance analysis.

**4.2 Modules**

The system consists of multiple interlinked modules where every module represents a different part of the automation process. Below is the elaboration of each module and how they work and are integrated to form a coherent system.

**4.2.1 Email Retrieval Module:**

Functionality: The module establishes a connection with the IMAP server to retrieve new emails. It authenticates with the email server and checks for unread or new messages using secure protocols.

Process Flow:

Connect securely to the IMAP server.

Scan inbox for new messages

Fetch metadata from email (from, subject, received-date) and body.

Save the pulled emails locally in temp storage to be used later.

Error Handling: If the connection cannot be established or if the server is down, the module retries connecting after an interval. Then it logs the error.

Efficiency: The module utilizes optimized queries to fetch only relevant data. Its resource usage is thereby minimized by the system.

**4.2.2 Content Analysis Module:**

Functionality: This module processes the content of the retrieved emails with the help of NLP techniques. It identifies phrases, sentiment, and intent in a message.

Process Flow:

Tokenize the email content into individual words.

Remove common stop words and apply stemming or lemmatization to simplify the words.

Analyze the remaining words to detect keywords and phrases.

Match the mined keywords to a pre-defined list of triggers.

Deep Analytics: The email is identifiable with a positive, negative, or neutral sentiment, which is useful in composing an appropriate response tone.

Configuration: The list of keywords with corresponding responses may be upgraded or customized as needed by business requirements.

Benefits: NLP enables this module to handle all kinds of linguistic variations - it can interpret many types of writing and vocabularies.

**4.2.3 Response Generation Module:**

Functionality: This module produces a machine-generated response as output based on the identified keywords from the email content

Process Flow:

Retrieve the matched keyword's associated response template.

Insert dynamic data where applicable (e.g., insert the name of the recipient or refer to specific details from the email).

The response is formatted, and ready for sending.

Customization: The system allows multiple response templates, which can be customized according to different scenarios or departments, for example, for customer service, sales questions, technical problems.

Error Handling: If the system could not generate the response due to the missing data or in case of template errors, it logs the error and sends an alert to the administrator.

**4.2.4 Email Sending Module:**

Functionality: This module is responsible for sending the responses it has generated using SMTP, covering both authentications, formatting, and error management related to sending.

Process Flow:

Authenticate using the sender's credentials.

Establish a secure connection with the SMTP server.

Send an email and await confirmation of receipt from the SMTP server.

When an email is successfully sent, log the interaction.

When an email fails to send, either retry or log the error for further investigation.

Security Measures: The module uses encrypted connections (SSL/TLS) to ensure the privacy and integrity of the emails being sent. It also implements measures to prevent unauthorized access and spam.

**4.2.5 Logging Module:**

Functionality: This module maintains a comprehensive record of all system activities, including email retrieval, content analysis, response generation, and email sending.

Data Logged:

Email metadata (e.g., sender, subject, timestamp).

Keywords identified during analysis.

The response generated and sent.

Any exceptions or errors that occur during the process.

Storage: They store logs in an organized manner with either a relational database or an Excel file. It makes querying about past interactions possible.

Benefits: The logging module provides transparency as well as accountability. Notably, it is an essential resource for debugging problems and improving the performance of the system over time.

**5. CONCLUSIONS**

This section summarizes the important outcomes and impact of the project to provide insight to readers about the effectiveness, reliability, and prospects for the future of the Automated Email Handling System.

**5.1 GENERAL**

The system developed in this project, Automated Email Handling System, has shown significant improvements in the process of streamlining and automating an email communication management process. The integrated modular design has done well in overcoming major challenges attributed to the conventional or traditional modes of email handling that are handled manually. During the project, reducing response times, increasing the accuracy of replies with automation while being scalable enough to maintain maximum adaptability into future upgrades was of paramount importance.

The system maximizes well-understood protocols such as IMAP for retrieval of e-mail and SMTP with encryption for secure and reliable transmission ofmessages. The use of NLP allows for the natural understanding and analysis of e-mail content, enabling intelligent and contextually correct responses. The addition of a sophisticated logging mechanism further increases the robustness of the system, making it more translucent, as it facilitates auditing, error analysis, and performance monitoring.

