

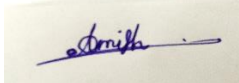

Sri Lanka Institute of Information Technology



Technical Paper

IE3042-Secure Software System

Year 3, Semester 2

IT NUMBER	NAME	CONTACT NUMBER	EMAIL	Signature
IT21167478	Nilupul S.A	0769665153	lt21167478@my.sliit.lk	
IT21167096	De Zoysa A. S.	0719895280	it21167096@my.sliit.lk	

Link

Mine 2024



VERSION 1.1

Secure Chat application by,
SLIIT CYBER SECURITY UNDERGRADUATE STUDENTS

Table of Contents

Abstract.....3

1. INTRODUCTION.....4

1.1 What is Link Mine?4

1.2 Main Features.....5

1.2.1 End-to-End Encryption:5

1.2.2 Real-Time Messaging:5

1.2.3 Create channel and chatting system:7

1.2.4 Secure Video Calling:.....8

1.2.5 Machine Learning Integration:.....8

1.2.6 User Authentication and Authorization:.....9

1.2.7 Data Encryption:9

1.2.8 User Presence and Status Indicators:9

1.2.9 Media Sharing:.....9

1.2.10 Push Notifications:9

1.2.11 User Profile Management:10

1.2.12 Administrative Controls:10

2. RESEARCH AREA.....11

2.1 Security challenges in Digital Communications11

2.2 Existing solutions and their limitations.11

2.3 Technological advancements.....11

3. DEVELOPMENT ENVIRONMENT12

3.1 Operating System:12

3.2 Front-end Development:12

3.3 Back-end Development:.....13

3.4 Video Calling:.....13

3.5 Machine Learning:13

3.6 Development Tools and Environment Setup:.....14

4. DISPLAY AND USER INTERFACE15

4.1 Design concept.....15

4.2 User Interface features15

5. SYSTEM ARCHITECTURE.....16

- 5.1 Component Breakdown 17
 - 5.1.1 Client-side (crypto.js) 17
 - 5.1.2 Server-side (Node.js, Express.js)..... 17
 - 5.1.3 Real-time Communication (Socket.io)..... 17
 - 5.1.4 Video calls (WebRTC)..... 18
 - 5.1.5 Database (MongoDB) 18
- 5.2 Security Measures 18
- 5.3 System Workflow 19
 - 5.3.1 User Registration and Authentication: 19
 - 5.3.2 Real-time Messaging: 19
 - 5.3.3 Video Calling:..... 19
 - 5.3.4 Machine learning enhancements: 19
- 6. TESTING AND EVALUATION 20
 - 6.1 Unit Testing..... 20
 - 6.2 Integration Testing. 20
 - 6.3 Secure Testing..... 21
- 7. Future Development..... 22
- CONCLUSION..... 23

Abstract

Link Mine is an innovative encrypted messaging program created to fulfill the rising demand for private and secure digital communication. Link Mine, built as a web-based platform, uses cutting-edge technologies such as Node.js, Socket.io for real-time communication, and WebRTC for video conversations. Link Mine, which uses machine learning to improve security and user experience, is available on both Windows and Ubuntu operating systems. This article describes Link Mine's thorough investigation, design, and implementation, focusing on its features, architecture, and rigorous testing techniques that assure its reliability and security.

1. INTRODUCTION

The increasing need for secure communication has driven the development of various messaging applications, where sensitive information is constantly being exchanged over various digital channels. Traditional messaging applications often fall short in providing robust security measures, leaving users vulnerable to data breaches, eavesdropping, and privacy violations. The need for a secure messaging platform that prioritizes end-to-end encryption, user privacy, and robust security features has become increasingly evident.

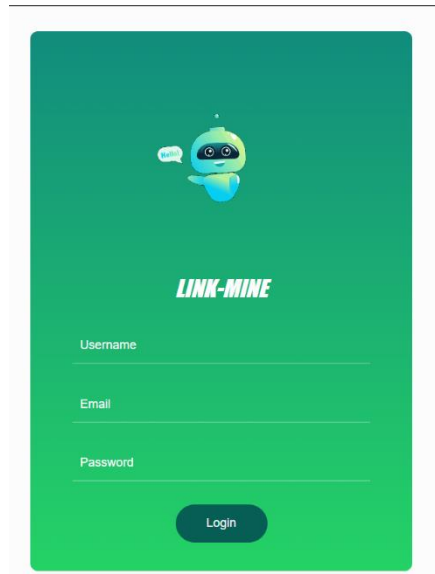
"Link Mine" is a secure messaging application designed to address these concerns. Developed using state of the art technologies such as Node.js, Socket.io, and WebRTC, "Link Mine" aims to provide a secure and user-friendly platform for real-time messaging and video calling. The application leverages machine learning techniques to enhance security and improve the overall user experience

1.1 What is Link Mine?

Link Mine is a secure chat application designed for individuals, who prioritize confidentiality in their communication. It offers a user-friendly experience while employing robust security features, real-time communication capabilities, and the potential for machine learning integration.

