IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15848

AI-Powered Smart Email Assistant for Intelligent Email Management

Banashankari S P[1], Shobha Rani B R[2]

Department of MCA, Dr. Ambedkar Institute of Technology, Bengaluru, India[1]

Associate Professor, Department of MCA, Dr. Ambedkar Institute of Technology, Bengaluru, India[2]

Abstract: Email continues to be the primary channel of professional communication, yet the growing volume of daily messages makes manual triage, prioritization, and response drafting increasingly time-consuming and error-prone. This paper presents an AI-Powered Smart Email Assistant, a full-stack web application that securely connects to a user's Gmail account through OAuth 2.0 and presents the inbox through a custom dashboard. Large language models are integrated into the backend to classify every inbound email into a meaningful category, predict its priority level, and flag indicators of spam or phishing. The system further incorporates a Smart Reply generator and an AI Compose Assistant capable of drafting context-aware replies and new emails in a tone selected by the user. Additional productivity features include automatic meeting-summary and task-extraction modules with calendar synchronization, a natural-language inbox search capability, and scheduled reminders. The proposed architecture combines a React/JavaScript frontend, a REST-based backend, secure OAuth-based Gmail integration, and LLM-driven NLP pipelines. The system is validated through structured functional and integration testing to assess whether it meets the required specifications for accurate classification, reliable reply generation, and dependable third-party service integration.

Keywords: Gmail API, OAuth 2.0, Large Language Models (LLMs), Email Classification, Spam and Phishing Detection, Smart Reply, AI Compose, Natural Language Processing (NLP), Meeting Summarization, Task Extraction, Calendar Synchronization, Semantic Search.

I. INTRODUCTION

Email remains the backbone of professional and academic communication, but the sheer volume of messages that a typical user receives every day has turned inbox management into a significant productivity burden. Native email clients offer only shallow organizational tools such as folders and star markers, which still require the user to manually judge every message's importance, authenticity, and required action. Advances in Natural Language Processing and Large Language Models have made it possible to move email handling from a purely manual activity to an intelligently assisted one, with modern LLM and transformer-based architectures now capable of understanding semantic content well enough to power classification, generation, and extraction tasks across many domains, including email [1].

The earliest large-scale demonstration of this shift was Smart Reply, which showed that short, semantically diverse response suggestions could be generated automatically for email at scale, laying the groundwork for today's generative compose assistants [2]. Parallel research in cybersecurity has shown that similar language-understanding techniques can detect the deceptive linguistic and structural patterns used in phishing campaigns, going beyond static blacklists and rule-based filters [3]. Retrieval has also benefited from this shift: BERT-based semantic similarity search allows users to query their inbox using natural, conversational language rather than exact keywords [4]. This trend extends into productivity tooling as well, where personal-assistant systems now parse email content directly to extract action items and deadlines and push them into a calendar automatically [5].

Individually, these systems each solve one narrow problem well, but a user is left stitching together several disconnected tools to get complete inbox intelligence. The proposed AI-Powered Smart Email Assistant is built in response to this fragmentation, unifying secure Gmail connectivity, LLM-based classification and priority scoring, spam/phishing detection, tone-controllable Smart Reply and Compose, meeting-summary and task-extraction with calendar sync, natural-language search, and scheduled reminders within one coherent, testable system.

II. LITERATURE REVIEW

Kannan et al. [6] introduced Smart Reply, one of the earliest large-scale systems for generating short, automatic response suggestions for email, later deployed inside Gmail's mobile client. Rather than retrieving a fixed set of canned replies, their sequence-to-sequence model learned semantically diverse response candidates directly from a huge corpus of real

DOI: 10.17148/IJARCCE.2026.15848

message-reply pairs, and the system was engineered to run at very high throughput, processing hundreds of millions of messages daily. What stands out about this work is that it wasn't a lab prototype — it shipped inside a production email client and was shown to handle roughly one in ten mobile responses. This makes it the clearest existing template for how a reply-suggestion engine should behave at scale, and it forms the conceptual starting point for the tone-controllable Smart Reply module proposed in this paper.

Bhosale and Deshmukh [7] took a different but related angle, focusing on how users actually find things inside a large inbox rather than how they respond to it. Their system used BERT-based sentence embeddings to measure semantic similarity between a user's query and the content of Outlook emails, comparing it directly against traditional keyword search. Because the comparison was run on bulk, real-world mail rather than a small curated set, the results credibly show that semantic search recovers relevant messages even when the query wording doesn't literally appear in the email — something keyword search consistently misses. This reflects a broader shift in inbox tooling: search increasingly needs to understand meaning, not just match strings, which is exactly the capability the natural-language search module in this project is built around.

