CHITKARA UNIVERSITY

PROJECT Proposal
ON
Chat-App

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FOR
Integrated Project(CS203)

SUBMITTED TO:
Ms Aditi Sharma Mam

SUBMITTED BY:
Prince Kumar(2210992090)

Index

PROJECT SYNOPSIS:

Olympic-Info : All you need to know about the summer Olympic games

Problem Statement :

In today's digital world, seamless communication is essential for personal and professional interactions. Existing chat platforms often lack **privacy, customization, and seamless integration** with specific user needs. Our proposed **Chat App** aims to bridge this gap by offering **secure, real-time messaging with an intuitive interface** and additional features like **group chats, multimedia sharing, and end-to-end encryption**.

Title of Project :

Chat-App: Seamless & Secure Real-time Communication.

Objective & Key

Objective: Chat App is designed to provide a secure, fast, and user-friendly messaging platform that supports real-time conversations, group interactions, and multimedia sharing. It will emphasize data privacy and ease of use, making it an ideal choice for both personal and business communication.

Key Learnings:

1. **Real-time Messaging System:** Learn to implement WebSockets for instant message delivery.
2. **User Authentication & Security**: Secure login with JWT authentication and end-to-end encryption for messages.
3. **Database Management:** Efficient storage and retrieval of messages and user data using MongoDB/PostgreSQL.
4. **Multimedia Sharing:** Ability to send images, videos, and documents within chats.
5. **Scalability & Deployment:** Deploy the app using Docker & cloud platforms like AWS or Firebase.

Technology Stack:

1. **Frontend:** React.js / Next.js – For building an interactive UI.
2. **Backend**: Node.js with Express.js – To handle API requests and real-time chat.
3. **Database: MongoDB / PostgreSQL –** For storing messages, users, and chat logs.
4. **Real-time Communication:** WebSockets (Socket.io) **–** To enable live chat functionality.
5. **Authentication & Security:** JWT (JSON Web Token), bcrypt **–** For user authentication and data security.
6. **Deployment:** Docker, AWS, or Firebase – To make the app scalable and accessible.

References:

- Official Documentation of React, Express.js, WebSockets, and MongoDB
- Security Best Practices from OWASP
- Guides and tutorials from platforms like GeeksforGeeks, MDN, and Stack Overflow

---