Link Mine gives needed to keep secure chat safe, focusing on:

- **Secure Communication:** End-to-end encryption ensures only the sender and recipient can access messages and calls.
- **Real-Time Interaction:** Socket.IO facilitates instant messaging and updates, fostering seamless communication.
- **Video Conferencing:** WebRTC enables integrated video calls directly within the application.
- **Machine Learning Potential:** The paper explores the possibility of integrating functionalities like spam detection, content moderation, and personalized recommendations.
- **Developed on the secure and stable Ubuntu operating system and Windows Operating System,** Link Mine utilizes HTML, CSS for the user interface, Node.js for the backend server, Socket.IO for real-time communication, and WebRTC for video calls. This combination ensures a secure, performant, and feature-rich chat application.



1.2 Main Features

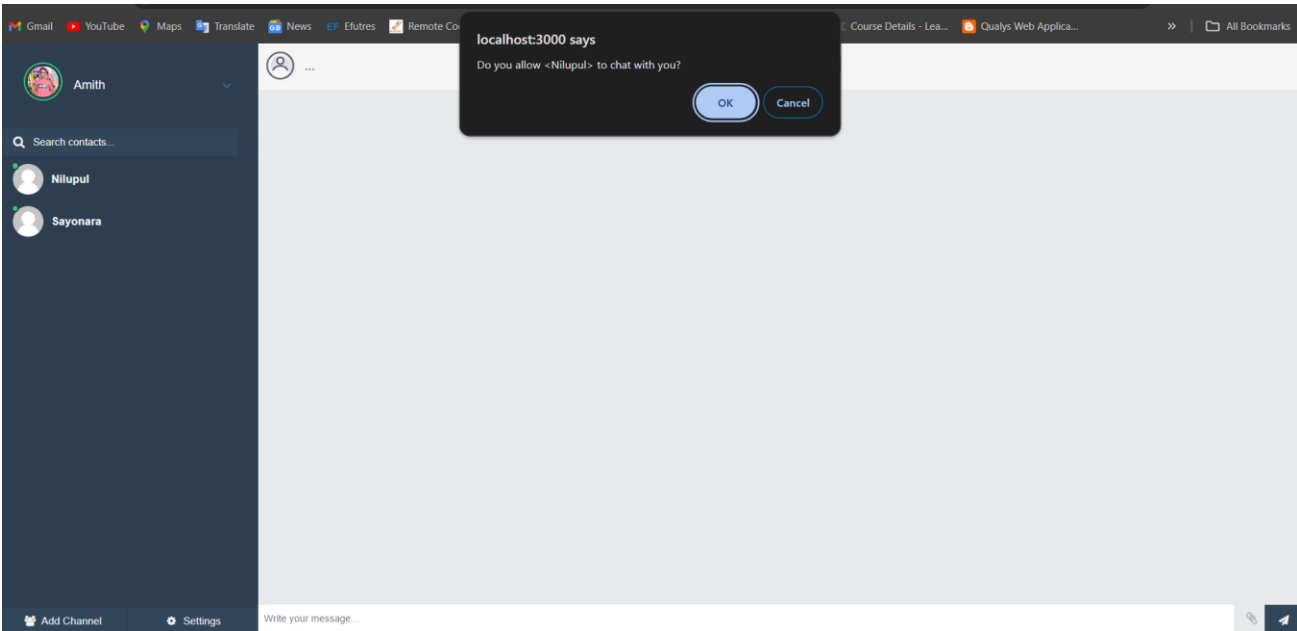
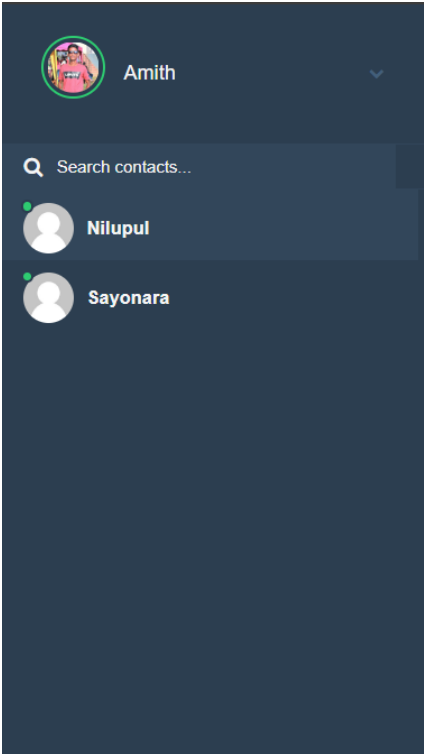
Link Mine full features set is available for windows workstation. Under the below features it provides secure your chats.