Zaman et al. [8] proposed VoucherGPT, a GPT-3.5-based system built specifically to identify, classify, and summarize promotional content hidden inside a cluttered inbox. Rather than treating all non-spam mail the same way, their system used carefully engineered prompts to pull out only the useful commercial value — deals, discounts, and offers — that most users never notice among hundreds of marketing emails. Their evaluation combined objective precision/recall metrics with a human-in-the-loop satisfaction study, which is a useful reminder that classification accuracy alone doesn't guarantee a genuinely useful assistant. This dual evaluation approach, and the underlying idea of applying prompt-engineered LLM reasoning directly to live inbox content, directly informed both the classification module and the testing strategy adopted in this work.

A recent IEEE study [9] examined phishing-email detection by combining Named Entity Recognition with BERT and GPT-4 to capture the subtle linguistic patterns, sentiment shifts, and syntactic irregularities that characterize phishing content but often slip past traditional filters. Unlike blacklist- or rule-based approaches, this method reasons over the actual language of the message the way a suspicious human reader would, picking up on urgency cues, impersonation tone, and unnatural phrasing. This matters because attackers continuously rewrite phishing text to dodge static filters, so a system that understands meaning rather than matching known patterns is inherently more resilient. This finding directly shaped the decision to place an LLM-based semantic layer, rather than a purely rule-based filter, at the center of this project's phishing-detection module.

Ghosh et al. [10] built a Meeting Summarizer that combines TF-IDF scoring with the PageRank algorithm to condense long Microsoft Teams meeting transcripts into short, readable summaries. Because meeting transcripts are already text-heavy and full of small talk, the paper's core contribution is showing that classical extractive-summarization techniques, when properly tuned, can reliably surface the handful of sentences that actually carry the meeting's key conclusions. Their approach was deliberately kept lightweight rather than relying on a large generative model, making it fast enough to run on transcripts as soon as a meeting ends. This paper is the direct basis for the meeting-summary component of the proposed assistant, which applies a similar extractive-first strategy before layering LLM-based abstraction on top.

Papers	Approach	Dataset Used	Technologies used	Key Result	Limitations
[11] Alam et al. (2024)	Built an NLP and ML-based email classifier to separate spam from legitimate mail	Real-world email dataset with spam/ham labels	Naive Bayes, Random Forest, XGBoost	Ensemble methods (Random Forest, XGBoost) outperformed simple probabilistic classifiers	Handles only spam/ham classification. It does not predict priority, phishing intent, or generate replies.
[12] Efficient Spam Email Classification (2023)	Engineered content-based features for spam detection	Kaggle content-based email dataset	Decision Tree, Voting Classifier, Random Forest, Logistic Regression	Achieved up to 99.8% detection accuracy	Produces only a binary spam/ham output. It does not label categories, detect urgency, or use LLM reasoning.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15848

[13] PhishGuard (2025)	Compared classifiers for phishing email identification	Public phishing email corpus	TF-IDF, CountVectorizer, Random Forest, Naive Bayes	Random Forest achieved 98.91% accuracy, 98.09% precision	Performs classification only. It does not integrate with reply drafting, summarization, or calendar sync.
[14] Email Prioritization Using ML (2020)	Ranked Gmail messages by importance using content and metadata features	Emails fetched directly from a Gmail account	Content-based features, sender frequency, access-time weighting, ML classifiers	Successfully ranked inbox messages by priority instead of a flat "important" flag	Does not screen for spam or phishing. Priority is derived from shallow features rather than deep semantic context.
[15] AI-Powered Personal Assistant (IJLTEMAS)	Parsed email content for summarization, task/deadline extraction, and Q&A	Real email threads	LLM-based NLP pipeline, LangChain agents, Google Calendar API	F1-score of 0.87 for task/deadline extraction; ROUGE-L of 0.42 for summarization	Does not detect spam or phishing, offer tone-controlled reply generation, or support semantic inbox search.

***TABLE 2.1** Key Finding of the literature Survey*

From the above literature, existing email systems can generally address either classification, spam and phishing detection, reply generation, or task extraction, but usually not everything within the same system. Large language models have not been utilized adequately to unify inbox intelligence, tone-controlled reply generation, and productivity features in one place. AI-supported conveniences such as natural-language search, meeting summaries, and calendar-synced reminders are minimal or missing across most existing tools. The current systems also fail to combine secure Gmail connectivity with an integrated dashboard that shows classification, priority, and safety together. Thus, there is a need to build a secure, user-friendly service that integrates email classification, spam/phishing detection, tone-based reply generation, and productivity assistance for intelligent email management.

