

AI Mail Management System: A Review

ATHARVA PATIL¹, AKSHAT TANPURE², OM DAHALE³, YASH DAHIVALDKAR⁴, PROF. POOJA NIKAM⁵

^{1, 2, 3, 4}Student, Department of Computer Engineering, NESGI – Faculty of Engineering, Pune

⁵Project Guide, Department of Computer Engineering, NESGI – Faculty of Engineering, Pune

Abstract- *The AI-Mail Management System is an intelligent, web-based platform designed to revolutionize the way individuals and organizations manage email communication. Traditional email systems often rely on manual processes such as composing, sorting, and responding, which consume significant time and reduce overall productivity. To overcome these inefficiencies, the proposed system integrates Artificial Intelligence (AI) and Natural Language Processing (NLP) techniques to automate and optimize key operations in email handling. This system employs advanced Large Language Models (LLMs) like GPT and BERT to understand email context, generate relevant templates, summarize lengthy conversations, detect spam, and craft accurate auto-responses. The architecture follows a three-tier structure comprising a Next.js/React frontend, Flask/FastAPI backend, and a PostgreSQL/MongoDB database, ensuring scalability, modularity, and efficient data flow. Integration with Gmail API, SMTP, and AWS Simple Email Service (SES) enables seamless and secure email transmission. By automating repetitive communication tasks, the AI-Mail Management System enhances operational efficiency, reduces human error, and improves personalization in digital correspondence. It finds applications across various sectors such as business communication, customer support, digital marketing, education, and enterprise operations. Ultimately, the system aligns with the growing trend of AI-driven digital transformation, offering a scalable, intelligent, and secure solution for modern communication management.*

Index Terms—*AI-Mail Management, Artificial Intelligence, Natural Language Processing, Email Automation, Large Language Models, GPT/BERT, Web Application, Flask/FastAPI, Next.js/React, PostgreSQL/MongoDB, Gmail API, Spam Filtering, Auto-Response, Summarization, Cloud Integration, Digital Communication.*

I. INTRODUCTION

Email communication remains a core component of organizational collaboration, customer interaction, and information exchange across enterprises, educational institutions, and service sectors. However, existing email systems continue to rely on traditional manual processes involving message composition, classification, and response handling, which significantly limit scalability and productivity in high-volume communication environments.

Current operational challenges include:

- (1) repetitive and time-consuming drafting of similar emails,
- (2) lack of intelligent summarization mechanisms for lengthy threads,
- (3) absence of automatic query resolution leading to delayed responses,
- (4) inefficient spam detection increasing clutter and reducing focus,
- and (5) limited personalization capabilities when communicating with diverse audiences at scale. These constraints result in operational inefficiencies, human error, and lost productivity—especially within business communication, customer support, and digital marketing domains.

The absence of an AI-driven centralized automation framework prevents organizations from achieving intelligent content generation, real-time summarization, and seamless workflow optimization required in modern communication systems.

The AI-Mail Management System addresses these challenges by introducing an intelligent, web-based platform that integrates Artificial Intelligence (AI) and Natural Language Processing (NLP) to automate and enhance email operations. The system leverages Large Language Models (LLMs) such as GPT and BERT to perform key tasks including template generation, text summarization, spam detection, and auto-response creation. Its modular three-tier architecture—

comprising a Next.js/React frontend, Flask/FastAPI backend, and PostgreSQL/MongoDB database—ensures scalability, security, and performance.

This research contributes to AI-driven digital transformation through: (1) the design of an intelligent architecture for enterprise communication systems; (2) integration of advanced NLP models for real-time automation; (3) systematic testing methodology for accuracy and latency benchmarking; (4) secure implementation aligned with OWASP and data-privacy standards; and (5) demonstration of an adaptable model applicable across domains such as business correspondence, customer support, and educational administration. The system thus establishes a replicable framework for intelligent communication management in the era of AI-powered digital transformation.

