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 6, Issue 3, May 2026
Impact Factor: 8.2

SecureChat: An AI-Powered Real-Time Messaging Platform with Intelligent Content Moderation and Video Communication

Raghav Turkar[1], Vaishnavi Ranjan[2], Sarthak Ukharde[3], Aakash Singh[4], Prof. S. V. Patil[5]
Department of Computer Engineering
Sinhgad College of Engineering, Pune, India
raghavturkar@scoeceducin, vaishnaviranjan@scoeceducin,
sarthakukharde@scoeceducin, aakashsingh@scoeceducin

Abstract: *Real-time communication platforms have become an essential part of personal, academic, and professional interaction. However, the increasing use of online messaging systems has introduced several security and trust-related challenges. Users may receive phishing links, inappropriate media, manipulated images, artificial intelligence generated text, suspicious files, and misleading information through chat applications. Traditional messaging platforms mainly focus on message delivery, media sharing, and user connectivity, but they often lack integrated intelligent content moderation mechanisms that can assist users in identifying unsafe or suspicious content during communication*

Keywords: Secure Chat, Artificial Intelligence, Real-Time Messaging, WebSocket, WebRTC, Content Moderation, Phishing Detection, Deepfake Detection

I. INTRODUCTION

Real-time communication systems have become a fundamental component of modern digital interaction. Messaging platforms are extensively used for personal communication, academic collaboration, and professional coordination. With the rapid growth of internet accessibility and mobile devices, users increasingly rely on chat-based applications for exchanging information in the form of text, images, videos, and documents. However, this widespread usage has also led to a significant rise in security threats and misuse of communication platforms.

To overcome these limitations, a secure and intelligent communication system named SecureChat is proposed. The system integrates real-time messaging with AI-powered content moderation and analysis features. It is designed to provide a safe communication environment by detecting harmful content and assisting users in understanding and verifying the information they receive. The platform supports user authentication, profile management, friend request handling, real-time messaging, file sharing, online presence tracking, typing indicators, read receipts, and WebRTC-based video calling.

II. MOTIVATION AND BACKGROUND

In recent years, real-time communication platforms have become an integral part of digital interaction across personal, academic, and professional domains. Applications such as messaging platforms and collaborative tools have significantly improved connectivity by enabling instant exchange of text, images, videos, and documents. However, this rapid growth has also introduced serious concerns related to security, authenticity, and reliability of shared information.

One of the primary motivations behind this research is the increasing exposure of users to unsafe and misleading content during digital communication. Modern messaging platforms often lack built-in mechanisms to proactively detect

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 6, Issue 3, May 2026
Impact Factor: 8.2

harmful elements such as phishing links, malicious URLs, inappropriate media, deepfake images, and artificially generated text

III. RELATED WORK AND LITERATURE SURVEY

The development of secure and intelligent communication systems has become an important area of research due to the increasing use of online messaging platforms. Existing communication systems provide features such as real-time messaging, file sharing, voice calling, and video calling. However, most of these systems mainly focus on communication efficiency and user connectivity. The integration of artificial intelligence based content moderation within real-time messaging platforms is still limited.

TABLE I: Comparison of Existing Systems and Proposed System

Parameter	Existing Chat Applications	Standalone AI Moderation Tools	Proposed Secure Chat System
Real-Time Messaging	Available	Not Available	Available
Friend Request Management	Available in Some Systems	Not Available	Available
File Sharing	Available	Not Available	Available
Video Calling	Available in some	Not Available	Available
NSFW Image Detection	Limited	Available	Available
AI Text Detection	Limited	Available	Available
Phishing Detection	Limited	Available	Available
Summarization	Limited	Separate Tool	Available

IV. SYSTEM ARCHITECTURE

The SecureChat system follows a client-server architecture designed to support real-time communication, efficient data management, and integration of artificial intelligence based services. The architecture is divided into three primary layers: presentation layer, application layer, and data and intelligence layer.

The presentation layer is developed using React. It provides user registration, login interface, chat interface, friend request management, file upload preview, video calling interface, and display of AI analysis results. The frontend communicates with the backend through REST APIs and WebSocket events. The data layer uses PostgreSQL for structured storage. It maintains information related to users, messages, friendships, AI message checks, hidden messages, and read statuses. The intelligence layer integrates AI services for image classification, deepfake detection, text classification, URL classification, translation, summarization, and image captioning.