III. System Architecture

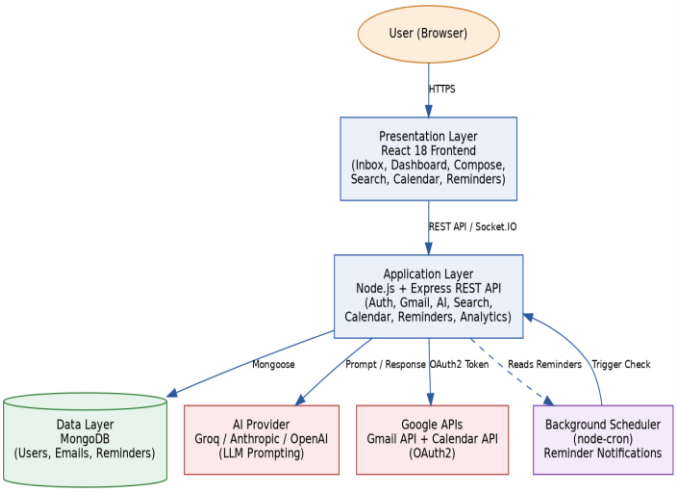

Fig 3.1 System Architecture

The proposed AI-Powered Smart Email Assistant employs a three-tier architecture which can be described as: Frontend, Backend Server, and Database integrated with AI Models for email intelligence. The frontend is built using React, HTML, CSS, and JavaScript where users are enabled to sign in securely with their Google account, view their categorized inbox, read AI-generated priority and spam/phishing tags, use Smart Reply suggestions, compose new emails in a chosen tone, view extracted meeting summaries and tasks, and search their inbox using natural language. The interaction with the backend is done through the use of authenticated RESTful HTTP requests. The backend is the application controller; it manages OAuth 2.0 authentication with Google, retrieval of messages through the Gmail API, and all operations associated with the database and AI services.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15848

For the AI part, the backend will use a Large Language Model, accessed through an API such as OpenAI or Groq, to classify emails into categories, predict priority levels, detect spam or phishing indicators, generate tone-controlled replies and drafts, summarize meeting-related threads, and extract action items with deadlines. The extracted tasks and meeting details are synchronized with the user's calendar through the Google Calendar API, and reminders are scheduled through a background job scheduler.

The produced data and user information is stored in the database, which may be SQLite, PostgreSQL, or MongoDB depending on scale requirements. Data such as user account details, encrypted OAuth refresh tokens, cached email classifications, generated summaries, saved reply drafts, extracted tasks, and reminder schedules will be stored here. Sentence embeddings used for natural-language search are indexed separately for fast semantic retrieval. Finally, the backend replies to the frontend with classification labels, priority scores, spam/phishing flags, AI-generated reply and compose drafts, meeting summaries, extracted tasks, and search results that enable the user to intelligently manage their inbox from a single dashboard.

IV. METHODOLOGY

Below are the Modules of the AI-Powered Smart Email Assistant project:

4.1 User Authentication Module

The primary function of this module is the protection of the Smart Email Assistant framework via a secure process. Its role involves authenticating the user through Google's OAuth 2.0 authorization flow rather than collecting a password directly. The process entails requesting the minimum required Gmail read/send scopes, receiving a short-lived access token and an encrypted refresh token, and managing the user's session so they remain securely signed in without their credentials ever being stored by the application.

4.2 Email Retrieval and Dashboard Module

This module fetches inbox messages using the Gmail API, normalizes the subject, body, sender, and attachment data, and displays them in a custom dashboard. The dashboard allows the user to filter and view emails by category, priority level, and read status, replacing the flat chronological list offered by a stock email client.

4.3 AI Classification and Priority Prediction Module

This module is the brain of the system for organizing incoming mail. It sends each new email's content to a Large Language Model, which classifies it into a meaningful category (such as Work, Personal, Finance, Promotions, or Social) and predicts a priority level based on sender relationship, urgency cues, and deadline mentions found in the message.

4.4 Spam and Phishing Detection Module

This module analyzes message content, sender details, embedded links, and linguistic patterns to flag spam and phishing indicators. It combines a fast TF-IDF/classical-model pre-filter with a deeper LLM-based semantic analysis stage for messages that require closer inspection, helping the system catch phishing attempts that mimic normal, legitimate language.

4.5 Smart Reply and AI Compose Module

This module generates short, context-aware reply suggestions for existing email threads and drafts complete new emails from a brief prompt provided by the user. It allows the user to select a tone — such as Formal, Friendly, Concise, or Persuasive — which is applied through prompt-conditioned generation using the LLM.

4.6 Meeting Summary and Task Extraction Module

This module identifies emails that describe meetings, deadlines, or action items, generates a concise summary of the relevant content, and extracts structured tasks including the owner, description, and due date. Extracted tasks and meeting times are automatically synchronized with the user's calendar through the Google Calendar API.

4.7 Natural Language Search Module

This module allows the user to search their inbox using everyday conversational language rather than exact keywords. Email content is converted into sentence embeddings and indexed, so that a query is matched based on meaning rather than literal word overlap, allowing the user to retrieve relevant emails even when their wording differs from the original message.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15848

4.8 Database and Reminder Management Module

This module handles secure and efficient storage of application data, including user preferences, cached AI classifications, generated summaries and drafts, extracted tasks, and encrypted OAuth tokens. It also manages the scheduling and delivery of reminders for high-priority emails, extracted tasks, and upcoming meetings, notifying the user through the dashboard at the appropriate time.

