

• Email: editor@ijfmr.com

Automated Social Media and Email Posting Using Flask and API Integration

Dr. M. Raghava Naidu¹, M. Baidhyuth², N. Vamsi kesava³, O. Srinivasarao⁴, SK. Mahammad Hussain⁵

¹Assistant professor Dept. of CSE, Krishna University College of Engg. & Tech, AP, India ^{2,3,4,5}UG Student, Dept. of CSE, Krishna University College of Engg, &Tech, A.P, India

Abstract:

In today's fast-paced digital world, timely and scalable communication is crucial for both individuals and organizations. This project introduces a unified, web-based platform that automates the process of sending bulk emails and posting content to social media—built using Flask and integrated with various API s. The platform allows users to log in securely using email credentials or social media access tokens. Once authenticated, users can upload recipient lists and media files, write personalized messages, and send them out efficiently. Whether it's emails with attachments or social media posts with images and videos, the system ensures smooth and streamlined communication. The backend is powered by Flask, while the frontend uses HTML, CSS, and JavaScript for a responsive and user-friendly experience. Real-time feedback, delivery tracking, and error handling are built in, reducing the chances of manual errors and saving time. Social media integration supports platforms like Facebook, Instagram, and LinkedIn— allowing posts to be scheduled or published instantly using secure credentials such as page IDs or URN s. By automating routine communication tasks, this solution boosts productivity, ensures consistency, and gives users full control over their outreach. It's a powerful tool for marketing teams, organizational announcements, and professional networking—offering a secure, scalable, and cost-effective alternative to relying on third-party platforms.

Keywords: Automated Communication, Bulk Emailing, Social media Posting, Flask, API Integration, Email Automation, Content Scheduling, Real-time Feedback, Secure Authentication, Facebook API, Instagram API, LinkedIn API, Recipient List Upload, Media File Upload, Message Personalization, Delivery Tracking, Scalable Communication, Marketing Automation, Web-based Platform, Cost-effective Solution.

1. INTRODUCTION

The "Automated Email Posting System" project addresses the repetitive and time- consuming nature of manual email communication by providing a streamlined web-based platform for sending bulk emails with ease and efficiency. Designed to automate the email dispatch process, the system allows users to authenticate using their email and API credentials, select recipient lists from uploaded .txt files, compose messages, and attach files—all within a single, intuitive interface. Built to support individuals and organizations that require frequent email outreach, the platform automates message delivery and tracks successful transmissions, reducing human effort and error. A key feature of this system is its ability to



E-ISSN: 2582-2160 • Website: <u>www.ijfmr.com</u> • Email: editor@ijfmr.com

dynamically incorporate user-specific credentials, ensuring secure and flexible email posting without reliance on third-party services. The solution is developed using Flask for the backend and a responsive frontend stack including **HTML**, **CSS**, and **JavaScript**, delivering a seamless and interactive user experience. By automating the workflow and enhancing usability, the system enables users to focus on communication strategies while the platform handles distribution, making it an ideal tool for marketing, notifications, and professional outreach at scale.

Additionally, the project includes an **Automated Social-Media Posting Platform**. This platform enables users to upload and post images or videos directly to social media channels **like Facebook**, **Instagram**, and **LinkedIn**. Users authenticate using their respective **access tokens**, **page IDs** (for Facebook), and **URNs** (for LinkedIn) to securely and efficiently share content across multiple platforms from a single web interface. This component streamlines social media management, reduces manual effort, and ensures consistent online presence for individuals and organizations.

2. Problem Statement

Traditional methods of sending bulk emails and managing social media posts are often manual, timeconsuming, and error-prone, limiting the efficiency of communication and increasing the risk of mistakes such as missed recipients, incorrect attachments, or missed post schedules. These approaches lack automation, personalization, and scalability, making them unsuitable for users who need to deliver information quickly and reliably to large audiences or across multiple social media channels.

