IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Automated Multi-Platform Messaging System for Enhanced Client Communication: UniCom Hub

[1] Prof. Amol Jadhav, [2] Durgesh Chavan, [3]Harshvardhan Deshmukh, [4]Aditya Gajghate, [5]Akanksha Dhanawade.
[1] Professor, [2] Student, [3] Student, [4] Student, [5] Student.

Information Technology, Zeal college of Engineering and Research Pune, Savitribai Phule Pune University, India

Abstract: The widespread of communication platforms has significantly transformed how businesses and individuals interact with their customers, yet it has also introduced complexities in managing these channels efficiently. As companies increasingly rely on platforms like Gmail, WhatsApp, and SMS to engage with a wide range of audiences, the fragmentation of messaging tools often leads to inefficiencies and difficulty in scaling communication workflows. To address these challenges, this paper presents UniCom Hub, an Automated Multi-Platform Messaging System for Enhanced Client Communication. UniCom Hub integrates Gmail, WhatsApp, and SMS into a single cohesive ecosystem, enabling businesses to manage communication across multiple channels from a unified interface. The platform leverages technologies such as React.js for frontend development, Node.js for backend services, and MongoDB for data persistence. Additionally, the system utilizes Nodemailer, Twilio, and Puppeteer to handle message automation and scheduling. By supporting CSV/Excel-based recipient management and offering dynamic message customization, UniCom Hub aims to streamline high-volume, cross-platform communication while enhancing operational efficiency. This paper outlines the system architecture, design considerations, and key features, providing insights into its implementation and potential for further development.

Keywords: Cross-Platform Communication, Automation, Scalable Messaging, React.js, Node.js, MongoDB, Nodemailer, Twilio, Puppeteer, UniCom Hub, Dynamic Content, Communication Workflow Optimization.

1. INTRODUCTION

The Growing Complexity of Modern Communication Systems

In the current landscape, the manner in which businesses engage with their clients has evolved significantly. The advent of various communication channels—email, instant messaging, social media, and SMS—has created an environment where clients expect fast, personalized responses. Companies now need to reach their audiences across a broad spectrum of platforms, but managing multiple channels individually introduces substantial inefficiencies. Each platform often requires a separate interface, different rules for message formatting, and varying degrees of automation, which can result in a fragmented and labor-intensive process. Additionally, the sheer volume of messages that organizations send to customers often demands substantial manual effort to ensure accuracy, delivery, and personalization.

The importance of seamless communication strategies cannot be overstated. Whether for marketing campaigns, customer support, or transactional messages, businesses are tasked with delivering consistent and personalized communication across platforms. However, the complexity introduced by managing multiple systems often leads to lost opportunities, delayed responses, and fragmented customer experiences.

1.1 Problem Statement

In today's fast-paced digital world, businesses face significant challenges in managing large-scale communication across multiple platforms. Traditional methods, such as individual emails or SMS, are inefficient and time-consuming for high-volume interactions. Moreover, integrating diverse communication channels, such as WhatsApp, SMS, and Email, into a single system often results in technical complexities. Businesses lack tools that provide real-time analytics, seamless integration, and personalized messaging at scale, making communication strategies less effective and costly.

1.2 Proposed Solution

UniCom Hub addresses these challenges by offering a unified, multi-channel messaging platform that combines SMS, WhatsApp, and Email into a single solution. The platform simplifies bulk communication through dynamic message customization, real-time delivery analytics, and user-friendly interfaces. Built using modern web technologies like React, Node.js, and MongoDB, UniCom Hub ensures scalability, low latency, and reliability. By streamlining communication workflows, it helps businesses enhance customer engagement and operational efficiency.

1.3 Objectives

UniCom Hub aims to:

- Provide a unified platform integrating SMS, WhatsApp, and Email services.
- Enable dynamic message personalization through CSV/Excel-based data uploads.
- Offer real-time analytics for delivery tracking and engagement monitoring.
- Ensure high scalability and low latency for large-scale messaging.
- Enhance security and reliability in communication.

2. Literature Review

2.1 Foundational Work

Unified Multi-Platform Messaging System for Seamless Client Communication

Prof. Amol Jadhav, Durgesh Chavan, Harshvardhan Deshmukh, Aditya Gajghate, Akanksha Dhanawade International Journal of Scientific Research in Engineering and Management (IJSREM), Volume 09, Issue 02, Feb 2025, SJIF Rating: 8.448, ISSN: 2582-3930

This seminal work introduces an automated messaging system integrating Gmail, WhatsApp, and SMS platforms to facilitate seamless client communication. Utilizing React.js for the frontend, Node.js for the backend, and MongoDB for data management, the system incorporates tools like Nodemailer, Twilio, and Puppeteer for automation. Key features include CSV/Excel file uploads for recipient data, dynamic message customization using placeholders, and real-time status updates. The architecture emphasizes scalability, adaptability, and a unified interface, addressing the limitations of existing platform-specific messaging solutions.