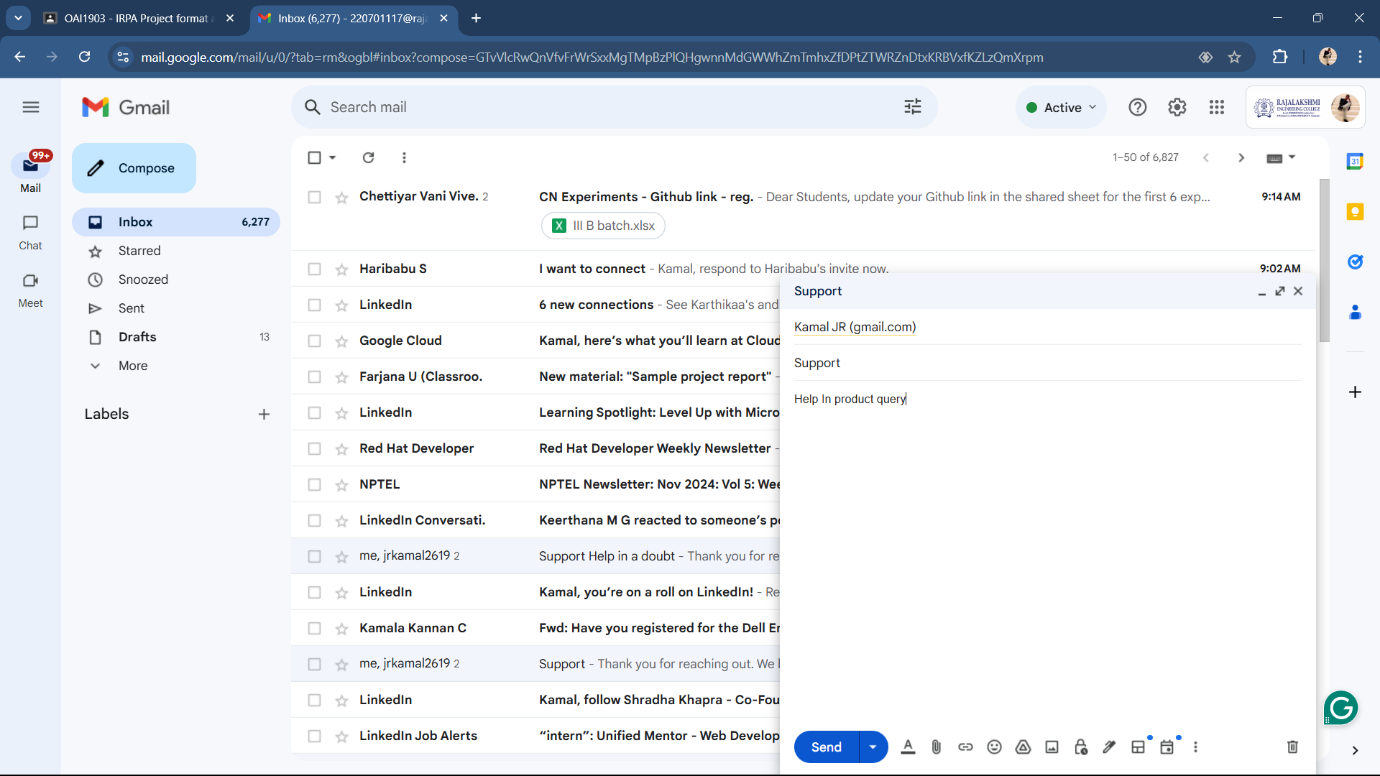
The project has successfully proven that automation in email handling leads to the proper usage of human resources, as employees can then focus their attention on more substantial tasks instead of just regular email management. It is further easy to integrate with the existing communication infrastructure, which places it as an appropriate solution for a broad array of organizational needs.

