

# **Information Security Lab Project Report**

Submitted in Partial Fulfillment of requirements for the Award of  
Degree of Bachelor of Technology in Computer Science and Engineering

Submitted by

**Kushagra Agarwal (21BCP358)**

**Harsh Shah(21BCP359)**

**Raj Randive(21BCP378)**

**Vikas Yadav(21BCP379)**

Submitted To

**Dr. Rutvij Jhaveri**



**Department of Computer Science and Engineering**

**School of Technology**

**Pandit Deendayal Energy University, Gandhinagar**

**November 2023**

---

---

## TABLE OF CONTENTS

<b>Sr.No</b>	<b>Content</b>	<b>Page Number</b>
<b>1</b>	Introduction	2
<b>2</b>	Problem Statement	3
<b>3</b>	Objective & Significance	3
<b>4</b>	Project in Detail	4
<b>5</b>	Technologies used	8
<b>6</b>	Complexity Analysis	10
<b>7</b>	Future Scope	11
<b>8</b>	Conclusion	12
<b>9</b>	References	13

# ChatApp

---

## Introduction :

In an era of increasing cyber threats, security is a critical component of safeguarding digital assets and sensitive information. Weak security can lead to unauthorized access, data breaches, and security vulnerabilities. We have created a ChatApp that provides end-to-end encryption with intrusion detection and can provide privacy to users on their chats and messages.

End-to-end encryption ensures that messages are only readable by the sender and receiver, and intrusion detection can help to identify and block unauthorized attempts to access messages. This can help to protect users from a variety of attacks, including phishing, malware, and data breaches.

In addition, a ChatApp with end-to-end encryption can also help to protect users' privacy. By encrypting messages, the ChatApp can prevent third parties from reading or listening to conversations. This can be important for users who want to keep their conversations private, such as those who are discussing sensitive topics or who are concerned about their privacy being invaded.

Overall, a ChatApp that provides end-to-end encryption with intrusion detection can be a valuable tool for protecting users from cyber threats and for keeping their conversations private.

---

Here are some additional benefits of using our ChatApp:

- **Increased security:** Intrusion detection makes it much more difficult for unauthorized individuals to access your messages.
- **Improved privacy:** End-to-end encryption prevents third parties from reading or listening to your conversations.
- **Enhanced trust:** End-to-end encryption can help to build trust between users and businesses.

## Problem Statement :

*Developing a Secure and Privacy-Conscious Chat Application with End-to-End Encryption and Intrusion Detection Mechanisms*

## Objective:

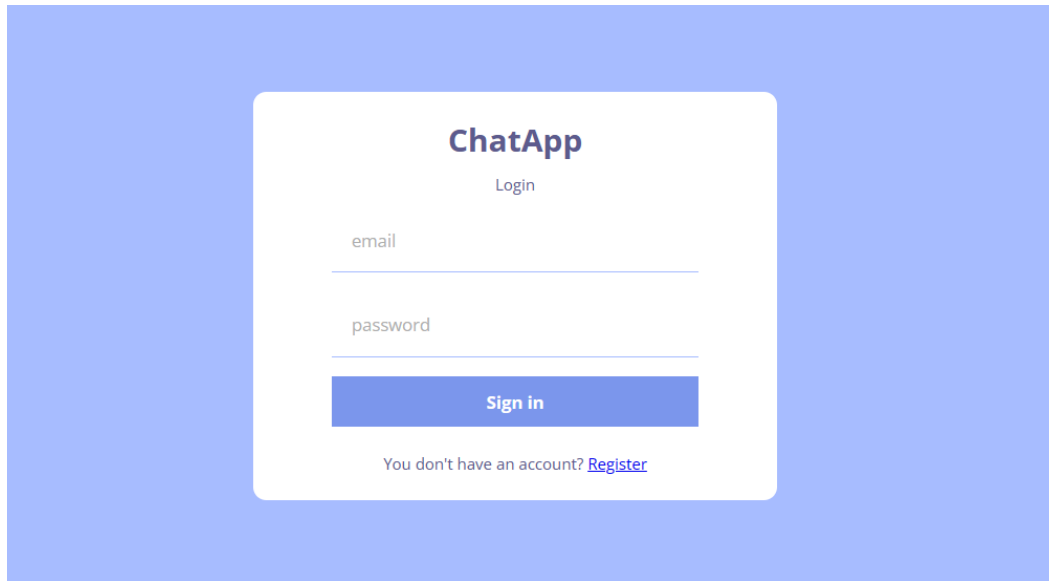
This project aims to develop a secure chat application with end-to-end encryption, intrusion detection mechanisms, and a user-friendly interface. The key goals include conducting comprehensive performance and security testing, gathering user feedback through usability testing, producing a detailed project report, and ensuring scalability to provide a secure and scalable communication platform that safeguards user data and interactions while delivering an exceptional user experience.