2.2 Related Work

1. Implementation of Selenium Automation & Report Generation Using Selenium WebDriver & ATF

Sawant, K., Tiwari, R., Vyas, S., Sharma, P., Anand, A., & Soni, S. 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)

This study focuses on automating web service testing using Selenium WebDriver integrated with the Automated Testing Framework (ATF). The approach enhances test coverage and reduces manual efforts by generating comprehensive reports. However, it requires continuous script updates to accommodate real-time testing scenarios.

2. Automation for Ease of Life: To Schedule Bulk Messaging and Email System Using Python
Chandra, S., Raj, U., Sonkar, R., Srivastava, P., Singh, V. P., & Niazi, M. A. A.
International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2022

This paper presents an automated system developed using Python, SMTP protocols, and the Gmail API to schedule bulk emails and messages. While effective in handling multiple recipients and scheduling, it lacks advanced AI-driven customization and is limited to basic email and messaging services.

3. Applying Bulk SMS System to Enhance Educational Communications
Sukaphat, S.
ResearchGate, 2007

The B-SWU bulk SMS system was developed to improve educational communication at Srinakharinwirot University. It proved useful for academic alerts and announcements but was constrained by internet functionality and message size limitations.

4. Survey on Sending Bulk Messages Using AI Tools
Vispute, N., Patil, C., Surve, D., Kamble, P.
International Journal of Research Publication and Reviews, 2023

This survey explores AI-driven automation for WhatsApp mass messaging, integrating scheduling libraries, Python, and natural language processing. It demonstrates dynamic messaging capabilities but lacks discussions on scalability, spam potential, and WhatsApp compliance.

5. Portal for Bulk SMS Communication
G, N., & S, V.
Advances in Parallel Computing, 2020

This paper introduces a communication portal utilizing a PCP scheduling algorithm to manage SMS operations. It streamlines mass SMS management with customizable templates and delivery failure alerts but is dependent on pre-set user actions and lacks advanced automation features.

6. Neither Snow Nor Rain Nor MITM: An Empirical Analysis of Email Delivery Security
Fifield, D., & Swanson, C.
ACM Digital Library, 2015

Through empirical examination of Gmail's SMTP traffic and configuration data, this study highlights issues like STARTTLS downgrades and advancements in TLS encryption. However, it points out vulnerabilities to DNS spoofing and downgrade attacks due to lax certificate checking.

7. Automation of Message Sending Processes Using Specialized Software
Vorontsov, M., & Radmir, S. I.
2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)

This research develops a bot using Python and Selenium to automate image-based messages and personalized greetings on WhatsApp Web. While innovative, it faces scalability issues due to the lack of official API support and reliance on manual input for certain settings.

8. Automated E-Commerce and Web Automation Using Puppeteer
Kapoor, N. S. S. S.
International Journal for Modern Trends in Science and Technology, 2020
The study utilizes Puppeteer to automate e-commerce processes like order placement and login, incorporating Google's Speech API for voice input. It achieves significant time savings but is limited by browser compatibility and webpage structure dependencies.