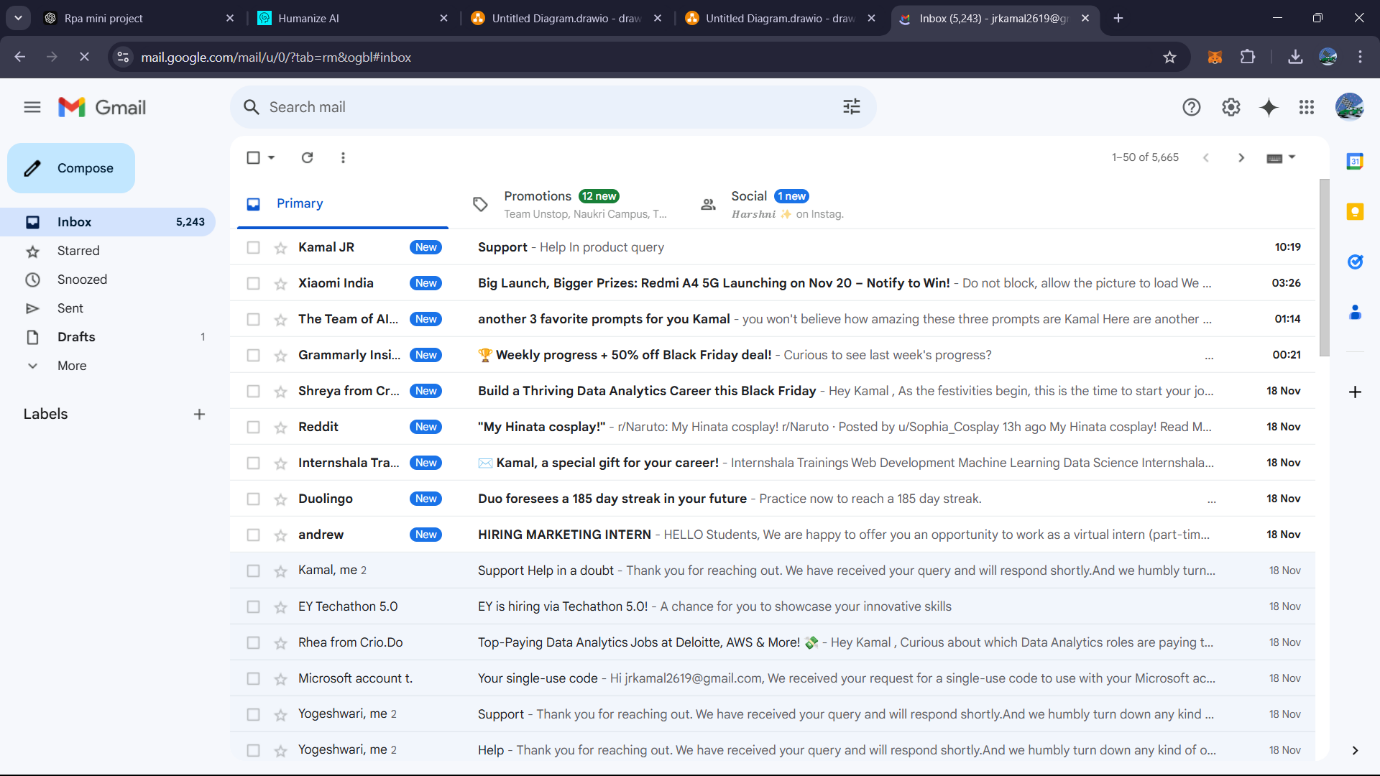
The project thus gives a solid foundation for further developments in automated management of emails, such as adaptive learning for more complex queries and multi-lingual support, which will broaden the scope of applications the system can be put to. The success story of this project clearly points out that automation in everyday business operations is immensely valuable. Such communication systems can become even smarter, efficient, and reliable in the future.