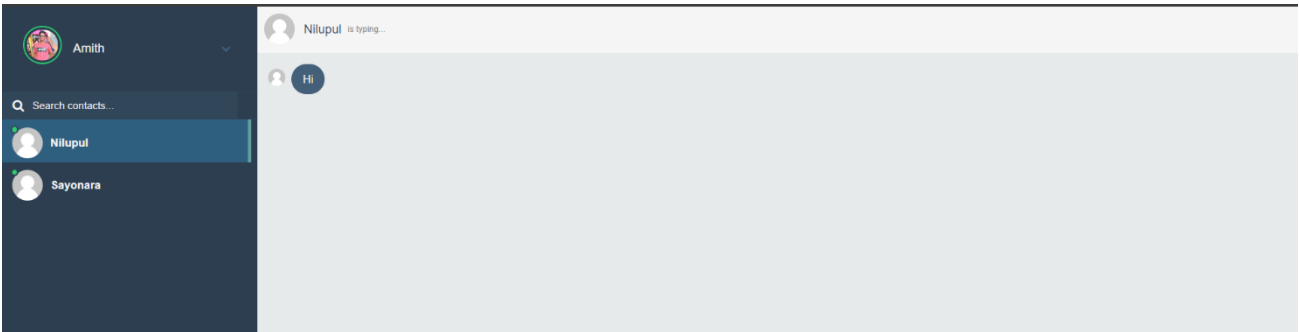
1.2.1 End-to-End Encryption:

Ensures that messages can only be read by the sender and the recipient, protecting the communication from eavesdropping.

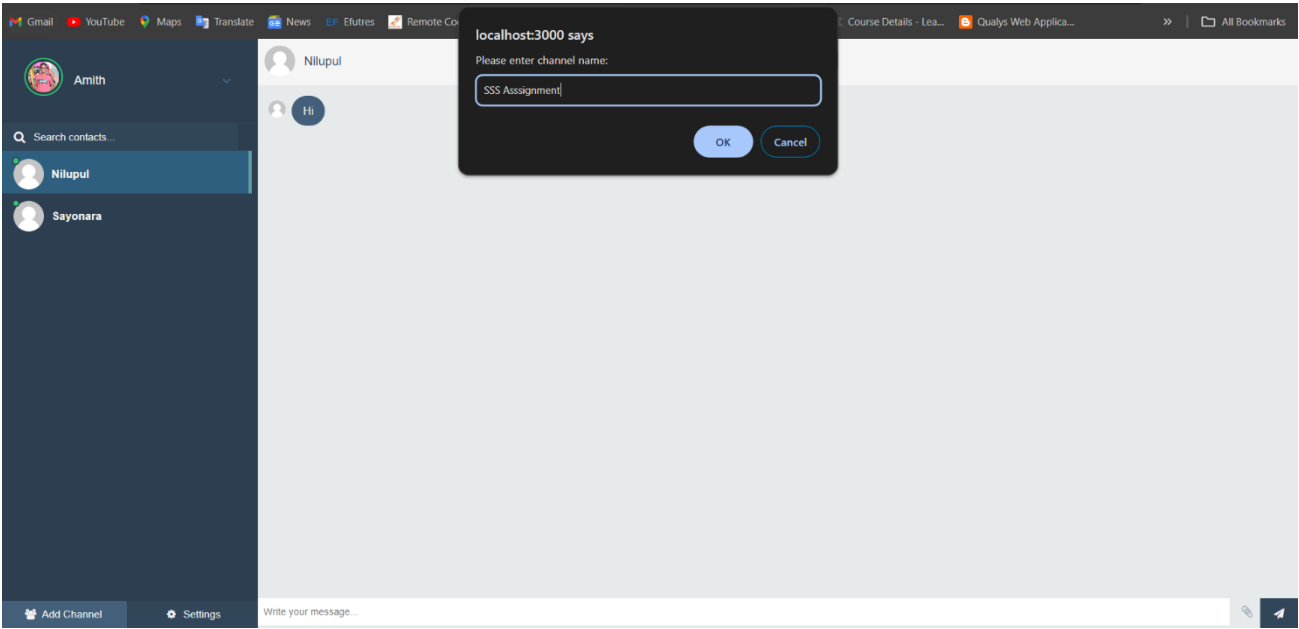
1.2.2 Real-Time Messaging:

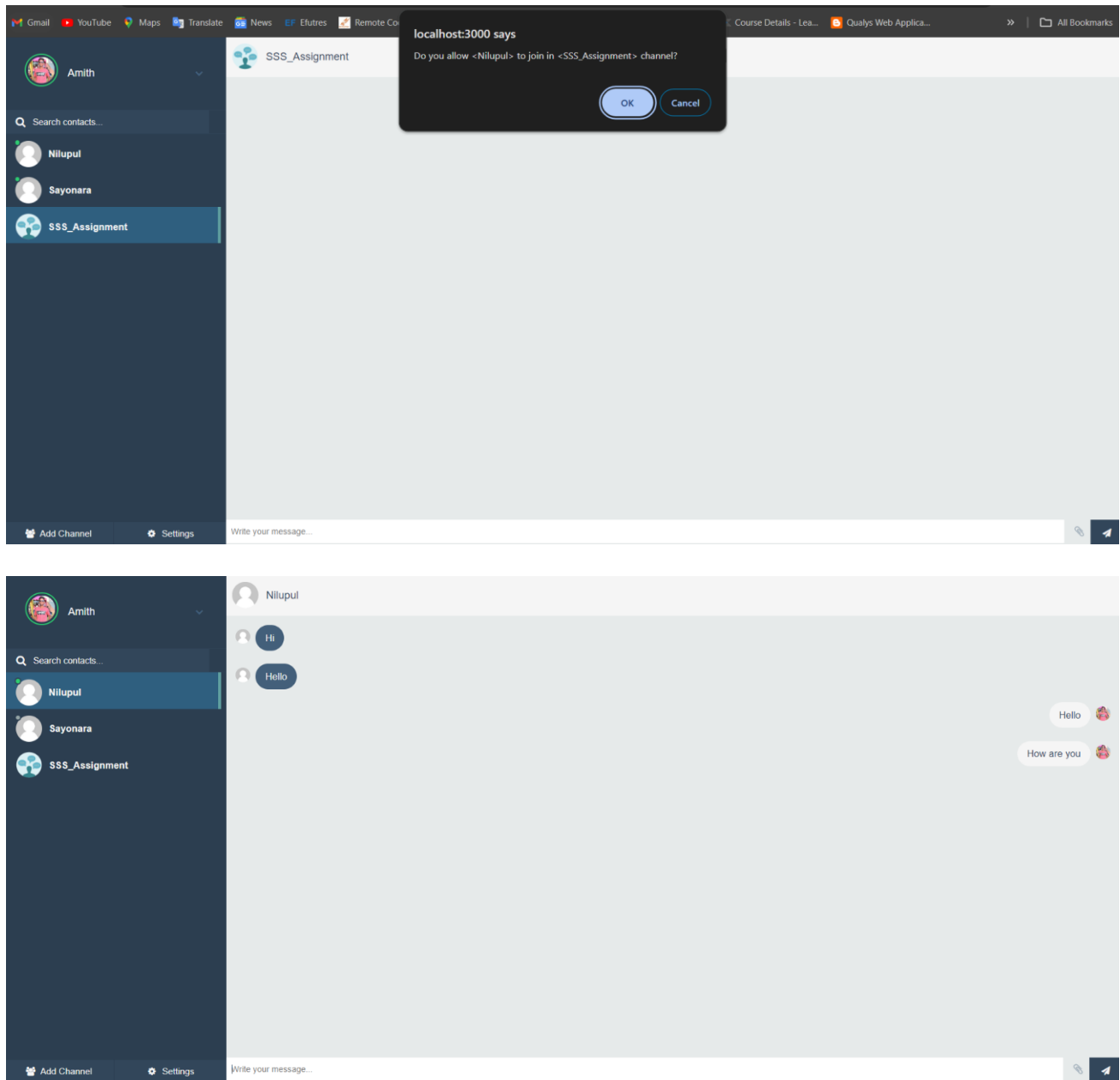
Utilizes Socket.io for low-latency, bidirectional communication, allowing users to send and receive messages instantly.





1.2.3 Create channel and chatting system:
Adding members and chat with another in this channel.





1.2.4 Secure Video Calling:

Integrates WebRTC for peer-to-peer video calling, ensuring real-time, high-quality video communication.