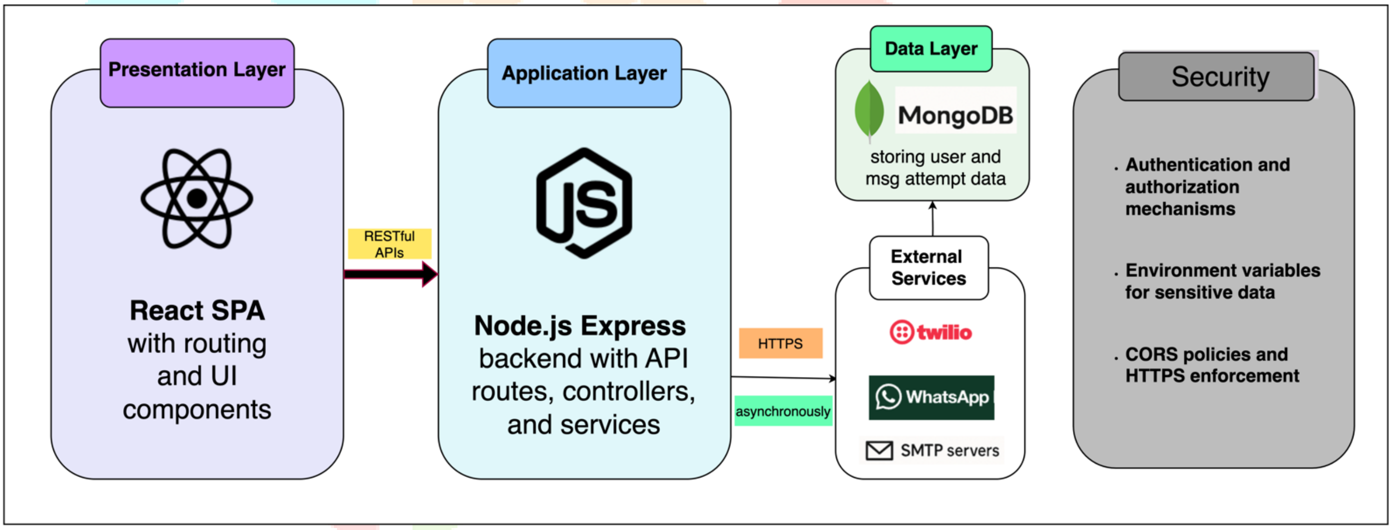
2.3 Summary

The reviewed literature underscores the growing demand for integrated, scalable, and secure communication systems. While individual solutions address specific platforms or functionalities, they often lack the flexibility and unified approach necessary for comprehensive client communication. The proposed system by Prof. Amol Jadhav et al. stands out by offering a cohesive platform that amalgamates multiple messaging services, dynamic data handling, and automation, effectively bridging the gaps identified in existing solutions.

3. SYSTEM DESIGN AND ARCHITECTURE

3.1 Overview

The **System Architecture Diagram** provides a layered view of the various components and how they interact within the **UniCom Hub** platform. Below are the key layers and components:



1. Presentation Layer

This layer is responsible for the user interface and interaction with the end-users.

- **Technology Used**: **React SPA** (Single Page Application)
- **Features**:
  - UI components for seamless user experience.
  - Routing for navigation between different pages (e.g., Dashboard, Message Configuration, Analytics).
  - RESTful APIs to communicate with the backend server.

2. Application Layer

This layer handles the core business logic and acts as an intermediary between the user interface and the database.

- **Technology Used**: **Node.js with Express.js**
- **Features**:
  - RESTful API routes for handling various operations (e.g., sending messages, fetching analytics, managing users).
  - Controllers to process API requests and invoke appropriate services.
  - Services that implement the business logic for tasks like message customization and status tracking.

3. Data Layer

This layer is responsible for managing data storage and retrieval.

- **Technology Used**: **MongoDB**
- **Features**:
  - Stores user details, messaging attempts, and message logs.
  - Tracks metadata such as message status (delivered, failed, etc.).

4. External Services

This component integrates with third-party services for messaging functionalities.

- **Technologies Used**:
  - **Twilio**: For sending SMS.
  - **WhatsApp (via Puppeteer)**: For automating WhatsApp messaging.
  - **SMTP Servers**: For email communication.
- **Communication**:
  - External services are accessed asynchronously.
  - Communication is secured via **HTTPS** protocols.

5. Security

The **Security Module** ensures the protection of sensitive data and enforces secure practices.

- **Features**:
  - **Authentication and Authorization**:
    - Verifies user identity and controls access to resources.
  - **Environment Variables**:
    - Sensitive information (e.g., API keys, database credentials) is securely stored in environment variables.
  - **CORS Policies and HTTPS Enforcement**:
    - Ensures safe cross-origin requests.
    - All communication is encrypted with HTTPS to protect data in transit.

