Jaypee Institute of Information Technology University, Noida

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING AND INFORMATION TECHNOLOGY

Major Project Title: MailzyAI

An End-to-End assistance for your Gmail

Enrolment No.	Name of Student
21803006	Tanya Vashistha
21803013	Vivek Shaurya
21803028	Sneha

Supervisor: Dr. Vikash

Submitted To: MS. ANUPAMA PADHA
DR. PARMEET KAUR

Course Name: Major Project 1

Program: Intg. Mtech CSE

2024 - 2025

Table of Contents

1. Introduction

1.1 Documents Conventions

This document outlines the conventions followed in the MailzyAi project, which is designed to deploy an AI-powered email assistant using a Large Language Model (LLM). The email assistant will categorize emails into four primary categories: personal, recruitment, spam, and offers. The assistant will also generate replies using generative AI (GenAI) to respond to emails in a contextually relevant manner. Additionally, the assistant will handle tasks such as sending emails to specified recipients and managing the Gmail inbox efficiently. A dashboard will be provided to track the assistant's usage and performance, offering an end-to-end solution for users to manage their mailbox effortlessly. Conventions followed in this project include consistent use of the term "LLM" when referring to the Large Language Model used for email classification and response generation. The system will employ standardized terms such as "categorize," "reply," "send," and "track" when describing the assistant's core functionalities. User interface elements like the "dashboard" will always be referenced in lowercase, while key project terms like "MailzyAi" will be capitalized consistently throughout the document. All code references will follow Python syntax and coding style guidelines, with adherence to best practices for API integrations and secure handling of email data. Any technical descriptions of the project’s architecture, including LLM interactions with the Gmail API, will be clearly documented using proper technical language. Furthermore, the document will maintain a logical structure, beginning with a project overview, followed by detailed sections on features, architecture, and user experience. Throughout, terminology will be precise to avoid ambiguity and maintain clarity for all stakeholders.

2. Purpose

2.1 Project Objective

The primary objective of the MailzyAi project is to develop and deploy a comprehensive, AI-powered email assistant that seamlessly integrates with Gmail to streamline inbox management and email communication. Utilizing a Large Language Model (LLM), MailzyAi will automatically categorize incoming emails into four essential categories: Personal, Recruitment, Spam, and Offers, allowing users to easily manage and prioritize their correspondence. Additionally, the assistant will leverage Generative AI to craft intelligent and contextually appropriate replies, eliminating the need for users to draft responses manually. MailzyAi will also feature the capability to automatically send emails, ensuring prompt communication without the constant oversight of the user.

A key component of the project is the integration of a dashboard that tracks the assistant's usage, providing detailed analytics on categorized emails, user interactions, and overall inbox activity. This will offer valuable insights into the user's email habits and the assistant's performance. By providing an end-to-end solution for email management, MailzyAi aims to significantly reduce the time and effort spent on daily email tasks, offering users a highly efficient, intuitive, and reliable tool for handling their email communications. This project focuses on enhancing productivity, improving communication efficiency, and ensuring a user-friendly experience.

2.2 Project Scope

The scope of the MailzyAi project encompasses the development of an AI-driven email assistant that integrates seamlessly with Gmail to manage inboxes efficiently. It includes categorizing emails into Personal, Recruitment, Spam, and Offers, generating AI-based replies, and automatically sending emails. Additionally, the project will feature a dashboard for tracking usage metrics and email activity, providing users with insights into inbox management. MailzyAi is designed to offer a full end-to-end solution for email communication, minimizing manual effort, improving productivity, and ensuring seamless handling of everyday email tasks. The project focuses on enhancing user experience and efficiency.

2.3. Literature Survey

1. An empirical study on email classification using supervised machine learning in real environments

   This paper addresses the challenge of spam email detection, a major issue for the Internet, and explores the application of supervised machine learning (SML) techniques in classifying legitimate and spam emails. While SML algorithms, such as decision trees and SVMs, have been widely used in research, many studies focus on datasets rather than real-world performance. To fill this gap, the authors conduct an empirical study across three different environments involving over 1,000 users. The results show that SML classifiers are effective and accepted by users in real email classification scenarios. The paper also highlights potential future research directions and insights to improve email classification using machine learning techniques.

