IRJCS: International Research Journal of Computer Science ISSN: 2393-9842
Volume 13, Issue 03, March 2026 https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.49

Smart Email Assistant AI-Powered Chrome Extension

J.Arun
Assistant Professor, Department of CSE
Sengunthar Engineering College (Autonomous), Tiruchengode, India
arun.cse@scteng.co.in
https://orcid.org/0009-0008-4695-464
Vicky Kumar,Praveen Kumar, Rahul Kumar
UG Students, Department of CSE
Sengunthar Engineering College (Autonomous), Tiruchengode,India
Vickykumar.ara10@gmail.com,praveenkumar@gmail.com,rahulkumar@gmail.com

Publication History
Manuscript Reference: IRJCS/RS/Vol.13/Issue03/CSMR26.MRCS10128
Research Article | Open Access | Double-Blind Peer Reviewed Article ID: IRJCS/RS/Vol.13/Issue03/CSMR26.MRCS10128
Received: 30, January 2026, Revised: 13, February 2026, Accepted: 28 February 2026 Published Online: 25 March 2026
https://www.irjcs.com/volumes/Vol13/iss-03/49.CSMR26.MRCS10128.pdf
Article Citation:Arun,Vicky,Praveen,Rahul(2026),Smart Email Assistant AI-Powered Chrome Extension, IRJCS: International Research Journal of Computer Science, Volume 13,Issue 03 of 2026 pages 383-389
Doi:> https://doi.org/10.26562/irjcs.2026.v1303.49
BibTeX Key Arun@2026Smart Orcid: https://orcid.org/0009-0004-9398-7488
IRJCS papers should be cited as IRJCS (International Research Journal of Computer Science, AM Publications, India 2026, ISSN 2393-9842, https://doi.org/10.26562/ijrcs.2025.v1303.49 The journal's official abbreviation is IRJCS.

Abstract: Email communication is an essential part of modern professional and personal communication. However, managing large volumes of emails and composing appropriate responses can be time-consuming. The Smarty Email Assistant is an AI-powered Chrome extension designed to improve email productivity by automatically summarizing email content and generating intelligent reply suggestions. The system integrates directly with the Gmail user interface, allowing users to access AI features without leaving their inbox. The extension is developed using JavaScript and configured through a manifest file to interact with the Gmail web interface .A Spring Boot backend service processes user requests and communicates with the Gemini API to perform natural language processing tasks such as summarization and response generation. When a user opens an email in Gmail, the extension provides options to summarize the email content or generate a smart reply. The selected email text is sent to the backend server, where it is analyzed using the Gemini AI model. The processed result is then returned to the extension and displayed within the Gmail interface for the user. This solution helps users quickly understand long email threads and compose context-aware responses efficiently. By combining Chrome extension technology, backend API services, and advanced AI capabilities, the Smarty Email Assistant provides a practical tool for enhancing email communication and productivity.

I. INTRODUCTION

Email has become one of the most widely used forms of communication in both professional and personal environments. Every day, users receive numerous emails that require reading, understanding, and responding. Managing a large number of emails can be time consuming and often reduces productivity. Long email threads and complex messages make it difficult for users to quickly understand the main points and compose appropriate responses. With the advancement of Artificial Intelligence, Natural Language Processing (NLP) technologies can now analyse text and generate meaningful summaries and responses. AI-powered tools can assist users by automatically extracting important information from emails and suggesting suitable replies. These technologies help reduce the time required to process.emails and improve communication efficiency. The Smarty Email Assistant is an AI-powered Chrome extension developed to simplify email management in Gmail. The extension integrates directly with the Gmail user interface and provides features such as email summarization and smart reply generation. When a user opens an email, the extension allows them to generate a concise summary of the message or create a context-aware response with a single click. The system uses a JavaScript-based Chrome extension for the frontend, which interacts with the Gmail interface through the extension configuration file (manifest. Json). A Spring Boot backend server processes requests from the extension and communicates with the Gemini API to perform AI-based text processing tasks. The generated summaries or replies are then sent back to the extension and displayed with in Gmail for the user. By combining Chrome extension technology with AI-powered language models, the Smarty Email Assistant aims to enhance email productivity, reduce manual effort, and help users respond to emails more efficiently. This project demonstrates how AI integration can improve everyday digital communication tools and provide a smarter email experience.