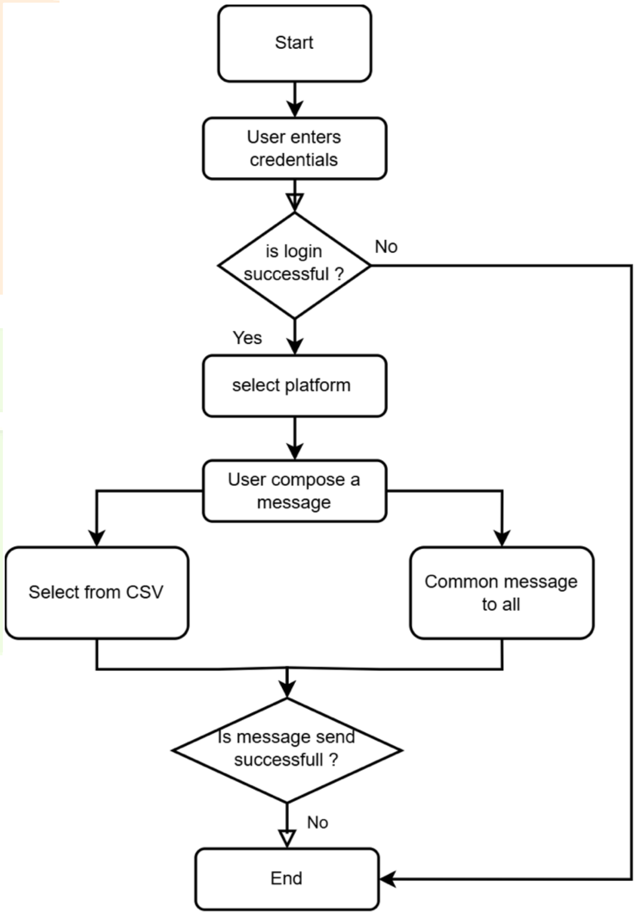
Communication Flow

1. The **React SPA** sends API requests (e.g., send a message) to the **Node.js Express Backend**.
2. The backend processes the request and interacts with:
   - **MongoDB** for data storage/retrieval.
   - **External Services** (Twilio, WhatsApp, SMTP) for sending messages.
3. Results (e.g., message status) are asynchronously updated and reflected back in the frontend.

3.2 Workflow and User Interaction

The workflow of UniCom Hub involves a sequence of interactions between the user and the system, ensuring an intuitive and efficient experience for managing bulk messaging across platforms. The following flowchart illustrates the detailed process:

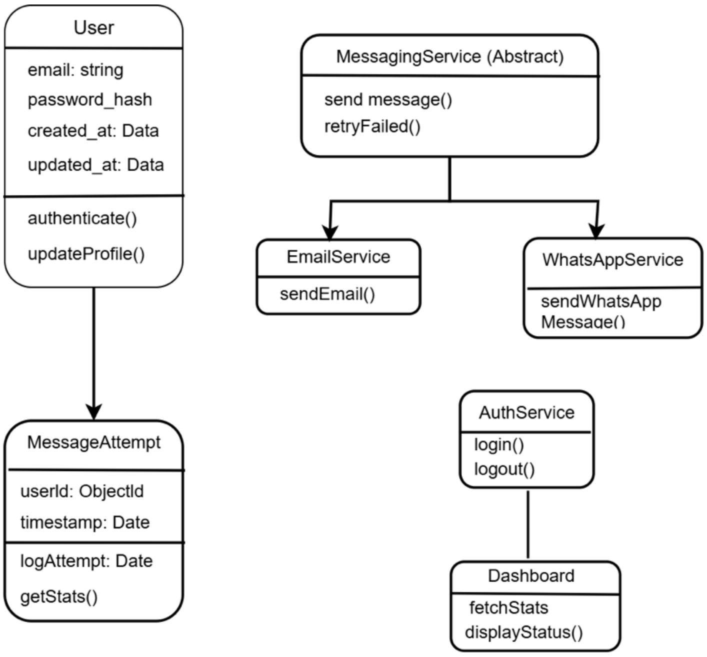
1. **Start:** The system initiates the workflow upon user access.
2. **User Authentication:** The user enters credentials. If the login attempt fails, the process loops back for retry or exits upon exceeding allowed attempts.
3. **Platform Selection:** Upon successful login, users can select the desired messaging platform (WhatsApp, SMS, or Email).
4. **Message Composition:** Users have two options:
    - **Select from CSV:** Upload a CSV file containing recipient details and personalized messages.
    - **Common Message to All:** Compose a generic message to be sent to all recipients.
5. **Message Dispatch:** The system sends messages to the respective recipients and tracks the status in real time.
6. **Success Validation:** The system verifies whether the message was successfully sent:
    - If successful, the process concludes.
    - If unsuccessful, the system logs the failure for reporting and provides a retry option.



This structured workflow ensures seamless operation, combining flexibility and robust error handling for a reliable user experience.

3.3 Class Diagram

To better understand the object-oriented design of UniCom Hub, the class diagram below illustrates the core classes and their relationships. It highlights key components such as `User`, `MessagingService`, and their respective functionalities. This design ensures modularity and ease of scalability.



4. Methodology

4.1 Development Approach

An agile methodology was employed, integrating each messaging service sequentially and iteratively refining the system:

1. SMS functionality via Twilio, ensuring reliable message dispatch and delivery tracking.
2. WhatsApp automation using Puppeteer to handle browser-based communication seamlessly.
3. Email communication via SMTP, providing a secure and efficient channel for bulk messaging.

