

# ChatApp

---

A real-time chat application built for modern web use.

## Table of Contents

About the Project

Features

Technology Stack

Installation

Configuration

Running the App

Usage

Folder Structure

Contributing

License

Contact

## About the Project

ChatApp is a full-stack real-time messaging web application. It lets users register/login, chat one-on-one or in groups, see presence status, send and receive messages instantly, and more. It is built by zabha (CEO of SolveXRwanda) and is ideal for demonstration, customization, or full deployment.

## Features

- User authentication (register/login/logout)



- Real-time messaging using WebSockets (or similar)
- Global and private chat rooms
- Online/offline status of users
- Message history & persistence
- Responsive UI (works on desktop & mobile)
- Optionally file / media sharing (if implemented)
- Easily deployable to your own server or cloud

## Technology Stack

Here's a typical stack:

- **Backend:** Node.js + Express (or your choice)
- **WebSockets / Real-time:** Socket.io (or similar library)
- **Database:** MongoDB (or other)
- **Frontend:** React (or Next.js)
- **Styling:** TailwindCSS / CSS Modules / Styled Components
- **Deployment:** Heroku, Vercel, AWS, DigitalOcean etc

(Edit the above to reflect what you actually used.)

## Installation

### Prerequisites

- Node.js (v14+ or latest)
- npm or yarn
- MongoDB (or other DB) or connection string
- A WebSocket-capable environment (for real-time)

### Steps

Clone the repository

```
git clone https://github.com/zabha5/chatapp.git  
cd chatapp
```



Install dependencies

### In the backend folder

```
cd backend  
npm install
```

### In the frontend folder

```
cd ../frontend  
npm install
```

## Configuration

Create a .env file in the backend folder (for example .env.example may exist). Add the following environment variables:

```
PORT=5000  
MONGODB_URI=your_mongo_connection_string  
JWT_SECRET=your_secret_key  
NODE_ENV=development
```

(Include any other required keys such as Cloudinary, AWS, etc if needed.)

## Running the App

From the backend folder:

```
npm run dev      # or npm start for production
```

From the frontend folder:

```
npm run dev      # launches frontend in dev mode
```

Then open your browser and go to <http://localhost:3000> (or whichever port you configured) to see the app.

## Usage

- Register a new user account.
- Log in with the account.



- Create or join a chat room.
- Send and receive messages instantly.
- See online/offline status of users.
- (Optional) Upload / share images or files if implemented.
- Log out when done.

## Folder Structure

Here's a sample of the structure. Edit based on your project actual tree.

```
chatapp/
|
└── backend/
    ├── controllers/
    ├── models/
    ├── routes/
    ├── middlewares/
    ├── server.js
    ├── .env
    └── ...
|
└── frontend/
    ├── public/
    ├── src/
    │   ├── components/
    │   ├── pages/
    │   ├── services/
    │   └── App.jsx
    └── ...
|
└── README.md
└── package.json
```

## Contributing

Contributions are welcome! Here's how you can help:

1. Fork the repo
2. Create a new branch: `git checkout -b feature/YourFeature`



- 3.** Make your changes
- 4. Commit and push:** `git commit -m "Add YourFeature"`
- 5.** Open a Pull Request and describe your changes

Please ensure your code follows the existing style and includes tests if applicable.

## License

This project is licensed under the MIT License — see the `LICENSE` file for details.

## Contact

zabha (Mbarushimana Emmanuel) – CEO of SolveXRwanda Feel free to open issues, ask questions or request features in this repository.

Feel free to modify any section to better reflect the details of your chatapp: the exact tech stack you used, any extra features, screenshots, deployment instructions, environment variables, etc.

