Modern Chat App - Installation Guide

A full-stack real-time chat application built with Vue 3, Node.js, Express, MongoDB, and Socket.io.

🚀 Technologies Used

Backend

- **Node.js** with **Express.js** framework
- **MongoDB** with Mongoose ODM
- **Socket.io** for real-time communication
- **express-async-handler** for error handling
- **JWT** for authentication
- **bcryptjs** for password hashing

Frontend

- **Vue 3** with Options API
- **Vue Router** for navigation
- **Axios** for HTTP requests
- **Socket.io-client** for real-time features
- **Vite** as build tool
- **Tailwind CSS** for styling
- **vue3-toastify** for notifications

📋 Prerequisites

Before you begin, ensure you have the following installed:

- **Node.js** (v18 or higher)
- **npm** or **yarn**
- **MongoDB** (local installation or MongoDB Atlas)

🛠️ Installation Steps

1. Clone or Create Project Structure

Create the following directory structure:

```
chat-app/
├── backend/
└── frontend/
```

2. Backend Setup

Navigate to the backend directory:

```bash
cd backend
```

Create `package.json` using the provided backend package.json file, then install dependencies:

```bash
npm install
```

Create the following files in the backend directory:

- `server.js` (main server file)
- `.env` (environment variables)
- `seed.js` (database seeding)

Create these directories and files:

```
backend/
├── models/
│   └── (User.js, Chat.js, Message.js combined in one file)
├── routes/
│   └── (auth.js, chat.js, users.js combined in one file)
├── middleware/
│   └── (auth.js middleware)
└── ...
```

Create a `.env` file:

```bash
NODE_ENV=development
PORT=3000
MONGODB_URI=mongodb://localhost:27017/chatapp
JWT_SECRET=your-super-secret-jwt-key-here-change-in-production
```

3. Frontend Setup

Navigate to the frontend directory:

```bash
cd ../frontend
```

Create `package.json` using the provided frontend package.json file, then install dependencies:

```bash
npm install
```

Create the following files and directories:

```
frontend/
├── src/
│   ├── views/
│   │   ├── Login.vue
│   │   ├── Register.vue
│   │   └── Chat.vue
│   ├── stores/
│   │   ├── auth.js
│   │   └── chat.js
│   ├── router/
│   │   └── index.js
│   ├── App.vue
│   ├── main.js
│   └── style.css
├── index.html
├── vite.config.js
├── tailwind.config.js
└── postcss.config.js
```

🗄️ Database Setup

Option 1: Local MongoDB

1. Install MongoDB on your system
2. Start MongoDB service:

   ```bash
   mongod
   ```

Option 2: MongoDB Atlas (Cloud)

1. Create a free account at MongoDB Atlas
2. Create a new cluster
3. Get your connection string
4. Update the `MONGODB_URI` in your `.env` file

🌱 Seed Database with Sample Data

From the backend directory, run:

```bash
npm run seed
```

This will create sample users and chats. Sample login credentials:

- **Email:** john@example.com
- **Password:** password123

Other sample users:

- jane@example.com / password123
- bob@example.com / password123
- alice@example.com / password123
- mike@example.com / password123

🚀 Running the Application

Start Backend Server

From the backend directory:

```bash
# Development mode with auto-reload
npm run dev

# Or production mode
npm start
```

The backend server will start on `http://localhost:3000`

Start Frontend Development Server

From the frontend directory:

```bash
npm run dev
```

The frontend will start on `http://localhost:5173`

🧪 Testing the Application

1. Open your browser and go to `http://localhost:5173`
2. You can either:
   - **Register** a new account
   - **Login** with sample credentials (john@example.com / password123)
3. Start chatting with other users!

📁 Project Structure

Backend Structure

```
backend/
├── server.js          # Main server file with Socket.io setup
├── models/            # MongoDB models (User, Chat, Message)
├── routes/            # API routes (auth, chat, users)
├── middleware/        # Authentication middleware
├── seed.js            # Database seeding script
├── package.json       # Dependencies and scripts
└── .env               # Environment variables
```

Frontend Structure

```
frontend/
├── src/
│   ├── views/         # Vue components (Login, Register, Chat)
│   ├── stores/        # State management (auth, chat)
│   ├── router/        # Vue Router configuration
│   ├── App.vue        # Root component
│   ├── main.js        # Application entry point
│   └── style.css      # Tailwind CSS styles
├── index.html         # HTML template
├── vite.config.js     # Vite configuration
├── tailwind.config.js # Tailwind CSS configuration
└── package.json       # Dependencies and scripts
```

🔧 Available Scripts

Backend Scripts

```bash
npm start        # Start production server
npm run dev      # Start development server with nodemon
npm run seed     # Seed database with sample data
```

Frontend Scripts

```bash
npm run dev      # Start development server
npm run build    # Build for production
npm run preview  # Preview production build
```

🌟 Features

- ✅ **Real-time messaging** with Socket.io
- ✅ **User authentication** (register/login)
- ✅ **Private and group chats**
- ✅ **Online status indicators**
- ✅ **Typing indicators**
- ✅ **Message history**
- ✅ **User search functionality**
- ✅ **Responsive design** with Tailwind CSS
- ✅ **Toast notifications**
- ✅ **Auto-scroll to latest messages**

🔒 Security Features

- JWT-based authentication
- Password hashing with bcryptjs
- Protected API routes
- Input validation and sanitization

🎨 UI/UX Features

- Modern gradient designs
- Smooth animations and transitions

- Responsive layout for all devices
- Custom scrollbars
- Real-time connection status
- Avatar generation for users

🛠️ Customization

Adding New Features

1. **Backend:** Add new routes in the `routes/` directory
2. **Frontend:** Create new Vue components in `src/components/`
3. **Database:** Add new models in the `models/` directory

Styling Modifications

- Modify `tailwind.config.js` for custom colors and themes
- Update `src/style.css` for global styles
- Component-specific styles can be added in individual Vue files

🐛 Troubleshooting

Common Issues

1. **MongoDB Connection Error:**
    - Ensure MongoDB is running
    - Check the connection string in `.env`
2. **Socket.io Connection Issues:**
    - Verify backend server is running on port 3000
    - Check CORS configuration in `server.js`
3. **Frontend Build Issues:**
    - Clear node_modules and reinstall: `rm -rf node_modules && npm install`
    - Check for version conflicts in package.json
4. **Authentication Issues:**
    - Verify JWT_SECRET is set in `.env`
    - Check token expiration settings

📝 API Endpoints

Authentication

- `POST /api/auth/register` - Register new user

- `POST /api/auth/login` - Login user

Chats

- `GET /api/chats` - Get user's chats
- `POST /api/chats` - Create new chat
- `GET /api/chats/:id/messages` - Get chat messages
- `POST /api/chats/:id/messages` - Send message

Users

- `GET /api/users` - Get all users
- `GET /api/users/search` - Search users

🚀 Deployment

Backend Deployment (Heroku/Railway/DigitalOcean)

1. Set environment variables on your hosting platform
2. Ensure MongoDB Atlas connection string is configured
3. Deploy using your platform's CLI or web interface

Frontend Deployment (Vercel/Netlify)

1. Build the frontend: `npm run build`
2. Deploy the `dist` folder
3. Configure environment variables for API endpoints

🤝 Contributing

1. Fork the repository
2. Create a feature branch
3. Make your changes
4. Test thoroughly
5. Submit a pull request

📄 License

This project is open source and available under the MIT License.

Happy Chatting! 🎉

For additional help or questions, please refer to the documentation of the individual technologies used in this project.