1.2.5 Machine Learning Integration:

Enhances security and user experience through features like spam detection, anomaly detection, and personalized recommendations.

1.2.6 User Authentication and Authorization:

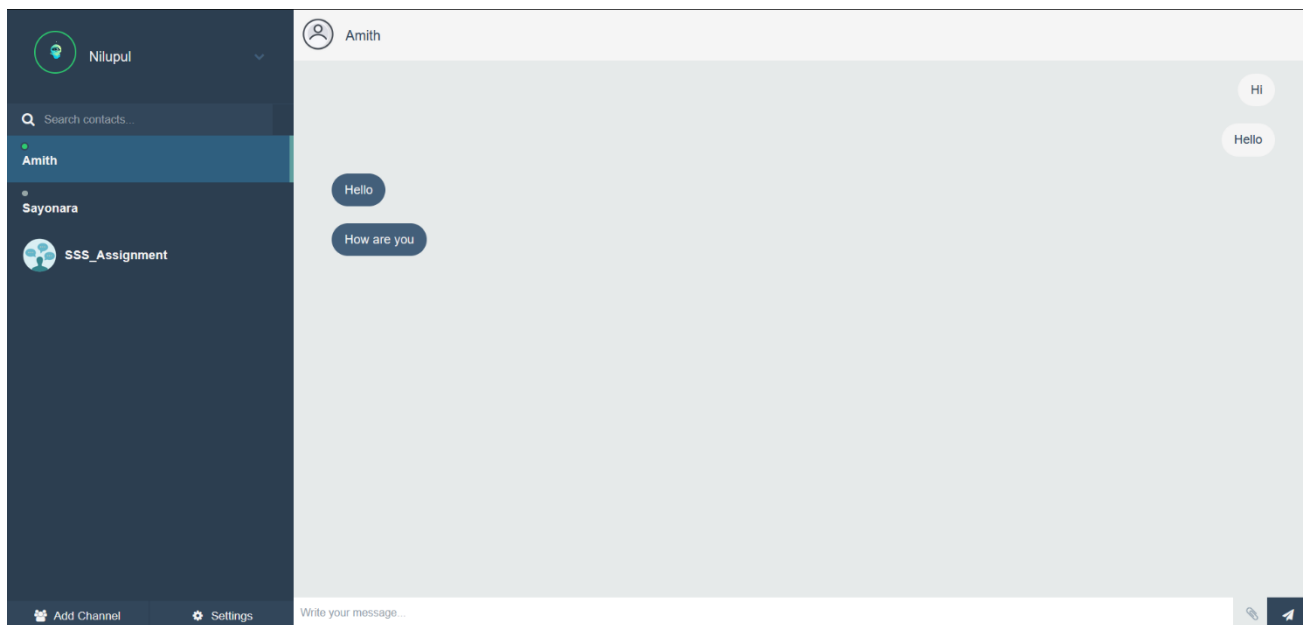
Implements secure user login and registration processes using JSON Web Tokens (JWT) and supports two-factor authentication (2FA) for an added layer of security.

1.2.7 Data Encryption:

Encrypts data at rest and in transit using Advanced Encryption Standard (AES), ensuring that user data is protected from unauthorized access.

1.2.8 User Presence and Status Indicators:

Shows the online status of users and indicates when someone is typing, enhancing real-time interaction.



1.2.9 Media Sharing:

Allows users to share photos, videos, and files securely within the chat application.

1.2.10 Push Notifications:

Keeps users informed about new messages and activities even when they are not actively using the application.

1.2.11 **User Profile Management:**

Enables users to manage their profiles, including setting , updating personal information, and managing privacy settings.

1.2.12 **Administrative Controls:**

Provides tools for administrators to manage users, and overall application settings, ensuring a controlled and secure environment.

2. RESEARCH AREA

2.1 Security challenges in Digital Communications

The demand for secure messaging applications has been on the rise, driven by growing concerns over privacy and data protection. Market analysis reveals a significant gap in the availability of user-friendly, highly secure chat applications tailored for individual users. Existing solutions often lack comprehensive security features or are too complex for everyday use. The rapid growth of digital communication platforms has highlighted a number of security concerns, including data breaches, illegal access, and privacy violations. These difficulties demand strong security measures to safeguard user data and communication integrity.

2.2 Existing solutions and their limitations.

Current communication systems provide varied levels of security, but many fall short of standards such as end-to-end encryption, user data protection, and cyber-attack resistance. By studying these constraints, Link Mine hopes to create a more secure alternative.

2.3 Technological advancements

WebRTC, machine learning, and real-time communication frameworks (such as Socket.io) have enabled more secure and efficient communication solutions. Link Mine leverages these technologies to provide stronger security features and a better user experience.

