SRM UNIVERSITY AP

Java Programming

Project Report on

“ Chat App”

Submitted in partial fulfillment for the award of the degree in

Bachelor of Technology
in
Computer Science and Technology

Submitted by:

Suman Kumar	-	AP21110010283
Ayon Sarkar	-	AP21110010286
Krish Srivastava	-	AP21110010302
Paras Agarwal	-	AP21110010324
Aniruddha Dewangan	-	AP21110011194
Nageshwar Prasad Yadav	-	AP21110011195

OBJECTIVE

The objective of the Java project "Chat App" is to develop a robust and user-friendly chat application that enables real-time communication between users. The goal is to provide a seamless and reliable platform for individuals to connect, chat, and share information in a secure manner.

- Real-time Communication: The primary objective is to provide instant messaging capabilities, allowing users to engage in real-time conversations. The Chat App will facilitate the quick and reliable exchange of text messages, images, files, and other media.

- Security and Privacy: A significant objective is to prioritize the security and privacy of users' data. The Chat App will implement robust encryption techniques and authentication mechanisms to safeguard sensitive information and ensure secure communication channels.

- Performance and Scalability: The objective is to create a high-performance Chat App capable of handling concurrent users efficiently. The application will optimize network communication, utilize efficient data structures, and employ scalable architecture to deliver reliable performance under various usage scenarios.

TOOLS AND TECHNOLOGY USED

The following tools and technologies were used to develop this project:

- Java: The main programming language used for developing the Chat App is Java. Java is a popular and versatile language for developing various types of applications.
- Swing: The repository utilizes the Swing framework, which is a graphical user interface (GUI) toolkit for Java. Swing provides a set of components and utilities for creating the graphical interface of the Chat App.
- Socket Programming: The project utilizes socket programming to establish network connections and enable communication between clients and servers. Socket programming allows for real-time messaging and data transmission over networks.
- Client-Server Architecture: The Chat App implements a client-server architecture, where clients connect to a server to exchange messages and data. This architecture allows for centralized communication management and facilitates real-time updates across clients.
- Networking: The project involves networking concepts and libraries for establishing connections, sending and receiving data, and handling network-related operations.

