

LeafChat Project Structure

LeafChat project consists of a frontend and backend, structured as follows:

Backend (Node.js, TypeScript, Prisma, Socket.io)

- backend/src/index.ts: Main entry point for the backend.
- backend/src/controllers/: Handles authentication (auth.controller.ts) and messaging (message.controller.ts).
- backend/src/routes/: Defines API routes for authentication (auth.route.ts) and messaging (message.route.ts).
- backend/src/db/prisma.ts: Manages database interactions using Prisma.
- backend/src/socket/socket.ts: Implements real-time messaging using Socket.io.
- backend/src/utis/generateToken.ts: Handles authentication token generation.

Frontend (React, TypeScript, Vite, Tailwind)

- frontend/src/main.tsx: Main entry point for rendering the React app.
- frontend/src/App.tsx: Core layout and routing.
- frontend/src/components/: Contains UI components:
 - messages/: Manages message display, input, and conversation view.
 - sidebar/: Manages sidebar, search, and logout features.
- frontend/src/context/: Handles authentication (Auth.context.tsx) and socket connections (SocketContext.tsx).
- frontend/src/hooks/: Custom hooks for API calls (useLogin.tsx, useSendMessage.tsx, etc.).
- frontend/src/pages/: Contains Home.tsx, Login.tsx, and SignUp.tsx.

Flow of the Application

User Authentication

- Users log in or sign up using useLogin.tsx and useSignup.tsx, which call auth.route.ts on the backend.
- Authentication tokens are managed using generateToken.ts and stored in Auth.context.tsx.

Real-Time Messaging

- Messages are sent via useSendMessage.tsx and displayed using Messages.tsx.
- socket.ts on the backend manages real-time message delivery using Socket.io.
- SocketContext.tsx ensures messages update in real-time on the frontend.

UI and State Management

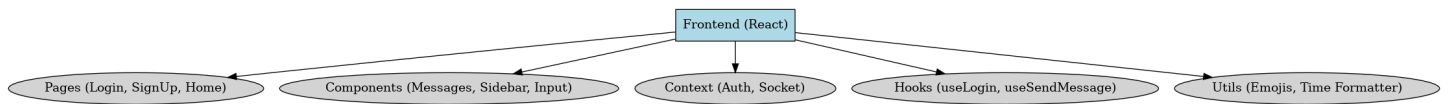
- UI components (sidebar, message input, conversations) interact with Zustand (`useConversation.ts`) for state management.
- Tailwind CSS is used for styling.
- Emojis and timestamps are processed with `emojis.ts` and `extractTime.ts`.

Sound & Theme

- Notification sounds (`notification.mp3`) play when a new message arrives.
- Falling leaves animation is handled via assets (`leaf_01.png`, etc.).

This structure ensures smooth authentication, real-time messaging, and a visually appealing UI.





Backend (Node.js, Express)

Controllers (Auth, Messages)

Routes (Auth, Messages)

Socket.io (Real-Time Chat)

Database (Prisma, PostgreSQL)

ORM Queries

