International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 03 | March - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

EmailEye: An Email Alert System Using a WebApp

Rashida C K[1], Reshma S[2], S Amritha[3],Shreya D Nair[4],Silja Varghese[5]

Student[1],Student[2],Student[3],student[4],Assistant Professor[5] of
Computer Science Engineering Department,
Nehru College Of Engineering and Research Centre(NCERC),Thrissur,India

---

Abstract - An automatic playback bot that can identify dangerous URLs is integrated into the suggested web-based email alert and monitoring system. Users will be informed of important occurrences or conditions thanks to this system's real-time monitoring and alerting features. In order to identify anomalous data, provide highly personalized and educational alerts, and make recommendations for further actions, the system makes use of artificial intelligence technologies. To lessen alert overload and make sure users pay attention to pertinent messages, the system also uses reinforcement learning to modify users' responses to prior alerts. The automatic replaying bot can react appropriately to a variety of situations because it is trained to obey precise instructions. Additionally, a dangerous link detection tool built into the system finds and flags dubious links, averting certain security risks. The overall goal of this system is to improve user experience and optimize workflows.

***Key Words*:** Email, Artificial Intelligence, Reply, Suspicious link, Machine learning.

1.INTRODUCTION

Effective alert notification systems are crucial for enterprises looking to uphold strict security standards and provide real-time control of crucial activities in the ever evolving digital world. By giving users a highly customizable notification platform, the proposed web-based email alert and monitoring system aims to empower users by enabling them to prioritize information based on their individual needs and responsibilities, set alert thresholds, and select notification types. This adaptability supports a more efficient and secure workflow by guaranteeing that users receive alerts that are both timely and highly relevant. This system, which was developed to take use of cutting-edge artificial intelligence (AI) and machine learning techniques, is designed to provide users with a smooth and safe experience while drastically lowering the dangers connected to malicious links and unusual activity.

An integrated automatic replaying bot is a crucial component of the system, helping to monitor and interact with possible threats in real time. This bot works precisely, finding malicious links in emails and spotting odd patterns that can point to malware or phishing risks. By examining the content and behavior of links, it is designed to identify possible dangers and stop users from accessing dangerous material. In order to shield users from unintentional security breaches, this proactive detection approach is essential. The solution helps businesses keep a safe communication environment and lowers the risk of cybersecurity problems by continuously checking incoming emails for malicious material and dangerous links.

AI-driven methods are also used by the email alert and monitoring system to track and examine anomalous data. The system detects anomalous patterns that diverge from typical behavior using sophisticated algorithms, which may point to possible dangers or mistakes in the data stream. Users receive instant alerts from this abnormality detection, enabling timely investigation and action. Because the alerts are highly tailored, users only receive the most pertinent and educational information based on their responsibilities and past replies. The method reduces the possibility of "alert fatigue," a typical problem where users are overloaded with messages and may miss important signals, by allowing users to choose the type and frequency of alerts. The system can improve the relevancy of upcoming notifications by using reinforcement learning to learn from users' previous interactions with alerts. When a user routinely ignores specific warning kinds, for instance, the system modifies its notification settings to give priority to other alerts that are probably more pertinent to that user. By reducing alert saturation, this adaptive strategy makes sure that users ignore less important notifications while being attentive and responding to high-priority alerts.

Emails can contain either positive or negative replies based on their tone, language, and intent. A positive reply generally expresses agreement, appreciation, or encouragement. It often includes words like "thank you," "great," "happy to assist," or "looking forward," indicating satisfaction or enthusiasm. Such responses foster good relationships and constructive communication.

Conversely, a negative response indicates rejection, disapproval, or disagreement. Words that convey disappointment or refusal, such as "unfortunately," "regret," "concern," or "not possible," may be included. Handling negative answers carefully is necessary to

preserve professionalism and prevent conflict. An example of a professional response is when an employer says, "Unfortunately, we have decided to proceed with another candidate," which indicates a rejection. To promote efficient communication, emails whether favorable or negative should always be courteous, professional, and unambiguous.