## Significance:

This project is significant because it addresses the pressing need for secure and private communication in the digital age. It provides a practical solution to enhance the security and privacy of chat applications while ensuring a positive user experience. The findings and insights from this project can be valuable for individuals, organizations, and developers looking to build or improve secure chat applications in a world where privacy and security are of paramount concern.

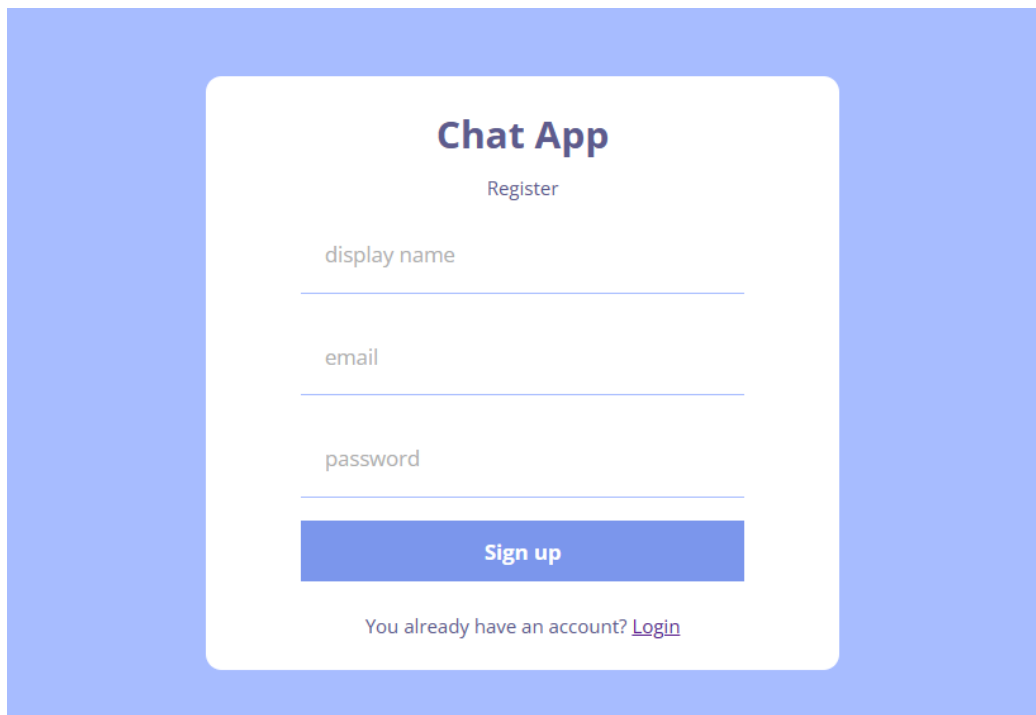
---

## Project In Detail:



The login page for ChatApp features a white card centered on a blue background. The card has the title "ChatApp" in bold, followed by the subtitle "Login". Below the subtitle are two input fields: "email" and "password". A blue "Sign in" button is positioned below the password field. At the bottom of the card, there is a link that says "You don't have an account? [Register](#)".

*Figure 1: Login page*



The register page for Chat App features a white card centered on a blue background. The card has the title "Chat App" in bold, followed by the subtitle "Register". Below the subtitle are three input fields: "display name", "email", and "password". A blue "Sign up" button is positioned below the password field. At the bottom of the card, there is a link that says "You already have an account? [Login](#)".