The artificial intelligence module performs analysis on text, images, and URLs. The process involves feature extraction, model inference, and result generation. To evaluate the performance of AI models, standard classification metrics are used:

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

$$\text{Precision} = TP / (TP + FP) \quad (2)$$

$$\text{Recall} = TP / (TP + FN) \quad (3)$$

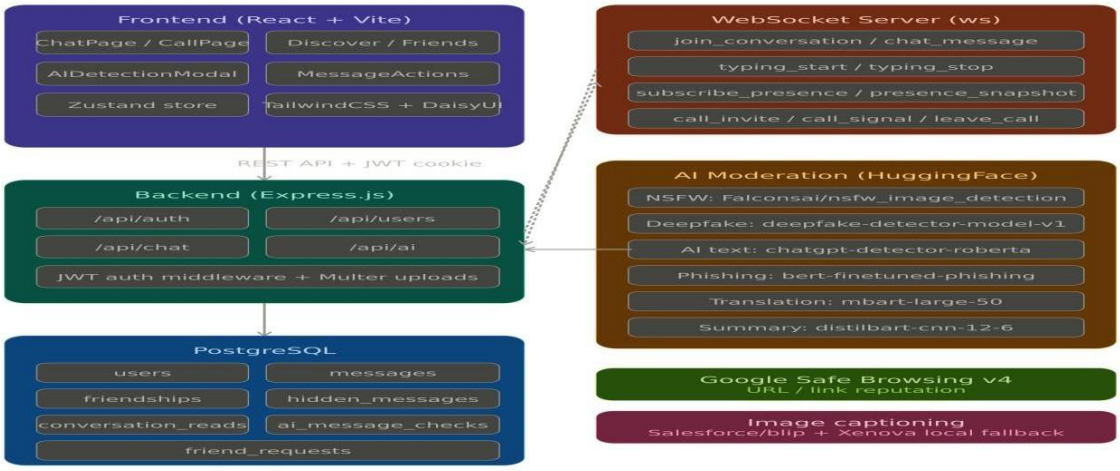
$$\text{F1 Score} = 2 \times \text{Precision} \times \text{Recall} / (\text{Precision} + \text{Recall}) \quad (4)$$

Text messages are processed using natural language processing techniques. Text similarity and semantic comparison can be represented using cosine similarity:

$$\text{Similarity} = A \cdot B / (||A|| \times ||B||) \quad (1)$$

DOI: 10.48175/IJARSCT-34967

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 6, Issue 3, May 2026
Impact Factor: 8.2



Images shared within the chat are analyzed using image classification models. The system evaluates images to determine whether they contain inappropriate content or are artificially generated. URLs shared in messages are analyzed to detect phishing attempts. The system extracts URLs from text and evaluates them using machine learning models and reputation-based APIs

V. FRONTEND: UX AND COMPONENTS

The frontend of the SecureChat system is designed to provide an intuitive, responsive, and user-friendly interface that enables seamless real-time communication while integrating artificial intelligence based insights. The user experience (UX) is carefully structured to ensure ease of navigation, minimal cognitive load, and efficient interaction with both communication and analysis features.

A. User Experience Design

The user experience is focused on simplicity, clarity, and responsiveness. The interface is designed to minimize user effort while performing common tasks such as sending messages, sharing files, and analyzing content.

Key UX considerations include:

- Clean and minimalistic chat interface
- Real-time updates without page refresh
- Visual indicators for typing, message delivery, and read status
- Clear highlighting of AI analysis results (e.g., phishing warning, unsafe content alert)
- Responsive design for different screen sizes

The system ensures that AI features do not interrupt communication flow but instead enhance user awareness by providing contextual insights.

B. Authentication and Onboarding Components

The authentication interface includes login and registration forms designed with validation mechanisms to ensure secure user input. Upon successful registration, users complete profile onboarding by providing details such as name, profile image, bio, and location.

Features include:

- Form validation for user inputs
- Secure authentication flow using tokens
- Profile setup interface
- Error handling and feedback messages

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 6, Issue 3, May 2026
Impact Factor: 8.2

VI. BACKEND: SERVICES AND ALGORITHMS

Backend: Services and Algorithms

The backend of the SecureChat system is responsible for handling core application logic, data processing, real-time communication, and integration with artificial intelligence services. It is implemented using Node.js and Express.js, providing a scalable and efficient server-side architecture.