3. DEVELOPMENT ENVIRONMENT

Link Mine prioritizes security and aims to be a versatile communication tool caters to individuals seeking a secure platform for casual conversations or exchanging sensitive information. The development environment for the secure web application "Link Mine" includes a variety of technologies and tools, as well as a customized operating system configuration. Here is a thorough description of the development environment:

3.1 Operating System:

Link Mine is designed to work with both Windows and Ubuntu operating systems. This dual compatibility allows the development team to work in their preferred settings while also testing the program on several platforms for increased robustness and dependability.

- The project was developed on both Windows and Ubuntu operating systems.
- Ubuntu is a popular open-source Linux distribution known for its security and privacy features, making it a suitable choice for developing a secure messaging application.
- Windows was also used during development, as it is a widely adopted platform, ensuring cross-platform compatibility and testing.

3.2 Front-end Development:

- React: React is a popular JavaScript library for building user interfaces, and it was chosen as the front-end framework for "Link Mine." React's component-based architecture and efficient rendering make it an excellent choice for developing responsive and performant web applications.
- Additional front-end libraries and tools, such as Router for handling client-side routing, and Redux for state management, may have been utilized to enhance the development process and application functionality.

3.3 Back-end Development:

- Node.js: Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine, allowing developers to run JavaScript on the server-side. It was chosen as the back-end technology for "Link Mine" due to its event-driven, non-blocking I/O model, which makes it highly efficient for building real-time applications.
- Socket.io: Socket.io is a library that enables real-time, bidirectional, and event-based communication between clients and servers. It was integrated into "Link Mine" to facilitate real-time messaging and communication between users.

3.4 Video Calling:

WebRTC (Web Real-Time Communication): WebRTC is an open-source project that enables real-time communication capabilities in web browsers, including video and audio streaming. It was utilized in "Link Mine" to enable secure video calling functionality without the need for intermediate servers or plugins.

3.5 Machine Learning:

- The project incorporates machine learning techniques to enhance security features and improve the overall user experience.
- Various machine learning libraries and frameworks, such as TensorFlow, PyTorch, or scikit-learn, may have been used in conjunction with Python or other programming languages to implement machine learning models.
- These models could be trained on relevant data to perform tasks like anomaly detection, user behavior analysis, and proactive security measures.

3.6 Development Tools and Environment Setup:

- **Code Editors:** Popular code editors like Visual Studio Code may have been used for writing and editing the application's codebase.
- **Version Control:** A version control system like Git would have been used to manage the project's source code, collaborate with team members, and track changes.
- **Build Tools:** The application's code, ensuring compatibility across different browsers and environments.
- **Development Servers:** Local development servers like the built-in Node.js server or tools like webpack-dev-server might have been used for running and testing the application during development.

4. DISPLAY AND USER INTERFACE

4.1 Design concept

Link Mine focuses a user-friendly and intuitive interface that promotes effective communication. The primary display platform, offering a comprehensive and feature-rich interface optimized for larger screens. With desktop web application. Here are the main design concepts that guide the application's interface:

- **Simplicity:** The interface will be simple and clear and with easy navigation and access to key functionality.
- **Easy Interaction:** Users should be able to easily navigate the program, with capabilities easily accessible without lengthy learning curves.
- **Flexibility:** The UI will be adaptable and adapt to multiple screen sizes, offering the best viewing experience on desktop, mobile, and tablet devices(local host)
- **Accessibility:** The application will be built with accessibility in mind, including features for users with vision problems and other accessibility requirements.