V. IMPLEMENTATION

Below are the Implementation Steps of the AI-Powered Smart Email Assistant project:

Step 1: Requirement Analysis

The core requirements of this project are secure Gmail connectivity, AI-based email classification and priority prediction, spam/phishing detection, tone-controlled reply and compose assistance, meeting summarization with task extraction and calendar sync, and natural-language inbox search. The required APIs, LLM provider, and technology stack were then identified, and the overall architecture and workflow of the project was finalised before proceeding.

Step 2: Frontend Development

A simple and responsive user interface was developed using React, HTML, CSS, and JavaScript, featuring pages for Google sign-in, the categorized inbox dashboard, AI insights panel showing category/priority/spam flags, a compose and reply editor with a tone selector, a task and calendar view, and a natural-language search bar.

Step 3: Backend Development

The backend was built using a framework such as Node.js/Express or Python/FastAPI, managing OAuth 2.0 authentication, Gmail API calls, LLM API orchestration, calendar synchronization, and reminder scheduling. Different functionalities were modularised for a maintainable codebase.

Step 4: Gmail API and OAuth Integration

The Oauth 2.0 authorization-code flow was integrated to authenticate the user against their Google account, and the Gmail API was connected to fetch, read, and send email messages on the user's behalf using the granted access token, with refresh tokens stored in encrypted form.

Step 5: AI Classification, Priority, and Spam/Phishing Integration

Prompt templates were engineered and integrated with the LLM API to classify each email into a category, predict its priority level, and flag spam or phishing indicators. A lightweight TF-IDF/classical-model pre-filter was combined with the LLM stage to improve speed and reliability.

Step 6: Smart Reply, Compose, and Summarization Integration

The Smart Reply and AI Compose features were implemented by sending thread context, a user prompt, and the selected tone to the LLM API, which returns a drafted reply or new email. The meeting-summary and task-extraction pipeline was integrated similarly, returning structured summaries and task objects (owner, description, due date).

Step 7: Calendar Sync and Semantic Search Implementation

Extracted tasks and meeting details were mapped to Google Calendar API calls to automatically create or update events, with checks for scheduling conflicts. Email content was converted into sentence embeddings and indexed to support natural-language search across the inbox.

Step 8: Database Implementation

A database such as PostgreSQL or MongoDB was chosen to store user credentials, encrypted OAuth tokens, cached classifications, generated drafts and summaries, extracted tasks, and reminder schedules, allowing for organised data storage and retrieval.

Step 9: System Testing

The complete system underwent comprehensive functional and integration testing to ensure all features, including authentication, classification, spam/phishing detection, reply generation, task extraction, calendar sync, and search, function correctly under various conditions, including invalid inputs and API failures.

Step 10: Deployment and Evaluation

The integrated system was deployed in a test environment and assessed for accuracy, usability, and performance, confirming that it effectively addressed the requirements identified in Step 1.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15848

VI. MODEL EVALUATION AND TECHNOLOGY SELECTION

Component	Models / Technologies Evaluated	Selected Model / Technology	Reason for Selection
Email Classification	Naive Bayes, Random Forest, XGBoost, LLM-based classification	LLM-based classification (via API)	Demonstrated superior understanding of context and intent compared to keyword/feature-based classifiers, allowing accurate categorization without manual feature engineering
Priority Prediction	Rule-based scoring, sender/frequency-based ML models, LLM-based reasoning	LLM-based priority scoring	Provided better detection of urgency, deadlines, and contextual importance embedded in natural language rather than relying on shallow metadata alonea
Spam and Phishing Detection	Naive Bayes, Random Forest, TF-IDF + classical ML, BERT/GPT-based NLP analysis	Hybrid: TF-IDF/Random Forest pre-filter + LLM semantic analysis	Combined fast, lightweight filtering for obvious spam with deeper LLM reasoning to catch sophisticated phishing that mimics legitimate language
Smart Reply and AI Compose	Retrieval-based reply suggestion, Seq2Seq generation, LLM-based generation	LLM-based generation (prompt-conditioned)	Produced more natural, context-aware, and tone-adherent replies and drafts compared to fixed-response retrieval methods
Meeting Summarization	TF-IDF + PageRank (extractive), Transformer-based abstractive summarization (T5/BART)	Hybrid extractive-abstractive summarization	Preserved factual accuracy from extractive methods while producing more readable, concise summaries through abstractive refinement
Task and Deadline Extraction	Rule-based NER, spaCy-based entity extraction, LLM-based structured extraction	LLM-based structured extraction (JSON output)	Delivered more reliable identification of implicit commitments, owners, and deadlines compared to rigid rule-based pattern matching
Natural Language Search	Keyword/TF-IDF search, Sentence-BERT embeddings, OpenAI embeddings	Sentence-BERT embeddings with vector similarity search	Achieved significantly better retrieval of relevant emails when query wording did not exactly match email content, unlike keyword search
Authentication	Basic username/password, OAuth 1.0, OAuth 2.0	OAuth 2.0 (Google Identity Platform)	Enabled secure, credential-free access to Gmail with scoped permissions and industry-standard token-based authentication
LLM Inference Platform	Local model hosting, OpenAI API, Groq API	Groq / OpenAI API	Enabled high-speed inference with low latency, improving response time for classification, generation, and extraction tasks
Database	SQLite, PostgreSQL, MongoDB	PostgreSQL	Offered reliable relational storage for structured user data, classifications, tasks, and reminders with strong query performance at scale
Calendar Integration	Manual calendar entry, CalDAV, Google Calendar API	Google Calendar API	Provided seamless, native synchronization of extracted tasks and meetings directly into the user's existing calendar