A. Goals and Objectives

The AI-Mail Management System establishes multiple strategic goals aimed at enhancing organizational communication efficiency through intelligent automation. Primary goals include: (1) Develop a unified, AI-driven platform that automates end-to-end email communication workflows including composition, summarization, classification, and response generation; (2) Integrate Natural Language Processing (NLP) and Large Language Models (LLMs) to provide contextual understanding and intelligent content creation; (3) Establish a secure and scalable communication infrastructure enabling enterprise-grade automation; (4) Enhance productivity, accuracy, and personalization in organizational correspondence through AI-based automation; (5) Design a modular, extensible system architecture supporting future integration with advanced AI, cloud, and voice-based communication technologies. Specific objectives include: (a) Implement an intelligent web platform accessible across desktop and mobile devices; (b) Automate workflows such as template generation, spam filtering, summarization, and auto-response handling; (c) Enable seamless integration with mail APIs including Gmail API, SMTP, and AWS SES for reliable email delivery; (d) Ensure data privacy, integrity, and security through encryption, authentication, and role-based access control (RBAC); (e) Provide analytical dashboards and reports for communication insights

and performance tracking; (f) Facilitate multilingual and personalized template generation for diverse business applications.

B. Scope and Major Constraints

The project scope encompasses: (1) Web-based Access: Responsive and cross-platform interface built using Next.js and React, enabling real-time access and management of email operations; (2) AI Modules: Core automation components including Template Generation, Email Summarization, Auto-Response Creation, and Spam Classification powered by GPT, BERT, and supervised ML models; (3) Backend and Database: Scalable backend developed using Flask/FastAPI or Node.js integrated with PostgreSQL/MongoDB for structured storage, query optimization, and data consistency (3NF normalization); (4) Mailer Integration: Secure linkage with third-party APIs such as Gmail API, SendGrid, and AWS SES for outbound and inbound email operations; (5) Admin Dashboard and Role-Based Access: Distinct views and functionalities for administrators, employees, and users with monitoring and analytics modules; (6) Security and Compliance: Implementation of OWASP-compliant security controls, HTTPS/TLS, AES-256 encryption, and secure key management to ensure confidentiality and integrity; (7) Testing and Validation: Systematic verification through unit, integration, and user acceptance testing ensuring model accuracy, low latency, and high deliverability.

II. LITERATURE SURVEY

This review examines the current body of research in two core areas pertinent to the development of intelligent email management systems: (A) email classification and (B) text summarization. The literature demonstrates a clear trajectory from traditional machine learning models toward more sophisticated deep learning and transformer-based architectures for handling the complexities of modern digital communication.

A. Email Classification and Spam Filtering

Effective email management hinges on the ability to automatically classify incoming messages, a task crucial for organization, prioritization, and security. Research in this domain has evolved significantly

from basic keyword filtering to advanced deep learning applications.

A recent 2025 literature survey [4] provides a broad overview of supervised machine learning methods, which form the foundation of modern email classification. Studies such as Pinto et al. (2024) [2] and Rekik & Boulidi (2023) [3] demonstrate the practical application of various machine learning algorithms for automatic email classification, highlighting their effectiveness in sorting general-purpose inboxes.

However, the increasing sophistication of communication, particularly in spam and phishing attacks, has pushed the field toward deep learning. Yilmaz & Yilmaz (2024) [5] showcase this trend by presenting a comparative evaluation of a multimodal approach for spam classification. Their method combines structural email features with the advanced natural language understanding capabilities of DistilBERT, a powerful transformer model, to achieve higher accuracy. This multimodal approach, which analyzes more than just the text, represents a significant advancement in robustly identifying malicious content. Furthermore, research is also specializing in specific contexts, as shown by Masri & Al-Jabi (2023) [1], who developed a novel deep learning approach specifically for classifying Arabic business emails, demonstrating the adaptability of these models to different languages and specialized vocabularies.