4.2 User Interface features

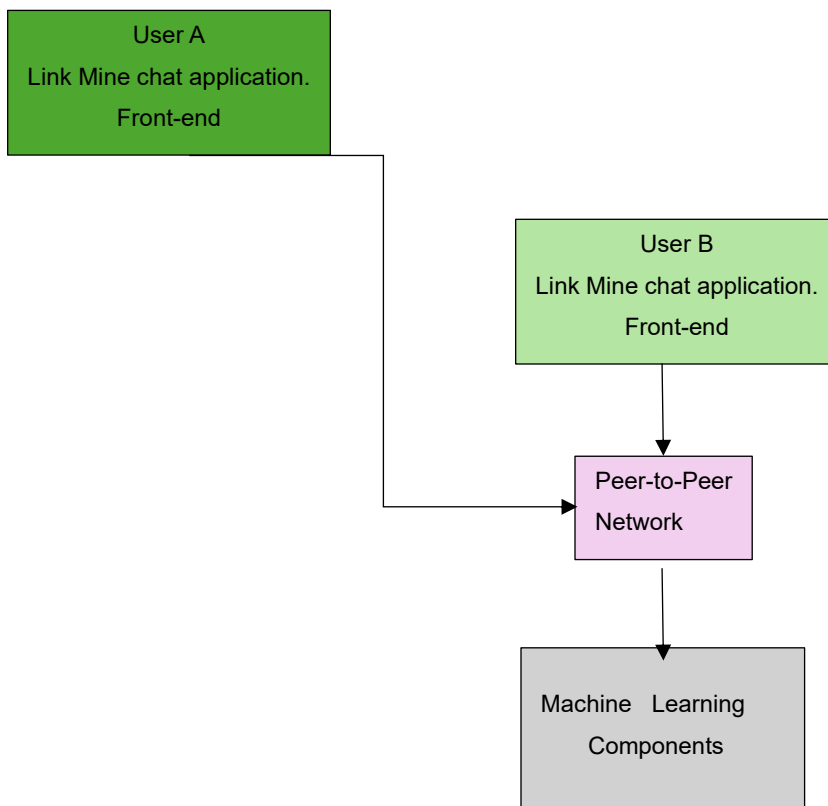
- **Login/Registration:** Existing users may access a secure login screen, while new accounts can be created using a registration form.
- **Contact List:** A specialized section for managing contacts, where users may add, search, and arrange their communication partners.
- **The Chat Interface** is the major means of communication, featuring individual chat chats. This includes message history, text input fields, file attachments
- **Video Call Button:** A visible button within the chat interface for starting video chats with contacts.
- **The Settings Menu** allows you to adjust user preferences such as notification settings, profile information, and perhaps future machine learning-powered capabilities (for example, spam filtering settings).

5. SYSTEM ARCHITECTURE

The system architecture of Link Mine is designed to ensure security, scalability, and performance. It operates on a client-server model, incorporating many technologies to give a comprehensive secure messaging solution.

The architecture of Link Mine can be divided into the following main components:

- Client-side implementation was made with crypto-js.
- Server-side development was carried out using Node.js and Express.js.
- Real-time communication is facilitated using Socket.io.
- Video calls are enabled using WebRTC.
- MongoDB is used for data storing.
- Machine learning is integrated to improve security and user experience.



5.1 Component Breakdown

5.1.1 Client-side (crypto.js)

The client side is in charge of displaying the user interface and managing user interactions. React is employed because of its component-based design, which enables reusable and maintainable programming.

- **Components:** Modular React components are built for various areas of the application, including login, chat interface, video call interface, and user settings.
- **State Management:** React manages the application's state, ensuring that it is consistent and predictable throughout.
- **Routing:** React Router is used to navigate between multiple views of an application.

5.1.2 Server-side (Node.js, Express.js)

The server manages business logic, database interactions, and API endpoints. Node.js was chosen because of its non-blocking, event-driven architecture, which makes it ideal for real-time applications.

Authentication: JWT (JSON Web Tokens) are used to provide secure user authentication and authorization. Several endpoints are defined for user administration, messaging, and video call signaling.

5.1.3 Real-time Communication (Socket.io)

Socket.io enables real-time, bidirectional communication between client and server. It is required for services such as instant messaging and presence notifications.

- **Connection Management:** Manages user connections and disconnections.
- **Message Broadcasting:** This ensures that messages are sent in real time to all participants in a conversation.
- **Attendance and Typing Indicators:** Updates and shows users' online statuses and typing activities.

5.1.4 Video calls (WebRTC)

WebRTC is used for peer-to-peer video communication, allowing users to make real-time video calls directly from their browser.

- Signaling: Socket.io exchanges signaling data to create and maintain WebRTC connections.
- Peer Connection: Manages direct media transfers between users.

5.1.5 Database (MongoDB)

MongoDB is a NoSQL database used for storing user data, message history, and other application data.

- User Data: Stores information like user profiles, authentication tokens, and settings.
- Message History: Keeps records of all messages exchanged between users, ensuring data persistence and retrieval.
- Indexes: Implemented for efficient querying and retrieval of data.

5.2 Security Measures

Security is a key component of Link Mine, with numerous levels of security integrated throughout the design.

- End-to-End Encryption: Encrypts communications and calls from sender to recipient, limiting unwanted access.
- Data Encryption: Uses AES to encrypt data both at rest and in transit.
- Two-factor authentication (2FA) provides an additional degree of protection for user accounts.

5.3 System Workflow

5.3.1 User Registration and Authentication:

- Users provide their personal information upon registering.
- Authentication is done via JWT.
- Two-factor authentication is optional, but encouraged.