2. A collaborative assistant for email

   This paper introduces a new paradigm for human-computer interaction, where software agents communicate and collaborate with users to accomplish complex tasks, complementing traditional graphical interfaces. The authors present a prototype agent designed to assist users with email management, based on research involving human assistants and Wizard-of-Oz experiments. The prototype leverages Collagen, a collaborative discourse modeling software developed by Lotus and Mitsubishi Electric, along with speech understanding technology from IBM Research. Users can interact with the agent through a combination of spoken language and graphical interface actions, which the agent observes and responds to. The agent maintains a model of the user's goals and activities, enabling it to act proactively and assist the user more effectively. The high-level goal modeling allows for natural, conversational speech interactions, enhancing the overall user experience.

3. Overall description

3.1 Product Features

1. Email Categorization:

- Emails are automatically categorised into four categories: Personal, recruitment, spam, and offers.

2. AI-Powered Reply Generation:

- Uses Generative AI (GenAI) to craft intelligent, contextually relevant responses to emails.

3. Automated Email Sending:

- Sends emails directly to recipients without manual intervention after generating appropriate responses.

4. Inbox Management:

- Organizes and prioritizes emails, allowing users to manage their Gmail inbox efficiently.

5. Dashboard with Usage Analytics:

- Provides a dashboard for tracking metrics such as email categorization, assistant activity, and user interactions.

6. End-to-End Email Solution:

- Handles the complete email workflow—from categorization to reply generation, sending, and inbox management.

7. Seamless Gmail Integration:

- Fully compatible with Gmail, ensuring smooth integration and user experience.

8. Time and Effort Optimization:

- Reduces the need for manual inbox handling and communication, boosting productivity.

3.2 User Classes and Characteristics for MailzyAi:

1. User Classes:

1. Individual Users:
   - Professionals or individuals managing personal and professional email accounts.
   - Users seeking to streamline inbox management and reduce time spent on emails.

2. Small Business Owners:
   - Entrepreneurs or small business owners handling customer inquiries, recruitment, and promotional emails.
   - Users aiming to automate routine email tasks to focus on core business activities.

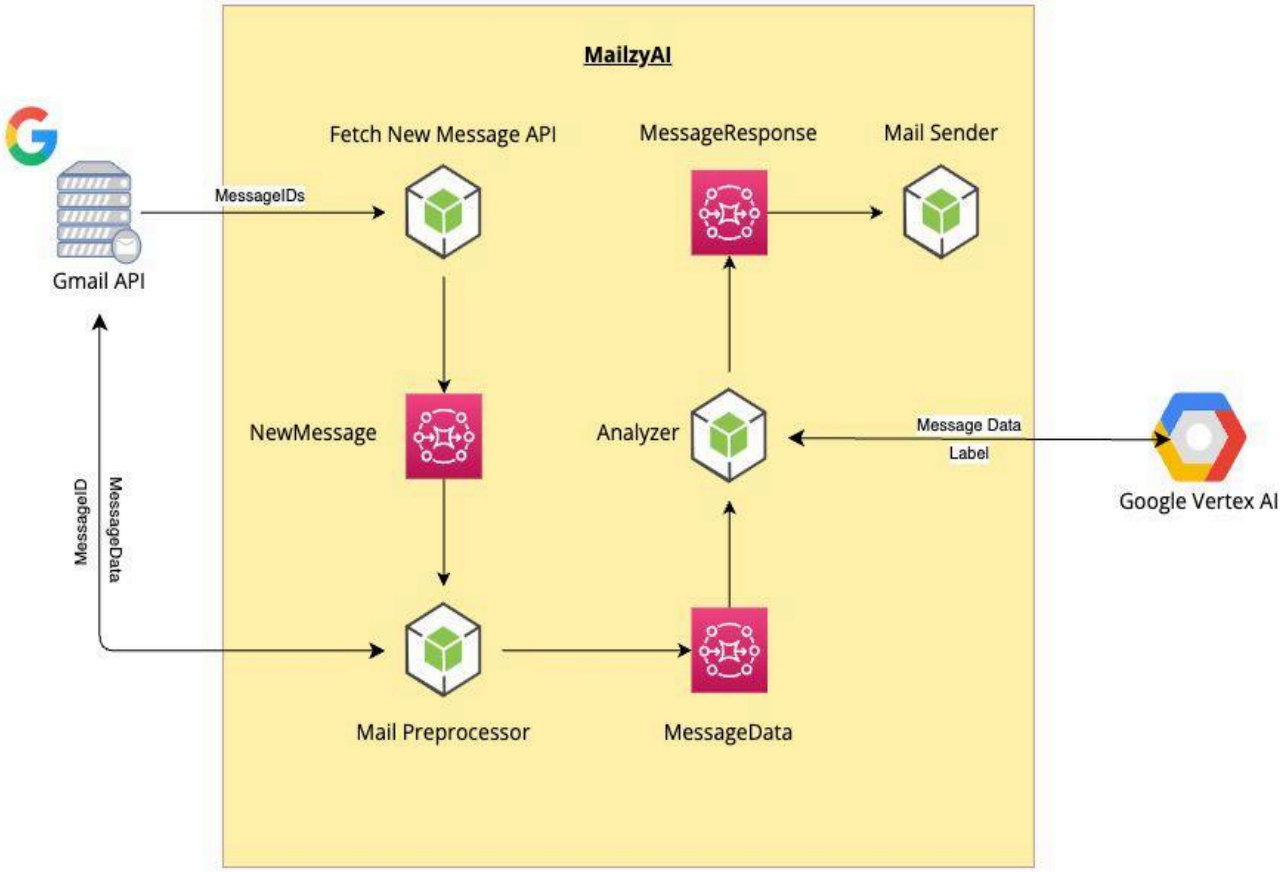
3. HR and Recruitment Teams:
   - Recruitment professionals receive large volumes of job applications and inquiries.
   - Users looking for efficient email categorization and automated replies for candidates.