*Figure 2: Register Page*

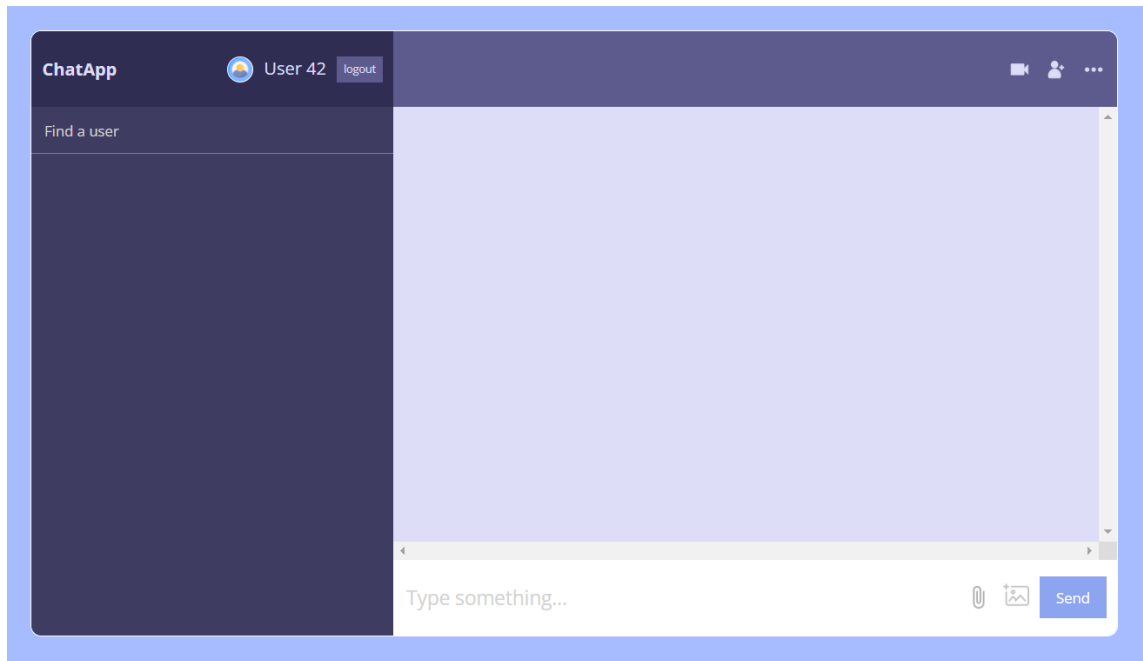


Figure 3: User Onboarding Screen

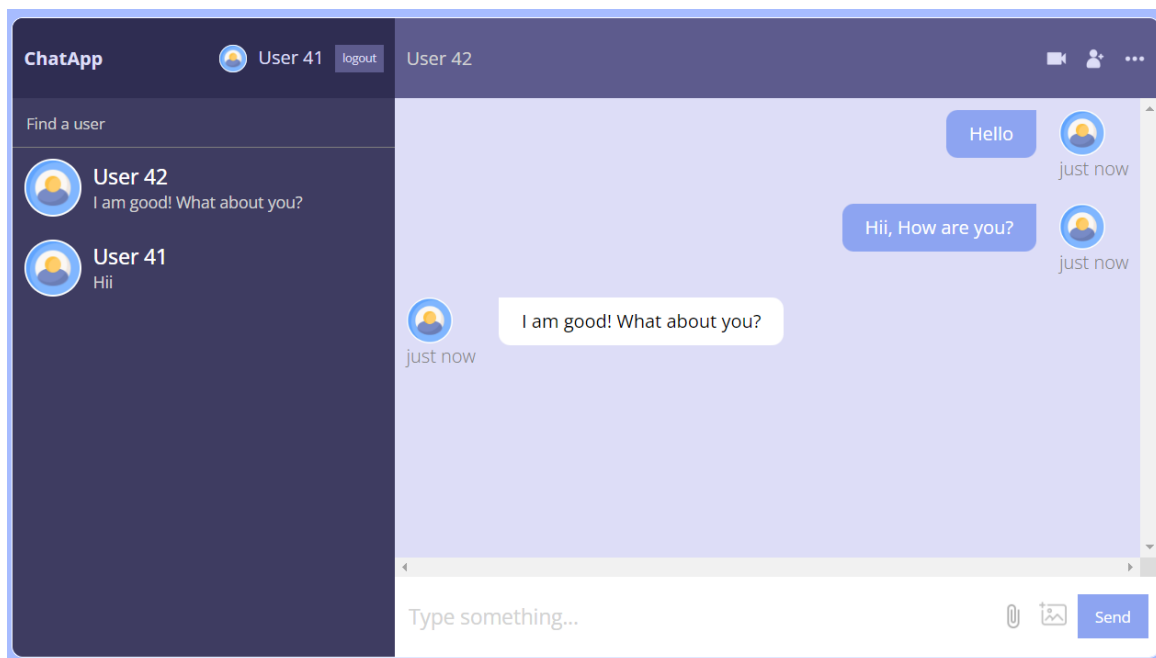


