**Simple Chat Application**

This guide walks you through the process of creating an application that sends messages back and forth between a browser and a server. WebSocket is a thin, lightweight layer above TCP. This makes it suitable for using “subprotocols” to embed messages. In this guide, we use STOMP messaging with Spring to create an interactive web application. STOMP is a subprotocol operating on top of the lower-level WebSocket.:

**Abstract:**

The **Chat Application** project focuses on creating a platform for seamless messaging and interaction. Users can exchange text messages, share files, and engage in group conversations. Here are the key points:

1. **User Registration**: New users can create accounts by providing basic details like username, email, and password.
2. **Real-Time Messaging**: The application enables instant messaging between users. Messages are delivered in real time.
3. **Group Chats**: Users can participate in group chats, allowing multiple people to communicate simultaneously.
4. **File Sharing**: The system supports file attachments, allowing users to share images, documents, and other files.
5. **Security**: Implementing secure authentication and encryption ensures user privacy.

**Implementation Details:**

* **Front-End Frameworks**: Chat applications can be built using various front-end technologies such as **Angular**.
* **Back-End Services**: Backend services can be developed using **Java (Spring Boot)**.
* **Database**: Choose a suitable database system like **MySQL**.
* **Real-Time Communication**: Implement **Web Sockets, Messaging (FCM)** for real-time messaging.
* **Deployment**: Deploy the application on platforms like **Tomcat.**

**Conclusion:**

The Chat Application project bridges the gap between individuals, allowing them to connect and communicate effortlessly. Whether you’re building a personal chat app or integrating chat features into an existing platform, understanding the architecture and design principles is crucial. [Dive into the full project documentation to explore the intricacies of building a robust chat application](https://www.slideshare.net/MuhammadAshiqurRahma/chat-application-full-documentation)

# System Configuration: -

# H/W System Configuration: -

# **Processor -** Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz 2.50 GHz

Speed - 1.1 Ghz

RAM - 16 GB

Hard Disk - 20 GB

Key Board - Standard Windows Keyboard

Mouse - Two or Three Button Mouse

Monitor - SVGA

For more understanding: https://spring.io/guides/gs/messaging-stomp-websocket