All things considered, our web-based email alert and monitoring system combines state-of-the-art AI technologies with user-focused design to provide a comprehensive approach to security and productivity. The system tackles the difficulties of contemporary digital communication and cybersecurity with its real-time monitoring, adaptive alerting, and automated response capabilities, guaranteeing that users are safe, knowledgeable, and capable of reacting to urgent situations. This service is a useful tool for businesses looking to protect their email correspondence and boost operational effectiveness by improving user experience and lowering security threats.

2. LITERATURE REVIEW

[1].The design and implementation of an email notification system aimed at alerting small user groups to changes in data sources that update dashboards based on user-defined criteria is the main topic of the paper "Development of Email notification system based on user criteria by Sai Charan, Vikranth Paluri, and Sowmya Nag." Without forcing users to constantly log into the application, the system seeks to deliver pertinent notifications to them. It minimizes any influence on the main business system by using a data pipeline to identify changes in data sources and notify users when updates take place. The major objective is to preserve system performance while making sure users are promptly notified of significant modifications.

Two methods for implementing this notification system are analyzed and compared: one using the Java-based Spring Boot framework combined with SendGrid, a cloud-based SMTP provider, and the other utilizing Azure Logic Apps, an automated workflow service. The Spring Boot and SendGrid solution offers more control over the email sending process, allowing for greater customization in terms of email content and delivery logic. However, it requires more infrastructure management. On the other hand, Azure Logic Apps provides a low-code approach that simplifies workflow creation and integrates seamlessly with other Azure services, though it may offer less customization than the Spring Boot/SendGrid method.

[2].S. Appavu alias Balamurugan, Aravind, Athiappan, Bharathiraja, Muthu Pandian, and Dr. R. Rajaram's paper "Association Rule Mining for Suspicious Email Detection: A Data Mining Approach" suggests a data mining technique for identifying suspicious emails, especially those pertaining to illegal activity. The authors categorize emails as either alert (foretelling future hazards) or instructive (depicting past instances) using Association Rule Mining with the Apriori Algorithm. Building an associative classifier that creates rules based on common word patterns comes after email preprocessing, which cleans and tokenizes the text. By examining important characteristics like action verbs, emotional phrases, and tense indicators, the system uses deception theory to identify information that appears suspicious. According to experimental data, this technique successfully and accurately identifies suspicious communications. The authors advocate more research into different categorization algorithms to enhance performance, but they also say that this technology can help security agencies effectively filter and identify possible threats.

[3]. A web-based client-oriented email application that actively detects, monitors, and controls email spoofing assaults is shown in the paper "An Email Application with Active Spoof Monitoring and Control" by T.P. Fowdur and L. Veerasoo. This method gives end users more control over spoof assaults than standard server-based solutions. In order to authenticate message sources, the program sends notifications concurrently with email transmissions and combines HTTPS and SSL for secure connection. To avoid mailbox clutter, the system sends a warning on the dashboard when it detects a spoof email and routes the questionable email to a different spoof filter. In order to create an active feedback loop, the recipient might also inform the actual sender about the spoofing incidence. The system, which was created with PHP and the Netbeans IDE, and MySQL for database administration, has an intuitive user interface and is readily deployable online. By guaranteeing worldwide scalability, the suggested solution overcomes significant drawbacks of previous systems, such as those restricted to particular subnets. This solution improves email security and offers a practical way to identify and handle spoofing attempts by integrating secure communication protocols, alert systems, and user-controlled feedback choices. In order to increase the application's usability and accessibility, the authors recommend future improvements include revamping the user interface or creating a plugin for well-known email services like Gmail and Yahoo Mail.