4. Sales and Marketing Teams:
   - Teams handling offers, promotions, and customer outreach via email.
   - Users needing quick categorization and automated follow-up emails to potential clients or customers.

5. Frequent Email Communicators:
   - Users who regularly send and receive a high volume of emails and want to minimize manual effort.
   - Those looking for a tool to handle routine communication and prioritize important messages.

6. Tech-Savvy Users:
   - Users comfortable with AI-powered tools who are looking to adopt productivity-enhancing technology for inbox management.

WORKFLOW DIAGRAM:



3.3 Design and Implementation Constraints for MailzyAi:

1. Gmail Integration Limits:

- The system must adhere to Gmail's API usage policies and rate limits, ensuring compliance with their terms of service and avoiding throttling.

2. Data Privacy and Security:

- MailzyAi must ensure strict adherence to data privacy regulations when accessing and processing users' emails, protecting sensitive information and ensuring secure data handling.

3. Real-time Performance:

- The email assistant must process incoming emails and generate replies in real-time or near real-time to provide a seamless user experience, without significant delays.

4. Scalability:

- The system needs to handle high volumes of emails efficiently for users with large inboxes, ensuring smooth performance across varying workloads.

5. AI Model Limitations:

- The LLM may occasionally generate irrelevant or incorrect responses, so additional checks or user review mechanisms may need to be implemented for critical communications.

6. Resource Consumption:

- The use of LLMs can be resource-intensive in terms of computing power and storage. The implementation must balance performance with resource usage to maintain cost-efficiency.

7. Cross-platform Compatibility:

- The assistant should function across different devices (mobile, desktop) and web platforms where users access their Gmail accounts.

8. User Customization Constraints:

- While providing personalized responses, the system may have limited ability to fully adapt to all individual user preferences without extensive training or feedback loops.

9. Dashboard Integration:

- The dashboard must provide real-time analytics and track email activity without causing performance lags or requiring significant backend resources for data visualization.

10. API Rate Limits for External Tools:

- Any third-party services (such as analytics tools) used in the dashboard or email tracking must function within the API limits and constraints imposed by those platforms.

3.4 Assumptions and Dependencies for MailzyAi:

Assumptions:

1. Gmail API Access:

- It is assumed that Gmail's API will be accessible and functional throughout the project's lifecycle, allowing integration for email management and communication.

2. User Permissions:

- Users will grant necessary permissions for MailzyAi to access their Gmail accounts and manage emails, including categorization, sending, and reply generation.

3. LLM Performance:

- The LLM used will perform as expected in generating contextually relevant responses and categorizing emails accurately, given proper training and fine-tuning.

4. Stable Internet Connection:

- Users will have a stable internet connection for real-time email processing and interactions with MailzyAi.

5. Data Accuracy:

- It is assumed that the data used for training and fine-tuning the LLM is accurate and representative of the email content it will handle.

Dependencies:

1. Gmail API and OAuth:
   - The project depends on Gmail's API for accessing and managing emails, and OAuth for user authentication and authorization.

2. LLM Service Provider:
   - Reliance on the LLM service provider (e.g., OpenAI) for model access, performance, and updates, including API rate limits and costs.

3. Third-Party Analytics Tools:
   - Integration with analytics tools for the dashboard depends on their API availability and functionality.

4. Compliance with Data Regulations:
   - The implementation must comply with relevant data protection regulations, which may affect data handling and user consent processes.

5. User Feedback:
   - Effective operation and continuous improvement of the system will depend on user feedback and interaction, requiring mechanisms for collecting and analyzing user input.

