Java Chat Application – Project Report

Introduction

The Java Chat Application is a real-time messaging system developed using Java, Socket Programming, Threads, and JavaFX. It allows multiple clients to connect to a server and communicate simultaneously in a secure and user-friendly manner. The application supports both group chat and private messaging.

Abstract

The project demonstrates the implementation of a client-server architecture where a server handles multiple client connections. Clients can send messages encrypted using a Caesar cipher, which the server decrypts before broadcasting. The GUI is built with JavaFX, providing an interactive platform for messaging. This application is a practical example of network programming, multithreading, and GUI development in Java.

Tools Used

- Java (JDK 11+): Programming language
- JavaFX 25: GUI development
- Socket Programming: Client-server communication
- Threads: Handling multiple clients simultaneously
- Command Prompt / IDE: Compile & run code

4. Steps Involved in Building the Project
5. Setup Server
    - Implemented ChatServer.java using ServerSocket to listen for client connections.
    - Created a ClientHandler thread for each client to manage communication.

6. Setup Client
    - Implemented ChatClient.java using JavaFX GUI.
    - Created input field, chat area, and send button for message input.
7. Connect Client to Server
    - Clients connect to server via TCP sockets on port 1234.
    - Server maintains a list of active clients.
8. Implement Messaging
    - Group messages broadcasted to all clients.
    - Private messages sent using @username message.
    - Messages encrypted using Caesar cipher for basic security.
9. Run Application
    - Start the server first.
    - Run multiple client instances to test group and private chat.
    - Verify message encryption and GUI functionality.

Conclusion

The Java Chat Application successfully demonstrates a real-time client-server chat system with GUI and encryption. It showcases essential concepts of network programming, multithreading, GUI design, and basic encryption techniques. The project can be extended to include file sharing, database storage, and advanced encryption in the future.