[4]. The absence of notification in conventional mailing systems can be addressed by creating an automated SMS alert mailing system, according to the study "Development of an Automated SMS Alert Mailing System" by Nanwin, Domaka Nuka, and Williams, Daniel Ofor. By alerting users when fresh mail arrives in their mailboxes, this system attempts to lower the possibility that users would overlook or be careless and

miss crucial correspondence. Incoming mail is registered at the post office, where the post administrator enters the mail information into the system. After that, the mail is sent out, and the receiving administrator verifies that it has arrived at its destination. Once verified, an automated SMS alert alerting the recipient of the mail's arrival is delivered to the registered phone number. An SMS API gateway is used by this system to guarantee effective and prompt alerts. The design ensures a systematic and ordered approach to system development by utilizing the object-oriented analysis and design (OOAD) methodology. With the exception of live SMS alerts because there was no active web hosting setup, the solution was successfully deployed in a simulated offline environment where the registration, confirmation, and notification processes operated as intended. The authors stress that by adding real-time notifications, increasing productivity, and resolving the frequent problem of missed or delayed mail retrieval, this system improves on conventional mailing services. Implementing SMS alerts for unsuccessful delivery is one of the suggested future improvements to make sure users are notified when their mail is not delivered.

[5]. The method shown in the paper "Email Alerts on WhatsApp" by Sakshi K, Darshan I, Maroof M, Sushant J, and Ms. M.K. Kute uses Twilio to integrate email notifications with WhatsApp, making email management easier. By creating an automation tool that retrieves email information based on user queries and transmits it as WhatsApp messages, the suggested solution tackles the problem of managing a large number of emails. This system makes use of Twilio's WhatsApp Sandbox, a platform that enables prototype testing for both inbound and outgoing messages and automates message delivery. The software is effective for low bandwidth conditions because it uses Python's imaplib package to retrieve emails over the IMAP protocol without downloading them. Given that many users spend a lot of time on WhatsApp, the new tool allows users to send and view emails directly through the messaging app, increasing convenience. The process entails setting up Twilio to automate WhatsApp, establishing email connections using the SMTP and IMAP protocols, and utilizing PyCharm IDE to deploy the system to a web server via Heroku.The system offers a user-friendly solution for efficient email management while guaranteeing flexibility, scalability, and performance. The project successfully illustrates the possibilities of integrating messaging automation with popular communication channels like WhatsApp, increasing productivity and boosting email accessibility, even if the Twilio platform requires a membership for heavy use.

[6]. In order to increase email accessibility and efficiency, a system that integrates email alerts with WhatsApp using Twilio is presented in the paper "Email Alerts on WhatsApp" by N. Lalitha, N. Neelima, S. Sravya, and G. Pavan Kumar. The system tackles the problem of email management, which can be difficult because of the volume of messages and newsletters. The suggested system uses the Twilio platform to query email data according to user-specified criteria and deliver the pertinent information as WhatsApp messages. By handling message delivery callbacks, replying to incoming messages, and sending outbound messages, the Twilio Sandbox for WhatsApp makes it possible to design and test this automated system. In order to ensure that emails are read without being downloaded—a feature that is particularly helpful when bandwidth is limited—the system uses Python's imaplib package to access emails via the IMAP protocol. The project procedure entails setting up Twilio for automation, establishing email connections using SMTP and IMAP protocols, and deploying the system to the Heroku web server using the PyCharm IDE. Performance, usability, scalability, adaptability, and availability are some of the system's primary attributes. The authors point out that by enabling users to manage emails within a platform they regularly use, the integration of email alerts with WhatsApp saves time. Despite its robust features, Twilio requires a premium subscription in order to be used more widely. This creative method efficiently optimizes email management, increasing user convenience and productivity when using WhatsApp to handle emails.

