

SYSTEM DESIGN DOCUMENT FOR CHAT APPLICATION (MERN STACK WITH SOCKET.IO)

BY-KARAN NEGI (IIT DHANBAD)

CONTENT

- 01** INTRODUCTION
- 02** ARCHITECTURE OVERVIEW
- 03** COMPONENTS
- 04** SETUP AND INSTALLATION
- 05** DEPENDENCIES AND LIBRARIES
- 06** CONCLUSION
- 07** ABOUT ME

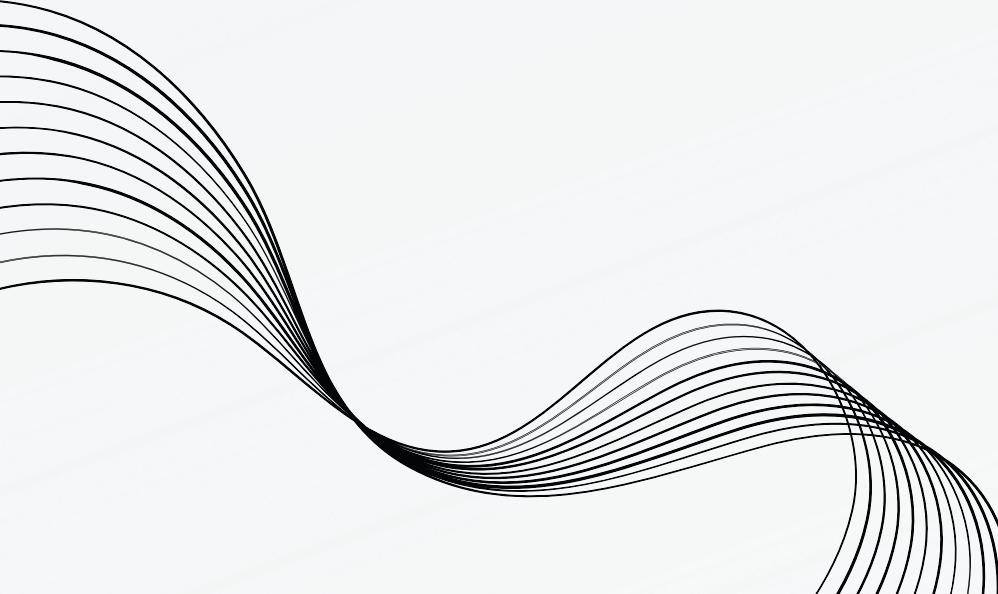
INTRODUCTION

- This document provides an overview of the system design for a chat application built using the MERN stack (MongoDB, Express.js, React, Node.js) along with Socket.IO for real-time communication. The application allows users to register, log in, send messages, and see user presence in real-time.

ARCHITECTURE OVERVIEW

The architecture consists of three main layers

- Frontend: Built with React, it provides a user interface for the chat application.
- Backend: Built with Node.js and Express, it handles API requests, authentication, and interactions with the database.
- Database: MongoDB is used to store user and message data.



COMPONENTS

Frontend

- React: A JavaScript library for building user interfaces.
- Tailwind CSS: Utility-first CSS framework for styling the application.
- Socket.IO Client: Enables real-time communication with the server.

Backend

- Node.js: JavaScript runtime for building server-side applications.
- Express.js: Framework for building APIs.
- Socket.IO: Library for real-time web applications, allowing for bi-directional communication.

Database

- MongoDB: NoSQL database for storing user and message data.

Real-Time Communication

- Socket.IO: Facilitates real-time, event-driven communication, enabling instant message delivery and user presence tracking.

SETUP AND INSTALLATION

Prerequisites

- Node.js: Ensure you have Node.js installed. You can download it from nodejs.org.
- MongoDB: Set up a MongoDB database. You can use MongoDB Atlas for cloud hosting or install MongoDB locally.

1-Clone the Repository

Copy code-
`git clone https://github.com/your-repo/chat-app.git
cd chat-app`

2-Backend Setup

1. Copy code-**cd server**
2. Copy code-**npm install**

3. Create a .env file in the server directory with the following variables: plaintext

Copy code-**DATABASE_URL=your_mongodb_connection_string**
PORT=4000
FRONTEND_URL=http://localhost:3000

4. Copy code-**npm run dev**

3-Frontend Setup

1. Copy code-**cd client**
2. Copy code-**npm install**
3. Copy code-**npm start**

4-Access the Application

Open your browser and navigate to <http://localhost:3000> to access the chat application.

DEPENDENCIES AND LIBRARIES

1-Backend

- Express.js: For building the API endpoints and managing server-side logic.
- Mongoose: For interacting with MongoDB, providing a schema-based solution to model the data.
- Socket.IO: For real-time web socket communication between client and server.
- Bcrypt.js: For hashing passwords and secure user authentication.
- dot env: For managing environment variables.

2-Frontend

- React: For building the user interface components.
- React Router: For navigation within the app.
- Axios: For making API requests to the backend.
- Tailwind CSS: For styling the application.
- Socket.IO Client: For connecting to the Socket.IO server for real-time messaging.

WHY THESE LIBRARIES?

- **MERN Stack:** Chosen for its popularity, scalability, and the ability to use JavaScript across both client and server.
- **Socket.IO:** Essential for enabling real-time features such as instant messaging and user presence.
- **Bcrypt.js:** Necessary for securely handling user authentication.
- **Mongoose:** Simplifies MongoDB interactions, making data manipulation easier and safer.

CONCLUSION

- This system design document outlines the architecture and components of the chat application using the MERN stack and Socket.IO. By following the setup instructions, developers can quickly get the application up and running, while the dependencies and libraries used provide a robust framework for real-time communication and data management.

ABOUT ME



Karan Negi

IIT Dhanbad

My name is Karan Negi, and I am currently pursuing my B.Tech from IIT Dhanbad. I am a passionate full-stack developer with a keen interest in the field of technology. I enjoy building innovative applications and constantly strive to enhance my skills in both frontend and backend development.

Remy
Marsh