**6. References:**

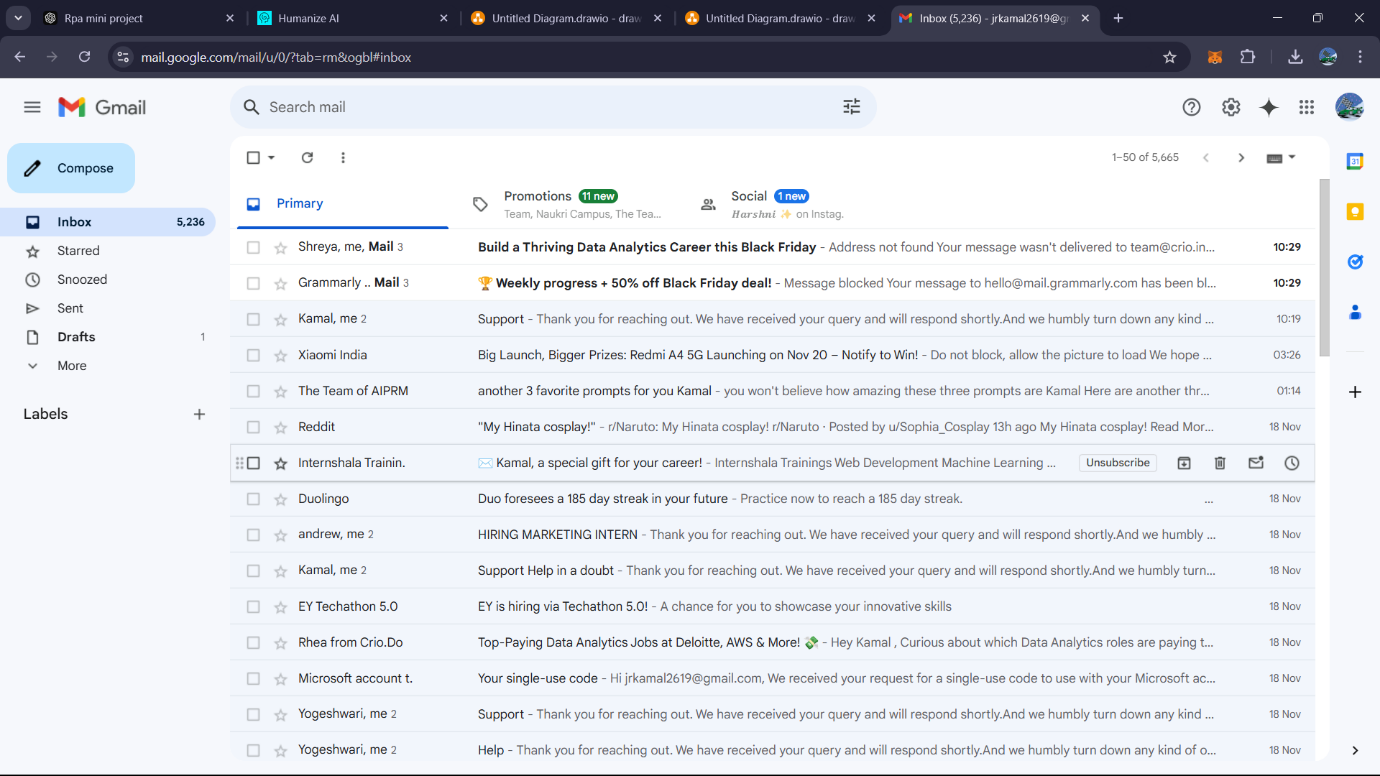
1. Katz, J. & Lindell, Y. (2021). Introduction to Modern Cryptography. This book provides an in-depth understanding of security protocols, which could be useful for understanding secure email communication using IMAP and SMTP.
2. Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. A great resource for understanding text analysis and NLP techniques, essential for the content analysis module in your project.
3. Jurafsky, D., & Martin, J. H. (2020). Speech and Language Processing. This textbook is a comprehensive guide to NLP techniques, which are central to your email content analysis module.
4. Postel, J. B. (1982). RFC 821 - Simple Mail Transfer Protocol (SMTP). This RFC document outlines the standard for SMTP, useful for understanding the underlying principles of sending emails.
5. Crispin, M. (2003). RFC 3501 - Internet Message Access Protocol (IMAP) Version 4rev1. A technical specification detailing how IMAP works, providing foundational knowledge for the email retrieval aspect of your project.
6. Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. This book is useful if you are implementing NLP using Python, particularly with the NLTK library.
7. Hunt, A., & Thomas, D. (2000). The Pragmatic Programmer: From Journeyman to Master. While not directly about email automation, this book offers valuable insights into software development best practices and modular system design.
8. Excel VBA Programming for Dummies (2021 Edition) by Michael Alexander & Dick Kusleika. If you are logging interactions in Excel, this resource could be helpful for automating and managing your data efficiently.
9. Yin, R. K. (2018). Case Study Research and Applications: Design and Methods. This resource can help you understand how to document and present the outcomes of your project systematically.
10. Scikit-learn Documentation (https://scikit-learn.org/stable/): If you are using machine learning libraries for NLP and data analysis, the official documentation is invaluable.
11. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. A foundational text for understanding the algorithms behind machine learning, which can be useful if you’re expanding your NLP capabilities.

