

# Real-Time Chat Application

## EchoVerse

### README

Pranjal Prajapati

## Contents

<b>1</b>	<b>Project Overview</b>	<b>1</b>
<b>2</b>	<b>Hosting the Application</b>	<b>2</b>
<b>3</b>	<b>Using the Application</b>	<b>4</b>
<b>4</b>	<b>Considerations</b>	<b>7</b>

## 1 Project Overview

- This project encompasses the task of developing a real-time chat application to facilitate communication between its users.
- It has been built using the following frameworks:
  - **Frontend:** React.js
  - **Backend:** Node.js with Express.js
  - **Real-Time Communication:** Socket.io
  - **Database:** MongoDB
- The application supports the following features:
  - **User Registration and Authentication:** Users can register and create their profiles with a unique username and password. The passwords are first hashed and then stored in the database to prevent them from being compromised on data breach. Authentication mechanisms are in place to ensure secure access to the chat application.
  - **Real-Time Messaging:** Application users can send and receive messages in real-time. The messages support both text and emojis.

- **Message History:** Chat history for a particular EchoRoom is maintained and is retrieved whenever the user joins the EchoRoom. This allows the users to view past messages when joining an EchoRoom.
- **User Profile Management:** Users can update their profiles with the application providing the option for changing passwords after registration.
- **Privacy of Messages:** Each EchoRoom upon creation gets assigned a unique identifier (visible to each member of the group), and users can join the room only upon entering the ID. Each EchoRoom only shows the messages which are sent by members of that room.
- **No Restriction on EchoRoom Names:** While usernames for each profile in the application have to be unique for security reasons, users are given the freedom of choosing any name for their EchoRooms.
- **Status of Other EchoRoom Members:** While inside an EchoRoom, members can view the usernames of all other members of that room which are online at the moment.

## 2 Hosting the Application

- To start / host the application, you will require the following to be installed in your system:
  - Node and Node Package Manager (npm)
  - MongoDB and MongoDB Shell (mongosh)
- Ensure that the mongod service is running on the system.
- Following are the steps for hosting the application on your local machine:
  1. Clone the git repository for this project titled EchoVerse-A-Real-Time-Chat-Application into a new directory. Now start a new terminal window and cd into the directory.

```
spidey@Spidey95:~$ cd EchoVerse/
spidey@Spidey95:~/EchoVerse$
```

2. Now, cd into the directory back-end and enter the command npm start. This will start up the http server at <http://localhost:3001> (default URL for the server).

