

# Chatribe

## project documentation

Chatribe is a minimal full-stack web application that enables users to connect and communicate through interest-based messaging. Users can easily create an account using their email, username, and password. Once logged in, users are presented with a list of all other registered users. The core functionality allows users to send interest requests to others. Recipients can either accept or ignore these requests. If the request is accepted, both users gain access to a private chat interface where they can exchange messages in real-time.

## Tech

### Frontend Development

- [React](#) - Main framework for building the user interface.
- [Framer Motion](#) - Used for adding smooth animations and transitions.
- [Lottie Files](#) - For rendering high-quality animations using lightweight JSON files.
- [Axios](#) - For handling HTTP requests to the backend API.
- [Tailwind CSS](#) - Utility-first CSS framework for styling components.

### Backend Development

- [Django](#) - High-level Python web framework used to build the backend.
- [Django REST Framework \(DRF\)](#) - For building robust and flexible RESTful APIs.
- [PostgreSQL](#) - Powerful, open-source object-relational database used to store application data.
- [Daphne](#) - ASGI server used to serve Django applications with support for WebSockets.
- [Redis](#) - In-memory data store used as a channel layer backend.
- [Django Channels](#) - Enables handling of WebSockets and real-time features like chat.

### Assumptions

- Only authenticated (logged-in) users can access the application and its core features.
- All registered users are visible to each other by default on the homepage.
- A basic user search functionality is provided to filter users by name.
- An interest request can be sent from one user to any other visible user.
- User can accept or ignore an incoming interest request – only accepted requests lead to a chat connection.
- Chat functionality is only available once both users have agreed (i.e., an interest is accepted).
- Real-time chat is implemented using **Django Channels**, **Redis**, and **Daphne**.
- No blocking or privacy filters were added – this was kept simple for demo purposes.
- All messages are exchanged in one-to-one private channels after a successful interest match.

## Installation

Frontend Install the dependencies and devDependencies and start the server.

1. Clone the frontend repository:

```
git clone https://github.com/Shamilnk812/chattribe-chat-application-frontend.git
```

2. Install dependencies:

```
npm install
```

3. Start the development server::

```
npm run dev
```

#### Backend Setup

4. Clone the backend repository:

```
git clone https://github.com/Shamilnk812/chattribe-chat-application-backend.git
```

5. Create a virtual environment:

```
python -m venv venv
source venv/bin/activate      # On macOS/Linux
venv\Scripts\activate        # On Windows
```

6. Install the required dependencies :

```
pip install -r requirements.txt
```

7. Run migrations :

```
python manage.py makemigrations
python manage.py migrate
```

8. Start the Daphne server :

```
daphne -b 0.0.0.0 -p 8000 chattribe.asgi:application
```

## Future Improvements

- **UI Enhancements:**

- Improve the overall user interface with a modern design, smooth transitions, and better color themes.
- Integrate more engaging animations using Framer Motion and Lottie.

- **User Profile Settings:**

- Add a dedicated profile settings page where users can:
  - Update their username and bio.
  - Upload or change their profile picture.

- **Privacy & Blocking Features:**

- Allow users to block or report other users.

- Introduce privacy settings to hide user profiles from public visibility if desired.