[7]. Siva Kumar a/l Subramaniam, Siti Huzaimah binti Husin, Yusmarnita binti Yusop, and Abdul Hamid bin Hamidon's article "Real-Time Mailbox Alert System via SMS or Email" offers a creative way to enhance the traditional mailbox system by incorporating contemporary electronic technology. The Mail Alert solution (MASYS), the suggested solution, is intended to rapidly alert users by email or SMS when fresh mail arrives in their mailbox. To send alerts, the system combines a GSM modem, an interface module, and a programmable logic controller (PLC). The mailbox has infrared sensors placed to identify mail delivery events, which causes the PLC to produce a message. The interface module then processes this message before sending it to the GSM modem, which notifies the designated receiver. Additionally, the MASYS enables users to remotely verify the status of their mailboxes by sending an authorized code via SMS. Because it eliminates the need for regular human checks, this method is especially advantageous for people in multi-story buildings or centralized mailbox sites. Enhanced dependability, efficiency, and convenience are provided by MASYS, which is made to operate independently without human assistance. The system is affordable, easy to use, and has potential uses for both home and business customers, according to the authors. In order to enhance user experience and provide prompt access to critical mail, the project shows how feasible it is to integrate contemporary communication technologies with conventional postal systems.

[8]. Mudit Shishodia, Sachin Pal, and Shyamsundar's paper "Email Alerts on WhatsApp" describes a novel method that uses Twilio technology to combine email notifications with WhatsApp. By sending email warnings straight to WhatsApp, the technology seeks to streamline email tracking and save users' time and effort. The suggested approach tackles the problem of effectively handling emails, particularly in light of the increasing volume of messages and newsletters. The system uses Twilio's WhatsApp Sandbox to automate message delivery callbacks, replying to incoming messages, and sending outbound messages. To ensure performance even in low-bandwidth situations, the system uses Python's imaplib package to read emails via the IMAP protocol, which fetches email data without downloading it. Emails can also be sent using the SMTP protocol. The project procedure entails setting up Twilio for automation, establishing email connections using the SMTP and IMAP protocols, and utilizing PyCharm IDE to deploy the system to a web server via Heroku. To guarantee a flawless user experience, the system is built with essential functional criteria including flexibility, scalability, availability, and usability. Despite being a premium product, Twilio successfully illustrates how integrating well-known services like WhatsApp may improve email accessibility and expedite conversation. This creative strategy serves users who spend more time on WhatsApp by providing a practical way to keep an eye on and reply to emails within their favorite messaging app.

[9]. A system that combines email alerts with WhatsApp to increase email accessibility and boost user productivity is presented in the paper "Email Alerts on WhatsApp" by Mrs. V. Abinaya and Ms. Shobika J. The idea is to connect a WhatsApp account to an email account so that users may get email notifications straight through the chat app. By guaranteeing that crucial messages are instantly accessible without requiring app switching, this integration solves the frequent problem of ignored emails. The system makes use of WhatsApp Sandbox, a pre-configured environment that enables message automation prototyping, including delivery callbacks, incoming message replies, and outgoing messaging. In order to efficiently retrieve emails without downloading messages, the system makes use of the IMAP protocol, guaranteeing seamless operation even when bandwidth is limited. Email sending is also controlled by the SMTP protocol. The authors list a number of benefits, including enhanced responsiveness to critical emails, the ease of getting alerts on a platform that is often used, and the capability to view communications that are kept in trash or junk folders. HTML and CSS are used for front-end development, JavaScript is used for automation, and PyCharm is used to deploy the project to the Heroku server. By combining email notifications into WhatsApp, the system seeks to streamline email management, improving user convenience and lowering the possibility of missing important messages. This creative strategy satisfies contemporary communication patterns, as consumers depend more and more on instant messaging apps like WhatsApp for better connectivity and real-time updates.