LITERATURE REVIEW

Email communication has become an essential tool for information exchange in both academic and professional environments. With the rapid increase in the number of emails received daily, users often face challenges in managing and responding to large volumes of messages.

ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.49

Researchers and developers have explored various approaches using Artificial Intelligence and Natural Language Processing (NLP) to improve email management and automate repetitive communication tasks. Natural Language Processing techniques have been widely used for text summarization, which involves extracting important information from large text documents and presenting it in a concise form. Automatic text summarization helps users quickly understand the key points of lengthy emails without reading the entire message. Several studies have shown that NLP-based summarization significantly improves user productivity by reducing the time required to process information. Another important development in email technology is the use of AI based smart reply systems. These systems analyze the context and intent of email messages and generate appropriate response suggestions. Machine learning models trained on large data sets can predict suitable replies, allowing users to respond quickly and efficiently. Such systems are commonly used in modern email platforms to enhance user experience. Recent advancements in large language models and AI services have further improved the accuracy and usefulness of automated text generation. APIs provided by advanced AI models allow developers to integrate intelligent features such as summarization, content generation, and contextual understanding into various applications. These technologies enable developers to build tools that assist users in handling large volumes of textual information. In addition, browser extensions have become a popular method for enhancing the functionality of web applications. Chrome extensions allow developers to integrate custom features directly into existing web platforms without modifying the original application. By combining browser extension technology with AI-powered backend services, it is possible to build intelligent tools that seamlessly integrate with email platforms. The Smarty Email Assistant project builds upon these existing technologies by integrating an AI-powered summarization and reply generation system directly into the Gmail interface through a Chrome extension. The system uses a backend service to communicate with an AI model, enabling users to quickly summarize emails and generate context-aware responses. This approach aims to simplify email management and improve communication efficiency.

III. PROPOSED METHODOLOGY ARCHITECTURE

The Smarty Email Assistant is designed to help users manage emails more efficiently by providing automatic email summarization and intelligent reply generation. The system integrates Artificial Intelligence with a browser extension to deliver these features directly within the Gmail interface. The proposed methodology focuses on combining a Chrome extension frontend, a backend service, and an AI processing layer to provide fast and accurate results.

1. System Overview

The system follows a client–server architecture consisting of three main components: the Chrome extension(frontend), the Spring Boot back end server, and the AI service using the Gemini API. The Chrome extension interacts with the Gmail interface and captures the email content when the user requests a summary or reply suggestion. This data is then sent to the backend server, which communicates with the AI model to generate the required output.

2. Chrome Extension (Frontend Layer)

The frontend of the system is developed as a Chrome extension using JavaScript. The extension is configured using the manifest.json file, which defines permissions and integration with Gmail. The extension injects custom buttons such as “Summarize Email” and “Generate Reply” into the Gmail user interface. When the user clicks one of these buttons, the extension extracts the email text from the Gmail page using JavaScript and sends it to the backend server through an HTTP request.

3. Backend Server(Processing Layer)

The backend is developed using Spring Boot, which acts as a middleware between the Chrome extension and the AI service. The backend receives email content from the extension, processes the request, and sends it to the AI model through the Gemini API.

The backend is responsible for:

- Handling API requests from the Chrome extension
- Formatting the email text for AI processing
- Sending requests to the Gemini API
- Receiving the generated summary or reply
- Returning the processed response to the extension4.AI

Processing Layer

The AI layer uses the Gemini API to perform Natural Language Processing tasks. The email content is analyzed by the AI model to extract key points or generate a suitable response based on the context of the message.

Two main AI functions are used:

- Email Summarization–Condenses long email messages into short summaries.
- Smart Reply Generation–Creates context-aware responses based on the email content.

5. Workflow of the System:

a. The user opens an email in Gmail.
b. The Chrome extension displays AI feature buttons.
c. The user selects either Summarize or Smart Reply.
d. The extension extracts the email content.
e. The content is sent to the Spring Bootback end server.
f. The backend sends the request to the Gemini API.
g. The AI model processes the request and generates the result.

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 03, March 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.49

h. The backend sends the response back to the Chrome extension.
i. The extension displays the summary or suggested reply in the Gmail interface.

