International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 179-182, May-2026

https://doi.org/10.47001/IRJIET/2026.105023

Smart Gmail Backup and Analytics System Using Gmail API

[1]Ketan Dilip Pazare, [2]Shreyas Dethe, [3]Rohan Borkar, [4]Kiran Jivtode, [5]Ravi Gorgate

[1,2,3,4]Student, Computer Science and Engineering, Shri Sai College of Engineering and Technology, DBATU University, Bhadrawati, Chandrapur, Maharashtra, India

[5]Assistant Professor, Computer Science and Engineering, Shri Sai College of Engineering and Technology, DBATU University, Bhadrawati, Chandrapur, Maharashtra, India

***Abstract* - Email communication has become an essential part of modern digital life for both personal and professional activities. Gmail is one of the most widely used email platforms and stores large amounts of sensitive information such as documents, images, financial records, and communication data. However, users often face issues including accidental deletion of emails, lack of offline access, dependency on cloud storage, and difficulty in managing large email datasets.**

This research paper proposes a Smart Gmail Backup and Analytics System using Gmail API that provides an automated, secure, and efficient solution for email backup and analysis. The system is developed using Python and PyQt5 with Gmail API integration for secure authentication and email retrieval. The proposed system enables users to backup emails and attachments to local storage, perform date-based filtering, automate scheduled backups, and generate analytical insights from email activity.

The system improves accessibility, enhances data security, and provides meaningful analytics related to email frequency, sender statistics, and attachment usage. Experimental testing shows that the system performs reliable backup operations with high accuracy and efficient processing speed. The proposed system offers a cost-effective and user friendly solution for secure Gmail data management and analytics.

Keywords: Gmail API, Email Backup, Python, Data Analytics, Cloud Storage, Automation, Email Management.

I. INTRODUCTION

In today's digital world, email has become one of the most important communication tools used in educational institutions, businesses, government organizations, and personal communication. Gmail is among the most popular email platforms because of its reliability, cloud storage support, and integration with various Google services.

Despite these advantages, users still face several challenges while managing email data. Accidental deletion of important emails, dependence on internet connectivity, limited offline access, storage management issues, and lack of advanced analytics are some common problems. Gmail provides cloud based storage but does not offer flexible local backup and detailed analytical features for understanding user email behaviour.

To solve these problems, this research proposes a Smart Gmail Backup and Analytics System using Gmail API. The system provides a secure and automated mechanism to backup emails and attachments into local storage while also generating analytical reports based on email activities.

The proposed system uses Gmail API for secure communication and OAuth 2.0 authentication for authorized access. The application is implemented using Python programming language and PyQt5 graphical user interface framework.

The main objectives of the proposed system are:

- To provide secure backup of Gmail emails and attachments.
- To allow offline access to important emails.
- To automate the backup process.
- To generate useful analytics related to email activities.
- To improve data accessibility and security.

II. RELATED WORK

A. Email Backup Systems

Several email backup systems have been developed to provide local storage support for cloud-based emails. Traditional backup systems mainly focus on downloading emails from servers and storing them locally. However, many of these systems lack automation, filtering options, and advanced user interfaces.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 179-182, May-2026

https://doi.org/10.47001/IRJIET/2026.105023

B. Gmail API-Based Applications

Google provides Gmail API for secure access to Gmail accounts. Many applications use Gmail API for sending, reading, and managing emails. Existing applications mainly focus on synchronization and email management but provide limited analytical capabilities.

C. Email Analytics Systems

Modern email analytics systems are designed to analyze communication patterns, sender behavior, and attachment usage. However, most available tools are enterprise-level applications and are not suitable for individual users or students due to high complexity and cost.

D. Limitations of Existing Systems

Existing systems generally suffer from the following limitations:

- Limited local backup support.
- Lack of intelligent analytics.
- Complex user interfaces.
- High dependency on internet connectivity.
- Limited attachment management features.

The proposed system overcomes these limitations by combining secure backup functionality with real-time analytics and a user-friendly interface.

III. PROPOSED SYSTEM

The proposed Smart Gmail Backup and Analytics System is designed to provide automated email backup and analytical reporting using Gmail API.

The system performs the following operations:

1. Authenticates users securely using OAuth 2.0.
2. Connects with Gmail API.
3. Fetches email data and attachments.
4. Stores data in local storage.
5. Generates analytical reports.
6. Displays insights through graphical dashboard.

The system provides both manual and automatic backup functionality

IV. SYSTEM ARCHITECTURE

The proposed system follows a three-layer architecture.

A. User Interface Layer

The user interface is developed using PyQt5 and provides interactive screens for login, backup configuration, analytics dashboard, and report generation.

Main features include:

- Gmail login interface
- Backup configuration panel
- Analytics dashboard
- Report visualization
- File management interface

B. Application Layer

The application layer processes user requests and handles all business logic.

Functions include:

- Email processing
- Attachment extraction
- Data filtering
- Backup scheduling
- Analytics generation

C. Database and Storage Layer

This layer stores emails, attachments, and metadata in structured format using local storage and SQLite database.

Functions include:

- Email storage
- Attachment management
- Data retrieval
- Backup history maintenance

V. SYSTEM MODULES

A. Authentication Module

- Secure login using Gmail OAuth
- Ensures authorized access

B. Email Backup Module

- Automatically fetches emails
- Stores emails locally or in database
- Supports attachments

C. Analytics Module

- Email frequency analysis
- Sender statistics
- Attachment analysis

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 179-182, May-2026

https://doi.org/10.47001/IRJIET/2026.105023

D. Storage Module

- Saves emails in structured format
- Ensures data safety and retrieval

VI. TECHNOLOGIES USED

Technology	Purpose
Python	Backend development
Gmail API	Email access and retrieval
PyQt5	Graphical User Interface
OAuth 2.0	Secure authentication
SQLite	Local database storage
JSON	Data formatting

VII. WORKING METHODOLOGY

The working process of the proposed system is explained below:

1. The user logs into the system using Gmail authentication.
2. OAuth 2.0 verifies user identity securely.
3. Gmail API fetches emails and attachments.
4. Selected emails are downloaded into local storage.
5. Metadata is processed for analytics.
6. Analytical reports are generated and displayed.
7. Scheduled backup operations are executed automatically.