3. Background

As digital communication becomes integral to professional workflows, automated email systems have emerged as essential tools for managing large-scale outreach efficiently. Whether used for marketing campaigns, notifications, or organizational updates, these platforms enable users to send personalized emails in bulk without the delays and errors associated with manual methods. By allowing users to compose messages, upload recipient lists, and attach files through a centralized interface, automated email systems streamline communication processes while saving time and resources

4. Motivation of the Project

The increasing need for scalable and reliable communication solutions inspired the development of this system. Professionals and organizations often face challenges in managing large email distributions efficiently without depending on costly third-party services. The motivation is to build a secure, user-friendly platform that automates bulk emailing, ensuring time efficiency, accuracy, and control over the messaging process.

5. Literature Review

The evolution of digital communication has dramatically improved how individuals and organizations exchange information, especially in scenarios requiring mass outreach. Traditional email-sending methods, reliant on manual entry and basic client tools, were limited by scalability, personalization, and delivery tracking.

The Automated Email and Social-Media Posting System builds upon this transformation by offering a web-based platform that automates both email and social media content delivery using Python (Flask), SMTP, and third-party social media APIs (such as LinkedIn, Facebook, Instagram, and Twitter).



E-ISSN: 2582-2160 • Website: <u>www.ijfmr.com</u> • Email: editor@ijfmr.com

Early bulk email tools such as Microsoft Outlook groups and services like Mailchimp focused on simplifying message delivery but often faced issues of cost, data limits, and dependency on external APIs (Kim & Basu, 2019). Similarly, early social media automation tools like Buffer and Hootsuite helped schedule posts but imposed limits or required expensive subscriptions (Sharma & Patel, 2021). These tools lacked deep user-level customization, especially for smaller organizations.

The backend for email uses Python's **smtp lib** and **email. Mime** modules, proven reliable for secure and structured email communication (Patel & Vora, 2020). For social media posting, APIs offered by platforms like LinkedIn, Facebook, and Instagram are integrated using OAuth2 authentication, ensuring secure token management and authorized posting.

According to Singh et al. (2021), SMTP systems paired with SSL/TLS encryption provide a high level of security for transmitting data across networks, aligning with our email sending process. Similarly, API-based OAuth authentication (Zhang et al., 2020) enhances the security of social media credentials by using short-lived access tokens.

For file-based recipient loading in email, the system allows uploading .txt files. For social media, the system allows users to schedule posts with text, images, and descriptions uploaded via the web platform. User feedback during sending and posting is emphasized—both modules provide real-time progress updates on the frontend using dynamic JavaScript elements.

Security remains a cornerstone: no persistent storage of email passwords or social media tokens beyond session use. This transient model mirrors best practices highlighted by Zhang et al. (2020).

By combining proven techniques in SMTP communication, secure OAuth integration, and real-time feedback mechanisms, the system offers a scalable, customizable, and cost-effective alternative to expensive third-party tools for both email and social media outreach.

6. Existing System

Current systems for bulk messaging and social media posting typically rely on manual methods or thirdparty services.

Email Systems: Platforms like Microsoft Outlook and Mailchimp allow sending to multiple recipients but often require manual recipient management, lack real-time sending feedback, and charge for additional features.

Social Media Systems: Tools like Buffer, Hootsuite, and Later allow users to schedule posts across platforms but are subscription-based and impose limitations on the number of posts/accounts, making them less viable for smaller organizations or individuals.

7. Proposed System

The **Automated Email and Social Media Posting System** addresses these challenges by offering a unified, self-hosted solution for bulk communication across both email and major social media platforms.

- 1. Email Automation
- 2. Social media Automation
- 3. Security: SMTP login, OAuth token
- 4. Frontend: HTML, CSS, JAVA SCRIPT

Email Module:



E-ISSN: 2582-2160 • Website: <u>www.ijfmr.com</u>

• Email: editor@ijfmr.com

- 1. Upload .txt file containing email addresses.
- 2. Input subject, message, and attachments.
- 3. Validate emails.
- 4. Authenticate via user SMTP credentials.
- 5. Send emails individually.

Display real-time progress/status.