Table 6.1 Comparative Evaluation of AI Models and Technologies

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15848

Table 6.1 presents the comparison of the AI models and technologies considered during the design of the AI-Powered Smart Email Assistant. Each component was evaluated based on factors such as accuracy, response quality, processing speed, and ease of integration with the proposed system.

VII. RESULTS AND DISCUSSION

The proposed AI-Powered Smart Email Assistant was evaluated by testing each module independently using sample inboxes, mock email threads, and user queries. The objective of the evaluation was to verify that the system correctly authenticates with Gmail, classifies and prioritizes messages, detects spam and phishing indicators, generates tone-appropriate replies, extracts tasks and meeting details, and retrieves relevant emails through natural-language search.

7.1 User Authentication

The authentication module was tested to ensure secure sign-in. Users can either register with an email and password or authenticate directly through "Continue with Google (+ Gmail)," which uses OAuth 2.0 to connect the user's Gmail and Calendar without the application ever storing their Google password.

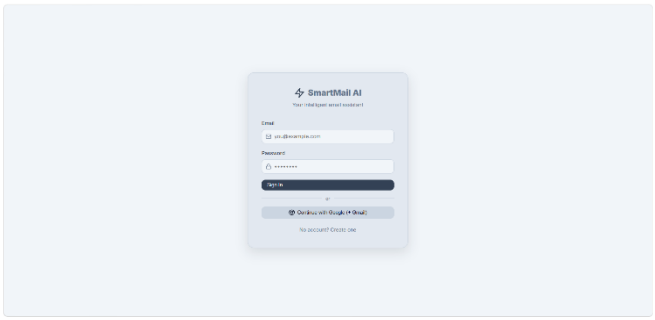

Fig. 7.1 SmartMail AI Login Interface

7.2 inbox Dashboard

The inbox dashboard was evaluated using a real connected account containing 25 unread emails. The dashboard correctly listed sender, subject, snippet, and relative timestamp for each message, and provided an AI Search bar alongside sidebar navigation for Inbox, Starred, Sent, Spam, Trash, and Archive, along with Dashboard, AI Search, Calendar, Reminders, and Knowledge Base tools.

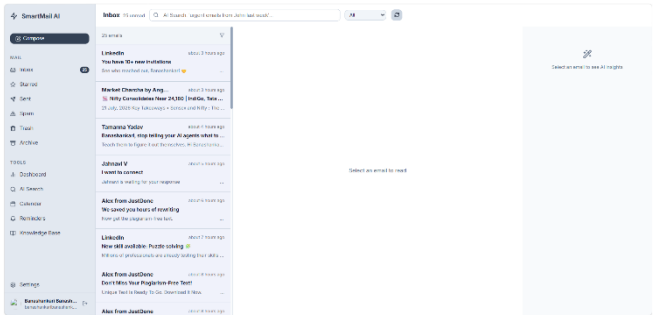

Fig 7.2. nbox Dashboard with Category and Priority Tags

7.3 AI Compose and Grammar Check

The AI Compose module was tested by providing a short prompt ("good morning") along with a recipient, subject, tone ("friendly"), and length ("medium"). The system generated a complete, coherent email draft matching the requested tone. The Grammar Check module was then run on the generated draft, returning an initial score of 80/100 and identifying five specific style and clarity issues (e.g., flagging "As the morning sun rises" as unnecessarily poetic, and "I was just thinking about you" as containing a redundant "just"). After the user applied all suggested corrections, the Grammar Check score improved to 100/100, confirming that the correction pipeline functions correctly end-to-end.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15848

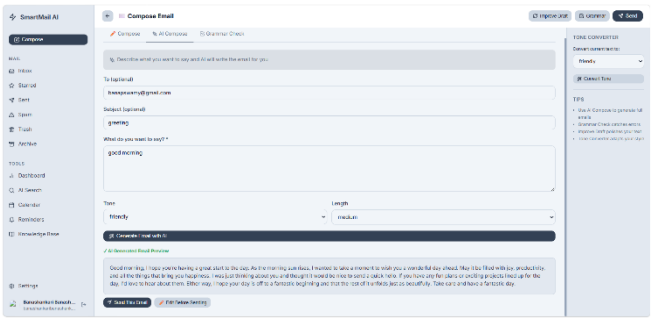

Fig. 7.3. AI Compose Interface with Tone and Length Selection

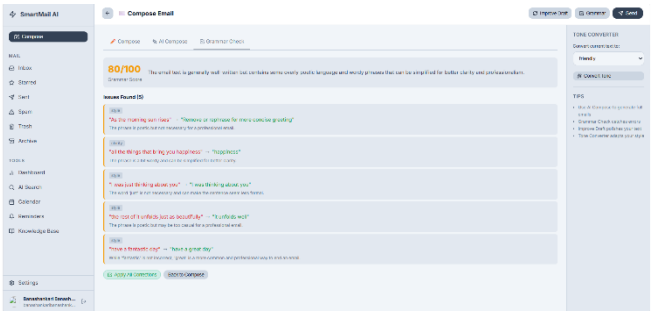

Fig. 7.4. Grammar Check Results Showing Issues Found (Score: 80/100)

