**Chat application**

**Description**

This is a real-time messaging application built with **React** and **Node.js**, designed for instant communication between users. The application ensures efficient messaging experience by leveraging modern web technologies and scalable infrastructure.

The core features include real-time messaging using **WebSockets**, allowing instant message exchange without delays. It also supports user authentication with **JWT-based** security, ensuring secure access and user identity verification. Additionally, the system provides message status indicators such as Delivered and Seen, giving users real-time feedback on message delivery. To enhance user experience, the application includes unread message counts, allowing users to keep track of new messages efficiently. As messages brokers, the application integrates **RabbitMQ/Kafka**, enabling efficient message handling and distribution across multiple services.

**C4 model diagrams:**

1. [**Level 1**](#_Level_1_(System)
2. [**Level 2**](#_Level_2_(Container)
3. [**Level 3**](#_Level_3_(Component)
4. [**Level 4**](#_Level_4_(Code)

# Level 1 (System context diagram):A diagram of a chat system AI-generated content may be incorrect.

# Level 2 (Container Diagram):**A diagram of a chat service AI-generated content may be incorrect.**

# Level 3 (Component Diagram):

Chat service:

**A diagram of a computer application

AI-generated content may be incorrect.**

User service:

**A diagram of a software company

AI-generated content may be incorrect.**

Notification service:

**A diagram of a service

AI-generated content may be incorrect.**

# Level 4 (Code Diagram):

Chat service:**A diagram of a message

AI-generated content may be incorrect.**

User service:

**A diagram of a computer program

AI-generated content may be incorrect.**

Notification service:

**A diagram of a service

AI-generated content may be incorrect.**