Social Media Module:

- 1. Authenticate user via OAuth to each platform.
- 2. Compose post (text, image, description).
- 3. Choose whether to post immediately or schedule.
- 4. Post using platform-specific API endpoints.

8. Use Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. those use cases



Figure: Use Case Diagram

Class Diagram

Classes:

User: user id, name, email, password

Register (), Login (), Logout

Social media account: Account id, Platform name Access Token

Connect account () disconnect account

Post: Post id, content, media URL, schedule time, status

Admin: Mange users (), manage post ()



Figure: Class Diagram

Sequence Diagram:

- 1. Login
- 2. Confirm login success
- 3. Create a new post
- 4. Connect social media account (OAuth)
- 5. Post content



- 6. Conformation
- 7. Show success messages



Fig: Sequence diagram

Data Flow Diagrams (DFDs)

- External Entities: User
- System: Automated Email & Social Media Posting
- Data Flows: Upload files, Input credentials, Compose posts/emails, Receive feedback
- Process 1: Email Management
- Upload .txt file \rightarrow Validate addresses \rightarrow Send emails \rightarrow Show status.
- Process 2: Social Media Management
- OAuth Authentication → Compose Post → Post to Platform → Confirm Status. Data Stores: Temporary session credentials, Uploaded files.



Figures: Data flow Architecture of the Automated social-media and multi email posting



E-ISSN: 2582-2160 • Website: <u>www.ijfmr.com</u>

• Email: editor@ijfmr.com

Activity Diagram:



Fig: Activity diagram

9. IMPLEMENTATION

Registration & Login System Registration & Account Activation

- User Registration Form:
- Collect user details: Email, Password, Full name, Phone number, etc.
- Store these details in the users table with an initial is active status set to FALSE.
- Admin Activation:
- Admins can view a list of pending users and approve or reject registrations.
- Once approved, the system:
- Updates is active to TRUE.
- Sets activated at with the current timestamp.
- Sends an email with an activation link.
- Account Activation Email:
- The email contains a link (e.g., http://example.com/activate/{activation_token}) to activate the account.
- On activation, users are directed to a success page and can proceed to login.

Login Process

- User Login:
- User provides email and password.
- System checks if:
- The email exists in the database.
- The account is active (is active = TRUE).
- Password is correct (compare hash values).
- Session Management:
- o Upon successful login, generate a session or JWT token for authentication.
- Optionally, log login metadata (timestamp, IP address) in login logs table for auditing.



Email Campaign Management

- Step 1: Select Email List
- File Upload:
- User uploads a .txt file containing email addresses (one per line).
- Validate the email addresses:
- Check for duplicate emails.
- Validate the email format (regular expression).
- Skip empty or malformed lines.
- Database Storage:
- Store valid email addresses in a recipients table with a status indicating whether it was successfully sent or skipped.
- Step 2: Compose Message
- 1.Email Composition:
- User enters the email subject and body.
- Support both plain text and HTML formats.
- Optional attachments: Files are uploaded and linked to the campaign.
- Step 3: Send Emails Individually
- Email Sending:
- Emails are sent individually to each recipient.
- Ensure each email is personalized and sent one-by-one to avoid spam or BCC leakage.
- Batch Process:
- The system sends emails in batches to optimize delivery, e.g., 50 emails per batch.
- Tracking and Logging:
- Log delivery results (success, failure) for each recipient in a delivery logs table.
- Automated Social Media Posting
- To enable automated social media posting, you can integrate APIs for popular platforms like Facebook, Twitter, or Instagram. Here's how:
- Post Composition:
- Users create a post (text, images, links).
- Optionally schedule the post for a future date/time.
- Integration with Social Media APIs:
- Use third-party libraries or APIs to post on social media (e.g., Twitter API, Facebook Graph API).
- Ensure the system authenticates using OAuth tokens for each platform.
- Scheduling:
- Allow users to schedule posts for a later time.
- Use Cron jobs or background workers to handle scheduling and publishing.
- Logging:
- Optionally log the post details, time of posting, and status (success/failure) for each platform in a social media posts table.
- Automated Multi-Email Posting