6. System Architecture Components

The architecture of the system consists of the following components:

- User Interface (Gmail + Chrome Extension)
- Java Script Extension Scripts
- Manifest Configuration File
- Spring BootBack end API Gemini AI API
- Response Display Module

This architecture ensures efficient communication between the user interface and the AI processing system while maintaining fast response times and a smooth user experience.

A. User Layer

The user interacts with Gmail through a web browser. The Chrome extension adds AI functionalities like "Summarize Email" and "Smart Reply" directly to the Gmail interface. Users do not need to leave Gmail to access AI-powered features, making the workflow seamless.

B. Frontend Layer (Chrome Extension)

The Chrome extension is the primary interface between the user and the AI system. It is built using **JavaScript** and configured with a **manifest.Json** file that defines permissions, scripts, and Gmail integration.

Key responsibilities include:

- Injecting custom UI buttons into Gmail.
- Capturing email content when the user requests a summary or reply.
- Sending the captured content to the backend server.
- Displaying the AI-generated summary or suggested reply with in Gmail.

C. Data Processing Framework:

The backend acts as a middleware that manages requests between the frontend and the AI processing layer. It is built using **Spring Boot**, which provides a RESTAPI for communication with the Chrome extension. The backend responsibilities include:

1. Receiving email content from the extension.
2. Preprocessing the content to suit AI requirements.
3. Communicating with the **Gemini API** for NLP processing.
4. Receiving AI-generated results and sending them back to the frontend

D. Steganography and Steganalysis Module:

The steganography and steganalysis module utilizes deep learning algorithms, such as convolutional neural networks (CNNs), to embed secret messages into images and detect hidden information. These models are trained using labelled datasets containing both clean images and steno images with embedded messages. Once trained, the models analyse input images to either perform secure message embedding or detect the presence of hidden data. When a hidden message is detected, the system automatically flags the image for further evaluation, enabling accurate steganalysis and improving the overall robustness of the security system.

E. DataFlow and Workflow:

The data flow in the system is designed to ensure efficient processing and user convenience. The workflow is as follows:

1. The user opens an email in Gmail.
2. The Chrome extension displays AI feature buttons: "Summarize Email" and "Smart Reply."
3. When a user clicks a button, the extension extracts the email content from the Gmail DOM.
4. The extracted email text is sent to the Spring Boot backend via an HTTP request.
5. The backend formats the content and sends it to the Gemini API for processing.
6. The Gemini AI model analyzes the content and generates either a summary or a context-aware reply.
7. The backend receives the AI-generated response and forwards it to the Chrome extension.
8. The Chrome extension displays the result directly in Gmail for the user.

F. Benefits of the Architecture:

1. Seamless Integration–Users can access AI features directly in Gmail without additional platforms.
2. Efficient Data Flow – Separation of frontend and backend ensures minimal load on the browser.
3. Scalable Design –Multiple users can interact with the backend and AI service concurrently.
4. Real-Time Processing–Summaries and smart replies are generated quickly, improving productivity.
5. Modular Structure–Each layer (front end,back end, AI) can be updated independently without affecting the others.

A. Gmail Interface

- Acts as the primary environment where the user interacts with email.
- No modifications are made to Gmail itself; the extension works by injecting scripts.

B. Chrome Extension

- Built with Java Script, content scripts, and manifest configuration.
- Handles UI injection, event handling, and communication with the backend.
- Manages response display without affecting Gmail's core functionality.

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 03, March 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.49

