IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, May 2025
Impact Factor: 7.67

Automated Client Communication System Using WhatsApp Business API and MERN Stack Integration

Gayatri Ambatkar[1], Ashwini Mahajan[2], Komal Naxine[3]
Student, Computer Science & Engineering[1]
Professor, Computer Science & Engineering[2,3]
Tulsiramji Gaikwad-Patil College of Engineering and Technology, Nagpur, India

Abstract: *In today's fast-paced digital era, businesses require seamless, scalable, and efficient communication channels to engage with their clients. This paper presents the design and implementation of an automated client communication system that integrates the WhatsApp Business API with a MERN (MongoDB, Express.js, React.js, Node.js) stack. The proposed system addresses a critical challenge faced during an industrial internship, where the manual process of sending messages to thousands of clients daily was inefficient and time-consuming. By leveraging WhatsApp's messaging infrastructure, the system enables sending personalized, template-based messages to large contact groups with a single click. Key features include real-time webhook configuration for message status updates, template management, group-wise contact uploads via Excel files, and a scheduling mechanism for future message delivery. This integration not only streamlines communication but also significantly reduces manual effort and operational time, thereby enhancing business outreach and productivity. The paper discusses the overall architecture, technical components, and implementation strategies, demonstrating the practicality and scalability of the solution in real-world business environments.*

Keywords: WhatsApp Business API, MERN Stack, Automated Messaging, Client Communication System, Message Scheduling, Contact Group Management, Real-Time Webhooks, Business Automation, Template Messaging, Node.js, React.js, MongoDB, Express.js, Digital Communication, CRM Integration

I. INTRODUCTION

In the era of digital transformation, effective and timely client communication has become a cornerstone for business success. With the proliferation of instant messaging platforms, businesses are increasingly adopting tools like WhatsApp to interact with clients in a more personalized and direct manner. WhatsApp, being one of the most widely used messaging platforms globally, offers a significant opportunity for businesses to enhance their customer engagement through the WhatsApp Business API. This API allows companies to send automated, structured messages to clients at scale, while also enabling real-time communication and updates.

The motivation behind this study stems from a real-world challenge faced during an internship at a company where the sales team was responsible for sending daily updates and promotional messages to over 10,000 clients, covering hundreds of services. The process of manually selecting contacts and sending messages was not only labor-intensive but also prone to errors and delays. This inefficiency highlighted the urgent need for an automated communication solution that could streamline the messaging workflow and reduce manual dependency.

To address this, a robust communication system was developed using the MERN stack—MongoDB, Express.js, React.js, and Node.js—integrated with the WhatsApp Business API. This system enables the user to upload Excel files containing client contact information, categorize contacts based on service types, and assign them to specific contact

DOI: 10.48175/568

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, May 2025 Impact Factor: 7.67

groups. Message templates can be pre-defined and scheduled to be sent on future dates, allowing for automated, bulk messaging to thousands of clients with a single click.

The development process began with thorough research on the WhatsApp Business API and its requirements, including the creation of a Meta Business account, business verification, and payment method activation to access full API functionality. Once verified, the API was integrated into the backend of the system, enabling the sending of messages, real-time tracking through webhooks, and support for message templates. On the frontend, a user-friendly interface was developed to manage contacts, templates, and scheduling with ease.

This research paper documents the complete journey of designing, developing, and implementing the system, highlighting the technical challenges, solutions, and impact. By automating client communication, the system not only enhances operational efficiency but also opens new avenues for scalable business outreach.

II. SYSTEM ARCHITECTURE AND DESIGN

The architecture of the proposed communication system is designed to be modular, scalable, and user-friendly. By using the MERN stack—MongoDB, Express.js, React.js, and Node.js—along with the integration of the WhatsApp Business API, the system ensures smooth and secure communication between businesses and clients.

System Architecture Overview

The system follows a three-layered architecture:

- **Frontend (Client-Side Interface):** The frontend is built using React.js and TypeScript to create a responsive and intuitive user interface. This interface allows users to manage contact groups, create and preview message templates, and schedule messages for future delivery.
- **Backend (Server-Side Logic):** Node.js and Express.js handle the server-side logic, including API routing, user authentication, and message scheduling. The backend also integrates with the WhatsApp Business API to send messages and handle delivery status.
- **Database (Storage):** MongoDB is used to store contact details, message templates, logs, and other system-related information. It provides efficient data storage and retrieval to support real-time operations.