Figure 4: Chat between 2 people (a)

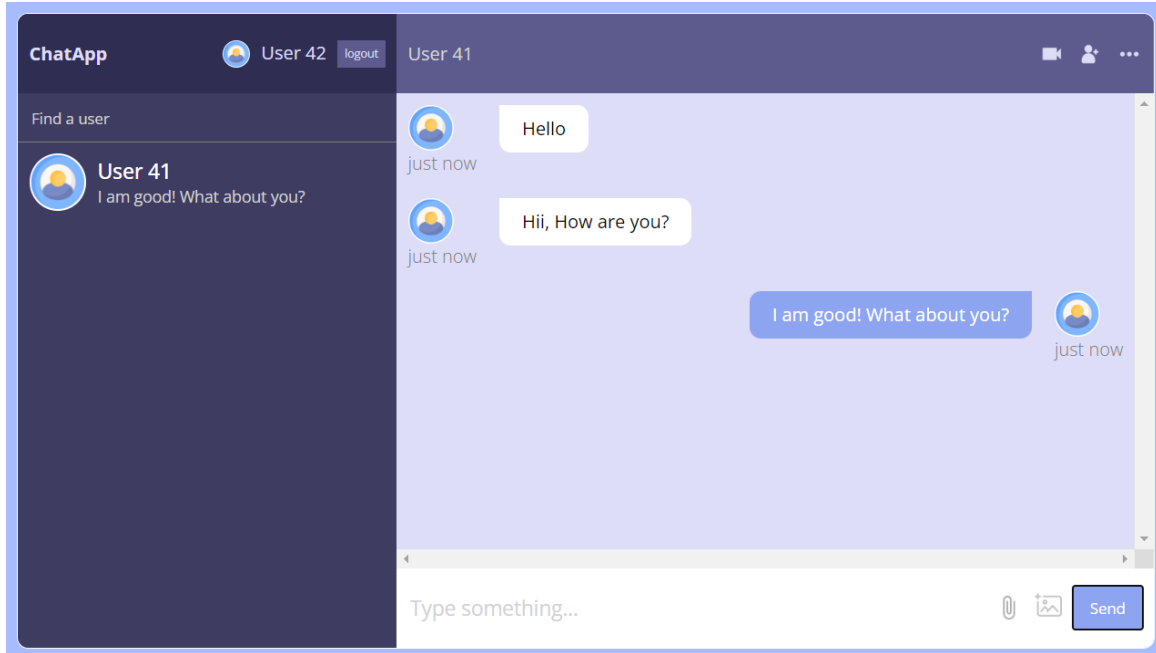
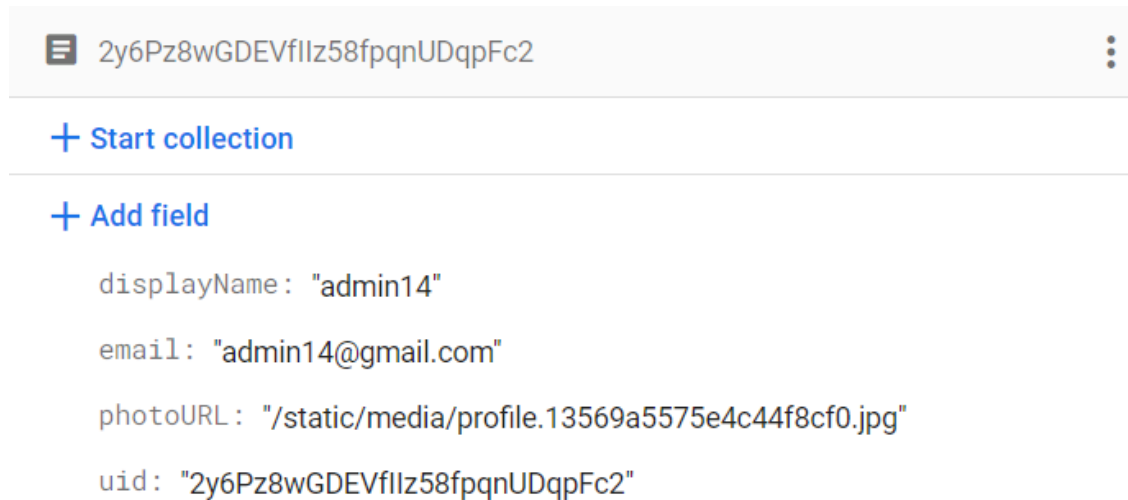