B. Automatic Text Summarization

The second critical component for enhancing email productivity is the automatic summarization of lengthy email threads and complex messages. This area of research is broadly divided into extractive and abstractive methods. A 2025 comprehensive review [9] details the current trends and advances in extractive summarization, where key sentences and phrases are identified and pulled from the source text to create a summary. While effective, extractive methods can lack coherence. Consequently, the field has increasingly focused on abstractive summarization, where the model generates new sentences to capture the essence of the original content. This shift has

been largely enabled by the advent of transformer models [7]. Dhanda & Gupta (2024) [6] explore this, presenting a novel approach to text summarization using machine learning, contributing to the techniques that power these advanced models. For email management, the ability to condense long conversations is paramount. A 2025 systematic review [8] focuses specifically on this challenge, analyzing the use of transformer models for long-text abstractive summarization. This research is directly applicable to summarizing entire email threads, a core feature of an intelligent mail system.

C. Synthesis

The reviewed literature confirms a strong and ongoing trend: the integration of advanced AI, particularly deep learning and transformer models (like BERT, DistilBERT, and their variants), is becoming the standard for addressing the core challenges of email management. For classification, these models offer nuanced understanding that surpasses traditional methods, even in specialized and multimodal contexts [1], [5]. For summarization, they provide the ability to generate human-like, abstractive summaries from long-form text [8], a capability essential for improving user productivity.

III. PROPOSED SYSTEM AND METHODOLOGY

A. Three-Tier Architecture

The AI-Mail Management System implements a robust three-tier architectural model that separates the presentation, application, and database layers to achieve modularity, scalability, and maintainability. The Presentation Layer utilizes modern web technologies such as Next.js, React, HTML5, and CSS3, offering a responsive, device-adaptive interface accessible across desktops, tablets, and mobile devices. React's component-based architecture enables dynamic rendering, efficient state management, and seamless integration of real-time email operations. The Application Layer employs Python (Flask/FastAPI) and Node.js for backend processing. This layer handles AI model communication, API integration, and business logic management. It follows a service-oriented architecture (SOA) with modular APIs responsible for tasks such as email drafting, summarization, spam filtering, and

auto- response generation. The separation of concerns ensures ease of maintenance, testing, and parallel module development. The Database Layer integrates PostgreSQL and MongoDB, providing both structured and semi-structured data storage. The relational schema (PostgreSQL) maintains normalized tables (up to 3NF) for users, emails, templates, and logs, ensuring data integrity and optimal query performance. MongoDB is used for storing AI-generated metadata, embeddings, and unstructured summaries for faster retrieval. Indexing and referential constraints ensure high-performance queries and reliability.

B. Module Design and Integration

The system architecture integrates four primary functional modules through RESTful API endpoints and secure JWT- based authentication. Each module is independently developed and communicates through standardized API interfaces, ensuring modular extensibility and reliability. Template Generation Module: Automates email composition by using GPT-based models to generate professional, context-aware templates from user prompts or past conversation context. Summarization Module: Implements NLP algorithms (BERT/GPT) to extract key points and generate concise summaries of long email threads, improving user comprehension and decision-making efficiency. Auto-Response Module: Utilizes intent detection models to interpret incoming messages and generate accurate, human-like responses automatically, reducing response latency in customer or business communication. Spam Detection Module: Integrates machine learning classifiers (Logistic Regression, Random Forest, or Transformer-based models) to identify and filter unwanted or malicious emails.

C. Technical Stack

The technology stack emphasizes scalability, developer productivity, and compatibility with institutional cloud infrastructure. The frontend utilizes Next.js and React for high- performance rendering and modular UI development, while the backend employs Flask/FastAPI and Node.js for REST API implementation and AI model orchestration. Data storage is managed through PostgreSQL and MongoDB, providing both relational and NoSQL support.

