

1. Project Overview

This project is a real-time messaging service that allows users to communicate instantly via a web-based application. It includes user authentication, message sending, and receiving, as well as group channels. The architecture is built using a client-server model, with a focus on scalability and real-time communication.

2. Features

- **User Authentication:** JWT-based authentication (login and registration).
- **Real-time Messaging:** Using WebSocket (Socket.IO) for instant communication.
- **Group Channels:** Users can create and join channels.
- **Message History:** View previous conversations in real time.
- **Media Uploads:** Support for uploading files and profile images (if implemented).

3. Technologies Used

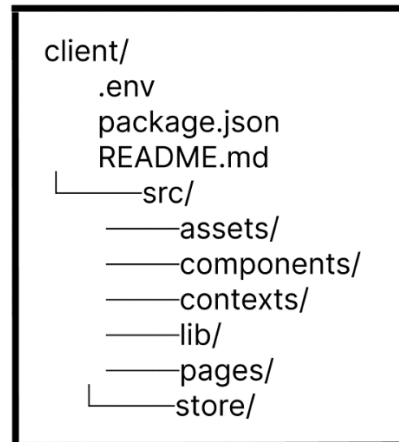
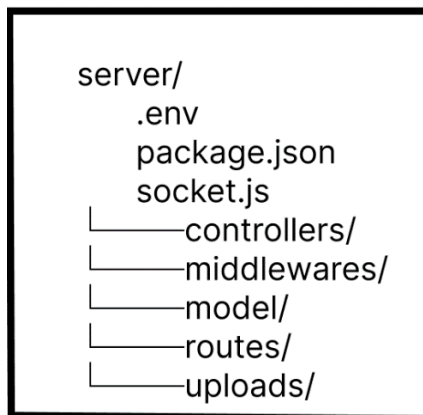
Frontend (Client):

- **React:** For building the user interface.
- **Tailwind CSS:** For styling.
- **Socket.IO:** For real-time communication with the server.
- **REST API:** For handling user authentication and data fetching.

Backend (Server):

- **Node.js:** Backend runtime environment.
- **Express.js:** For building REST APIs.
- **Socket.IO:** For WebSocket-based real-time communication.
- **JWT:** For secure authentication.
- **MongoDB/MySQL:** For database storage (adjust based on what you used).
- **Multer:** For handling file uploads (optional).

4. Folder Structure



5. Setup Instructions

Clone the Repository

Backend Setup (Server)

1. Navigate to the server directory.

```
cd server
```

2. Install the required dependencies.

```
npm install
```

3. Create a .env file in the server/ directory with the following variables:

```
PORT=8747
```

```
JWT_KEY="I am beside u"
```

```
ORIGIN="http://localhost:5173"
```

```
DATABASE_URL="mongodb://localhost:27017/wechat"
```

4. Start the backend server:

```
npm start
```

Frontend Setup (Client)

1. Navigate to the client directory.

```
cd client
```

2. Install the required dependencies.

```
npm install
```

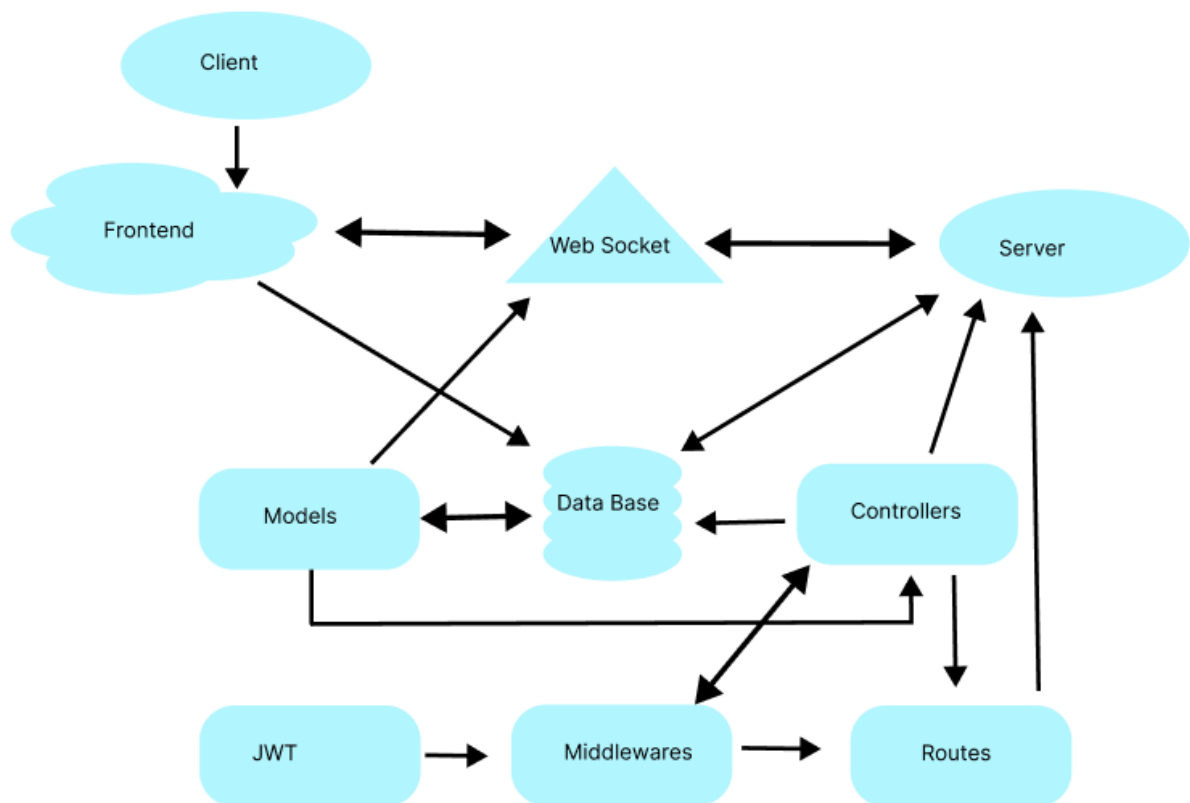
3. Create a .env file in the client/ directory with the following variables:

- VITE_SERVER_URL="http://localhost:8747"
- VITE_SOCKET_URL="http://localhost:3001"

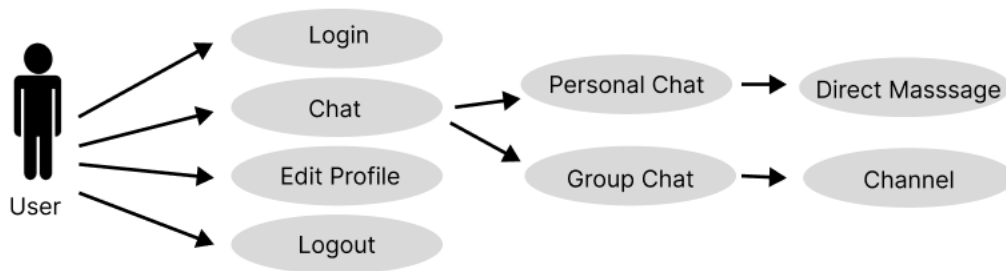
4. Start the frontend development server:

`npm run dev`

Architecture and workflow:



Use Case Diagram:



User Interface