[10]. Vincent Bazinette, Norman H. Cohen, Maria R. Ebling, Guerney D. H. Hunt, Hui Lei, Apratim Purakayastha, Gregory Stewart, Luke Wong, and Danny L. Yeh's paper "An Intelligent Notification System" describes a system that is intended to provide users with notifications in an effective and context-aware way. By alerting users only when certain events or circumstances are satisfied, the Intelligent Notification System tackles the increasing difficulty of managing enormous volumes of information. Users can specify which devices they would want to get alerts on, including pagers, email, instant messaging, and cell phones. Using information from multiple sources, including online feeds, stock tickers, and news wires, the system efficiently handles content aggregation. The three primary components of its architecture are the Secure Context Service, which securely manages user context information to guarantee that notifications are delivered at the right times and through the right devices; the Universal Notification Dispatcher, which sends alerts via the appropriate communication channels based on user preferences and current context; and the Trigger Management Service, which keeps an eye on content and finds conditions that match user-defined triggers. The system makes use of technologies such as DB2 for data storage, Gryphon (a content-based publish-subscribe system), and Java APIs. This system, which was created to reduce information overload and boost productivity, makes sure that users receive pertinent and timely updates without requiring continual manual oversight. The system is a strong option for individualized information management in a dynamic computing environment, according to the authors, who also highlight its scalability, adaptability, and privacy protections.

3. PROBLEM STATEMENT

Users in today's digital environment suffer from "alert fatigue" as a result of receiving too many unprioritized messages, which causes them to miss important alerts and result in ineffective workflows. Existing alert systems lack relevance-based prioritization and are based on static rules that are not responsive to user activity.

Managing large email volumes while reducing security threats is another challenge for organizations. Although email is still a vital tool for communication, it is also a major entry point for cyberthreats like phishing links, which can result in data breaches and interruptions to business operations. Due to its lack of real-time reactivity, traditional email monitoring increases risk exposure and delays threat detection.

Furthermore, personally replying to a high volume of emails takes time and is unreliable, taking focus away from important work. An intelligent, real-time email monitoring and alert system that reduces alert overload through adaptive notifications and mitigates security threats is required.

An AI-powered email bot that makes context-aware response recommendations can increase response efficiency, boost security, and streamline communication. An email environment that is safer and more productive would result from addressing these issues.

4. PROPOSED SYSTEM

Overview: In organizational settings, the suggested web-based email alert and monitoring system improves security and simplifies communication. By combining artificial intelligence (AI) and machine learning, it can identify phishing links, instantly scan email content, eliminate low-level threats, and raise serious dangers. By prioritizing warnings according to user responsibilities and preferences, its adaptive notification system lessens alert fatigue. An efficient answering bot also makes context-aware response recommendations. Constructed using Flask, MongoDB, and Python, the system provides cross-platform compatibility and an intuitive user interface. Future improvements will guarantee strong security and effective email handling, including multilingual support and sophisticated machine learning techniques.

Key Features:

1.Email alert and monitoring system using the web app.

2.We are an automated replaying bot according to our instructions.

3.If there are any harmful links the model will detect it.

Module Description:

1.User Authentication & Database Module:

- Secure authentication mechanism for user access.
- Stores user credentials, email data, and alert settings.

2.Email Processing Module:

- Connects to Gmail for accessing emails.
- Retrieves incoming emails in real time.
- Categorizes emails based on priority and relevance.

3. Automated Replying Bot Module:

- Activates AI-based auto-reply functionality.
- Processes email content to generate appropriate responses.
- Sends context-aware replies based on NLP analysis.

4.Suspicious Link Detection Module:

- Scans emails for phishing and malicious links.

