Project Documentation: Regnum- Elite Couples Chat Room

1. Introduction

This project aims to develop a sophisticated chat room exclusively for elite couples, offering an array of features to enhance their interaction, including video calls, text messaging, and engagement activities such as quizzes.

2. System Architecture

The system architecture of the chat room is divided into several key components:

1. User Sign Up and Authentication

2. Database Storage and Management

3. Features Implementation

3. User Sign Up and Authentication

Frontend Development:

- Sign-Up and Login Forms: The frontend is built using a modern JavaScript framework such as React. Users can sign up or log in through forms that collect essential information like username, password, and email.

- Input Validation: Client-side validation ensures that the user input meets all necessary criteria before submission.

Backend Development:

- JWT (JSON Web Tokens):

- When a user signs up or logs in, the backend (built using Node.js and Express) creates a JWT, which is then sent to the client.

- This token is stored securely on the client side, typically in local storage or a cookie, and is included in the header of every subsequent request to authenticate the user.

Flow:

1. User Sign-Up: User inputs are validated and sent to the backend.

2. User Authentication: Backend validates the credentials.

3. Token Generation: A JWT is generated and sent back to the client.

4. Token Storage: The client stores the token and uses it for authenticated requests.

4. Database Storage and Management

Database Selection: MongoDB

- User Data Storage: MongoDB is used to store user information, including profile details, authentication credentials, and activity logs.

- Chat History: All chat messages, including text, video call records, and engagement activities, are stored in MongoDB collections.

Database Structure:

1. Users Collection:

- UserID (unique identifier)

- Username

- Email

- Password (hashed)

- Profile Information

- JWT tokens (for active sessions)

2. Chats Collection:

- ChatID (unique identifier)

- Participants (UserIDs)

- Messages (array of message objects)

- Timestamps

3. Activities Collection:

- ActivityID (unique identifier)

- Type (e.g., Quiz, Poll)

- Participants (UserIDs)

- Results

- Timestamps

5. Features Implementation

1. Video Call using WebRTC:

- WebRTC: Real-Time Communication (RTC) is achieved using WebRTC. This technology enables peer-to-peer audio and video communication directly between users’ browsers.

- Signaling Server: A signaling server built with Node.js and WebSockets is used to coordinate the connection between peers.

- STUN/TURN Servers: These servers help in establishing the connection and handling NAT traversal.

2. Texting using WebSockets:

- WebSocket Protocol: For real-time messaging, WebSockets are implemented. This allows full-duplex communication channels over a single TCP connection.

- Message Broadcasting: The server handles broadcasting messages to the appropriate participants in a chat room.

3. Engagement Activities:

- Quizzes and Polls: Interactive quizzes and polls are implemented to engage couples. The data from these activities are stored in MongoDB for analytics and user feedback.

- Real-Time Updates: WebSockets are used to ensure that results and updates are broadcasted to all participants in real-time.

6. Detailed Workflow

Sign-Up and Authentication Flow:

1. User Registration:

- User submits registration form.

- Frontend sends a POST request to the backend.

- Backend validates data and creates a new user in the MongoDB.

- A JWT is generated and sent back to the frontend.

2. User Login:

- User submits login form.

- Frontend sends a POST request with credentials to the backend.

- Backend verifies credentials, generates JWT, and sends it to the frontend.

- Frontend stores the JWT for future requests.

Real-Time Communication Flow:

1. Video Call:

- User initiates a video call.

- WebRTC APIs are used to set up the peer connection.

- Signaling data is exchanged via the signaling server (using WebSockets).

- Media streams are directly exchanged between peers.

2. Text Messaging:

- User sends a text message.

- Message is sent to the backend via WebSocket.

- Backend broadcasts the message to all participants in the chat room.

Engagement Activities Flow:

1. Quiz Participation:

- User participates in a quiz.

- Frontend sends quiz responses to the backend.

- Backend processes and stores responses.

- Results are broadcasted in real-time using WebSockets.

7. Conclusion

The elite couples chat room is designed with a robust architecture to ensure seamless user experience and secure communication. By leveraging modern technologies such as JWT for authentication, MongoDB for database management, WebRTC for video calls, and WebSockets for real-time messaging, this platform aims to provide an engaging and interactive environment for its users.