E-ISSN: 2582-2160 • Website: <u>www.ijfmr.com</u> • Email: editor@ijfmr.com

- To implement automated multi-email posting, you can adapt the existing campaign system to handle multiple campaigns simultaneously.
- Multiple Campaigns:
- Allow users to select multiple email lists and compose different messages for each list.
- o System should send emails in parallel, processing each list individually while maintaining
- o proper logs.
- Campaign Scheduler:
- Users can schedule different email campaigns to be sent at different times.
- Cron jobs or background workers can be used to automate this process.
- Email Batch Management:
- Support multiple email templates.
- Track each campaign separately, ensuring there's no overlap or duplication of recipient

10. OUT PUT & SCREEN SHOTS



Fig: Login page



Fig: Register page



Fig :Interface page



E-ISSN: 2582-2160 • Website: <u>www.ijfmr.com</u> • Email: editor@ijfmr.com



Fig: social media account

	ties these		
	and an other state of the state		
	State of Sta		
1		10	

Fig Email posting



Fig: welcome page

11. CONCLUSION

In today's fast-paced digital era, timely and efficient communication plays a crucial role in both professional and personal environments. With the increasing need for mass communication, especially in corporate, academic, and marketing sectors, automation has become essential for enhancing productivity and minimizing manual effort. To address these needs, two independent systems were developed: the Automated Email Posting System and the Automated Social-Media Posting System, each focusing on specific communication channels. The Automated Email Posting System simplifies the traditional process of sending emails by allowing users to register and log in securely, upload recipient email lists via text files, compose personalized messages with attachments, and dispatch emails with real-time feedback. Meanwhile, the Automated Social-Media Posting System streamlines the scheduling and publishing of posts across LinkedIn (using URN and Access Token), Facebook (using Page ID and Access Token), and Instagram (using Username and Password). It also uses Flask for backend operations, integrating secure API-based authentication to protect user credentials while allowing users to manage posts efficiently through an intuitive web interface. Both systems maintain clean, responsive designs and emphasize strong



security practices like transient credential handling and organized file management. Together, the Automated Email Posting System and the Automated social media Posting System offer a comprehensive, scalable, and robust solution for modern digital communication, making them valuable tools for organizations and individuals aiming to enhance outreach through separate, specialized platforms.

12. Reference

Email Automation & SMTP Handling:

- 1. Python Software Foundation. (2024). *Smtp lib SMTP protocol client*. Retrieved from <u>https://docs.python.org/3/library/smtplib.html</u>
- 2. Real Python Team. (2024). *Sending Emails with Python*. Retrieved from <u>https://</u>realpython.com/python-send-email/
- 3. D'Souza, R. (2024). *Automating Emails Using Python and Gmail*. Towards Data Science. Retrieved from <u>https://towardsdatascience.com/automate-email-sending-with-python-using-</u> <u>gmail-api-8421c8dba6f3</u>
- 4. Geeks for Geeks. (2024). *Send Mail Using Flask*. Retrieved from <u>https://</u> www.geeksforgeeks.org/send-mail-using-flask/

Web Development (Flask, HTML, CSS, JS):

- 5. Flask Documentation Team. (2025). *Flask Documentation*. Retrieved from <u>https://</u><u>flask.palletsprojects.com/</u>
- 6. W3Schools. (2025). *Flask Tutorial*. Retrieved from <u>https://www.w3schools.com/python/</u> <u>python_flask.asp</u>
- 7. Mozilla Developer Network (MDN). (2025). *HTML Documentation*. Retrieved from <u>https://developer.mozilla.org/En-US/docs/Web/HTML</u>
- 8. Mozilla Developer Network (MDN). (2025). CSS Documentation. Retrieved from <u>https://</u> developer.mozilla.org/En-US/docs/Web/CSS
- 9. Mozilla Developer Network (MDN). (2025). *JavaScript Guide*. Retrieved from <u>https://developer.mozilla.org/En-US/docs/Web/JavaScript/Guide</u>