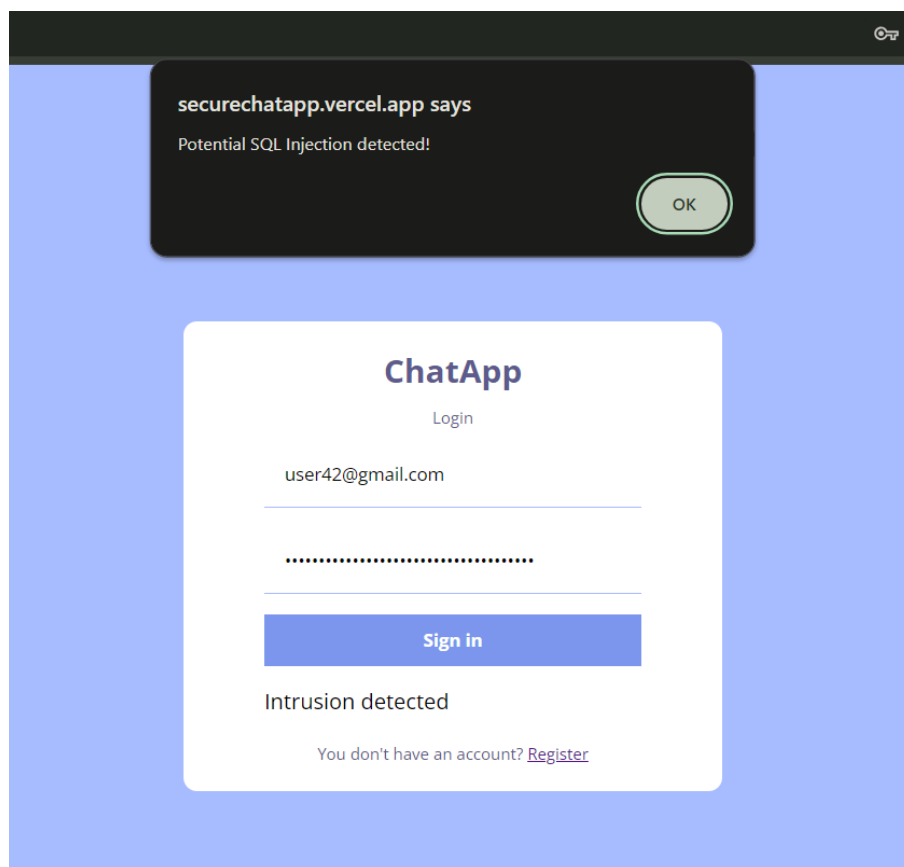
Figure 5: Chat between 2 people (b)