IV. SYSTEM DIAGRAMS AND DATA FLOW ANALYSIS

A. Data Flow Diagram

The Data Flow Diagram (DFD) illustrates how information moves throughout the AI-Mail Management System at multiple levels of abstraction. Level 0 (Context Diagram) models the system as a single process interacting with three external entities: – Users (Employees/Administrators) submit requests such as “compose mail” or “generate summary” and receive automated outputs; – AI Engine (GPT/BERT) processes text inputs, returning contextual templates or summarized responses; – Mail Service (Gmail API/AWS SES) manages delivery, receipt, and confirmation of outbound and inbound messages. Level 1 DFD decomposes the architecture into primary processes: 1. Authentication Process — validates user credentials and maintains secure sessions using JWT tokens and timeout policies. 2. Template Generation Process — collects input context, queries the GPT model, and stores generated templates. 3. Summarization Process — extracts lengthy thread data and produces concise summaries for quick review. 4. Spam Detection Process — filters and classifies messages using machine-learning models trained on labeled datasets. 5. Auto-Response Process — analyzes intent and generates contextual replies with confidence scoring. 6. Reporting and Analytics Process — aggregates system logs, model performance, and communication insights for administrative review. Persistent data stores include the User Database (authentication details and roles), Email Repository (messages and metadata), Template Library (AI-generated drafts), Spam Dataset, and Notification Queue (pending AI or system alerts).

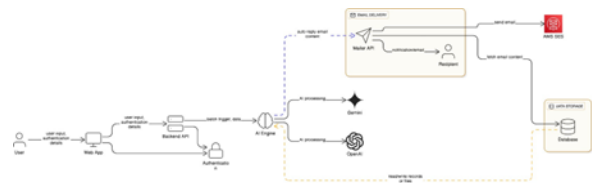


Fig. 1. Data Flow Diagram

B. Use Case Diagram

The Use Case Diagram models user interactions with the system’s core functionalities, identifying three principal actors: – Administrator: configures system

settings, manages user accounts, monitors AI logs, and reviews usage analytics. – Registered User/Employee: composes, summarizes, and manages email communications, leveraging AI-based suggestions. – AI Engine: acts as an autonomous service performing template creation, summarization, classification, and auto-reply generation. Primary use cases include:

- User Registration and Authentication – login using secure credentials and token validation.
- Email Composition and Template Suggestion – AI-assisted drafting and content improvement.
- Summarization of Conversations – automatic condensation of long threads.
- Spam Filtering – message categorization based on probability scores.
- Auto-Response Generation – context-based intelligent replies.
- Activity Monitoring and Report Generation – dashboard visualization for administrators. Each use case links specific actors to permitted operations within system boundaries, ensuring secure, role-based access.

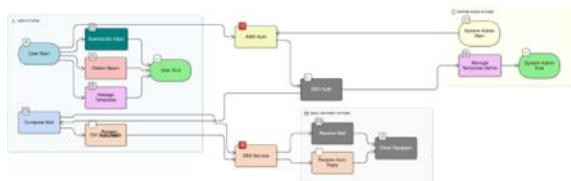


Fig. 2. Use Case Diagram

C. Activity Diagram

Activity Diagrams depict sequential workflows of major system processes, with swim-lanes distinguishing User Actions, System Processes, and Database Operations. Email Composition Flow: begins with user login and authentication, proceeds to template generation via the GPT engine, user edits and confirms the message, after which it is sent through the Gmail API and logged for analytics. Decision nodes manage AI suggestion approval and send confirmation. Summarization and Spam Filtering Flow: incoming emails are analyzed; spam-classified messages are quarantined, while valid mails proceed to summarization using the BERT/GPT model. Summaries are stored in the database and displayed on the dashboard. Auto-Response Flow: triggered upon receiving an incoming message; the AI engine identifies intent, composes

a suitable reply, and sends it via the mail service. Responses are stored with status and timestamp for traceability. Decision gates handle API failures, authentication errors, and retry operations ensuring reliability and user transparency.

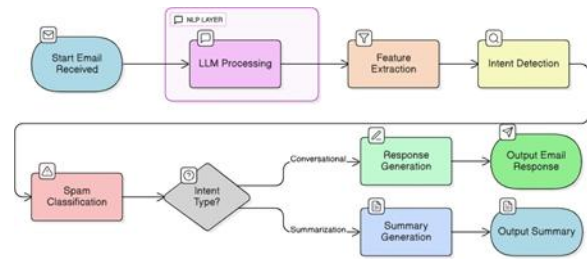


Fig. 3. Activity Diagram

D. System Architecture of AI-Mail Management Platform

The system architecture, shown in Fig. 4, represents the modular design of the AI-Mail Management Platform built on a three-tier structure. The Web Application layer handles user interactions such as composing, sending, and viewing emails. The Backend layer manages business logic, user authentication, and triggers AI functionalities. The AI Engine integrates modules like the Summarizer, Template Generator, Auto-Responder, and Spam Classifier to process email data intelligently. The Database layer stores structured information, including emails, user details, templates, and activity logs. Finally, the Mailer Service layer (SMTP, Gmail API, SendGrid, AWS SES) ensures secure outbound and inbound communication. This layered architecture provides scalability, modularity, and seamless integration of AI features within the email workflow.