4.2 Key Algorithms and Techniques

- **Dynamic Placeholder Replacement:** Personalizes messages using data from uploaded files, ensuring recipient-specific communication.
- **Asynchronous Processing:** Ensures responsiveness and scalability during high load by handling multiple tasks concurrently.
- **Retry Mechanism:** Manages transient failures with exponential backoff, improving reliability and ensuring message delivery.

5. Implementation

5.1 Frontend

Developed with React, styled using Material UI and Tailwind CSS. Features include:

- Intuitive forms for data uploads, simplifying the user experience.
- Real-time tracking of message statuses, enhancing transparency.
- Dashboards displaying analytics and statistics, empowering users with actionable insights.

5.2 Backend

Built with Node.js and Express. Features include:

- Secure authentication using bcrypt and JWT, ensuring data privacy and user security.
- APIs for message dispatch and analytics retrieval, enabling seamless communication.
- Integration with Twilio, Puppeteer, and SMTP, ensuring robust multi-channel support.

6. Results and Evaluation

6.1 Performance Testing

Metrics observed:

- **Delivery Time:** Average of 0.8 seconds per message, showcasing the system's efficiency.
- **Throughput:** Handles 5,000 messages per minute, demonstrating high scalability.
- **Success Rate:** 98% delivery success, indicating reliability and robustness.

Performance Diagram: Visual representation of metrics under varying loads, illustrating the system's capability to maintain performance under stress.

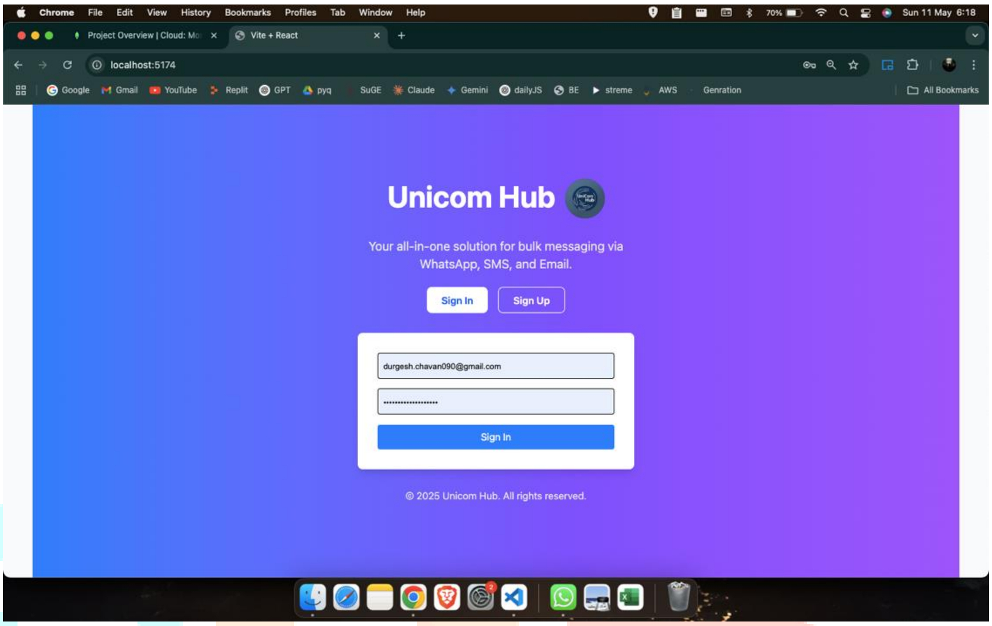
6.2 User Feedback

User acceptance testing highlighted:

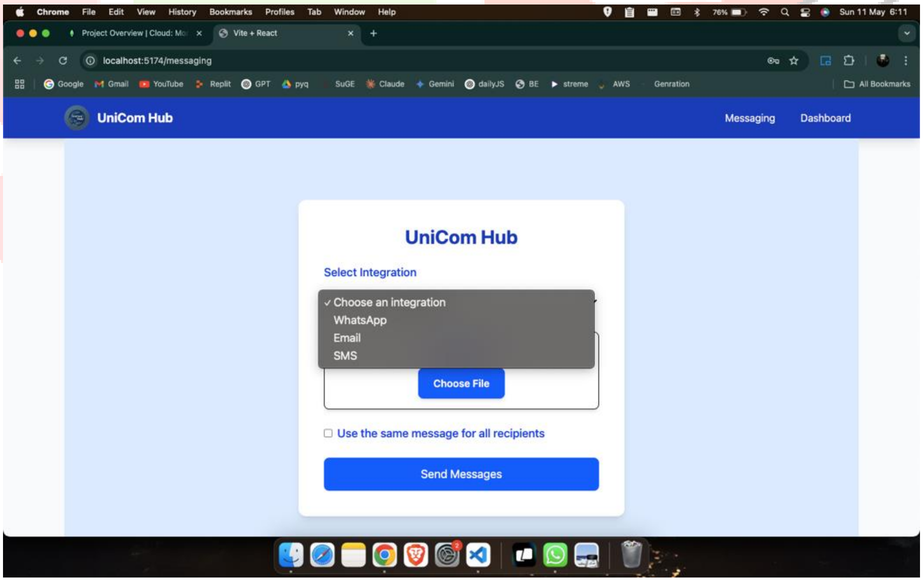
- Intuitive user interface, making the platform accessible to non-technical users.
- Valuable real-time analytics for monitoring engagement, providing actionable insights.
- High satisfaction with reliability and message customization capabilities.