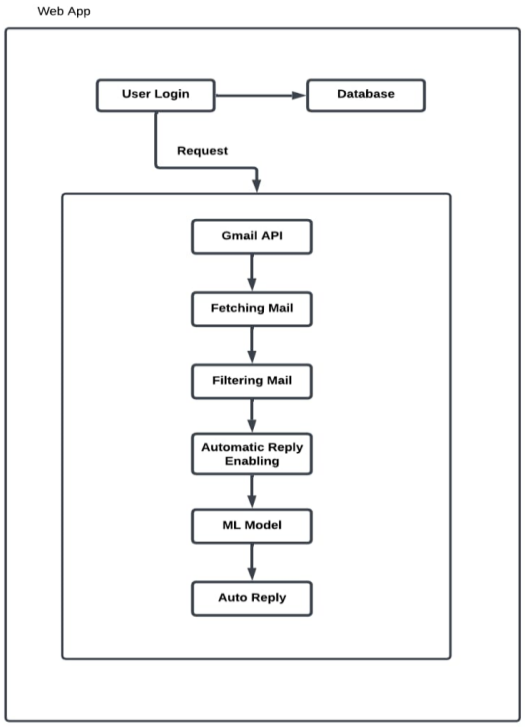


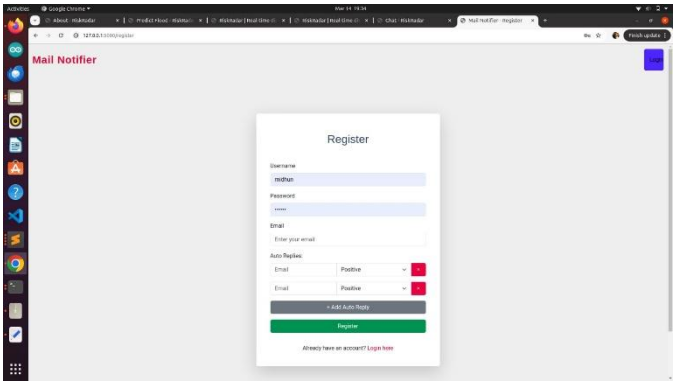
Fig 1: Block Diagram

5.RESULT AND DISCUSSION

To improve cybersecurity and risk assessment, the created system incorporates intelligent alert mechanisms, real-time suspicious link detection, and automatic email monitoring. With an auto-reply feature that guarantees timely responses and lessens manual labor, the email monitoring module classifies incoming emails according to predefined rules, spam identification, and phishing analysis. By identifying dangerous links, it stops data breaches and phishing attempts. To stop security breaches, the suspicious link detection module uses machine learning models and sophisticated URL analysis

to identify and flag potentially harmful URLs. Users are notified via email and a web dashboard. Real-time email analysis, automated responses, security alerts, and an intuitive dashboard with email records, security alerts, and risk forecasts are all features of the web-based monitoring system. To improve security, users can set up security settings and email filtering rules. The technology enhances online safety, streamlines email management, and raises cybersecurity awareness by combining these features. Multilingual assistance, AI-driven threat intelligence, and enlarged risk categories are possible future improvements.