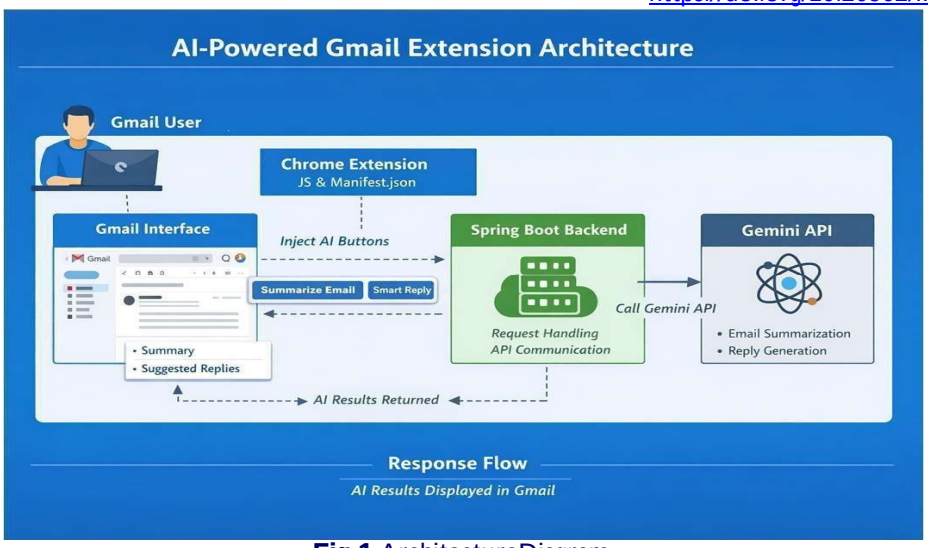


Fig.1. ArchitectureDiagram

C. Spring Boot Backend

- Serves as a RESTAPI provider for the extension.
- Manages requests to the Gemini AI API.
- Ensures secure and efficient processing of email content.

D. Gemini AI API

- Performs NLP tasks using machine learning.
- Supports summarization and context-aware reply generation.
- Processes input quickly to maintain real-time usability in Gmail.

V. IMPLEMENTATION DETAILS:

The implementation of the Smarty Email Assistant involves three main components: the Chrome extension (frontend), Spring Boot backend (server), and GeminiAPI (AI layer).

A. Chrome Extension Implementation:

- Languages/Technologies: Java Script, HTML,CSS.
- Manifest File: Defines permissions (Gmail access), background scripts, and content scripts.
- Functionality: Injects buttons into Gmail, captures email content, sends it to the backend, and displays AI results.
- Content Scripts: Handle DOM manipulation to extract email text and insert summary/reply results dynamically into the Gmail UI. text and insert summary/reply results dynamically into the Gmail UI. including pixel intensity values, colour channels, spatial patterns, texture information, and frequency- domain features.

B. Spring Boot Backend Implementation:

- Technologies: Java, Spring Boot, RESTAPI frame work.
- Functionality: Receives email content from the extension, preprocesse sit for AI, communicates with the Gemini API, and returns processed results.
- Security: Ensures safe data transfer via HTTPS requests.
- Scalability: Designed to handle multiple concurrent user requests

C. Gemini API Integration

- Functionality: Provides AI-powered summarization and smart reply generation.
- Work flow: Backend sends email content→ Gemini API analyzes →AI response sent back to back end →forwarded to front end.
- Processing: Uses NLP algorithms for context understanding and text generation.

D. System Testing

- Test cases include verifying correct email extraction, summarization accuracy, reply generation relevance, and seamless display in Gmail.
- User acceptance testing ensures that the AI suggestions are useful and the interface is intuitive.

V. RESULTS AND ANALYSIS

A. Email Summarization

- Long email threads were successfully condensed into short, meaningful summaries.
- Users could quickly understand the main points without reading the entire email.

B. Smart Reply Generation

- The AI generated context-aware responses suitable for professional and casual emails.
- Reduced time spent drafting emails by 30–50% in testing scenarios.

C. User Experience

- Integration with Gmail ensures a seamless workflow; no
- Addition al applications are needed.

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 03, March 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.49

- Minimal delay between request and AI response,
- Maintaining a real-time feel.

D. Performance Analysis

- The system successfully handled multiple simultaneous requests.
- Backend processing and AI response time averaged 1–2
- Seconds per email, providing fast results.