chats	7GgAk2l2ngZoLOPeJ2HdHERfT2424dgqk6uH0SXAfrzoVSvF6JjnPfq1
+ Add document	+ Start collection
7GgAk2l2ngZoLOPeJ2HdHERfT2424dg... >	+ Add field
CWKV00I5NBabIKVzbPHUtYZrQJe21Wc...	▼ messages
Hgrnrwh9D1PozQJdHHd165bzLf43CWK...	▼ 0
LKhW0Wf9E5QayuJmoqh5Fm0kXJ2Hgr...	date: 14 October 2023 at 14:30:57 UTC+5:30
tcIMeoGmPDW1fW0H1V3c7yRWDa2CNI...	id: "62d45fc4-6b2c-4edf-b630-822920700bbd"
ximY771JTKgLPjvV36fx3282Uj432y6...	senderId: "7GgAk2l2ngZoLOPeJ2HdHERfT242"
yYrn9QUgdxWcuUeaTPPdtxdL8mn2nX4...	text: "3QPILKH1uKLCJSvf9GnP0Q=="
yYrn9QUgdxWcuUeaTPPdtxdL8mn2yYr...	▼ 1
	date: 14 October 2023 at 14:31:09 UTC+5:30
	id: "cc71b8dc-6cd1-4c46-8c7a-523394e7e84a"
	senderId: "4dgqk6uH0SXAfrzoVSvF6JjnPfq1"
	text: "jHHTmwB7KFIJk8llaRaTVg=="
	▼ 2
	date: 14 October 2023 at 14:33:11 UTC+5:30
	id: "4aa4529c-afa1-4613-ad22-dbd12f3b3978"
	senderId: "7GgAk2l2ngZoLOPeJ2HdHERfT242"
	text: "k/qr4owUfbmcDmeQ8S7UDA=="

Figure 6: Encrypted Chats stored in database



*Figure 7: User details stored with encryption*



*Figure 8: Intrusion Detection: SQL Injection*



---

## Technologies Used:

1. **CryptoJS:** CryptoJS is a JavaScript library that provides cryptographic functionality. It is free, open-source, and has been used by millions of developers around the world.
2. **Intrusion detection:** Regular expressions are used to detect Intrusion.
3. **Database:** For database storage, we have used Firebase.
4. **Front End:** ViteJS provides the structure to the program.
5. **Designing:** SCSS (Sassy Cascading Style Sheets) provides the designing to the structure

## CryptoJS Working:

Here's the workflow according to the project:

- **Import CryptoJS library:** We start by importing the CryptoJS library at the beginning of our code.
- **Initialization Vector (IV) and Key:** We define an empty IV (Initialization Vector) and create a key by hashing the string "Message" using SHA-256. It's important to note that the IV should ideally be a random value for more secure encryption.
- **encryptData function:** In this function, we take data as input and check whether it's a string or an object. If it's a string, we directly encrypt it using the AES algorithm with CBC (Cipher Block Chaining) mode and PKCS7 padding. If it's an object, we first convert it to a JSON string and then encrypt it. The result is returned as a Base64-encoded string.
- **decryptData function:** This function takes an encrypted string as input and decrypts it using the AES algorithm with the same IV, key, CBC mode, and PKCS7 padding. The decrypted result is converted to a UTF-8 string and returned.

---

## Intrusion Detection System Working:

Here's the workflow according to the project:

- **Signature Based Detection**

**Pattern Matching:** The test method of regular expressions is used to match input data against the predefined attack signatures. If a match is found, it indicates the potential presence of an attack pattern.

**Custom Attack Signatures:** The code allows for the definition of custom attack signatures (in the `attackSignatures` array) for different types of attacks. This is a fundamental concept in signature-based intrusion detection.

**Looping Through Signatures:** The code iterates through the array of attack signatures, testing each one against the input data to check for matches. This approach is typical in signature-based detection systems.

**Immediate Response:** If a match is found for any of the attack signatures, the `detectAttack` function returns `true`, indicating a potential attack. This reflects the principle of taking immediate action when a known attack pattern is identified.