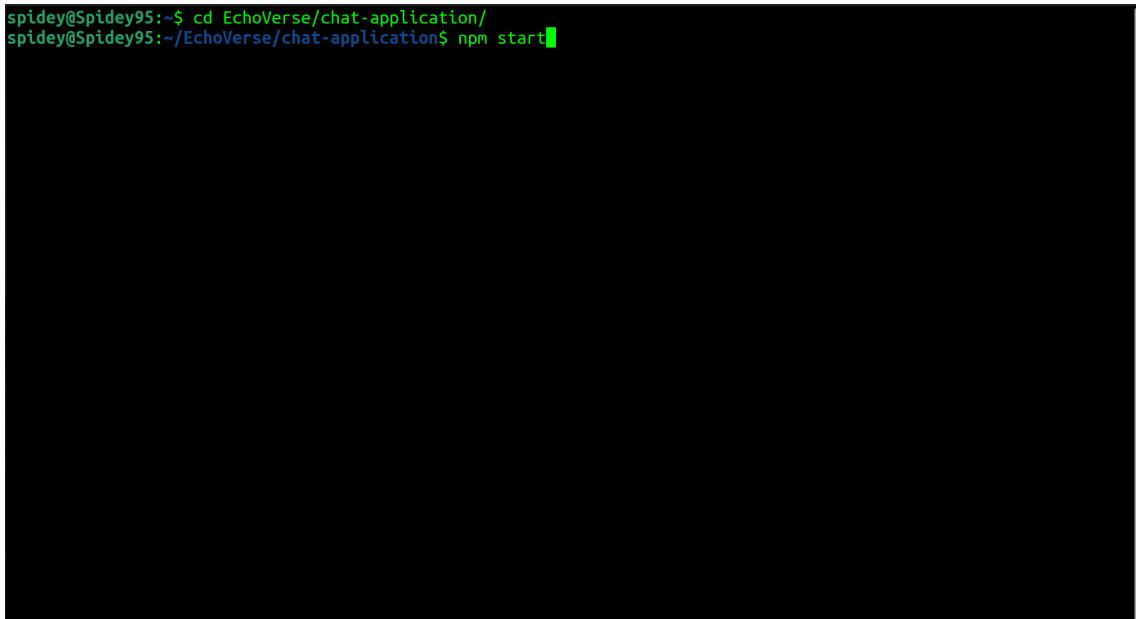
```
spidey@Spidey95:~$ cd EchoVerse/
spidey@Spidey95:~/EchoVerse$ cd back-end/
spidey@Spidey95:~/EchoVerse/back-end$ npm start

> sockets@1.0.0 start
> node server.js

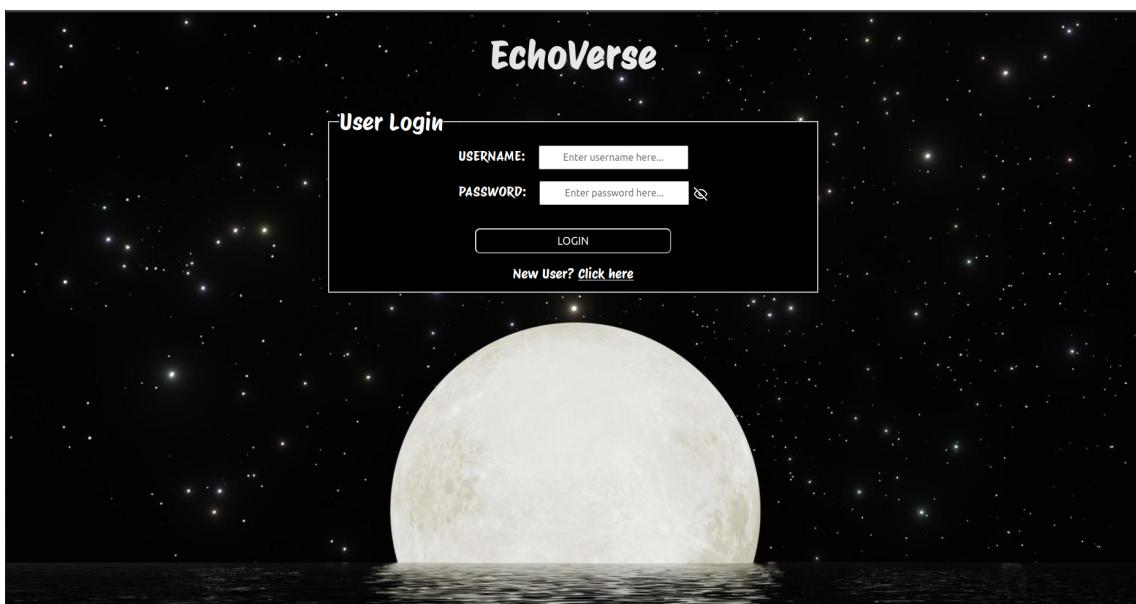
HTTP server listening on 3001
```

3. Keeping this terminal window open, start a new window and cd into the chat-application directory of the repo. Enter the command npm start to start up the application at <http://localhost:3000> (default URL for the application)

```
spidey@Spidey95:~$ cd EchoVerse/chat-application/  
spidey@Spidey95:~/EchoVerse/chat-application$ npm start
```