The system ensures secure communication and efficient processing during all operations.

VIII. RESULTS AND DISCUSSION

The proposed system was tested using multiple Gmail accounts with different email volumes.

Performance Results

Metric	Result
Total Emails Tested	500+
Average Backup Time	42 Seconds
Attachment Retrieval Accuracy	98% Accurate
Analytics Processing Speed	Efficient
User Satisfaction	Positive Feedback

IX. OBSERVATIONS

- The system successfully backed up emails and attachments.
- Date-based filtering reduced unnecessary backup size.
- Analytics dashboard provided meaningful insights.
- OAuth authentication improved account security.
- Backup scheduling reduced manual effort.

The experimental results demonstrate that the proposed system provides reliable performance and efficient email management.

X. ADVANTAGES OF THE PROPOSED SYSTEM

- Secure Gmail authentication.
- Offline access to emails.
- Automated backup scheduling.
- Real-time analytics dashboard.
- Easy attachment management.
- User-friendly graphical interface.
- Improved data safety and accessibility.

XI. LIMITATIONS

- Requires internet connection during synchronization.
- Backup speed depends on email volume.
- Gmail API request limits may affect large-scale operations.

XII. FUTURE SCOPE

The future improvements planned for the system include:

- AI-based email categorization.
- Spam detection system.
- Mobile application support.
- Cloud synchronization.
- Machine learning-based analytics.
- Multi-email platform integration.

XIII. CONCLUSION

This research paper presented a Smart Gmail Backup and Analytics System using Gmail API for secure and automated email backup. The proposed system provides efficient email management by combining backup functionality with analytical reporting.

The system successfully performs secure Gmail authentication, retrieves emails and attachments, stores data locally, and generates useful analytics related to email activity. The implementation using Python and Gmail API ensures flexibility, reliability, and scalability.

The proposed solution improves accessibility, enhances data security, and reduces the risk of data loss. Experimental results indicate that the system performs efficiently with high data accuracy and user satisfaction.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our project guide and faculty members of the Department of Computer Science and Engineering, Shri Sai College of

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 179-182, May-2026
https://doi.org/10.47001/IRJIET/2026.105023

Engineering and Technology, for their valuable guidance, encouragement, and support throughout this research work. We also thank our institution for providing the facilities and resources required for successful completion of this project.

CREDIT AUTHORSHIP CONTRIBUTION STATEMENT

Ketan Dilip Pazare: System development, Gmail API integration, backend implementation, writing original draft.

Shreyas Dethe: User interface development, testing, analytics module implementation.

Rohan Borkar: Database management, backup module development, documentation.

Kiran Jivtode: System testing, data analysis, graphical report generation.

Asst. Prof. Ravi Gorgate: Supervision, methodology validation, review and editing.

DATA AVAILABILITY STATEMENT

The datasets generated and analysed during the current study are available from the corresponding author upon reasonable request.

DECLARATION OF COMPETING INTEREST

The authors declare no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

REFERENCES

[1] Google Developers, "Gmail API Documentation," Google LLC, 2025. [Online]. Available: https://developers.google.com/gmail/api

[2] Python Software Foundation, "Python Programming Language," 2025. [Online]. Available: https://www.python.org/

[3] Riverbank Computing, "PyQt5 Documentation," 2025. [Online]. Available: https://www.riverbankcomputing.com/software/pyqt/

[4] Google Developers, "OAuth 2.0 Authentication," Google LLC, 2025. [Online]. Available: https://developers.google.com/identity/protocols/oauth2

[5] SQLite Documentation, "SQLite Database Engine," 2025. [Online]. Available: https://www.sqlite.org/index.html

[6] R. Elmasri and S. Navathe, Fundamentals of Database Systems, 7th ed., Pearson Education, 2016.

[7] Ian Sommerville, Software Engineering, 10th ed., Pearson Education, 2015.

[8] M. Lutz, Learning Python, 5th ed., O'Reilly Media, 2013.

[9] A.Silberschatz, P. B. Galvin, and G. Gagne, Operating System Concepts, 10th ed., Wiley, 2018.

[10] A.Tanenbaum and D. Wetherall, Computer Networks, 5th ed., Pearson Education, 2013.

[11] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed., Pearson, 2021.

[12] Google Workspace Developers, "Best Practices for Gmail API," Google LLC, 2025. [Online]. Available: https://developers.google.com/workspace/gmail/api/guides

[13] OAuth 2.0 Industry Protocol, "Authorization Framework Standard," 2025. [Online]. Available: https://oauth.net/2/

[14] PyPI Community, "Google API Python Client Library," 2025. [Online]. Available: https://pypi.org/project/googleapi-python-client/

[15] Microsoft Documentation, "Local Data Backup and Recovery Techniques," 2024. [Online]. Available: https://learn.microsoft.com/

Citation of this Article:

Ketan Dilip Pazare, Shreyas Dethe, Rohan Borkar, Kiran Jivtode, & Ravi Gorgate. (2026). Smart Gmail Backup and Analytics System Using Gmail API. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 10(5), 179-182. Article DOI https://doi.org/10.47001/IRJIET/2026.105023

****