Component Design and Flow

The user authentication module ensures that only authorized personnel can send or schedule messages. This is achieved using bcrypt for password hashing and jsonwebtoken for secure authentication.

The contact group management feature allows users to upload Excel files containing client contact information. These contacts are grouped based on services, making it easy to send messages to specific groups. The xlsx and multer packages are used to parse and store contact information.

Message templates are predefined and stored within the system to save time on repetitive tasks. Once a template is created, users can preview it before sending and assign it to specific contact groups. These templates can be scheduled for future delivery, providing flexibility in the timing of communication.

Once a message is scheduled, the system automatically handles its delivery using node-cron jobs. The real-time status of each message is tracked through Meta's webhook functionality, providing updates on message delivery, read status, and any delivery failures.

Security and Validation

The system employs robust security measures such as helmet for setting secure HTTP headers, cors to manage cross-origin requests, and express-validator for input validation. These measures ensure that the system remains secure and reliable.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, May 2025
Impact Factor: 7.67

API Integration Workflow

The integration with the WhatsApp Business API requires a Meta Business account and a verified WhatsApp number. After authentication, access tokens are used to authorize requests to the WhatsApp API. The /v1/messages endpoint is used to send messages, while webhook events are leveraged to capture real-time updates on message delivery and status.

This architecture ensures the system is secure, scalable, and capable of handling thousands of messages with minimal manual intervention.

III. IMPLEMENTATION AND FUNCTIONALITY

The implementation of this system is the result of a practical approach to solving real-world communication inefficiencies within a business environment. Built on the MERN stack, the system brings together the robustness of backend processing with a seamless frontend experience. The WhatsApp Business API acts as the backbone for message transmission, enabling the core functionality of sending bulk messages through a single click.

Frontend Implementation

The frontend interface is developed using React.js with TypeScript, ensuring strong typing and scalability. Tailwind CSS is used for styling, offering utility-first components that are responsive and highly customizable. The interface includes various pages and functionalities, such as:

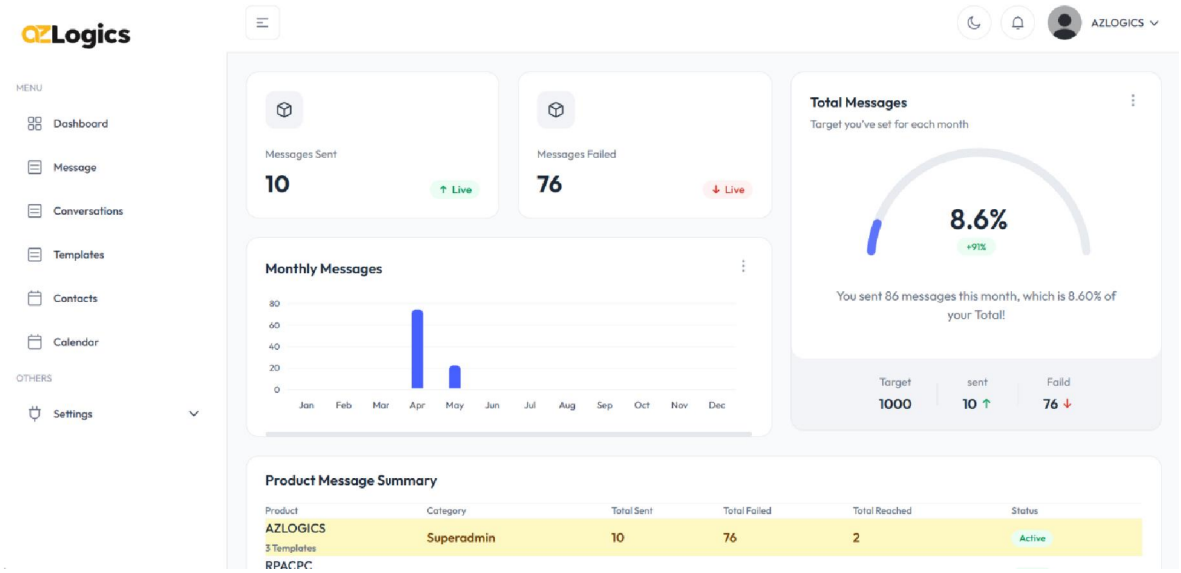

Fig.1A dashboard displaying total messages sent, scheduled tasks, and delivery statistics.