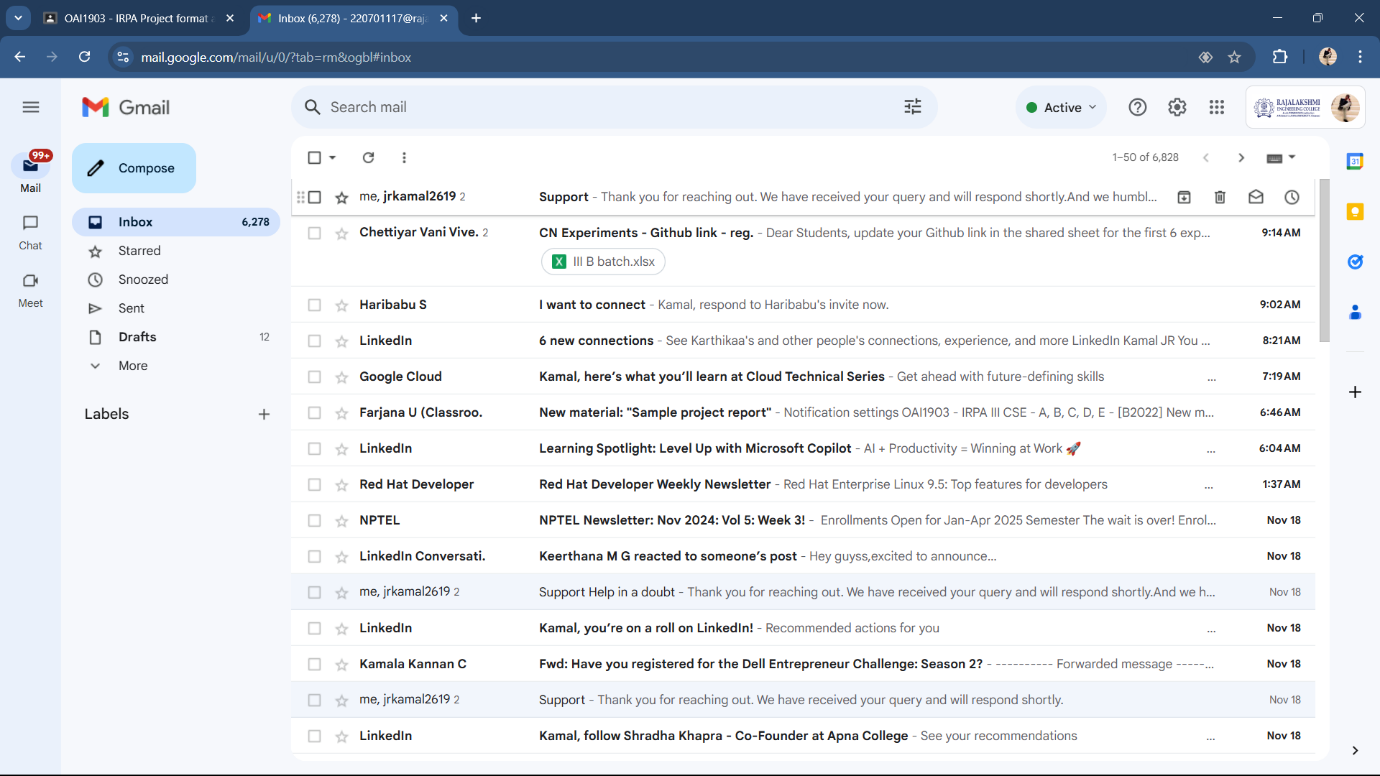
**Output**

**Sender Mail**

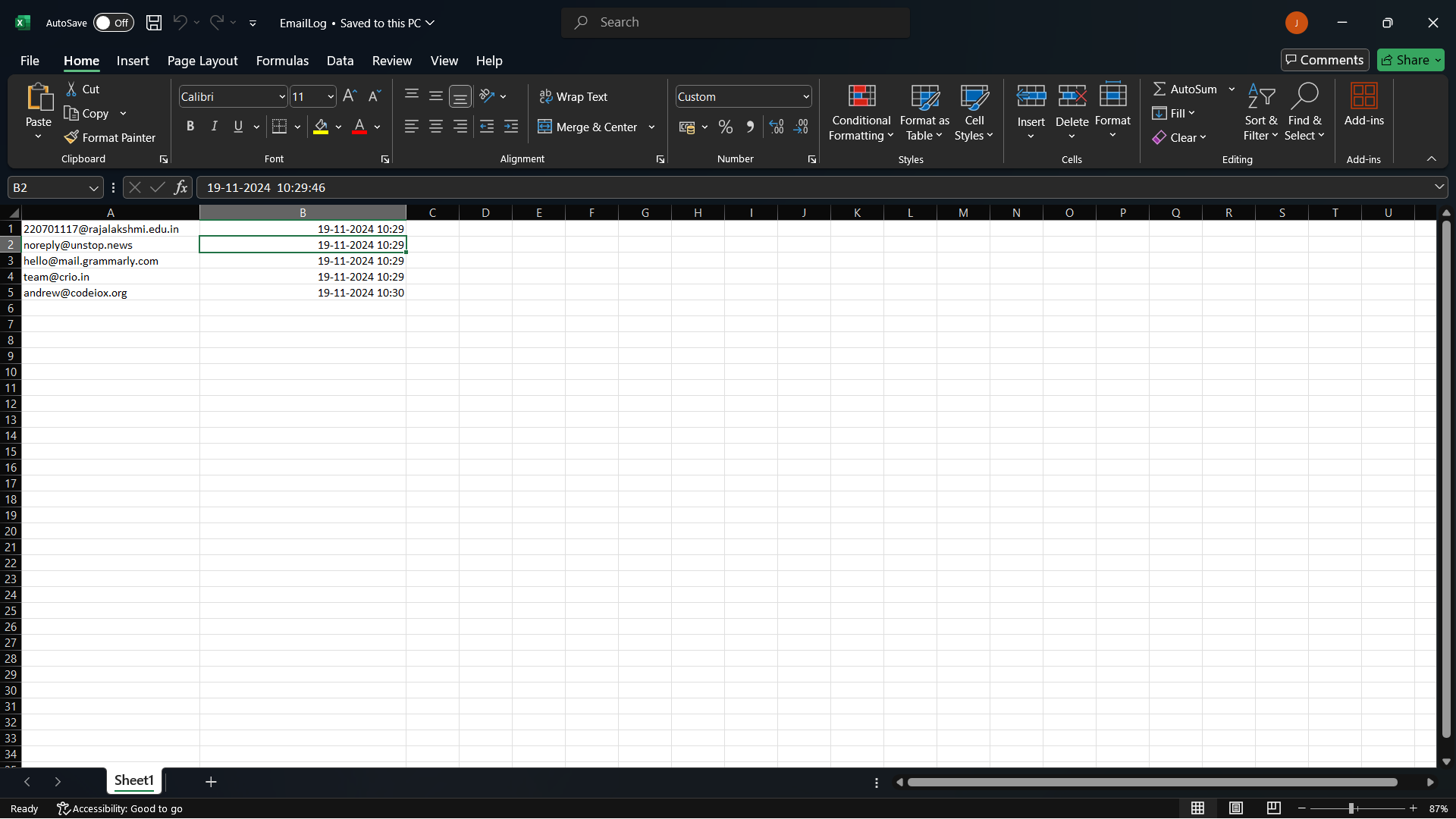