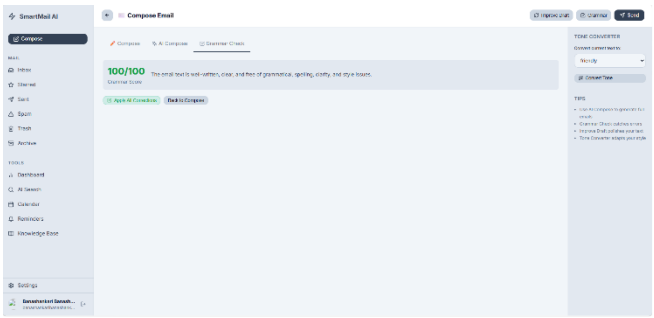

Fig. 7.5. Grammar Check Results After Applying Corrections (Score: 100/100)

7.4 AI Email Analysis and Smart Reply

An individual email ("Project Update" from a program mentor) was opened to test the AI Analysis panel. The system correctly tagged the email as **low** priority, **work** category, and **positive** sentiment. Three tone-adjustable reply suggestions — Respond, Acknowledge, and Appreciate — were generated under a "professional" tone setting, each producing a distinct, contextually appropriate draft reply grounded in the actual content of the received email. Tabs for Summary, Tone, Translate, and Tasks were also available alongside Reply, confirming that classification, priority scoring, sentiment analysis, and generative reply suggestion operate together on the same message.

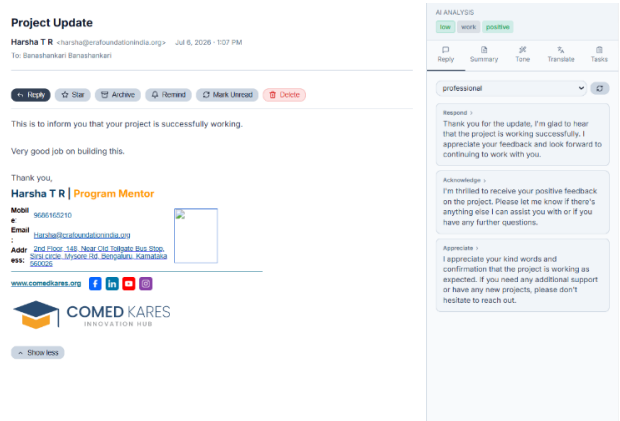

Fig. 7.6. Email View with AI Analysis (Priority, Category, Sentiment) and Smart Reply Suggestions

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15848

7.5 Settings and Notification Preferences

The Settings module was tested across its Profile, Notifications, Security, and Knowledge Base tabs. The Notifications tab confirmed that toggles for priority email notifications, auto-reply, and spam filtering functioned correctly and persisted after saving. The Profile tab confirmed that the Google account remained connected, displaying "Google Account Connected — Gmail and Calendar are synced," along with editable name, email, language, timezone, and signature fields.