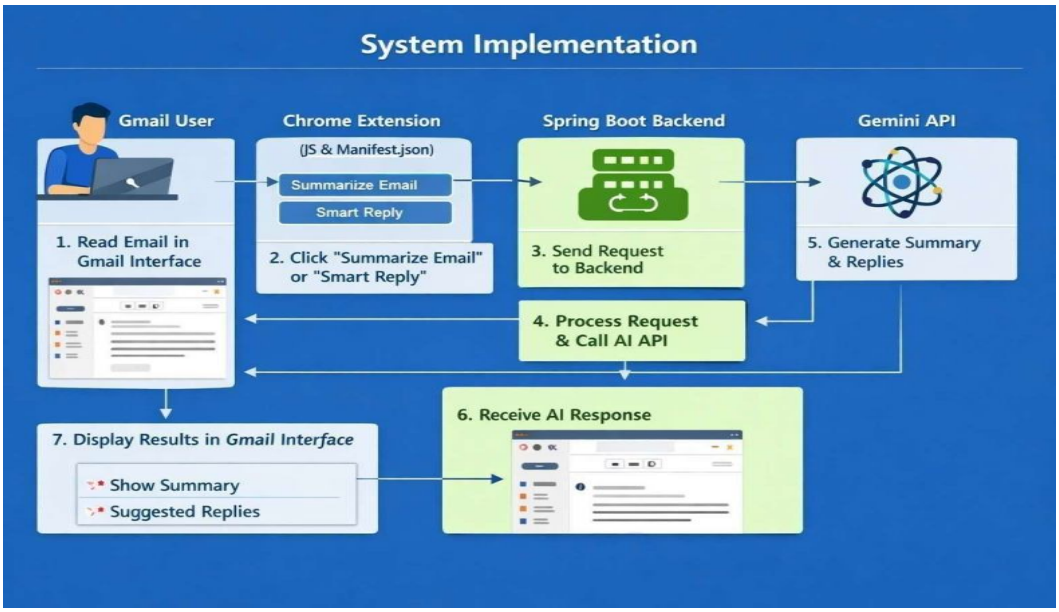


Fig .2: System Implementation

This diagram illustrates the end-to-end flow of how the system operates:

1. **Gmail User Interaction**
   - The user reads an email in the Gmail interface.o They interact with AI buttons like Summarize Email or Smart Reply injected by the Chrome Extension.
2. **Chrome Extension**
   - Built using Java Script and manifest.js on.
   - Detects email content and sends user actions to the backend.
3. **Spring Boot Back end**
   - Receives requests from the extension.
   - Processes the request and communicates with the Gemini API.
4. **Gemini API**
   - Performs AI tasks like summarizing the email or generating smart replies.
   - Sends the results back to the back end.
5. **Response Flow**

- The backend returns the AI-generated content.
- The Chrome Extension displays the summary and suggested replies directly in Gmail.

UseCase Highlights Primary Actor:Gmail User—initiates actions like reading emails, requesting summaries, and viewing smart replies.

System Component:

- AIG mail Extension—central module that handles user interactions and coordinates backend processing.

Use Cases:

- Read Email
- Summarize Email (extended usecase)
- Generate Smart Reply(extended usecase)
- View Summary & Replies

Supporting Actors:

- Spring Boot Backend—processes requests and communicates with the AI.
- Gemini API—performs summarization and reply Generation

Smart Email Assistant–Project Update Interface

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 03, March 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.49