****

**Receiver Mail (Unread)**

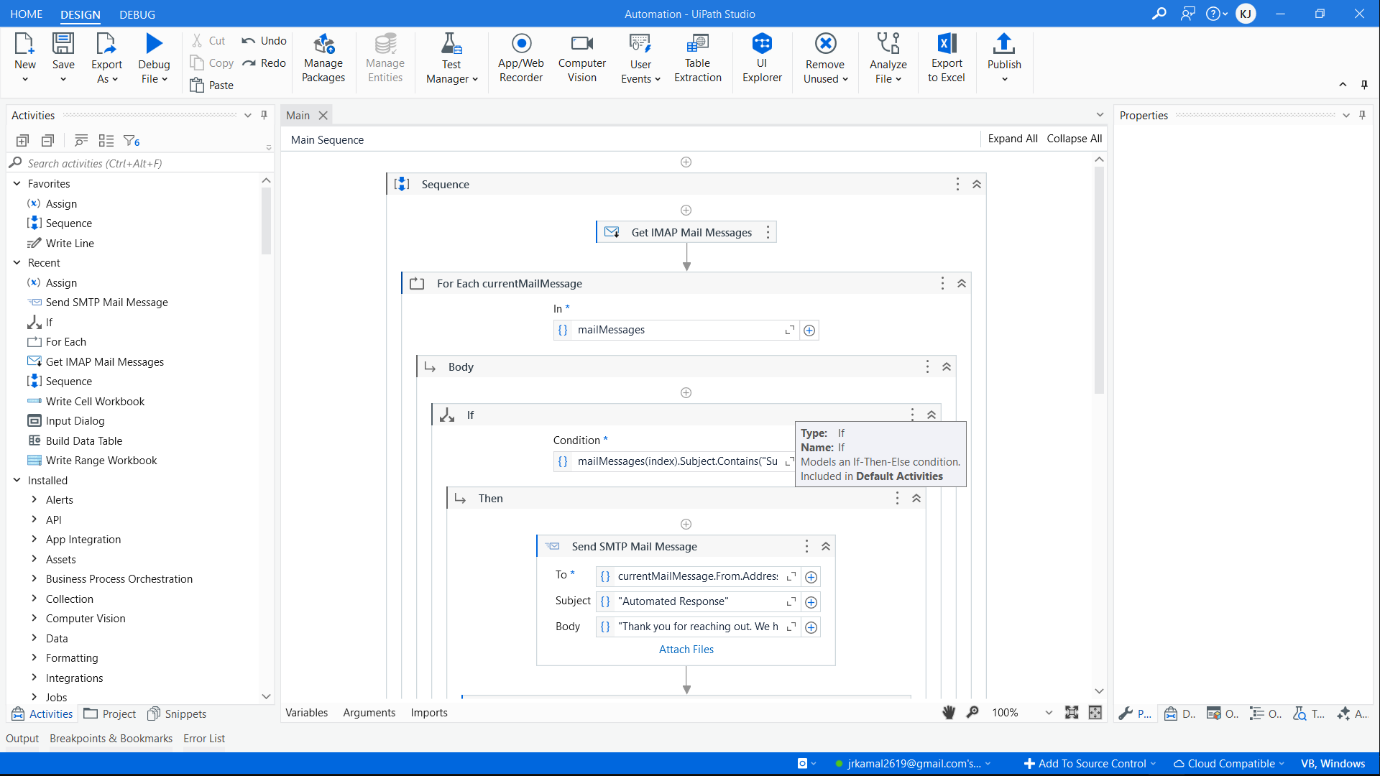
****

**  
Mail Read**

**Mail Acknowledged**

****

**Mail Sent Excel Logging**

**Appendix   
Process Workflow**