A. Backend Architecture

The backend follows a modular architecture where different services are responsible for specific functionalities.

Key modules include:

- Authentication service
- Messaging service
- Friend management service
- File handling service
- AI processing service

This modular design improves maintainability and scalability.

B. Authentication Service

The authentication service ensures secure access to the system using JSON Web Token (JWT) based authorization.

Features include:

- User registration and login
- Token generation and validation
- Secure session handling
- Password hashing for security

C. URL and Phishing Detection Algorithm

The system analyzes URLs using probabilistic methods:

$$P(\text{Phishing}\backslash\text{URL}) = P(\text{URL})P(\text{URL}\backslash\text{Phishing}) \cdot P(\text{Phishing})$$

This helps determine whether a URL is safe or malicious

VII. DATA PERSISTENCE AND MANAGEMENT

Data Persistence and Management

The SecureChat system employs a robust data persistence layer to ensure reliable storage, retrieval, and management of user data, messages, and artificial intelligence analysis results. The database is designed to support real-time operations while maintaining data integrity, scalability, and efficient query performance. PostgreSQL is used as the primary database management system. It stores structured data including user profiles, messages, friendships, and AI analysis logs. The database schema is normalized to reduce redundancy and improve consistency across related entities.

The system stores artificial intelligence analysis results in a dedicated structure that includes metadata such as:

- userId: Identifier of the user initiating the analysis
- messageId: Reference to the associated message

Future work can focus on extending system capabilities. Possible enhancements include: • result: Output classification

- confidenceScore: Model confidence
- timestamp: Time of analysis

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 6, Issue 3, May 2026
Impact Factor: 8.2

VIII. SECURITY, PRIVACY, AND COMPLIANCE

Security and privacy are critical components of the SecureChat system, given the sensitive nature of communication data. The system is designed to protect user information, ensure secure data transmission, and comply with standard data protection practices.

User authentication is implemented using JSON Web Tokens (JWT), ensuring secure session management. Passwords are stored using strong hashing algorithms, such as bcrypt, to prevent unauthorized access even in case of data breaches.

IX. EVALUATION AND EXPERIMENTS

The SecureChat system was evaluated through a prototype implementation to analyze its performance, usability, and effectiveness of artificial intelligence integration.

The evaluation focused on:

- Real-time communication efficiency
- Accuracy of AI-based content moderation
- User experience and responsiveness

Testing was conducted with multiple users interacting through the platform over a defined period. The results demonstrated that the system provides near real-time message delivery with minimal latency due to the use of WebSocket communication.

X. ADVANTAGES, LIMITATIONS, AND FUTURE WORK

The SecureChat system provides several advantages in terms of functionality, security, and usability. The integration of real-time communication with artificial intelligence enables users to detect harmful content such as phishing links, inappropriate images, and AI-generated messages. The system also enhances user experience through features like translation, summarization, and image captioning.

The modular architecture allows scalability and easy maintenance, while caching mechanisms improve performance by reducing redundant computations. The use of modern technologies such as WebSocket and WebRTC ensures efficient communication and low latency.

- Implementation of group chat functionality
- Integration of end-to-end encryption for improved security
- Development of mobile applications
- Integration of context-aware AI models
- Advanced probabilistic models for content classification

These improvements can further enhance system performance, scalability, and user experience.

ACKNOWLEDGMENTS

The authors express their sincere gratitude to the faculty members of the Department of Computer Engineering at Sinhgad College of Engineering for their continuous guidance and support throughout the development of this project.

Special thanks are extended to peers and pilot users for their valuable feedback, which contributed significantly to improving the system design and functionality. The authors also acknowledge the contributions of open-source communities and platforms that provided essential tools and resources used in the implementation of the SecureChat system.

REFERENCES

1) React Documentation. Meta Platforms Inc. Available: https://react.dev
2) Node.js Documentation. OpenJS Foundation. Available: https://nodejs.org
3) Express.js Documentation. OpenJS Foundation. Available: https://expressjs.com

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 6, Issue 3, May 2026
Impact Factor: 8.2

4) PostgreSQL Documentation. PostgreSQL Global Development Group. Available: https://www.postgresql.org
5) WebRTC Documentation. World Wide Web Consortium (W3C). Available: https://webrtc.org
6) HuggingFace Model Hub. HuggingFace Inc. Available: https://huggingface.co