6. Platform Updates:
   - The project's functionality may be influenced by updates to Gmail or external APIs, which could affect integration or require adjustments to maintain compatibility.

7. System Resources:
   - Adequate computing resources and storage are needed for running the LLM and handling large volumes of email data efficiently.

4. External Interface Requirements for MailzyAi:

4.1 User Interfaces:

1. Dashboard Interface:

- Overview Dashboard:
  - Provides a high-level view of email activity, including categorized emails, response statistics, and overall usage metrics.
- Category Breakdown:
  - Displays a detailed breakdown of emails sorted into Personal, Recruitment, Spam, and Offers categories.
- Performance Metrics:
  - Offers insights into response times, email handling efficiency, and user interaction data.
- Settings & Customization:
  - Allows users to adjust preferences, manage notification settings, and customize the assistant's behavior.

2. Email Inbox Interface:

- Categorized Email View:
  - Shows emails organized by their assigned categories (Personal, Recruitment, Spam, Offers) with easy navigation between categories.
- Email Details:
  - Provides a detailed view of individual emails, including sender information, subject, and content, with options to view or reply.
- Search & Filter:
  - Includes search functionality to quickly locate emails and filtering options to refine results based on criteria such as date or category.

3. Email Reply Interface:

- Response Generation:
  - Displays suggested replies generated by the LLM, allowing users to review and edit before sending.
- Reply Management:
  - Provides options to send, discard, or save replies as drafts.

- Quick Actions:
  - Features buttons for common actions such as replying, forwarding, or archiving emails.

4. Email Sending Interface:
- Compose Email:
  - Allows users to compose new emails with fields for recipient addresses, subject lines, and message content.
- Attachment Handling:
  - Includes functionality for attaching files or documents to outgoing emails.
- Send Confirmation:
  - Provides a confirmation screen or notification once an email has been successfully sent.

5. Settings Interface:
- Account Management:
  - Allows users to manage account settings, including email account connections and authentication.
- Notification Preferences:
  - Provides options to configure notification settings for different email activities and assistant updates.
- AI Preferences:
  - Lets users customize the behavior and tone of the AI-generated responses based on their preferences.

6. Help and Support Interface:
- FAQs and Guides:
  - Offers access to frequently asked questions, user guides, and troubleshooting tips.
- Contact Support:
  - Provides a way to contact customer support for assistance with issues or inquiries.

These interfaces aim to create a user-friendly experience, ensuring efficient email management, easy navigation, and effective communication through MailzyAi.

4.2 Hardware Interfaces:

- Internet Connection:
  - A stable internet connection is crucial for accessing MailzyAi, ensuring seamless functionality as the platform operates as a web-based application.
- Accessibility Tools:
  - For users with impaired vision, compatibility with magnifying software is ensured to enhance text visibility on the website.

4.3 Software Interfaces:

- Integrated Development Environment (IDE):

- Development was carried out using Visual Studio Code (VS Code) with languages such as HTML, CSS, JavaScript, ReactJS, and TailwindCSS.

- Database System:

- MongoDB was employed as the backend database system, managing user login information, product details, quantities, prices, and other essential data.

4.4 Communication Interfaces:

- Communication Standards:

- Communication functions, including web browser interactions, electronic forms, and feedback submissions, adhere to industry standards such as FTP or HTTP.

- Security Measures:

- Java scripts handle communication security and encryption issues, ensuring the protection of user data and preventing unauthorized access.

5. Other Non-Functional Requirements for MailzyAi:

5.1 Database:

- MongoDB

5.2 Tools:

- VS Code, Github, Llama 3.1, RabbitMQ

5.3 Languages:

- HTML, CSS, JavaScript, React JS, Node JS, Python(Pytorch)

5.4 Security:

- Data security will be provided

5.5 Usability:

- Easy to use

5.6 Maintenance:

- Maintenance can be used easily

1.2 References

1. https://colorwhistle.com/top-e-learning-web-apps/ [Accessed: Aud-2024].
2. https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document [Accessed: Aug-2024].
3.https://en.wikipedia.org/wiki/Use_case#:~:text=Use%20cases%20are%20a%20technique,that%20contributes%20to%20its%20goals. [Accessed: Sep-2024].
4. https://rpaframework.org/contributing/conventions.html [Accessed: Sep-2024].
5. https://www.rabbitmq.com/docs [Accessed: Sep-2024].
6. https://ollama.com/library/llama3.1 [Accessed: Sep-2024].
7. https://github.com/huggingface/peft [Accessed: Sep-2024]