UNIVERSITY INSTITUTE OF ENGINEERING

Full Stack Project

PulseChat – Real Time Messaging & Collaboration System

Subject Code – 23CSP-339

Submitted To:
Faculty Name: Er. Juned Alam
(E18479)

Submitted By:
Name: Suhan Kumar Singh
UID: 23BCS13955
Section: KRG - 2B
Semester: 5th

CU
CHANDIGARH
UNIVERSITY

Module 4 — Admin + UX polish + quality (read receipts, typing, search)

Goal: admin tools, basic UX signals, search, and tests.

What to do

1. Admin

- Role-based endpoints: GET /api/admin/users, PUT /api/admin/users/{id}/role, DELETE /api/admin/users/{id}.
- Frontend: simple Admin Dashboard to list users and deactivate.

2. Read receipts & typing indicator

- Read receipts: when chat window opens, send STOMP /app/chat.read with messageIds -> backend updates readBy[] -> broadcast updated message state to chat subscribers.
- Typing: send ephemeral STOMP /app/chat.typing with isTyping -> others show "X is typing…".

3. Search and filters

- Create text index on Message text field in MongoDB: db.messages.createIndex({ text: "text" }).
- Endpoint GET /api/search/messages?q=...&chatId=....
- Frontend: search bar in ChatWindow.

4. **Tests and QA**
    - Backend unit tests for auth, chat persistence.
    - Manual test checklist (register/login, message persistence, file upload, group join/leave).
    - Write a small E2E smoke script (Postman collection or Playwright test if time).

Deliverables

- Admin can manage users.
- Read receipts and typing indicator functional.
- Message search works.

Acceptance criteria

- Admin actions enforced by role checks.
- Read receipts update message state for participants.
- Search returns relevant messages.