SCREENSHOTS



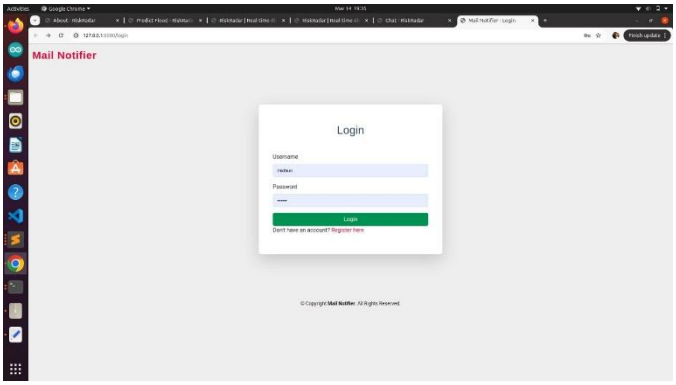


Fig 2: Login Page

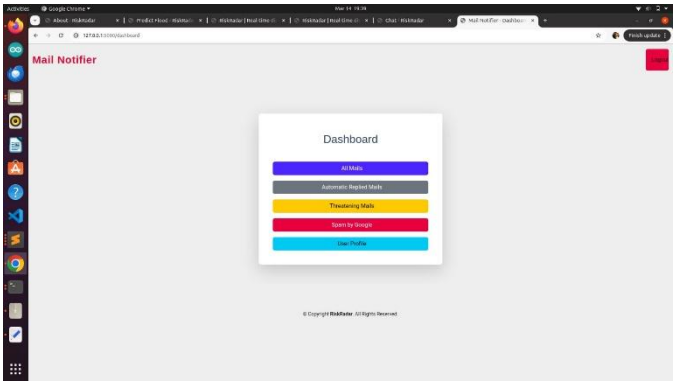


Fig 3: Dashboard

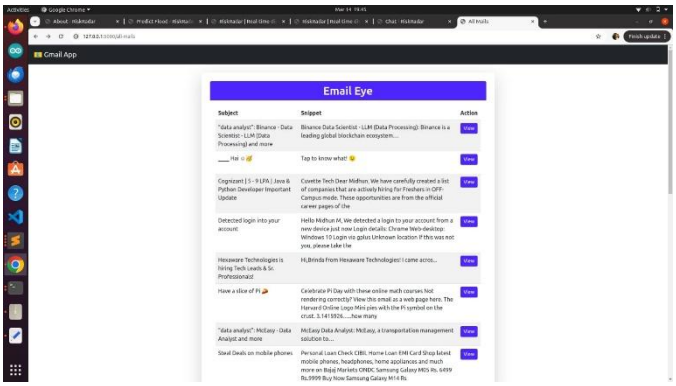


Fig 4: All Mails Page

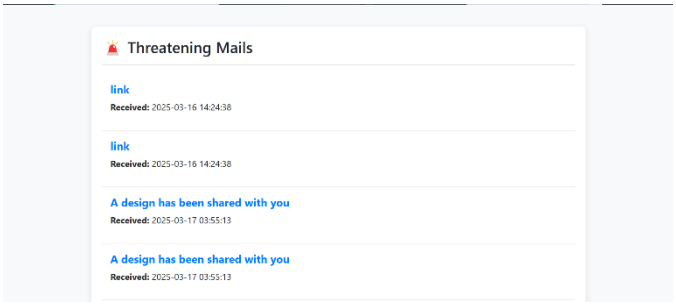


Fig 5: Threating Mails Page

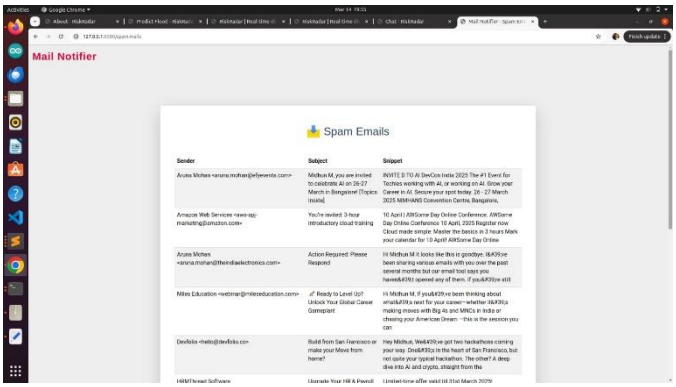


Fig 6: Spam Emails Page

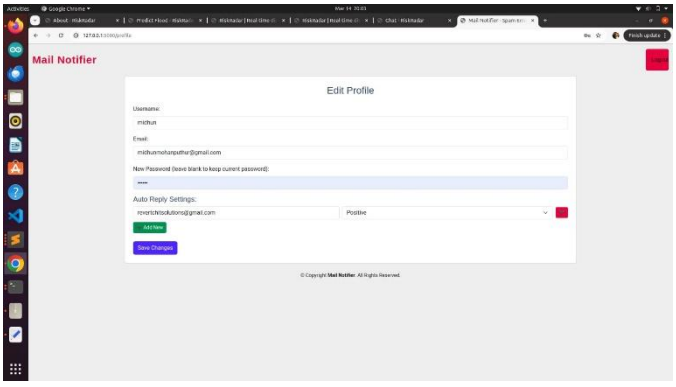


Fig 7: Edit Profile Page

6.CONCLUSION

To sum up, the suggested web-based email alert and monitoring system is a major step forward for email security and communication effectiveness. By including clever features like real-time hazardous link detection, configurable alert notifications, the solution helps users manage their email correspondence while protecting against potential risks by providing detection and an automated response bot. Artificial intelligence and machine learning greatly improve user experience and productivity by enabling context-aware answers and adaptive alert prioritizing. Additionally, the system's ability to run on both Windows and Ubuntu, as well as its strong support for MongoDB databases, guarantees a smooth deployment in a variety of settings. With businesses depending more and more on digital communication, this technology not only reduces the risk of cyberattacks but also optimizes processes so that users can concentrate on critical tasks. In the future, the system has a lot of promise. Expecting upcoming improvements that will further cement its position as an essential tool in contemporary email management, such as cross-platform integration, multilingual support, and predictive analytics. In the end, this solution has the potential to empower people, promote secure communication, and adjust to the always shifting cybersecurity issues, making it a priceless tool for businesses in the current digital era.

REFERENCES

[1]Abinaya, V., & Shobika, J. (2023).Email alerts on WhatsApp.EPRAInternationalJournalofResearch&Development(IJRD),8(5). https://eprajournals.com/IJSR/article/10537

[2]Paluri, S. C. V., & Nag, S. K. (2022). Development of email notification system based on user criteria. International Research Journal of Engineering and Technology,9(7),2393-2396. https://www.irjet.net/archives/V9/i7/IRJET-V9I7439.pdf

