

Assignment: Backend Developer - Chat Application API

Objective

The goal of this assignment is to evaluate your backend development skills by implementing a chat application API that supports real-time messaging, group chats, user authentication, and file attachments.

Due date and timings : 06-03-2025 , 3:00 PM IST.

Requirements

You are required to develop a backend service using **Node.js** with **Socket.io** for real-time communication and a database of your choice (**MongoDB, PostgreSQL, or MySQL**). The API should support the following features:

1. User Authentication

- Users should be able to **register** and **login** using JWT authentication.
- JWT tokens should be used for secure API access.

2. Real-Time Chat (One-to-One & Group)

- Implement real-time messaging using **Socket.io**.
- Support **one-to-one messaging** between users.
- Users should be able to create **groups** and invite other users.
- Group members should be able to send messages within the group.
- Store chat history in the database.

3. File Attachments

- Allow users to **send and receive attachments** (PDF, JPEG, PNG).
- Store files using **cloud storage** (e.g., AWS S3, Firebase, or local storage).
- Save file metadata (URL, type, sender) in the database.

4. API Endpoints

Design RESTful APIs for:

- **User Management**
 - `POST /auth/register` - Register a new user.
 - `POST /auth/login` - Authenticate and return a JWT token.
 - `GET /users` - Get a list of users (protected route).
- **Chat Functionality**
 - `POST /chats/send-message` - Send a message.

- `GET /chats/:chatId` - Fetch chat history.
- **Group Management**
 - `POST /groups` - Create a new group.
 - `POST /groups/:groupId/add-user` - Add a user to a group.
 - `GET /groups/:groupId/messages` - Fetch group messages.
- **File Uploads**
 - `POST /uploads` - Upload and send a file attachment.

5. WebSockets (Socket.io)

- Establish a **real-time** connection with Socket.io.
- Handle events for:
 - `connect / disconnect`
 - `send_message / receive_message`
 - `create_group / join_group / group_message`
 - `send_file / receive_file`
- Ensure messages and files are **broadcasted** to the correct recipients.

7. Deployment & Documentation

- Deploy the application on **Heroku, Render, or Vercel**.
- Provide **Postman API documentation**

Submission Instructions

- Upload your code to a **public GitHub repository**.
- Provide a **README.md** with setup instructions.
- Share a **Postman collection** for testing your APIs.

Good Luck! 🚀