File Handling & Attachments:

- 10. Flask Documentation Team. (2025). *File Upload Patterns in Flask*. Retrieved from <u>https://</u><u>flask.palletsprojects.com/En/2.2.x/patterns/file uploads/</u>
- 11. Geeks for Geeks. (2024). *File Uploading in Python Flask*. Retrieved from <u>https://www.geeksforgeeks.org/file-uploading-in-python-flask/</u>

Security & Session Management:

- Bhandari, A., & Singh, D. (2019). Secured Login Using OTP and Captcha. ResearchGate. R e t r i e v e d f r o m <u>h t t p s : //www.researchgate.net/publication/</u> <u>335943545</u> Secured Login Using OTP and Captcha
- 2. Flask Documentation Team. (2025). *Session Management in Flask*. Retrieved from <u>https://flask.palletsprojects.com/en/2.2.x/quickstart/#sessions</u>
- 3. OWASP Foundation. (2025). *Web Application Security Best Practices*. Retrieved from https://owasp.org/www-project-top-ten/

Web Technologies & System Performance:

4. Flask Framework Team. (2025). Flask Documentation (Latest). Retrieved from https://



E-ISSN: 2582-2160 • Website: <u>www.ijfmr.com</u> • Email: editor@ijfmr.com

flask.palletsprojects.com/en/latest/

- 5. Flask-Mail Team. (2025). *Flask-Mail Documentation*. Retrieved from <u>https://flask-mail.readthedocs.io/en/latest/</u>
- 6. Mozilla Developer Network (MDN). (2025). *Front-End Web Development (HTML, CSS, JavaScript)*. Retrieved from <u>https://developer.mozilla.org/</u>
- 7. Bootstrap Team. (2025). Bootstrap Documentation. Retrieved from https://getbootstrap.com/ docs/
- 8. SQLite Consortium. (2025). *SQLite Official Documentation*. Retrieved from <u>https://sqlite.org/docs.html</u>
- 9. Python Software Foundation. (2024). *sqlite3 Module Documentation*. Retrieved from <u>https://docs.python.org/3/library/sqlite3.html</u>

Fault-Tolerant Systems (Optional Advanced Reference):

10. Lee, H., & Park, S. (2021). *Integrated Fault Reduction Scheduling (IFRS) for Reliable Systems*. IEEE Xplore. Retrieved from https://ieeexplore.ieee.org/document/9631096

Additional Email Automation Tutorials:

- 11. Malek, P., & Djuric, I. (2024). Flask Send Email: Tutorial with Code Snippets. Mail trap Blog.
- 12. Gigous, B. (2024). *How to Send Emails with Python + Flask-Mail*. Coding Nomads.
- 13. Mishra, J. (2022). *Email-Automation-using-Flask*. GitHub Repository. Retrieved from https://github.com/
- 14. Odyntsov, Y. (2024). Flask Send Email Gmail: Tutorial with Code Snippets. Mail trap Blog.
- 15. Tennyson, M. (2021). *How to Send Emails While Using Python Flask Fully Asynchronously*. DEV Community.
- Social Media Automation (LinkedIn, Facebook APIs):
- 16. LinkedIn Corporation. (2025). *LinkedIn Marketing Developer Documentation*. Retrieved from https://learn.microsoft.com/en-us/linkedin/marketing/
- 17. Facebook Developers. (2025). *Graph API Documentation*. Retrieved from <u>https://</u><u>developers.facebook.com/docs/graph-api/</u>
- 18. Instagram Graph API Team. (2025). Instagram Graph API Documentation. Retrieved from https://developers.facebook.com/docs/instagram-api
- 19. Postman. (2025). Using APIs for Automation. Retrieved from <u>https://learning.postman.com/</u> <u>docs/getting-started/introduction/</u>
- 20. Geeks for Geeks. (2024). *OAuth 2.0 Authorization Explained Simply*. Retrieved from <u>https://www.geeksforgeeks.org/oauth-2-0-authorization-code-flow/</u>