DESIGN DOCUMENT:

1. Architecture Overview

Echo Hub is built upon the MERN (MongoDB, Express.js, React.js, Node.js) stack:

- **Frontend:** Developed with React.js, Echo Hub provides interfaces for real-time messaging, user authentication, and chat management.
- **Backend:** Powered by Express.js and Node.js, the backend offers RESTful APIs for user authentication, message handling, and database interactions.
- **Database:** MongoDB serves as the database to store user data, chat messages, and related information.

2. Database Design

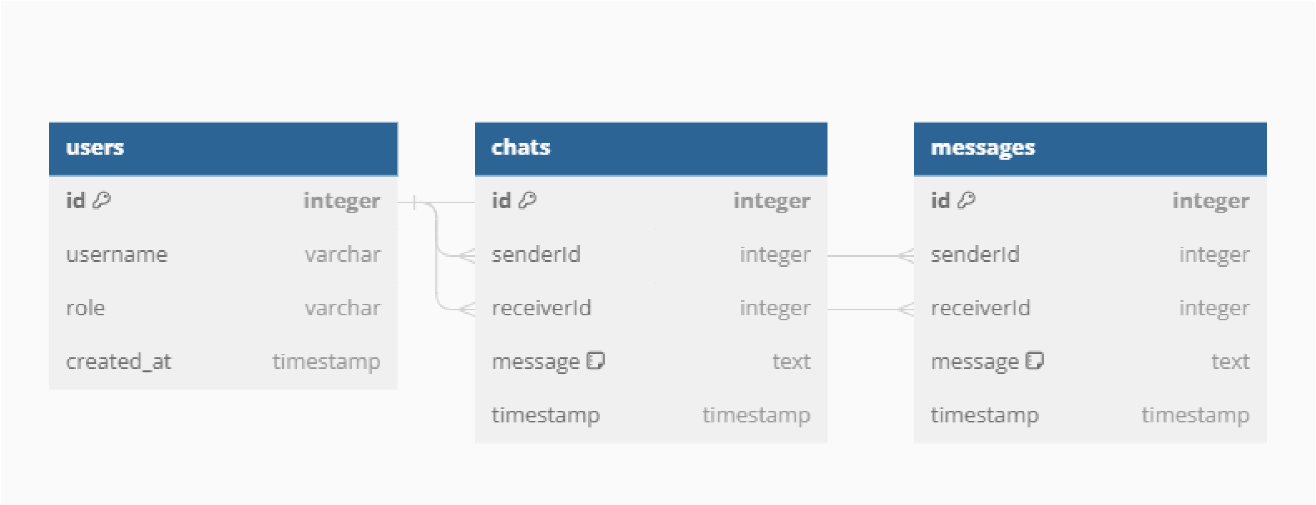
Echo Hub's MongoDB database consists of the following collections:

- **Users Collection:** Stores user information, including name, email, password hash, and preferences.

- **Chats Collection:** Contains information about individual chat sessions between users, including participants and metadata.

- **Messages Collection:** Stores individual messages exchanged within the chat sessions, including sender, receiver, content, and timestamps.

3. Schema Models



users	
id	**integer**
username	varchar
role	varchar
created_at	timestamp

chats	
id	**integer**
senderId	integer
receiverId	integer
message	text
timestamp	timestamp

messages	
id	**integer**
senderId	integer
receiverId	integer
message	text
timestamp	timestamp

4. API Design

Echo Hub's backend offers RESTful APIs for various functionalities:

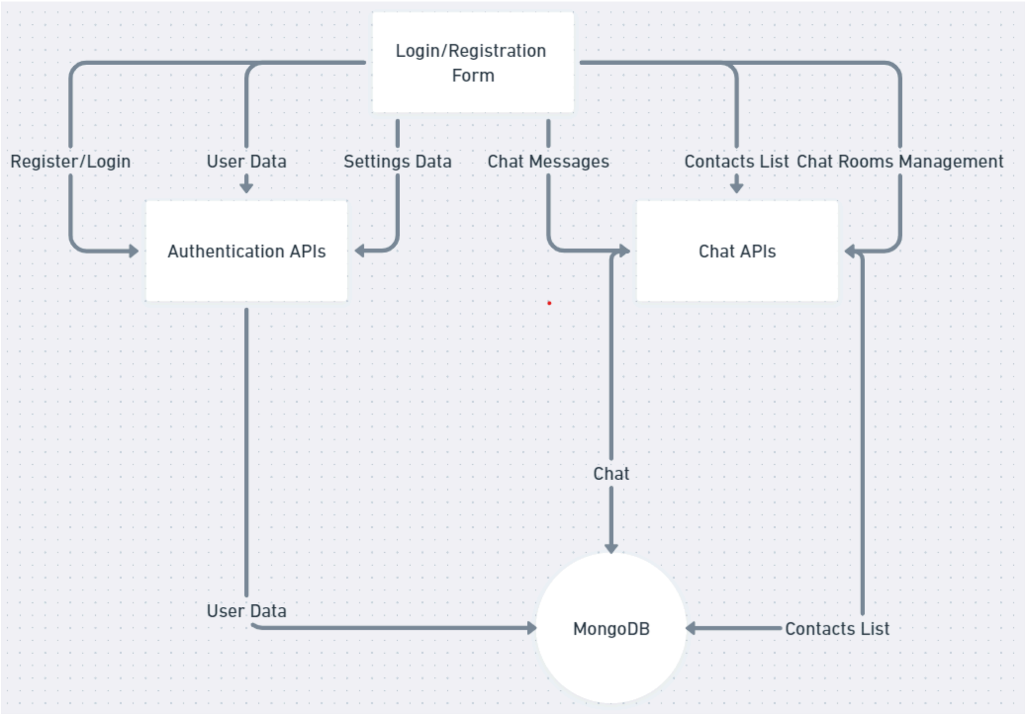
- **Authentication APIs:**
  - `/api/auth/register`: Register a new user.
  - `/api/auth/login`: Authenticate user credentials and generate a JWT token.
  - `/api/auth/user`: Retrieve current user details.
- **Chat APIs:**
  - `/api/chat/messages`: Retrieve and send chat messages.
  - `/api/chat/contacts`: Retrieve user's contacts list.
  - `/api/chat/rooms`: Create, join, and leave chat rooms.

5. Frontend Design

Echo Hub's frontend, developed using React.js and styled with CSS, includes the following components:

- **Login/Registration**: Facilitates user authentication and registration.
- **Chat Interface:** Provides a user-friendly interface for real-time messaging.
- **Contacts List:** Displays the user's contacts for easy communication.
- **Chat Rooms:** Allows users to create, join, and manage chat rooms.
- **Settings:** Enables users to customize their profile and account settings.

6. Data Flow Diagram



7. Deployment

Echo Hub can be deployed on cloud platforms like Heroku, AWS, or DigitalOcean. Frontend hosting services such as Netlify or Vercel, coupled with MongoDB Atlas for database hosting, offer a robust deployment solution.

8. Testing

Unit tests for frontend components using Jest and backend APIs using tools like Mocha/Chai ensure robustness. Additionally, integration tests with Cypress cover end-to-end scenarios.

9. Scalability and Performance

Echo Hub is designed for scalability, employing caching mechanisms, load balancing, and horizontal scaling strategies. Performance optimizations guarantee fast response times and efficient resource utilization.

10. Security

The app prioritizes security by encrypting sensitive data, implementing input validation, safeguarding against common web vulnerabilities, and utilizing secure authentication mechanisms like JWT tokens.

11. Accessibility

Adhering to web accessibility standards ensures usability for all users. Features such as keyboard navigation support and semantic HTML markup enhance accessibility.

12. Maintenance and Support

Regular maintenance, updates, and a robust support system address user inquiries, bug reports, and feature requests promptly, ensuring a seamless user experience.

13. Conclusion

Echo Hub aims to provide users with a secure and efficient platform for real-time communication. By following the outlined design principles and implementing suggested features, the app seeks to enhance the user experience and foster meaningful connections.