5.3.2 Real-time Messaging:

- Users submit messages using the client interface.
- Messages are delivered to the server using Socket.io.
- The server sends the messages to their designated recipients.
- Messages are saved in MongoDB.

5.3.3 Video Calling:

- Users may initiate a video call from the chat interface.
- Signaling data is transferred via Socket.io.
- WebRTC creates a peer-to-peer connection for video calls.
- Audio and video feeds are immediately sent between users.

5.3.4 Machine learning enhancements:

- Machine learning algorithms examine incoming communications to detect spam.
- User behavior is analyzed for irregularities in order to discover potential security concerns.
- Users receive personalized recommendations.

6. TESTING AND EVALUATION

Link Mine uses thorough testing methodologies to verify its dependability, security, and performance. This section describes the testing procedures used for each component, highlighting key technologies and libraries that are important for the program.

6.1 Unit Testing

Unit testing checks the functionality of individual components in isolation. Link Mine's unit tests cover the following:

Check API endpoints in Node.js and Express.js to ensure they handle HTTP requests correctly and provide anticipated results. Middleware, check that middleware services like authentication and data validation work properly.

Socket.io

Test Socket.io's server for proper handling of user connections and disconnections. Ensure that events are emitted and received as meant by it.

jQuery

DOM Manipulation, test jQuery methods that manipulate the DOM to verify they work properly. Event Handling, ensure that event listeners added using jQuery reply properly.

6.2 Integration Testing.

Integration testing ensures that the various components of the program function together effectively.

Use Node.js, Express.js, and MongoDB for user authentication. Test the whole authentication process, from user registration to login and token validation.

Message Handling ensures that messages are properly saved in MongoDB and used via API calls.

Socket.io & WebRTC

Real-Time Messaging: Ensure that messages sent over Socket.io are delivered to the intended users in real time. Verify the signaling procedure for WebRTC video calls.

6.3 Secure Testing

Security testing discovers and addresses possible vulnerabilities.

Crypto.js

- Data Encryption: Ensure that data encryption and decryption using AES are carried out appropriately.
- Hashing: Before storing passwords, ensure that they are properly hashed.

General Security ensures JWTs are issued and verified appropriately. Storage Security ensures that important data is not kept in local Storage, but rather in more secure storage systems.

7. Future Development

- Advanced Machine Learning Models: Using more advanced models to improve security and user experience.
- Blockchain Integration: Investigating the usage of blockchain for decentralized communications and enhanced security.
- Cross-Platform Compatibility: Bringing the application to more platforms, such as desktop apps and different mobile operating systems.
- Screen sharing and group adding options. Status update option development

CONCLUSION

"Link Mine" represents a significant advancement in the realm of secure messaging applications, addressing the growing need for privacy and data protection in digital communication. Through research, thoughtful design, and implementation, this application showcases the potential of cutting-edge technologies to deliver a secure, user-friendly, and intelligent messaging platform.

The development of "Link Mine" involved a comprehensive approach, incorporating industry-leading practices and innovative techniques. The implementation of robust end-to-end encryption mechanisms, coupled with a decentralized peer-to-peer network architecture, ensures that user data remains confidential and secure throughout the communication process. By eliminating the need for a central server to manage and store sensitive information, the application mitigates the risks associated with centralized systems and minimizes the potential for large-scale data breaches.

Furthermore, the integration of machine learning techniques into "Link Mine" represents a pioneering step towards enhancing security features and improving the overall user experience. By leveraging advanced algorithms and models, the application can perform tasks such as anomaly detection, user behavior analysis, and proactive security measures, contributing to a more intelligent and adaptive messaging platform.

The adoption of state-of-the-art technologies, including React for the front-end user interface, Node.js for the back-end server, Socket.io for real-time communication, and WebRTC for secure video calling, demonstrates the project's commitment to delivering a robust and scalable solution. This technological stack not only ensures seamless real-time communication but also facilitates cross-platform compatibility and future extensibility.

Throughout the development process, rigorous testing and evaluation procedures were employed to ensure the reliability and resilience of "Link Mine." Extensive security audits and penetration testing were conducted to identify and mitigate potential vulnerabilities, further reinforcing the application's commitment to data protection and user privacy.

As the demand for secure communication channels continues to grow, "Link Mine" stands as a pioneering solution, catering to the needs of individual users seeking a safe and private messaging environment. With its robust architecture, cutting-edge technologies, and integration of machine learning techniques, "Link Mine" sets a new standard for secure messaging applications, paving the way for future innovations in the field of secure digital communication.