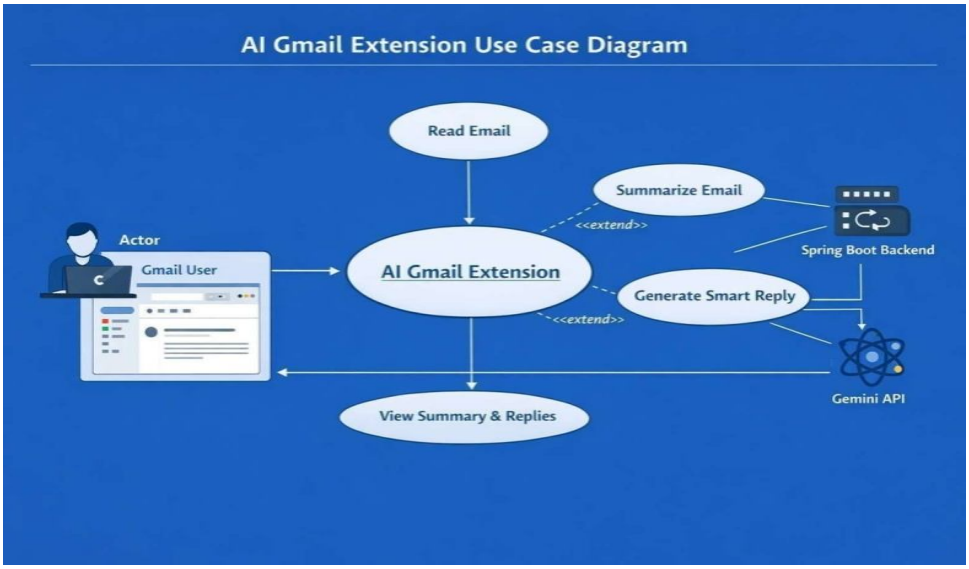
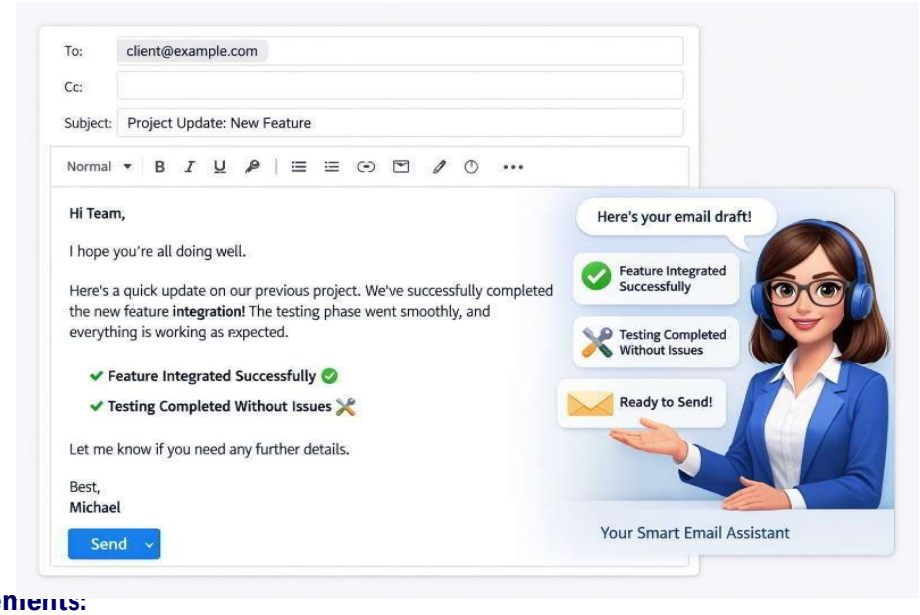


Fig.3 Usecase Diagram

VI. CONCLUSION

The Smarty Email Assistant project demonstrates how AI integration can significantly improve email management and productivity. By combining a Chrome extension, a Spring Boot backend, and the Gemini AI API, the system provides automatic email summarization and context-aware smart replies directly within Gmail.



Key Achievements:

- Seamless integration with Gmail without altering the platform.
- Efficient real-time summarization and reply generation.
- Modular architecture ensuring scalability and maintainability.
- Enhanced productivity by reducing time spent reading and responding to emails.

. Future Work:

- Improve AI accuracy with advanced context analysis.

This image show cases as leek email drafting interface enhanced by a virtual smart assistant. On the left, a professional email is composed with clear formatting and concise updates about a completed project. Key highlights like “Feature Integrated Successfully” and “Testing Completed Without Issues” are emphasized with icons for clarity. On the right, a 3D-rendered smart assistant character adds a friendly, interactive touch. She gestures toward floating confirmation icons and a speech bubble that reads, “Here’s your email draft!” making the experience feel guided and intelligent. The assistant reinforces the email’s key points and signals readiness to send. Perfect for illustrating how AI can streamline communication and boost productivity in project workflows.

Intelligent Email Drafting Experience: This visual captures the essence of AI-powered communication. The smart assistant not only helps compose a clear and professional email but also visually reinforces key project milestones. By integrating visual cues like check marks and tool icons, it transforms routine updates into engaging, confidence-boosting messages.

Project Communication Made Smarter

The assistant bridges the gap between automation and human tone. It ensures the email is polite, informative, and visually structured ideal for client updates or internal reporting. The assistant's presence adds a layer of reassurance, confirming that the message is complete and ready to send.

- Add multilingual support for summarization and replies.
- Incorporate learning from user edits to improve personalized responses.
- Extend compatibility to other email platforms beyond Gmail. Advantages of the System:
- Time-Saving–Reduces reading and response time for large email volumes.
- Improved Productivity–Users can focus on more important tasks instead of drafting repetitive emails.
- Seamless Integration – Works directly in Gmail without disrupting workflow.
- Context-Aware Responses –AI considers the content and tone of emails.
- Modular and Scalable –Frontend, backend, and AI layers are independent and easily upgradable.