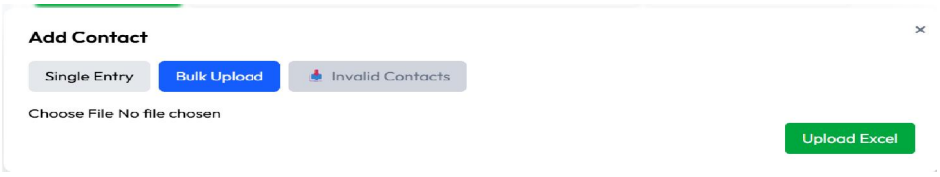

Fig. 2 Contact group management, allowing users to upload .xlsx files with client data.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, May 2025
Impact Factor: 7.67

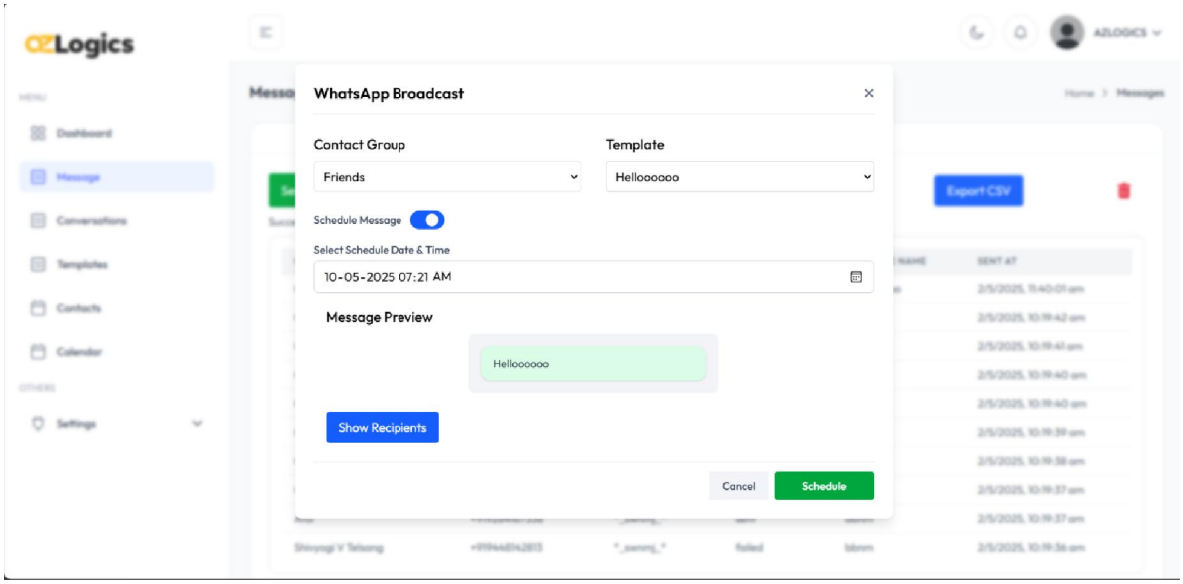

Fig. 3 Message template creation with fields for dynamic message input.