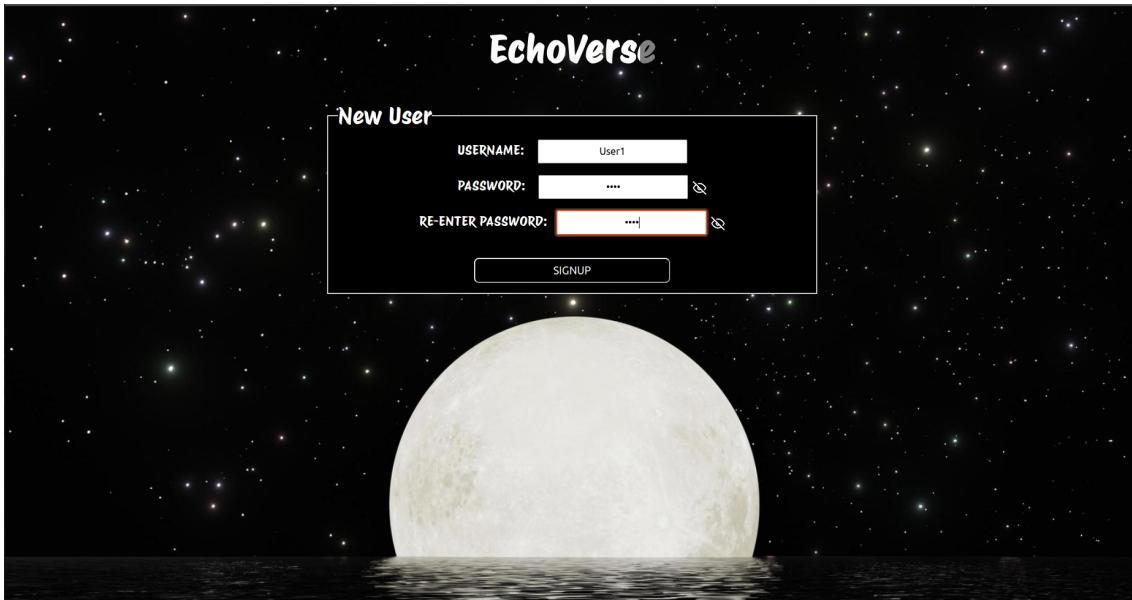


4. You should be able to view the login page of the application at the mentioned URL. Congratulations! You have successfully started the application.



### 3 Using the Application

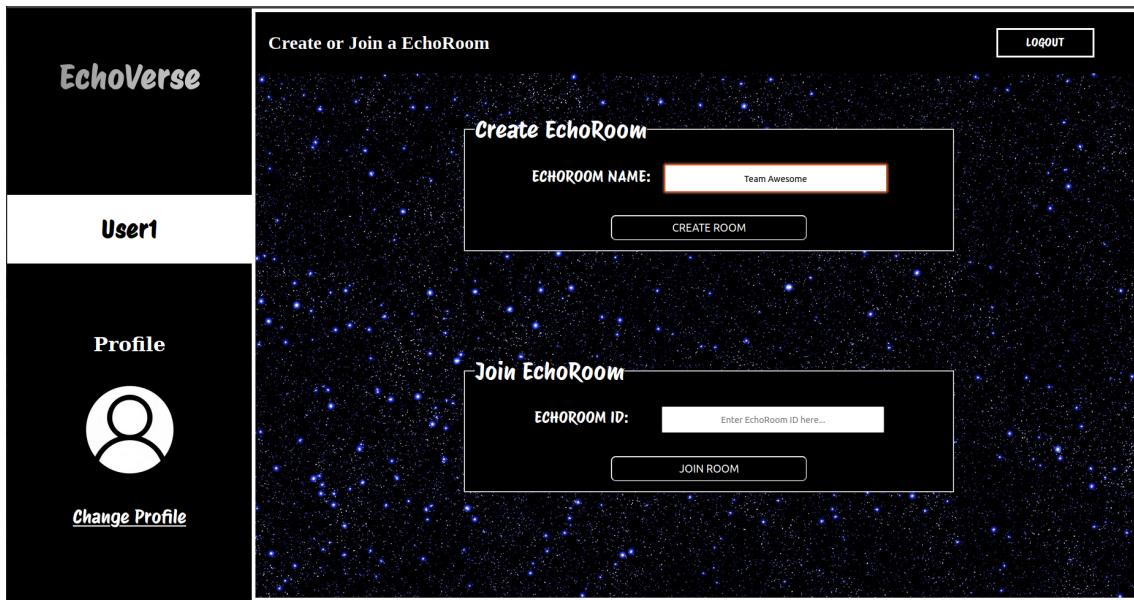
- While the application is quite user-friendly and easy to work with, this shall act as a small demonstration on using it.
- Click on the link for creating a new user. Enter the profile details in the fields given.



