**Title: -Talk-Hub**

**Aim:**

To develop a secure, user-friendly, and interactive chat application that facilitates real-time messaging and media sharing, equipped with essential user management and notification features.

**Description:**

Talk-Hub is designed to provide an engaging platform for users to connect and chat in real time. The app supports functionalities such as profile customisation, online presence indication, secure login and signup processes, and password recovery through email OTP verification. The application’s light and dark modes enhance the user experience, while notifications keep users informed of incoming messages and user activity.

**Modules:**

* **Authentication Module:** Includes login, signup, forgot password, and OTP-based password reset.
* **User Profile Management:** Users can edit profiles, set profile pictures, and add personal information.
* **Chat Functionality:** real-time text and media messaging, along with message timestamps.
* **Notification System:** Pop-up notifications for new messages and online user status.
* **Theme Customisation:** Options for dark and light mode to suit user preferences**.**
* **Friend System:** The user can connect with friends and check online presence**.**
* **AI-Based Chatbot**: Including an AI chatbot for user support and interaction.

**Hardware Specifications: -**

* **RAM:** Minimum 4GB (8GB recommended)
* **Processor:** Intel i3 Processor or Higher
* **Hard Disk :** 20GB and above.
* **Internet**: stable broadband connection.

**Software specifications:**

* Any operating system (Windows, Linux, MAC, etc..).
* Google Chrome/Firefox is recommended for this project for better performance and a good interaction with the end user.
* Browsers and other Internet client access to the web applications.

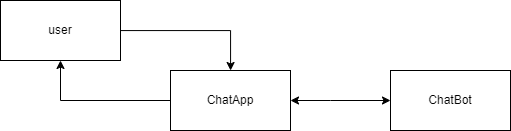
**Front-end (Client-side):**

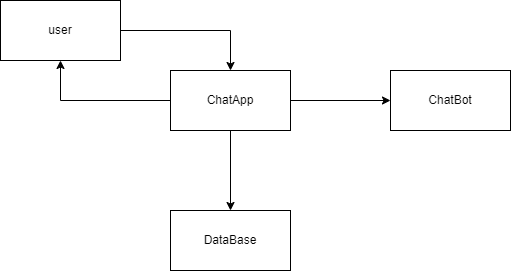
* **React.js**: For creating an interactive, responsive user interface.
* **HTML5 & CSS**: For structuring and styling the application.
* **JavaScript**: For handling client-side logic.

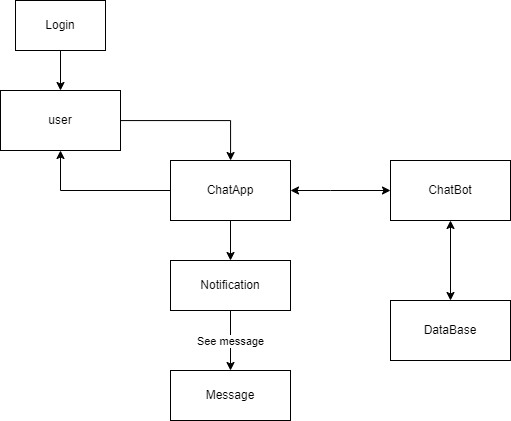
**Back-end (Server-side):**

* **Node.js**: For handling server-side operations and logic.
* **Express.js**: To manage servers and routing.
* **MongoDB**: A NoSQL database to store user profiles, chat history, and settings.
* **Cloudinary API**: For uploading and storing profile pictures and media files shared within chats.

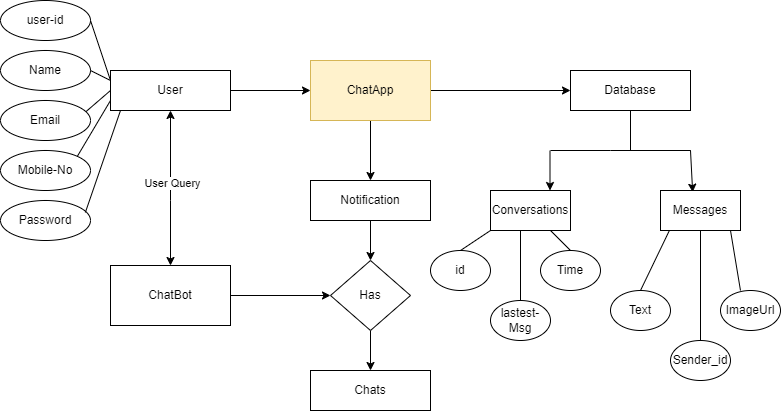
**Data Flow Diagram**

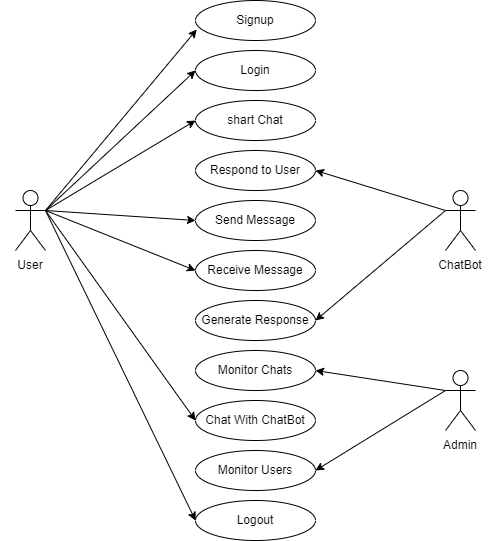
**DFD LEVEL 0**

**DFD LEVEL 1**

**DFD LEVEL 2**

**ER Diagram**

****

**UseCase Diagram**

**Conclusion:**

**Talk-Hub** successfully delivers a modern, real-time communication platform that integrates advanced functionalities with a user-friendly interface Designed to social connections, the application combines instant messaging with the power of an AI-driven chatbot, bridging the gap between human communication and artificial intelligence. user-friendliness is at the core of Talk-Hub's design. The application features an intuitive interface that is easy to navigate, ensuring accessibility for users of all technical backgrounds. a scalable architecture, making it capable of handling a growing user base and increased traffic without compromising performance. Its modular design, powered by the MERN stack (MongoDB, Express.js, React.js, and Node.js), allows for easy maintenance and future enhancements. The application is adaptable and ready for new features, such as voice and video calling, media sharing, and advanced AI integrations.

**Future Scope:**

The future scope of the project circles around maintaining information regarding.

* **Group Chat**: Adding group messaging features.
* **Video and Voice Calls**: Integrating VoIP functionality.
* **Message Reactions**: Allowing users to react to messages.
* **End-to-end encryption**: Enhancing security and privacy for user messages.
* **Location Sharing**: Allowing users to share their real-time location.
* **Voice-Based Chatbot Interaction**: Allow users to communicate with the chatbot through voice commands, making the platform more accessible and engaging.
* **Cloud-Based Scalability**: Use cloud platforms like AWS, Google Cloud, or Azure to support high user traffic, ensuring consistent performance during peak usage.