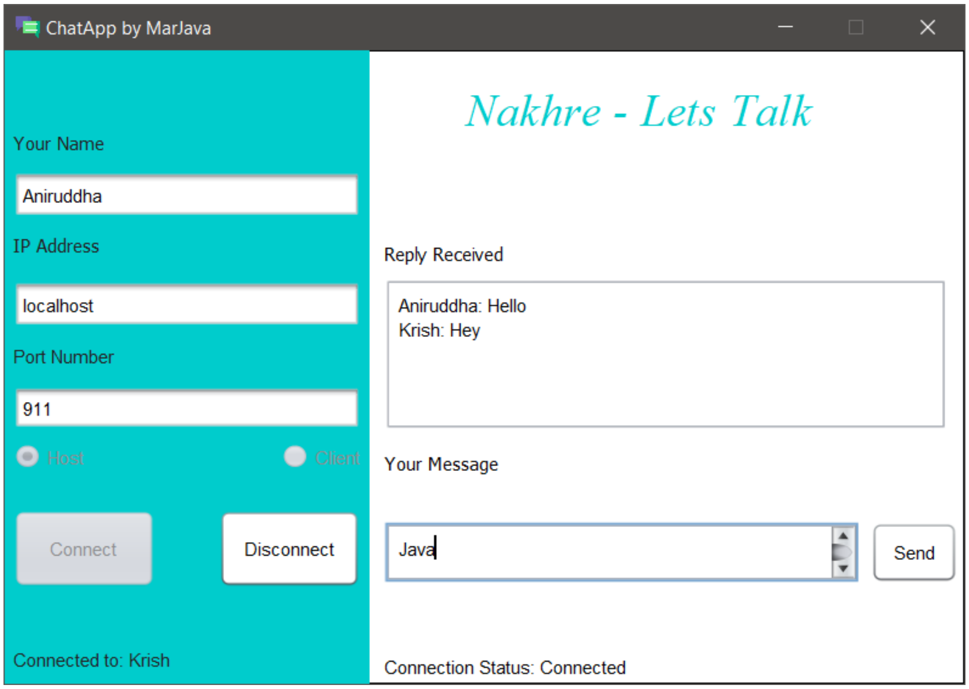
SOURCE CODE

GitHub Repository – https://github.com/retr0-kernel/JAVA-CHATAPP

Main Code - https://github.com/retr0-kernel/JAVA-CHATAPP/blob/main/ChatApp.java

SAMPLE OUTPUT

- Host Side –



Your Name: Aniruddha

IP Address: localhost

Port Number: 911

Reply Received

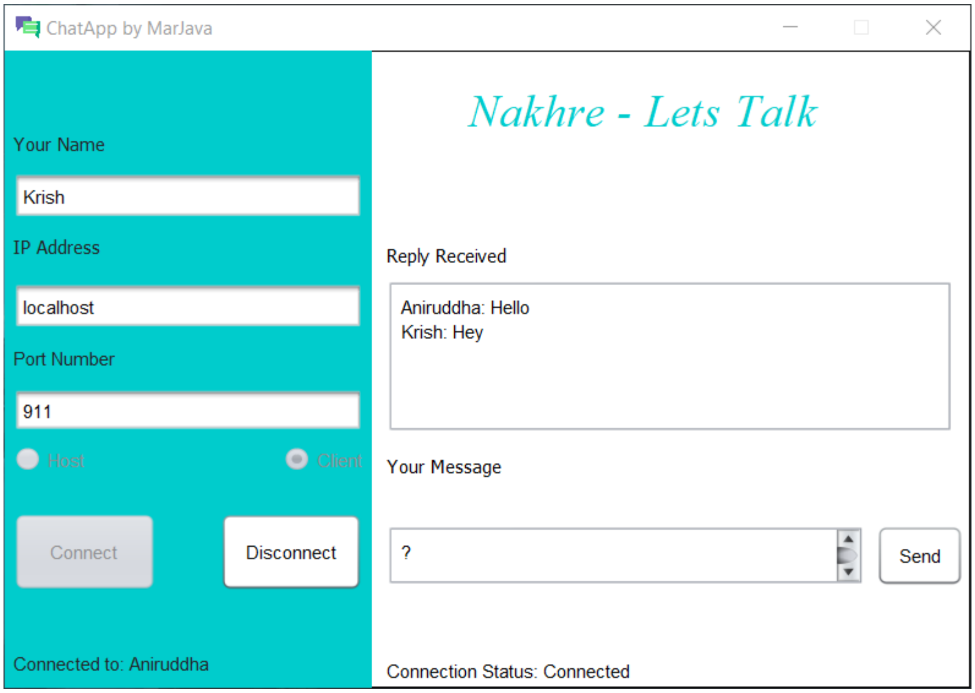
```
Aniruddha: Hello
Krish: Hey
```

Your Message: Java

Connected to: Krish

Connection Status: Connected

- Client Side –



Your Name: Krish

IP Address: localhost

Port Number: 911

Reply Received

```
Aniruddha: Hello
Krish: Hey
```

Your Message: ?

Connected to: Aniruddha

Connection Status: Connected

FUTURE WORK

There are several avenues for future development and enhancement of the "JAVA-CHATAPP" project:

- User Authentication: Implement a secure user authentication system to ensure that only authorized users can access the Chat App. This can include features like username/password login.

- Encryption: Enhance the security of the Chat App by incorporating end-to-end encryption for message transmission. This would protect the privacy and confidentiality of the users.

- File Sharing: Extend the functionality of the Chat App to support file sharing between users. This would allow users to send and receive files of various formats.

- Group Chats: Implement group chat functionality, enabling users to create and participate in group conversations. This feature would enhance the social aspect of the Chat App.

- User Interface Enhancements: Continuously improve the user interface by incorporating user feedback and conducting usability tests. Enhance the visual aesthetics, streamline navigation, and introduce features that enhance user experience.