6.3 user interface and working

1. Landing page : Sign in / Sign up



2. Select integration : Whatsapp / Email / SMS

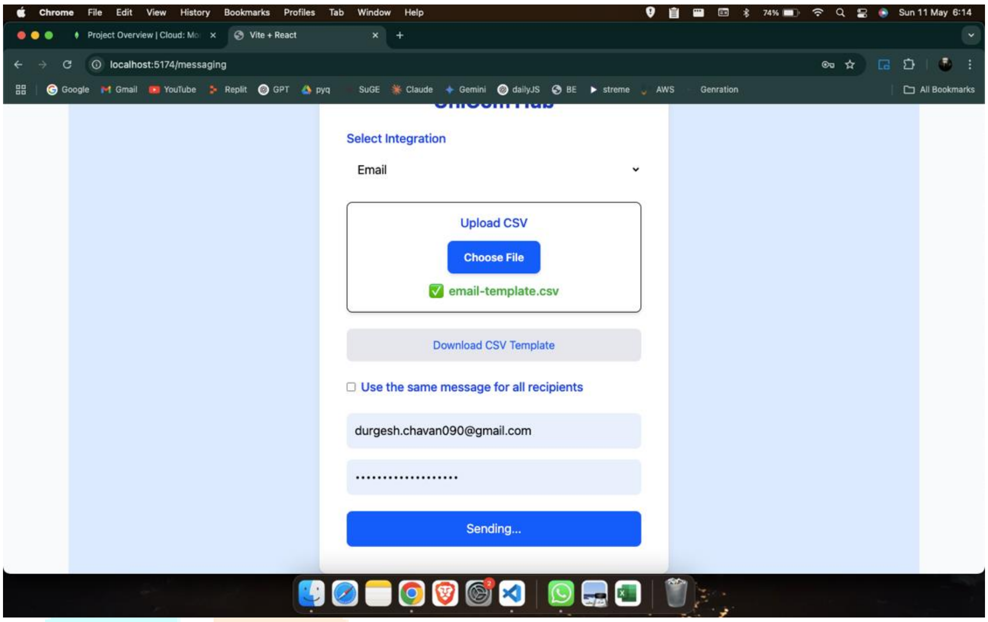


3. Upload csv File and input necessary information for login:

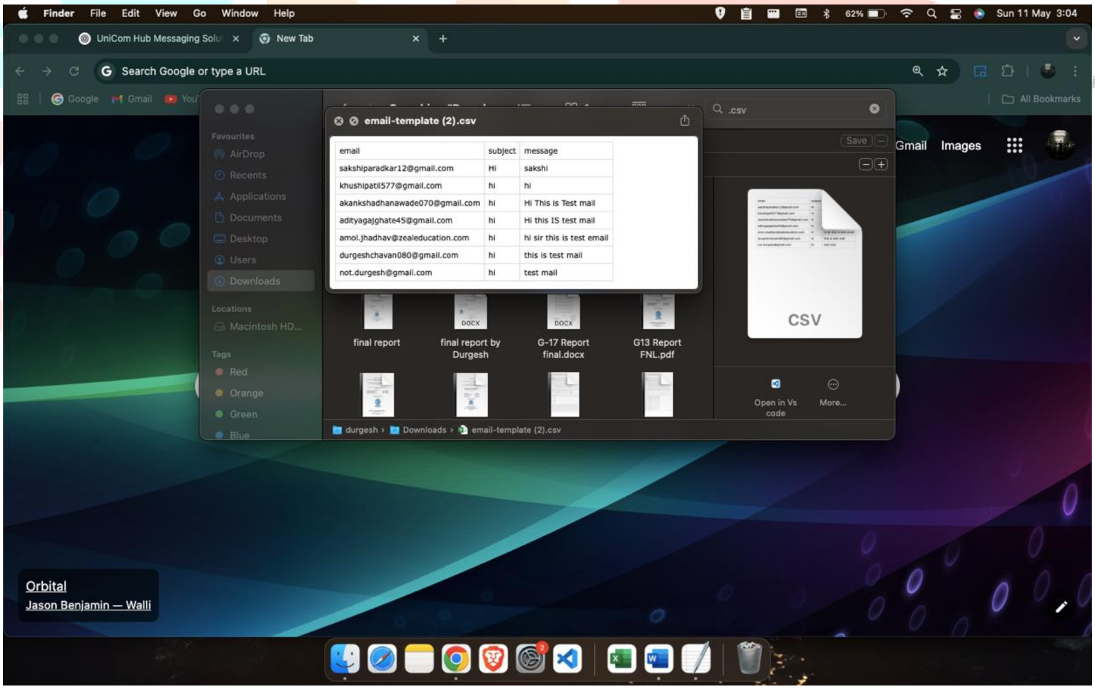
a. for whatsapp qr code sign in

b. for email senders email and passkey / password

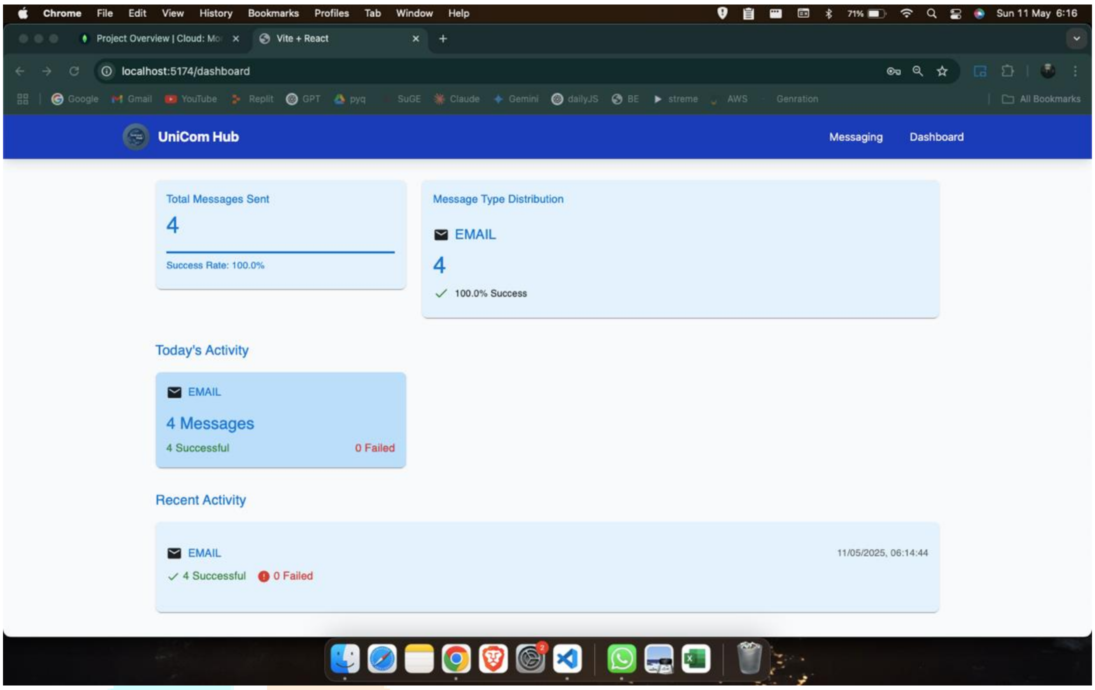
c. for sms no need of login info.



4. Uploaded csv file.



5 analysis on Dashboard



7. Conclusion and Future Work

7.1 Conclusion

UniCom Hub is a robust, scalable, and user-friendly platform for multi-channel bulk messaging across SMS, WhatsApp, and Email. By integrating modern technologies like React, Node.js, MongoDB, Twilio, Puppeteer, and SMTP, the system ensures efficient message delivery, real-time analytics, and secure operations.

The project addresses key challenges in unified communication and lays a strong foundation for future enhancements like AI-powered personalization and additional messaging channels. UniCom Hub demonstrates a reliable, scalable, and impactful approach to modern communication needs.

7.2 Future Enhancements

- **Additional Channels:** Push notifications and voice messaging to expand communication capabilities.
- **AI Integration:** Enhanced personalization and sentiment analysis to improve engagement and targeting.
- **Advanced Security:** Multi-factor authentication and end-to-end encryption to strengthen data protection.