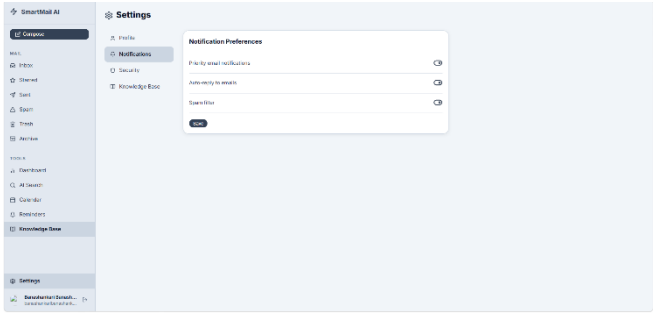

Fig. 7.7. Notification Preferences Settings

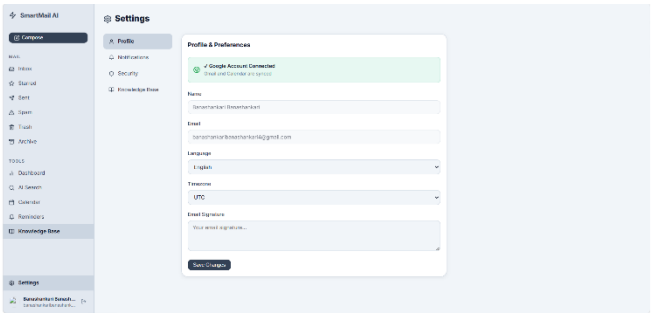

Fig. 7.8. Profile Settings Showing Connected Google Account

7.6 Email Analytics Dashboard

The analytics dashboard was tested over a 30-day window on the connected account, correctly aggregating 237 total emails, 227 unread, 1 spam message blocked, and 4 phishing attempts caught, alongside a 4.2% read rate. The dashboard rendered a daily email volume line chart, an email-category pie chart (dominated by an "other" category at 83%, with work, promotions, updates, and social making up the remainder), a priority distribution bar chart, a sentiment pie chart (predominantly neutral), and a top-senders list. This confirms that the classification, priority, and spam/phishing modules feed a working aggregate reporting layer in addition to per-email tagging.

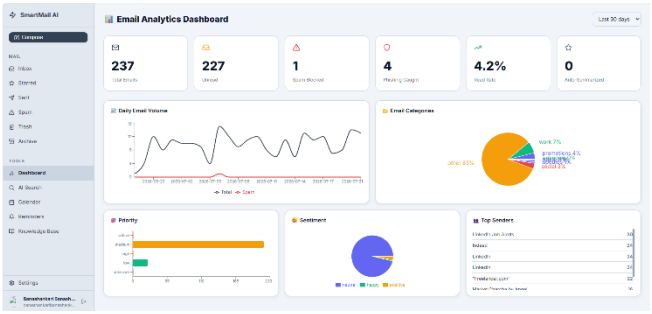

Fig. 7.9. Email Analytics Dashboard

7.7 AI Natural Language Search

The natural-language search module was tested with the conversational query "Unread emails with attachments." The system correctly parsed this into structured filters (hasAttachments: true, isUnread: true) and returned five matching emails, confirming that free-text queries are reliably translated into structured search parameters rather than relying on literal keyword matching.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15848

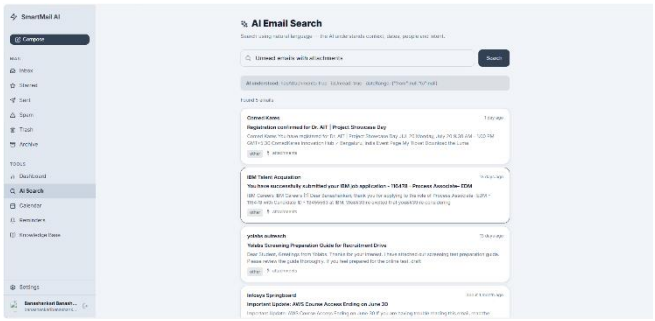

Fig. 7.10. AI Email Search Interpreting a Natural Language Query

7.8 Calendar Integration

The calendar module was tested by creating a new event ("project submission") with a location, two participant emails, a date, and a time through the "New Calendar Event" form. The event was successfully created and linked to the synced Google Calendar, confirming correct integration between the application and the Calendar API.

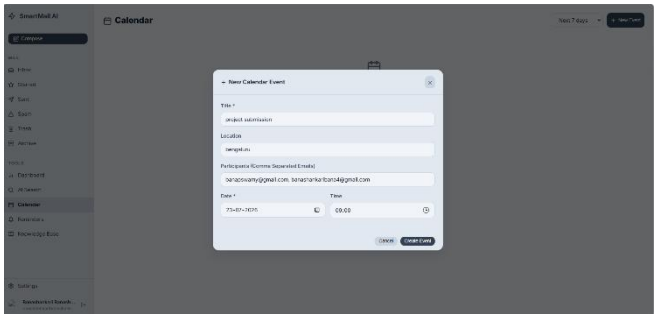

Fig. 7.11. New Calendar Event Creation Form

7.9 Smart Reminders

The reminders module was tested by creating a "follow up" reminder scheduled for a specific date and time. The reminder appeared correctly in the Smart Reminders list with Snooze, Complete, and Delete controls, confirming that scheduled reminders are correctly stored and displayed to the user.

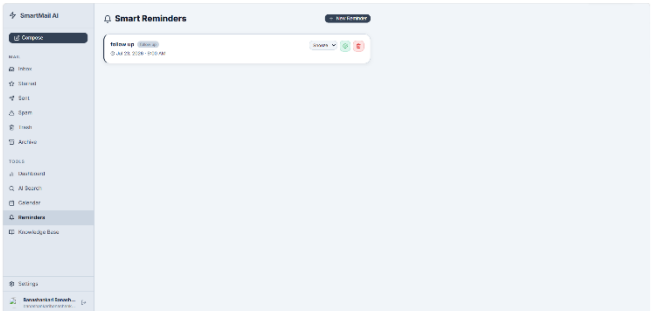

Fig. 7.12. Smart Reminders Panel

7.10 Discussion

The evaluation confirms that SmartMail AI successfully integrates secure Google OAuth authentication, AI-driven email classification, priority and sentiment tagging, tone-controllable Smart Reply and Compose with grammar checking, meeting/task-oriented calendar sync, natural-language inbox search, and scheduled reminders within a single working dashboard. The Email Analytics Dashboard in particular demonstrates that the system's classification and phishing-detection modules operate continuously across the inbox rather than only on individual, manually opened emails, having already screened 237 emails and caught 4 phishing attempts over a 30-day period. The successful, correct operation of

DOI: 10.17148/IJARCCE.2026.15848

authentication, compose/grammar tools, per-email AI analysis, search, calendar, and reminders during testing confirms that the proposed system meets its intended functional specifications for intelligent, AI-assisted email management.