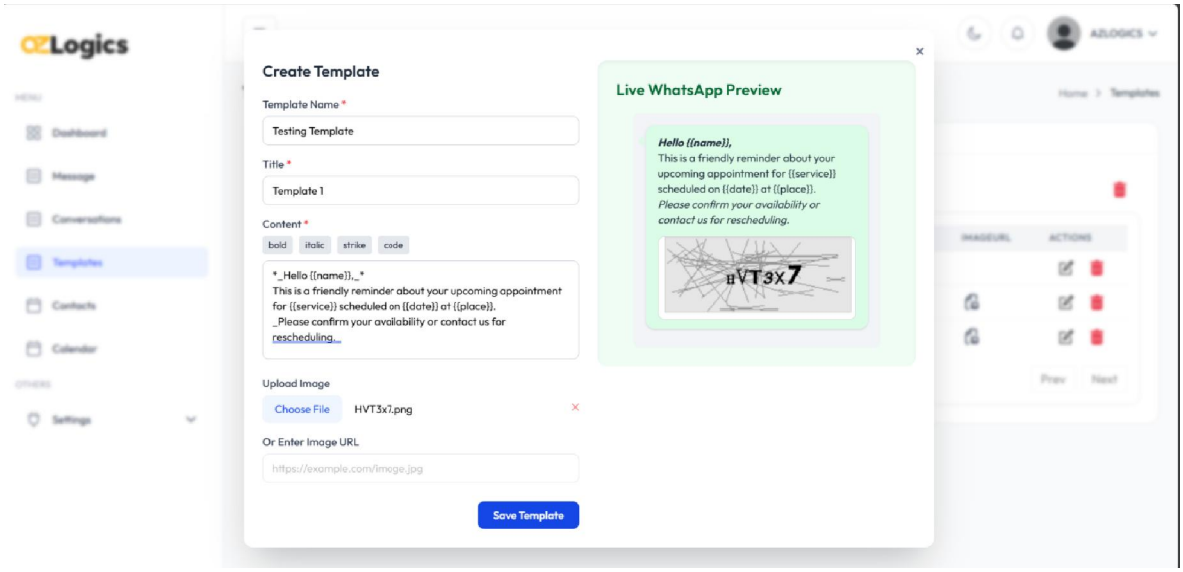

Fig. 4 A scheduling interface where messages can be queued for future dates and times.

Key packages like react-datepicker, react-dropzone, and papaparse enhance interactivity and file handling on the client side. jwt-decode is used to maintain session-based authentication, while react-router-dom is used for smooth routing across components.

Backend Implementation

The backend is powered by Node.js and Express.js, forming the core of the business logic and API interactions. MongoDB serves as the database, with Mongoose used as an ORM to model data schemas such as user credentials, contact groups, message templates, and message logs.

The backend exposes RESTful APIs for:

- Uploading and categorizing contact data.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, May 2025
Impact Factor: 7.67

- Creating and managing message templates.
- Scheduling and sending messages through the WhatsApp Business API.
- Handling authentication and user roles securely.

Important backend packages include:

- bcrypt and jsonwebtoken for secure authentication.
- xlsx and multer for parsing and handling file uploads.
- node-cron for scheduled messaging tasks.
- moment-timezone to standardize message timing across regions.
- express-validator for validating incoming API requests.

WhatsApp API Integration

The integration with Meta's WhatsApp Business API is achieved through the use of HTTP POST requests. Each scheduled message uses a pre-approved message template and recipient contact number. Access tokens are securely stored using environment variables via dotenv.

Webhook endpoints are configured to receive real-time delivery statuses of messages such as "sent," "delivered," "read," or "failed." This allows the backend to update the status of messages in the MongoDB database and display this data in the frontend dashboard.

Automation and Scheduling

One of the most powerful features of the system is its automation capability. Using node-cron, the backend queues scheduled messages and sends them at the specified date and time. This reduces manual intervention and ensures timely client communication without user input.

The user can also set recurring messages or promotional broadcasts based on calendar events. Once configured, the system handles these automatically, improving operational efficiency

IV. CONCLUSION

The integration of the WhatsApp Business API into a communication system, as demonstrated in this project, addresses a critical need for efficient, scalable, and automated client communication. By leveraging the MERN stack—comprising MongoDB, Express.js, React.js with TypeScript, and Node.js—this system provides a powerful solution that simplifies the process of sending bulk messages, managing contact groups, and scheduling communications across various business services.

Through rigorous research, practical development, and real-time implementation, the system has successfully eliminated the manual bottlenecks faced by sales and service teams in client outreach. It enables the sending of thousands of messages within minutes, while also allowing message personalization, template-based formatting, and future scheduling—features that significantly boost productivity and reduce human error.

Security has been maintained through proper authentication and validation, while real-time feedback from WhatsAppwebhooks ensures that the user is always informed about the status of each message. Moreover, the interface is user-friendly, allowing even non-technical users to manage complex communication tasks effortlessly.

This project not only solves the immediate problem of bulk messaging but also lays the foundation for integrating further automation and analytics into business communication strategies. It demonstrates how modern web technologies and cloud APIs can work together to transform traditional communication workflows into dynamic, data-driven processes.