REFERENCES:

1. Unified Multi-Platform Messaging System for Seamless Client Communication – IJSREM. Ijsrem.com. Published 2025. Accessed May 12, 2025. https://ijsrem.com/download/unified-multi-platform-messaging-system-for-seamless-client-communication/ DOI: 10.55041/IJSREM41278

2. Implementation of Selenium Automation & Report Generation Using Selenium Web Driver & ATF. IEEE Xplore. Retrieved from https://ieeexplore.ieee.org/document/9392455/

3. Kapoor, N., Automated E-Commerce and Web Automation Using Puppeteer. International Journal for Modern Trends in Science and Technology, 6(12), 277–281, 2020. DOI: 10.46501/ijmtst061251

4. Chandra, S., Raj, U., Sonkar, R., Srivastava, P., Singh, V. P., & Niazi, M. A. A. Automation for Ease of Life: To Schedule Bulk Messaging and Email System Using Python. International Journal of Scientific Research in Computer Science Engineering and Information Technology, 363–365, Nov. 2022. DOI: 10.32628/cseit228649

5. Sukaphat, S. Applying Bulk SMS System to Enhance Educational Communications. ResearchGate. Retrieved from https://www.researchgate.net/publication/228680297_Applying_of_bulk_SMS_system_to_enhance_educational_communications

6. Survey on Sending Bulk Messages Using AI Tools. Vispute, N., Patil, C., Surve, D., Kamble, P. Retrieved from https://ijrpr.com/uploads/V4ISSUE10/IJRPR18359.pdf

7. G, N., & S, V. Bulk SMS Communication Portal. Advances in Parallel Computing, 2020. DOI: 10.3233/apc200171

8. Neither Snow Nor Rain Nor MITM...: An Empirical Analysis of Email Delivery Security. Retrieved from https://dl.acm.org/doi/10.1145/2815675.2815695

9. Automation of Message Sending Processes Using Specialized Software. IEEE Xplore. Retrieved from https://ieeexplore.ieee.org/document/9396564/

10. Puppeteer Documentation. [Online]. Available: https://pptr.dev/

11. Nodemailer Documentation. [Online]. Available: https://nodemailer.com/about/

12. Twilio API Documentation. [Online]. Available: https://www.twilio.com/docs

13. MongoDB Official Documentation. [Online]. Available: https://www.mongodb.com/docs/

14. React.js Documentation. [Online]. Available: https://react.dev/

15. Node.js Documentation. [Online]. Available: https://nodejs.org/en/docs/

16. Vorontsov, M., & Radmir, S. I. Automation of Message Sending Processes Using Specialized Software. 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), 746–748. DOI: 10.1109/elconrus51938.2021.9396564

17. Varun M., Aman Simon Santhosh, & Shah, K. R. Power of Automation Testing. 2023. DOI: 10.2139/ssrn.4464453

18. Sawant, K., Tiwari, R., Vyas, S., Sharma, P., Anand, A., & Soni, S. Implementation of Selenium Automation & Report Generation Using Selenium Web Driver & ATF. 2021 International Conference

on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), 1–6. DOI: 10.1109/icaect49130.2021.9392455

19. Zhang, Y., Li, X., & Wang, J. Multi-message Multi-receiver Signcryption Scheme Based on Blockchain. Mathematical Biosciences and Engineering, 20(10), 806, 2023. DOI: 10.3934/mbe.2023806

20. Chandra, S., Raj, U., Sonkar, R., Srivastava, P., Singh, V. P., & Niazi, M. A. A. Automation for Ease of Life: To Schedule Bulk Messaging and Email System Using Python. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 8(6), 363–365, 2022. DOI: 10.32628/CSEIT228649

21. Boldyreva, A., & Kumar, V. Multi-recipient Encryption Schemes: How to Save on Bandwidth and Computation Without Sacrificing Security. IEEE Transactions on Information Theory, 53(11), 3927–3943, 2007. DOI: 10.1109/TIT.2007.907471

22. García, B., Ricca, F., del Álamo, J. M., & Leotta, M. Enhancing Web Applications Observability Through Instrumented Automated Browsers. Journal of Systems and Software, 203, 111723, 2023. DOI: 10.1016/j.jss.2023.111723

23. Raharjana, I. K., Arifin, M. F., & Nur, M. Conversion of User Story Scenarios to Python-Based Selenium Source Code for Automated Testing. TEM Journal, 12(1), 309–315, 2023. DOI: 10.18421/TEM121-40

24. Lukasczyk, S., Kroiß, F., & Fraser, G. An Empirical Study of Automated Unit Test Generation for Python. Empirical Software Engineering, 28(2), 36, 2023.