Software Engineering Lab 2020
Project by the accidental coders.
Chat application "Legal, privately owned child of hangouts and discord."
By group().

TABLE OF CONTENTS

Introduction

Purpose

- The purpose of this project is to implement a real time "chatting" application with a fully functional NodeJS based server, where users with internet connection can engage in a one-on-one conversation or group conversation between multiple users.
- The development of this project is centered on the development of a message protocol that would allow the users to properly log in users, send messages and perform system maintenance

Intended audience and reading suggestions

- This project is a prototype of a chat application that allows multicast chatting and it is restricted within the college premises. This has been implemented under the guidance of college professors and teaching assistants. This app is useful for the conversation between the college residents.

Project Scope

- The purpose of the online chatting app is to create a convenient and easy way to chat with other users. It also simulates multicast chatting. Every chat or data will be secure with end to end encryption. We will try to provide a bot which you can talk to ,ask questions and much more. Above all , we hope to provide users with a comfortable experience .

Project Perspective

- The system to be developed here is a Chat facility. It is a centralized system. It is a Client-Server system with centralized database server. All ocal clients are connected to the centralized server via LAN.
- There is a two way communication between different clients and servers.
- This chat application can be used for group discussion. It allows users to find other logged in users.

- Can work in areas with just a Local Area Network and does not need a proper internet connection. This system is useful when nearby users feel to chat among themselves (eg.in a conference hall).

References

Google and TA support

Overall Description :-

Product features

- Instant chat service. Both online and over LAN.
- Users can send and receive multimedia messages to other users privately or over a chat room.
- Businesses can implement their own bots to cater to the needs of their audiences.

Operating Environment

- Centralised Database (hosted at mongodb.com).
- Operating System : Android, iOS
- Database Type : NoSQL.

Design and implementation constraints

~To write schema~

Data Flow Diagram

Data Flow Diagram

System Features

External Interface Requirements

User Interfaces

- Front-end software: React native
- Back-end software: MongoDB, ExpressJS, Nodejs

Hardware Interfaces

- Android Phones, iOS powered devices.

Software Interfaces

Following software used for the chat application

Software used	Description
Database	MongoDB is chosen for its flexibility and easy scalability. It serves as an online storage for chats and user-details.
Server-side Back-end	ExpressJS framework is used because not only it is simpler and faster, but also the most popular node framework for server-side applications.
GUI	React Native, which is used extensively to build cross platform mobile applications with javascript.
Operating System	Android as well as iOS

Communication Interfaces

This Project works on both android and iOS