ACKNOWLEDGMENT

First and foremost, I would like to express my sincere gratitude to the Almighty for giving me the strength, determination, and clarity to complete this project and research work.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, May 2025 Impact Factor: 7.67

I extend my heartfelt thanks to Prof. AshwiniMahajan and Prof. KomalNaxine, Computer Science & Engineering Department, TGPCET, Nagpur, for their valuable guidance, encouragement, and continuous support throughout this project. Their insightful suggestions and timely feedback were instrumental in the successful completion of this work.
I am also grateful to the Head of the Department and the entire faculty of the Computer Science & Engineering Department for providing a supportive academic environment and necessary resources.
Special thanks to my internship company for presenting the real-world problem that inspired this project. The experience significantly enhanced my understanding of client communication systems and business automation.

REFERENCES

[1]. Meta for Developers – WhatsApp Business Platform Documentation
Comprehensive guide covering API setup, message types, webhooks, and best practices for the WhatsApp Business Platform.
https://developers.facebook.com/docs/whatsapp

[2]. Meta for Developers – Message Templates Documentation
Detailed information on creating and managing message templates, including components like headers, bodies, footers, and buttons.
https://developers.facebook.com/docs/whatsapp/business-management-api/message-templates

[3]. Insider – The Ultimate Guide to WhatsApp Business API
An in-depth guide exploring the advantages of the WhatsApp Business API for marketers, setup instructions, and expert insights on using WhatsApp for business.
https://useinsider.com/the-ultimate-guide-to-whatsapp-business-api/

[4]. Gupshup – WhatsApp Business API Guide
Comprehensive overview of WhatsApp Business API features, including messaging categories, pricing models, and commerce capabilities.
https://www.gupshup.io/resources/guide/whatsapp-business-api-guide

[5]. Sinch – Your Complete Guide to WhatsApp Business API
Insights into the latest developments in WhatsApp Business API, including the Marketing Messages Lite API and its implications for businesses.
https://sinch.com/blog/whatsapp-business-api/

[6]. A Study on Modern Messaging Systems – Kafka, RabbitMQ, and NATS Streaming
An analytical comparison of modern messaging systems, discussing their scalability, fault tolerance, and suitability for real-time applications.
https://arxiv.org/abs/1912.03715

[7]. Riposte: An Anonymous Messaging System Handling Millions of Users
Presents Riposte, a system designed for anonymous broadcast messaging, focusing on scalability and protection against traffic-analysis attacks.
https://arxiv.org/abs/1503.06115

[8]. Quarks: A Secure and Decentralized Blockchain-Based Messaging Network
Introduces Quarks, a blockchain-based messaging system aiming to eliminate centralized control and enhance security attributes in group communications.
https://arxiv.org/abs/2308.04452

[9]. XRD: Scalable Messaging System with Cryptographic Privacy
Discusses XRD, a metadata-private messaging system that provides cryptographic privacy while scaling to support millions of users.
https://arxiv.org/abs/1901.04368

[10]. Callbell – 17 Message Templates for WhatsApp Business API
Provides a collection of message templates to improve response times and service quality in customer

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, May 2025
Impact Factor: 7.67

interactions via WhatsApp Business API.
https://www.callbell.eu/en/message-templates-for-whatsapp-business-api/

[11]. EnableX – 10 Essential WhatsApp Business Template Messages for All Businesses
Offers a list of essential message templates tailored to diverse business needs, designed to drive impactful results.
https://www.enablex.io/insights/10-essential-whatsapp-business-template-messages-for-all-businesses/

[12]. EnableX – How to Create Message Templates in WhatsApp Business API
A step-by-step guide on creating effective message templates, detailing components like headers, bodies, footers, and interactive buttons.
https://www.enablex.io/insights/how-to-create-message-templates-in-whatsapp-business-api/

[13]. Qiscus – 11 Ways to Use Message Templates on WhatsApp Business API
Explores various applications of message templates in WhatsApp Business API, including account updates, auto-replies, and payment notifications.
https://www.qiscus.com/en/blog/11-ways-to-use-message-templates-on-whatsapp-business-api/

[14]. EnableX – How to Create Message Templates in WhatsApp Business API
Guides businesses on structuring message templates for personalized, targeted messaging, enhancing customer experience through the WhatsApp Business API.https://www.enablex.io/insights/how-to-create-message-templates-in-whatsapp-business-api/

[15]. Qiscus – 11 Ways to Use Message Templates on WhatsApp Business API
Provides insights into leveraging message templates for various business scenarios, improving communication efficiency and customer engagement.
https://www.qiscus.com/en/blog/11-ways-to-use-message-templates-on-whatsapp-business-api/