# **Bak-Bak**

Design Document

**Overview:** Bak-Bak is a web-based chat application designed to facilitate real-time conversations among users. It will enable seamless text-based communication, potentially with future expansions into file sharing and even voice/video chat capabilities.

**Target Audience:** The target audience for Bak-Bak is broad. It can serve individuals seeking casual conversation, communities looking to connect around shared interests, and potentially even businesses who want to integrate chat functionality for customer support or team collaboration.

## **1. Features**

### Core Features:

* User Registration/Login: Secure system for new users to create accounts and existing users to log in.
* Chat Rooms: Creation and management of public and private chat rooms, with clear controls over participation.
* Direct Messaging: One-on-one messaging between users.
* Real-time Messaging: Messages sent and received immediately using technologies like Socket.IO.
* Message History: Searchable logs of past conversations within rooms and direct messages.

### Additional Features:

* File Sharing: Ability to send images, GIFs, documents, and other file types.
* Voice/Video Calls: Integrated support for real-time voice or video calls within chat rooms or direct messages.
* Rich Text Formatting: Expanded text editing features beyond bold, italics, and underline.
* Customization: User profiles, themes, and notification preferences.

## **2. Architecture**

Bak-Bak follows a client-server architecture, with the frontend and backend components interacting via RESTful APIs.

### Frontend:

* Framework: React.js
* State Management: Redux
* Routing: React Router
* Styling: CSS and TailWind
* WebSockets: Socket.IO
* HTTP Requests: Axios

### Backend:

* Framework: Express.js
* Database: MongoDB (via Mongoose)
* Authentication: Passport.js
* WebSockets: Socket.io
* API Documentation: Postman

### Deployment:

* Frontend: Hosted on a platform like Vercel
* Backend: Hosted on a platform like Heroku

## **3. User Interface Design**

* Focus on Usability: Intuitive layout, clear navigation, and accessible design.
* Group/Channel List: Well-organized display with search functionality
* Chat Area: Clean presentation of messages, with user identifiers and timestamps.
* Responsive: Adapt to various screen sizes (desktop, tablet, mobile).

## **4. Security Considerations**

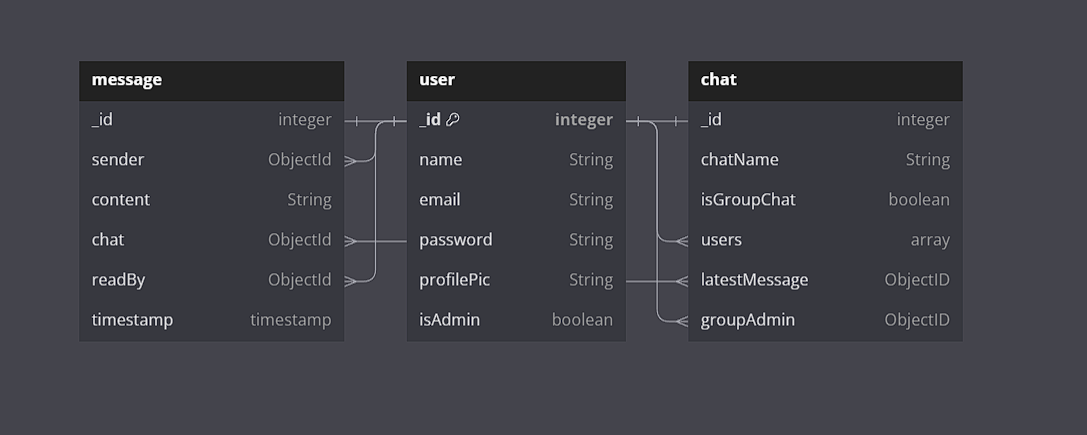
* Input Sanitization: Prevent XSS and other injection attacks.
* Data Encryption: Protect sensitive data at rest and in transit (SSL/TLS).
* Authentication/Authorization: Robust user authentication and controls over room access.
* Moderation Tools (Optional): If public rooms are a core feature, develop filtering, reporting, and banning mechanisms.

## **5. Testing Strategy**

* Conduct integration testing to ensure seamless interaction between frontend and backend.
* Test user flows and edge cases to identify and resolve any bugs or issues.

**6. Database Schema:**

* **User:**
  + **\_id integer**
  + **Name string**
  + **Email string**
  + **Password string**
  + **profilePic string**
  + **isAdmin boolean**
* **Chat:**
  + **\_id integer**
  + **chatName String**
  + **isGroupChat boolean**
  + **Users array of users**
  + **latestMessage message**
  + **groupAdmin user**
* **Message:**
  + **\_id integer**
  + **Sender user**
  + **Content String**
  + **Chat chat**
  + **readBy user**
  + **timeStamp timestamp**

****

## 

## **7. API EndPoints:**

* **Users:**
  + POST /api/users/register: Register a new user.
  + POST /api/users/login: Log in an existing user.
  + POST /api/users/logout: Log out from user account.
  + GET /api/users/:userId: Get user profile.
  + PUT /api/users/:userId: Update user details.
  + DELETE /api/users/:userId: Delete user account
* **Chat:**
  + GET /api/messages/:userId: get messages for a particular user
  + GET /api/chats/:userId: get all the people that user has chatted with
  + POST /api/chats/:userId: send a message to a user

## 

## **8. Technology Stack:**

## **9. Conclusion**

Bak-Bak aims to be a user-friendly and versatile chat platform. Careful attention to security, scalability, and a polished user experience will be crucial for its success.