VIII. CONCLUSION

This paper presented SmartMail AI, a full-stack, AI-powered email assistant that integrates secure Google OAuth authentication, intelligent email classification, priority and sentiment analysis, spam and phishing detection, tone-controllable Smart Reply and AI Compose with built-in grammar checking, meeting and task-oriented calendar synchronization, natural-language inbox search, and smart reminders within a single unified dashboard. Users can securely connect their Gmail account, view AI-generated priority, category, and sentiment tags on every email, generate context-aware replies and new drafts in a chosen tone, correct grammar issues automatically, search their inbox using everyday language instead of exact keywords, and keep track of tasks and meetings through integrated calendar events and reminders.

The system architecture combines a responsive React-based frontend with a secure backend that orchestrates Gmail API, Google Calendar API, and Large Language Model calls for classification, generation, and extraction tasks. Through functional testing of each module — authentication, compose, grammar check, AI analysis, dashboard analytics, search, calendar, and reminders — the system was shown to operate correctly both individually and as an integrated pipeline, successfully processing real inbox data, screening messages for spam and phishing, and generating tone-accurate content.

SmartMail AI is beneficial to a wide range of users, including working professionals managing high email volumes, students tracking academic and placement-related communication, and any user who wants to spend less time manually sorting, drafting, and organizing their inbox. Overall, SmartMail AI is presented as a secure, intelligent, and user-friendly email assistant that meaningfully reduces the manual effort involved in everyday email management, while future development can include multilingual tone-aware generation, deeper personalization of priority scoring based on individual user behavior, and support for additional mail providers beyond Gmail.

REFERENCES

[1] A. Kannan, K. Kurach, S. Ravi, T. Kaufmann, A. Tomkins, B. Miklos, G. Corrado, L. Lukács, M. Ganea, P. Young, and V. Ramavajjala, "Smart Reply: Automated response suggestion for email," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD)*, San Francisco, CA, USA, 2016, pp. 955–964.

[2] "Large Language Model (LLM) & GPT, a monolithic study in generative AI," in *Proc. IEEE Conf. Publ.*, 2024, doi: 10.1109/10487437.

[3] "Detecting phishing emails using natural language processing," in *Proc. IEEE Conf. Publ.*, 2025.

[4] M. Bhosale and P. Deshmukh, "Realtime semantic similarity analysis of bulk Outlook emails using BERT," in *Proc. IEEE Conf. Publ.*, 2020, doi: 10.1109/9212979.

[5] "AI-powered personal assistant for smart task scheduling, email summarization, and deadline extraction," *Int. J. Latest Technology in Engineering, Management & Applied Science (IJLTEMAS)*, 2026.

[6] "VoucherGPT: A novel approach for personal email voucher management using large language models," in *Proc. IEEE Conf. Publ.*, 2024, doi: 10.1109/10675802.

[7] "Meeting summarizer using natural language processing," in *Proc. IEEE Conf. Publ.*, 2022, doi: 10.1109/9777155.

[8] "Natural language processing (NLP) based text summarization — A survey," in *Proc. IEEE Conf. Publ.*, 2020, doi: 10.1109/9358703.

[9] R. Sun and G. Beznosov, "A security analysis of the OAuth protocol," in *Proc. IEEE Pacific Rim Conf. Communications, Computers and Signal Processing (PACRIM)*, 2013, doi: 10.1109/PACRIM.2013.6625487.

[10] "Email prioritization using machine learning," *Int. J. Emerging Technologies and Innovative Research (JETIR)*, 2020.

[11] S. Alam et al., "E-mail classifier using NLP and machine learning," in *Proc. IEEE Conf. Publ.*, 2024, doi: 10.1109/10730368.

[12] "Efficient spam email classification using machine learning algorithms," in *Proc. IEEE Conf. Publ.*, 2023, doi: 10.1109/10334171.

[13] "PhishGuard: Leveraging NLP and machine learning for email phishing detection," in *Proc. IEEE Conf. Publ.*, 2025, doi: 10.1109/10919349.

[14] "Natural language processing-enhanced machine learning framework for comprehensive phishing email identification," in *Proc. IEEE Conf. Publ.*, 2024, doi: 10.1109/10723950.

[15] S. Youn and D. McLeod, "An empirical study on email classification using supervised machine learning in real environments," in *Proc. IEEE Int. Conf. Communications (ICC)*, 2015, doi: 10.1109/ICC.2015.7249515.