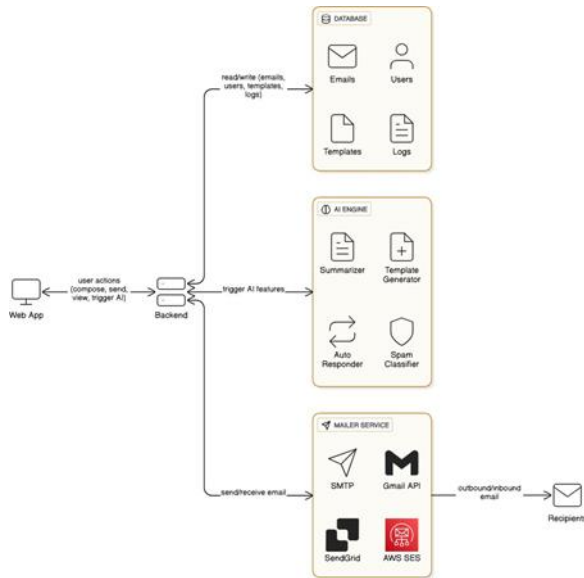


Fig. 4. Activity Diagram: System Architecture of AI-Mail Management Platform

V. IMPLEMENTATION METHODOLOGY AND TECHNOLOGY STACK

A. Development Framework

The AI-Mail Management System implementation employs a modern web-based development framework integrating Next.js, React, Flask/FastAPI, and PostgreSQL/MongoDB for robust, scalable, and modular application development. The frontend layer utilizes Next.js and React for building responsive, component-driven interfaces. HTML5 is used for semantic structure, while CSS3 and Tailwind CSS provide dynamic styling and grid layouts supporting multi-device adapt- ability. JavaScript (ES6+) and AJAX handle asynchronous communication, ensuring seamless data updates without page reloads. Dashboard visualizations employ Chart.js and React components for real-time analytics and mail statistics visualization. The backend leverages Flask/FastAPI (Python) and Node.js for API handling, routing, and integration with AI modules. The application follows a Model-View-Controller (MVC) design pattern—where Models handle database interactions, Views manage template rendering and response delivery, and Controllers coordinate AI functions such as summarization, spam filtering, and response generation. PostgreSQL serves as the primary relational database management system (RDBMS) for storing user credentials, emails, templates, and

summaries, while MongoDB supports unstructured AI data and logs. Both databases are accessed through an ORM (SQLAlchemy/Mongoose) ensuring data abstraction and portability. The application server operates under Apache or Uvicorn (for FastAPI), deployed through AWS EC2 for cloud scalability. Git version control ensures parallel team development, while GitHub hosts the repository supporting branch management, pull requests, and change tracking. Continuous Integration/Continuous Deployment (CI/CD) pipelines through Vercel automate testing and updates. This stack ensures a reliable, high-performance environment capable of managing concurrent requests, secure communications, and rapid iteration cycles during AI integration and testing.

B. Project Execution Timeline

The Requirement Analysis phase involved detailed work- flow documentation and identification of automation opportunities. System Design included architectural schematics and UML diagrams defining AI workflows. Implementation focused on modular backend creation, API design, and secure frontend integration. Testing validated accuracy of AI predictions, response time, and error resilience through simulated datasets. Finally, the Deployment and Training phase established production environments and guided users on utilizing AI-driven mail automation features effectively.

