**INITIAL PROJECT REPORT**

*Dissertation submitted in fulfilment of the requirements for the Degree of*

# BACHELOR OF TECHNOLOGY in

**COMPUTER SCIENCE AND ENGINEERING**

**– DATA SCIENCE WITH MACHINE LEARNING**

**TOPIC :: SIMPLE CHAT APPLICATION**

By

**Name: Vishwanth Reddy**

**Registration No: 12210506**

Supervisor

**Waseem Ud Din Wani**



**School of Computer Science and Engineering**

Lovely Professional University

Phagwara, Punjab (India)

October 2023

# DECLARATION STATEMENT

I hereby declare that the work reported in the Assignment Project entitled "SIMPLE CHAT APPLICATION” in partial fulfilment of the requirement for the award of Degree for Bachelor of Technology in Computer Science and Engineering – Data Science with Machine Learning at Lovely Professional University, Phagwara, Punjab is an authentic work carried out under supervision of my research supervisor Waseem Ud Din Wani. I have not submitted this work elsewhere for any degree or diploma.

I understand that the work presented herewith is in direct compliance with Lovely Professional University’s Policy on plagiarism, intellectual property rights, and highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this dissertation represents authentic and honest research effort conducted, in its entirety, by me. I am fully responsible for the contents of my dissertation work.

**VIshwanth Reddy**

**R.No:12210506**

# TABLE OF CONTENTS

1. Introduction
2. Objectives And Scope of Project
3. Methodology
4. Conclusion

# INTRODUCTION

In a digitally connected world, communication is at the heart of human interaction. Instant messaging and chat applications have become an integral part of our daily lives, enabling us to connect with friends, family, colleagues, and even strangers across the globe in real-time. These applications have revolutionized the way we communicate, providing a platform for quick, efficient, and engaging conversations. A simple chat application is a lightweight, user-friendly program designed to facilitate text-based communication between individuals or groups over the internet. It serves as a virtual space where users can exchange messages, share information, and engage in discussions. These applications are used for a wide range of purposes, from personal conversations to professional collaboration.

# OBJECTIVES AND SCOPE OF THE PROJECT

Objectives:

The objectives of the simple chat application project are as follows:

* Basic Networking Understanding: Gain a foundational understanding of networking concepts, including sockets and client-server architecture.
* Real-time Communication: Implement a chat system that enables realtime text-based communication between multiple clients.
* User Authentication: Create a basic user authentication system to allow users to log in with a username or handle.
* Message Exchange: Develop the ability for users to send and receive text messages within the chat application.
* Chat History: Store and display a chat history to allow users to view previous messages and conversations.
* User-Friendly Interface: Design a simple, user-friendly interface for both the client and server sides to ensure an intuitive user experience.
* Error Handling: Implement error-handling mechanisms to manage

network issues, disconnections, or unexpected events gracefully.

* Scalability: Structure the project in a way that makes it easy to scale up and add additional features or capabilities in the future

Scope:

* Server-Client Architecture: The project focuses on creating a basic serverclient architecture where clients connect to a central server to exchange messages.
* Text-Based Communication: The chat application supports text-based messages, excluding multimedia or file sharing capabilities.
* Limited User Authentication: User authentication is limited to the provision of usernames or handles and is not a comprehensive user management system.
* Single Chat Room: The project operates as a single chat room with all connected clients in the same environment. It does not include features for creating multiple chat rooms or private conversations.
* Basic Error Handling: Error handling mechanisms are implemented to ensure the application handles disconnections or network issues to some extent.
* Simple User Interface: The project includes a basic, text-based user interface for clients and a console-based interface for the server.
* Java Programming: The entire project is implemented in Java, focusing on the Java Socket API for networking.
* No Security Features: The project does not address security concerns such as encryption or authentication. It's meant for educational purposes and does not offer production-level security.
* Single Chat History: The chat application maintains a single chat history that is not saved beyond the current session. It does not include features for exporting chat logs or advanced search capabilities.
* Standalone Application: The project aims to create standalone chat application code that can serve as a starting point for further development or learning but does not include integration into larger systems.

# METHODOLOGY

1. Define Requirements : Understand the project's objectives and scope. Define the core features of the chat application.

Consider user requirements, such as real-time messaging, chat history, and user authentication.

2.Select Technology Stack:

choose the programming language and libraries or frameworks for development.

For a basic chat application, Java and Java's Socket API are suitable choices.

3.Server-Client Architecture:

Design the server-client architecture. The server acts as a central hub for message exchange among clients.

4.Networking Basics:

Learn and understand fundamental networking concepts, including sockets, IP addresses, and ports.

5.Server Development:

Start with server development:

Create a server socket that listens for incoming client connections.

Handle client connections and set up communication channels.

Implement user authentication or username registration.

Develop message broadcasting and chat history management.

6.Client Development:

Begin client development:

Create a client application to connect to the server.

Establish socket connections to the server.

Implement user interface components, such as input fields and message display areas.

Set up message sending and receiving.

7.User Authentication:

Develop a basic user authentication system. Users may register a username or handle for identification.

8.Real-Time Communication:

Enable real-time communication between clients. Implement message sending and receiving functionalities.

9.Chat History:

Store chat history on the server for users to access. You can use data structures like ArrayList to manage the chat history.

10.Error Handling:

Implement error-handling mechanisms on both the client and server sides to gracefully manage issues like disconnections or network errors.

**Algorithm Implementation (Pseudocode):**

Start

|

|--[User]--> Enter a username

| |

| |--[User]--> Click "Connect"

| | |

| | |--[Client]--> Attempt to connect to the server

| | | |

| | | |--[Server]--> Accept the connection

| | |

| | |--[Client]--> Show connected message

| | |

| | |--[User]--> Type a message

| | | |

| | | |--[User]--> Click "Send"

| | | | |

| | | | |--[Client]--> Send the message to the server

| | | | | |

| | | | | |--[Server]--> Broadcast the message to all clients

| | | | |

| | | | |--[User]--> Continue typing or type "exit" to leave

| | |

| | |--[User]--> Type "exit" to leave the chat

| | | |

| | | |--[Client]--> Send "exit" to the server

| | | | |

| | | | |--[Server]--> Broadcast "username has left the chat"

| | | | |

| | | | |--[Client]--> Close the connection

| | |

| | |--[User]--> Click "Connect" to start a new session

|

|--[User]--> Exit the application

|

End

# CONCLUSION

The development of our simple chat application has been a rewarding journey that has allowed us to apply various data structures and algorithms while creating a practical and functional software system. This project has served as an educational experience that has enhanced our understanding of key programming concepts and provided valuable insights into software design, user experience, and real-world application development