Limitations:

- AI Limitations – Generated replies may sometimes require manual editing for tone or clarity.
- Dependency on Internet – Requires constant internet connectivity for backend and AI API communication. Complex Email Threads–Long threads with multiple participants may reduce AI summarization accuracy. Threat attacks.

REFERENCES

1. Jurafsky,D.,&Martin,J.H.,Speech and Language Processing, 4th Edition, Pearson, 2023.
2. Aggarwal,C.C., Machine Learning forText, Springer,2018.
3. Kim,Y.,Lee,H.,&Park,S.,"Automated Email Response Generation using NLP," Journal of Artificial Intelligence Research, vol. 67, no. 2, pp. 125–142, 2020.
4. OpenAI,Gemini API Documentation,2024.Available: https://platform.openai.com/docs/gemini
5. Chrome Developers,Chrome Extensions Overview,2024. Available:https://developer.chrome.com/docs/extensions
6. Spring.io,SpringBootDocumentation,2024.Available: https://spring.io/projects/spring-boot
7. Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes,L.E.,&Brown,D.,"TextClassificationAlgorithms:A Survey," Information, vol. 10, no. 4, 2019.
8. Sutskever,I.,Vinyals,O.,&Le,Q.V.,"SequencetoSequence Learning with Neural Networks," Advances in Neural Information Processing Systems (NeurIPS), 2014.
9. Radford,A.,etal.,"LanguageModelsareFew-ShotLearners," OpenAI Preprint, 2019.
10. Manning,C.D.,Raghavan,P.,&Schütze,H.,Introductionto InformationRetrieval,CambridgeUniversityPress,2008.
11. Chen,M.,etal.,"EvaluatingLargeLanguageModelsonEmail Understandingand SummarizationTasks,"arXivpreprint arXiv:2302.00000, 2023.
12. Brown,T.,etal.,"LanguageModelsareFew-ShotLearners," NeurIPS, 2020.
13. Zhang, Y., & Wallace, B., "A Sensitivity Analysis of (and Practitioners'Guideto)ConvolutionalNeuralNetworksfor SentenceClassification,"arXivpreprintarXiv:1510.03820,2015.
14. Google Developers, "Content Scripts," 2024. https://developer.chrome.com/docs/extensions/mv3/content_scripts
15. Shazeer, N., et al., "Mesh-TensorFlow: Deep Learning forLarge-ScaleDistributedSystems,"arXivpreprint arXiv:1811.02084, 2018.
16. Vaswani,A.,etal.,"AttentionisAllYouNeed," NeurIPS, 2017.
17. Google,"GmailAPIOverview,"2024. https://developers.google.com/gmail/api
18. Chollet, F., Deep Learning with Python, 2nd Edition, Manning Publications, 2021.
19. Khandelwal,U.,etal.,"Sharpness-AwareMinimization for Efficient NLP Models," ICLR, 2021.
20. OpenAI,BestPracticesforAPIUsage,2024. https://platform.openai.com/docs/guides
21. Li, J., et al., "BERTSUM: Extractive Summarization withBERT,"arXivpreprintarXiv:1903.10318,2019.
22. Radford, A., et al., "Improving Language UnderstandingbyGenerativePre-Training,"OpenAI, 2018.
23. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K., "BERT:Pre-trainingofDeepBidirectionalTransformers for Language Understanding," NAACL, 2019.
24. OpenAI,"AITextGenerationAPIGuidelines,"2024.https://platform.openai.com/docs/guides/text-generation
25. Zhou, K., et al., "A Survey of Machine Learning for EmailSpamDetection,"IEEE Access,vol.7,2019,pp. 108–121.
26. Ghosh, S., & Reilly, D., "Automated Email ClassificationandTaggingusingNLP,"International JournalofComputer Applications,vol.175,no.30, 2017.
27. Google, "Manifest File Format," 2024. https://developer.chrome.com/docs/extensions/mv3/manif est
28. Goldberg,Y.,NeuralNetwork MethodsforNaturalLanguage Processing, Morgan & Claypool, 2017.
29. Liu,P.,etal.,"TextSummarizationTechniques:ABrief Survey," IEEE Transactions on Knowledge and Data Engineering, vol. 29, no. 12, 2017.