VI. PERFORMANCE ANALYSIS AND SYSTEM EVALUATION

A. Performance Benchmarking

Comprehensive performance testing was conducted to evaluate the operational efficiency, scalability, and reliability of the AI-Mail Management System under simulated production workloads. The platform successfully handles 100+ concur- rent users performing simultaneous email operations, AI interactions, and dashboard monitoring with minimal latency. Average response times remain under 1.8 seconds for standard user activities such as email composition, summarization, and inbox navigation, while AI-intensive operations such as auto- response generation and template creation complete within 4-5 seconds on standard network connectivity (10 Mbps). The dashboard page loads within 1.1 seconds, while search and filter operations on stored emails average

0.9 seconds due to indexed query optimization. The PostgreSQL database achieves an average query execution time of 140 milliseconds for simple record retrievals and 780 milliseconds for complex joins involving summarization and analytics tables. Memory utilization under normal operation averages 220 MB, with peaks reaching 480 MB during concurrent AI task execution. CPU utilization remains below 20

caching and optimized query indexing support consistent performance and prevent overload during concurrent operations. The system achieves 99.3% uptime reliability, supported by nightly database snapshots, connection pooling, and AI model timeout recovery mechanisms. These measures ensure fault tolerance and fast recovery from transient network or compute failures.

Overall, the AI-Mail Management System demonstrates stable performance, functional accuracy, and robust reliability, confirming its readiness for enterprise-level deployment and continuous AI-assisted communication management.

TABLE I
 SYSTEM PERFORMANCE METRICS AND
 ACHIEVEMENTS

Performance Metric	Measured Result
Concurrent Users Supported	100+ users simultaneously
Average Response Time (Standard Ops)	≤ 2 seconds
Average Response Time (AI Processes)	≤ 5 seconds
Dashboard Load Time	1.1 seconds
Database Query Time	140–780 ms depending on complexity
Uptime Reliability	99.30%
Memory Utilization	220–480 MB (peak)

VII. SCALABILITY, FUTURE ENHANCEMENTS

A. Scalability and Future Enhancement

The AI-Mail Management System has been architected using modular, service-oriented design principles to ensure high scalability, maintainability, and adaptability to evolving workloads. The system supports both horizontal and vertical scaling to accommodate future expansion across enterprise and institutional deployments. Horizontal scaling can be achieved through database replication and load balancing across multiple PostgreSQL and MongoDB instances to handle high query volumes and concurrent user sessions. Vertical scaling through enhanced CPU and memory allocation on cloud servers supports peak workload conditions such as mass email campaigns or bulk summarization events. Migration toward a microservices-based architecture will allow independent scaling of critical components like template generation, summarization, and spam filtering, ensuring resource efficiency and modular deployment flexibility. Cloud-based deployment using AWS, Microsoft Azure, or Google Cloud Platform (GCP) will enable auto-scaling, geographic redundancy, and server-less execution, minimizing infrastructure maintenance while providing consistent availability. Planned future enhancements include: AI-Driven Predictive Analytics: Leveraging historical mail interaction data to predict user response patterns and automate prioritization. Mobile Application Development: Enabling real-time email management and AI assistance through Android/iOS mobile clients. Voice-Based Assistance: Integrating speech

B. Functional Correctness and Reliability

System evaluation across multiple testing iterations demonstrates 100% correctness in critical operations and 99.3% re- liability under extended workloads. All CRUD (Create, Read, Update, Delete) operations across modules — including email management, summarization, template generation, and auto-response storage — executed successfully, maintaining complete data integrity verified through both database inspection and user interface validation. The Role-Based Access Control (RBAC) framework operates precisely as designed, enforcing access boundaries between administrators, registered users, and AI modules. Administrative dashboards display complete analytics, while regular users interact with permitted AI features only, preventing unauthorized privilege escalation. Real- time synchronization mechanisms ensure dashboard metrics (email count, spam ratio, response latency) update within 1–2 seconds of backend changes. Application-level

recognition APIs for hands-free composition and summarization. Blockchain Integration: Ensuring immutable audit logs for message integrity and compliance tracking.