[3]Shishodia, M., Pal, S., & Shyamsundar. (2022). Email Alerts on Whatsapp. International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), 2(2), 112 . doi: 10.48175/568.

[4]Bazinette, Vincent & Cohen, Norman & Ebling, Maria & Hunt, Guerney & Lei, Hui & Purakayastha, Apratim & Stewart, Gregory & Wong , Luke & Yeh, Danny.(2001)An intelligent notificationsystem.RCComputerScience.22089. https://www.researchate.net/publication/228970929_An_intelligent_notification_syste m

[5]Sakshi.K, Darshan.I, Maroof M, Sushant.J, & Kute, M.K. (2023). Email alerts on WhatsApp. EPRA International Journal of Research and Development (IJRD), 8(5). https://doi.org/10.36713/epra2016

[6]Lalitha, N., Neelima, N., Sravya, S., & Kumar, G. P. (2021). Email alerts on WhatsApp. Journal of Emerging Technologies and Innovative Research, 8(7). Retrieved from https://www.jetir.org/papers/JETIR2107092.pdf

[7]Rector, K., & Hailpern, J. M. (2014). MinEMail: SMS alert system for managing critical emails. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 783-792). ACM. https://doi.org/10.1145/2556288.2557182

[8]Nanwin, D. N., & Williams, D. O. (2018). Development of an automated SMS alert mailing system. RIK International Journal of Science and Technology Research, 8(3), 110-115. Retrieved from https://www.researchgate.net

[9]T. P. Fowdur and L. Veerasoo, "An email application with active spoof monitoring and control," 2016 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2016, pp. 1-6, doi: 10.1109/ICCCI.2016.7480002.

[10]S. K. a. Subramaniam, S. H. b. Husin, Y. b. Yusop and A. H. b. Hamidon, "Real time mailbox alert system via SMS or email," 2007 Asia-Pacific Conference on Applied Electromagnetics, Melaka, Malaysia, 2007, pp. 1-4, doi: 10.1109/APACE.2007.4603963.

[11]S. Appavu, Muthu Pandian and R. Rajaram, "Association Rule Mining for Suspicious Email Detection: A Data Mining Approach," 2007 IEEE Intelligence and Security Informatics, New Brunswick, NJ, USA, 2007, pp. 316-323, doi: 10.1109/ISI.2007.379491.

[12] Survey Paper: Email Alert and Monitoring System Using a Web App Rashida C K, Reshma S, S Amritha, Shreya D Nair, Silja Varghese of Computer Science Department, Nehru College Of Engineering and Research Centre, Thrissur, India.
DOI: 10.55041/IJSREM42568