📄 Project Requirements Document

Project Title: Multi-User Chat App with Message History and Authentication

Intern Duration: 1.5 months (6 weeks)

Intern Name: Adithya V
Intern Role: Full-stack Developer Intern
Supervisor: Veer Yadav and Sandeep Rawat

🧭 Objective

Develop a web-based real-time multi-user chat application that includes user authentication and message history. The app should allow users to register, log in, join a global chat room or individual rooms, and send/receive messages in real time with message history persisted.

🧩 Key Features

1. User Authentication

- Register / Login / Logout
- Password hashing (e.g., bcrypt)
- Session management or JWT tokens
- Basic profile info (e.g., username, email)

2. Real-Time Chat

- One global chatroom (basic)
- Support for multiple rooms or direct messaging
- Socket-based communication (e.g., using **Socket.IO**)

3. Message History

- Messages stored in a database
- Load recent messages when a user joins the room
- Timestamps on messages

4. Frontend

- Simple and clean UI using HTML/CSS and a JS framework (Angular)
- Login/Register pages
- Chat interface with real-time updates
- Message input and display

🛠️ Tech Stack (Suggested)

Area	Technology
Frontend	HTML, CSS, JavaScript (Angular)
Backend	Node.js with Express
Real-Time Comm	Socket.IO
Database	MongoDB & Mysql
Auth	JWT / Express-session
Version Control	Git + GitHub

🧪 Deliverables

Phase	Week	Deliverables
1. Planning & Setup	Week 1	- Finalize tech stack - Set up project repo - Plan architecture and UI
2. Auth System	Week 2	- User registration and login - Authentication middleware - Basic UI for login/register
3. Real-Time Chat	Week 3	- Set up Socket.IO - Enable sending/receiving messages - Basic chat UI

Phase	Week	Deliverables
4. Message Persistence	Week 4	- Store messages in database - Load message history on login/room entry
5. UI Improvements & Rooms (optional)	Week 5	- Add chat rooms or user-to-user messaging - Improve UI/UX
6. Testing & Documentation	Week 6	- Fix bugs - Write README - Deploy on free hosting

---

🔎 Evaluation Criteria

- Functional login and registration system
- Ability to send and receive messages in real time
- Message history works after refresh/login
- Code structure and comments
- Git commit quality and documentation
- Final presentation/demo to the team

📚 Resources

- Socket.IO Docs
- JWT Auth Guide
- MongoDB Docs
- React Docs

✅ Expectations

- Minimum 4-5 commits/week

- Weekly check-ins with mentor
- Open communication for blockers
- End-of-internship presentation/demo