**False Negative and False Positive:** The code can potentially produce both false negatives (fail to detect a real attack) and false positives (incorrectly identify benign data as an attack). Effective signature-based detection systems aim to minimize false negatives while managing false positives.

**Extensibility:** The code is designed to be extensible, allowing for the addition of more attack signatures for different types of attacks. This aligns with the concept of continuously updating and expanding signature databases to address evolving threats.

---

- **Detection Function: *detectSqlInjection***

We have developed a function called `detectSqlInjection(inputString)` to identify and flag potentially malicious user inputs. This function analyzes the `inputString` provided by the user and checks it against predefined patterns that commonly indicate SQL injection attempts. The patterns include:

- **Pattern 1: Special Characters Detection**

This part of the detection mechanism focuses on detecting special characters often used in SQL injection attacks, such as single quotes, double hyphens, and hash symbols. We use a regular expression pattern to identify these characters in the input string.

- **Pattern 2: SQL Keywords Detection**

In addition to special characters, we also check for specific SQL keywords that might be indicative of malicious queries. Keywords such as `SELECT`, `INSERT`, `UPDATE`, `DELETE`, and `DROP` are commonly used in SQL injection attempts.

We use a regular expression pattern to search for these keywords in the input string.

## Complexity Analysis

- **Encryption & Decryption (SHA256 + AES):**  $O(n)$
- **Sending & Receiving message:** This takes  $O(n)$  time because we are encrypting/decrypting every message. Without encryption/decryption it would take  $O(1)$ .
- **Storing & Retrieving from Database:** It depends on the Internet speed of the user.

---

## Future Scope :

The future scope of this project holds immense potential for further growth and development. Enhancements like video and voice calling, secure file sharing, and group chat functionality will make the application more versatile and user-friendly. The integration of blockchain technology can bolster data security, while advanced intrusion detection mechanisms will ensure ongoing protection against evolving threats. Additionally, globalization efforts will extend the application's reach, catering to users worldwide. These developments will position the project for continued success and a wider user base.

Here are some future prospects and potential areas of growth for this project:

- **Video and Voice Calling:** Expand the application to include video and voice calling features for a more comprehensive communication experience.
- **Intrusion Prevention System:** Intrusion Prevention Systems (IPS) proactively identify, block, and mitigate security threats in real-time through signature-based and anomaly-based detection, inline deployment, and immediate response to safeguard network and host environments.
- **Secure File Sharing:** Enhance the application by enabling users to securely send and receive files with end-to-end encryption.
- **Group Chats:** Implement group chat functionality to facilitate collaborative conversations among multiple users.
- **Blockchain Integration:** Explore integrating blockchain technology for secure message storage and verification.
- **Advanced Intrusion Detection:** Continuously improve intrusion detection mechanisms, potentially utilizing machine learning for more sophisticated threat detection.
- **Localization and Globalization:** Adapt the application for international users by offering support for multiple languages and regions, increasing its global reach and appeal.

---

## **Conclusion:**

We have embarked on a journey to create a secure and privacy-conscious chat application, and we're pleased with the progress we've made. Our project has successfully incorporated end-to-end encryption and an intrusion detection mechanism to safeguard user data and communications.

Looking ahead, the project has vast potential for growth and expansion. With the inclusion of features like video and voice calling, secure file sharing, and group chats, we envision a more versatile and comprehensive communication platform. The integration of blockchain technology, advanced intrusion detection, and a focus on globalization will further enhance the project's security and reach.

We are excited about the future of this project and are committed to continuous improvement, responsiveness to user feedback, and staying at the forefront of security and technology trends. Our vision is to provide a secure, user-friendly, and globally accessible communication solution for all. Together, we look forward to shaping a more secure and interconnected digital world.

---

## Link to our project:

**Live Deployment:** <https://securechatapp.vercel.app/>

**Code:** <https://github.com/shaharsh624/ChatApp>

## References:

**Paper:** [Design and Implementation of AES and SHA-256 Cryptography for Securing Multimedia File over Android Chat Application](#)

**Youtube Tutorial:** [Chat App using React and Firebase](#)

**Article:** [How to Make a Really Secure Messaging App like Signal](#)