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

Module 3 — Group chats, file sharing, notifications (priority after MVP)

Goal: groups + attachments + instant notifications.

What to do

1. **Groups**
   - Extend Chat model to support group type and name, adminId, members[].
   - Endpoints: POST /api/groups, PUT /api/groups/{id}/members, GET /api/groups/{id}.
   - Frontend: UI to create group, add members, show members list.
2. **File upload (keep it simple)**
   - Option A (fast): local file storage in server /uploads and serve static files — implement file size limit and whitelist (images, pdf, mp4).
   - Option B (recommended if comfortable): AWS S3 or similar. But for a month deadline start with local storage; you can swap later.
   - Endpoint: POST /api/files/upload -> returns fileUrl, type, id. Save metadata in Files collection.
   - When sending a message with attachment, include attachments array with fileUrl.
3. **Notifications**
   - Use WebSocket to publish notification events to a user-specific topic /user/{userId}/queue/notifications or broadcast in chat-topic with a notification DTO.
   - Frontend: show in-app small badge; also use the browser Notification API for desktop (request permission).

Deliverables

- Ability to create group chats and send files.
- Notifications shown on new messages.

Acceptance criteria

- File attached in chat and downloadable/viewable.
- Group members receive messages in real-time.
- Browser shows a notification (if permission allowed) on incoming message.