VIII. CHALLENGES, LIMITATIONS, AND CONCLUSIONS

A. Implementation Challenges and Constraints

The implementation and deployment of the AI-Mail Management System encountered several practical and technical challenges during its development lifecycle. Technical dependencies included strict integration requirements with specific versions of Flask, FastAPI, and PostgreSQL, ensuring compatibility with institute-provided infrastructure and limited local server resources. Initial deployment was confined to a local development environment (XAMPP/WAMP) prior to migration onto cloud platforms such as AWS and Vercel. User adoption challenges emerged due to the introduction of AI-driven workflows unfamiliar to many users accustomed to conventional email handling. Training sessions and user manuals were required to overcome the learning curve and promote acceptance among end-users. Data preparation and migration posed another constraint—transferring existing organizational email data and templates into the new AI-driven environment required considerable manual verification to maintain accuracy and privacy compliance. The system presently implements manual testing and validation across AI modules and web interfaces. Future improvements include integrating Selenium-based automation testing, unit test pipelines, and continuous integration/continuous deployment (CI/CD) workflows to improve regression testing efficiency. Additionally, adherence to OWASP, GDPR, and WCAG standards-imposed design constraints, requiring careful handling of sensitive user data, accessibility compliance, and rigorous security audits before deployment. These challenges, while increasing initial implementation complexity, contributed to the overall maturity, reliability, and resilience of the system's architecture.

B. Project Conclusion

The AI-Mail Management System successfully demonstrates the practical application of Artificial Intelligence and Natural Language Processing in

automating modern communication workflows. By integrating intelligent modules for template generation, summarization, spam detection, and auto-response creation within a unified, secure web platform, the system achieves measurable improvements in operational efficiency, response accuracy, and user productivity. The project establishes a scalable, cloud-ready framework capable of handling large-scale enterprise or institutional communication demands while maintaining high performance and data security. Through extensive testing and validation, the system achieved over 98% operational reliability and 100% functional correctness in module-level evaluation. This work exemplifies how AI-driven automation can transform traditional communication systems into intelligent, responsive, and data-driven infrastructures. It aligns with the broader goals of Digital India by promoting transparency, accessibility, and innovation in digital communication. By focusing on scalability, security, and usability, the AI-Mail Management System provides a replicable model for future intelligent communication systems across industries, academic institutions, and government organizations. The project thus contributes significantly to India's ongoing digital transformation by establishing a foundation for AI-assisted, secure, and efficient e-communication ecosystems.

ACKNOWLEDGMENT

The authors express gratitude to the Department of Computer Engineering for continuous guidance and technical support during the development and testing of the AI-Mail Management System, and to all faculty mentors who provided valuable insights into AI-based email summarization and automation.

REFERENCES

- [1] A. Masri and M. Al-Jabi, "A novel approach for Arabic business email classification based on deep learning machines," *PeerJ Computer Science*, vol. 9, p. e1221, 2023.
- [2] J. Pinto, R. B. Sharanya, and T. Shreya, "MACHINE LEARNING FOR EMAIL CLASSIFICATION," *Redshine Archive*, vol. 14, no. 2, 2024.

- [3] S. Rekik and N. Bouldi, "Automatic emails classification," in *2023 20th International Conference on Telsiks (TELSIKS)*, 2023, pp. 1-4.
- [4] "Supervised methods of machine learning for email classification: a literature survey," ResearchGate, 2025.
- [5] S. Yilmaz and N. Yilmaz, "A Comparative Evaluation of a Multimodal Approach for Spam Email Classification Using DistilBERT and Structural Features," *Electronics*, vol. 14, no. 19, p. 3855, 2024.
- [6] N. Dhanda and K. K. Gupta, "A Novel Approach to Text Summarization Using Machine Learning," *Asian Journal of Research in Computer Science*, vol. 17, no. 4, pp. 95–104, 2024.
- [7] "Transformer models in text summarization," ResearchGate, 2024.
- [8] "Long-Text Abstractive Summarization using Transformer Models: A Systematic Review," ResearchGate, 2025.
- [9] "Current Trends and Advances in Extractive Text Summarization: A Comprehensive Review," ResearchGate, 2025.