Talky Chat

(Mern Chat App)

📌 Project Overview:

Talky Chat is a real-time chat application backend built using **Node.js**, **Express**, **MongoDB**, and **Mongoose**. It supports essential chat functionalities like user authentication, chat management, and messaging. Authentication is secured using **JWT tokens**, while email-based OTP verification ensures user identity. Users can chat privately, manage their profiles, and receive timely updates.

🔍 Key Functionalities Provided:

- 🔐 **OTP-Based Email Verification** to prevent spam or fake registrations.
- 📝 **User Registration & Authentication** with login and secure password change.
- 👤 **Profile Management**, including username, profile picture, and status updates.
- 💬 **One-on-One Chat System** with support for fetching and sending messages.
- ✉ **Message Handling**, with support for tracking sender and seen status.
- 🔄 **JWT Token Management** (Access and Refresh tokens).
- 🗑 **Account Deletion and Logout** handling.
- 📥 **Welcome Emails** to enhance user onboarding experience.

The backend is modular, scalable, and follows industry best practices, making it ready for future integrations like group chat, push notifications, and video calling.

🔐 Authentication APIs

1. Create OTP

- **Endpoint:** `POST /api/auth/create-otp`
- **Description:** Sends a 4-digit OTP to the user's email.
- **Request Body:** `{ email: string }`

2. Verify OTP

- **Endpoint:** `POST /api/auth/verify-otp`
- **Description:** Validates the OTP sent to the email.
- **Request Body:** `{ email: string, otpCode: number }`

3. Register User

- **Endpoint:** `POST /api/auth/register`
- **Description:** Registers a user only if OTP is verified.
- **Request Body:** `{ username, email, password, phoneNumber }`

4. Login User

- **Endpoint:** `POST /api/auth/login`
- **Description:** Logs in user via email or phone number.

5. Refresh Token

- **Endpoint:** `POST /api/auth/refresh-token`
- **Description:** Issues a new access token from refresh token.

6. Logout User

- **Endpoint:** `POST /api/auth/logout`
- **Description:** Logs out the user and marks them offline.

👤 User APIs

1. Get My Profile

- **Endpoint:** `GET /api/user/profile`
- **Description:** Returns profile details of the logged-in user.

2. Update Profile

- **Endpoint:** `PUT /api/user/update-profile`
- **Description:** Updates user profile fields like username, status, and profile picture.

3. Change Password

- **Endpoint:** `PUT /api/user/change-password`
- **Description:** Allows users to update their password securely.

4. Get All Users

- **Endpoint:** `GET /api/user/all`
- **Description:** Fetches all users except the current logged-in one.

5. Delete Account

- **Endpoint:** `DELETE /api/user/delete-account`
- **Description:** Deletes user account and all related data.

💬 Chat APIs

1. Access Chat

- **Endpoint:** `POST /api/chat/access-chat`
- **Description:** Returns existing one-on-one chat or creates a new chat.
- **Request Body:** `{ userId: string }`

2. Get All Chats

- **Endpoint:** `GET /api/chat`
- **Description:** Retrieves all chats for the logged-in user.

✉ Message APIs

1. Send Message

- **Endpoint:** `POST /api/message/send`
- **Description:** Sends a message to the specified chat.
- **Request Body:** `{ content: string, chatId: string }`

2. Get All Messages

- **Endpoint:** `GET /api/message/:chatId`
- **Description:** Fetches all messages under a chat ID.

🛠 Planned Enhancements

- ⏳ Auto-expire used OTPs after registration.
- 🙅 User block/unblock functionality.
- 👨‍👩‍👧 Group chat with admin and member management.
- 👀 Typing indicators & read receipts via WebSockets (Socket.IO).
- 📣 Notification system for messages.
- 🎥 Voice & Video calls using WebRTC integration.

🧠 Developer: Anirudh T. Anil

🛠️ Tech Stack: Node.js, Express.js, MongoDB, Mongoose, JWT, Nodemailer

🕒 Last Updated: June 28, 2025