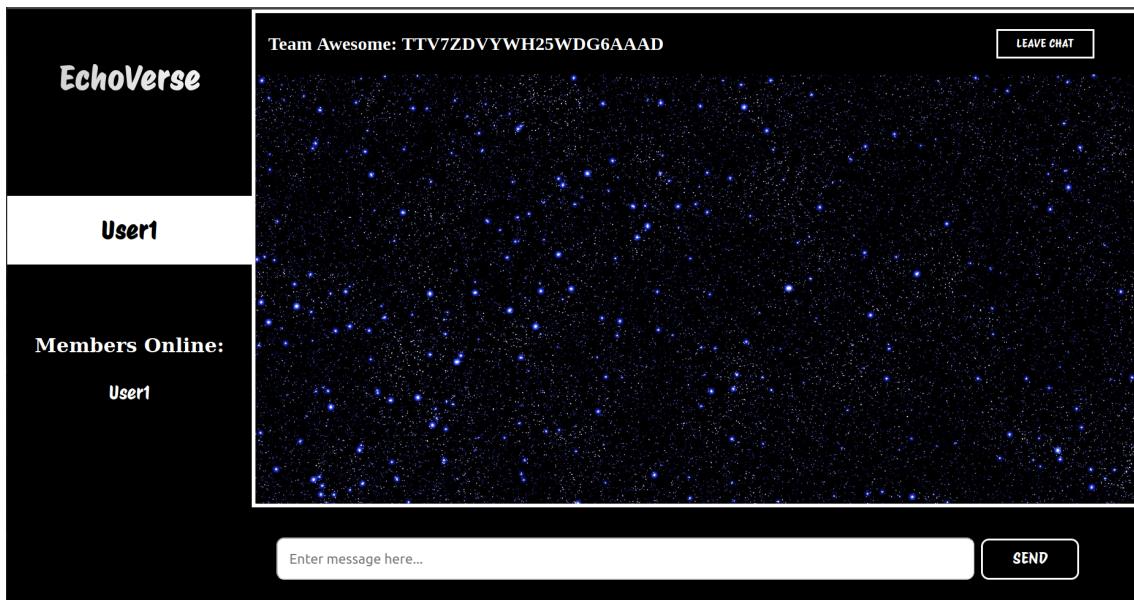
- Click on the SIGNUP button once you're done to get to the main page of the application.

A screenshot of the EchoVerse application's main interface. On the left is a sidebar with "EchoVerse" at the top, followed by "User1", "Profile" with a user icon, and a "Change Profile" link. The main content area has a dark blue starry background. It shows two forms: "Create or Join a EchoRoom" with a "Create EchoRoom" section (text input for ECHOROOM NAME and a "CREATE ROOM" button) and a "Join EchoRoom" section (text input for ECHOROOM ID and a "JOIN ROOM" button). A "LOGOUT" button is in the top right corner of the main content area.

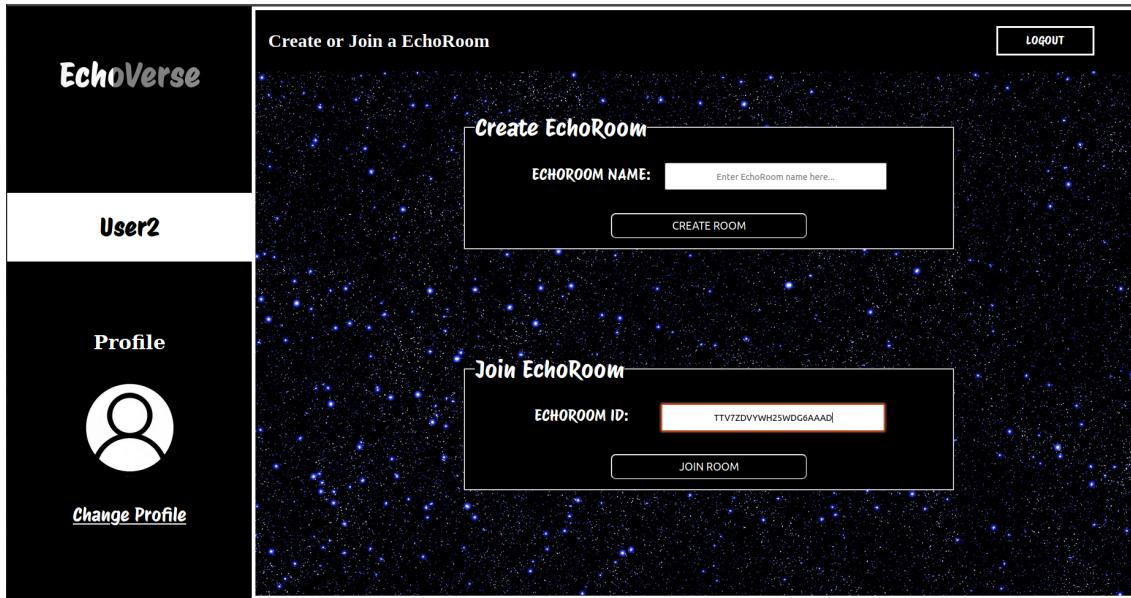
- To start a new EchoRoom, enter the name of the EchoRoom in the field given.



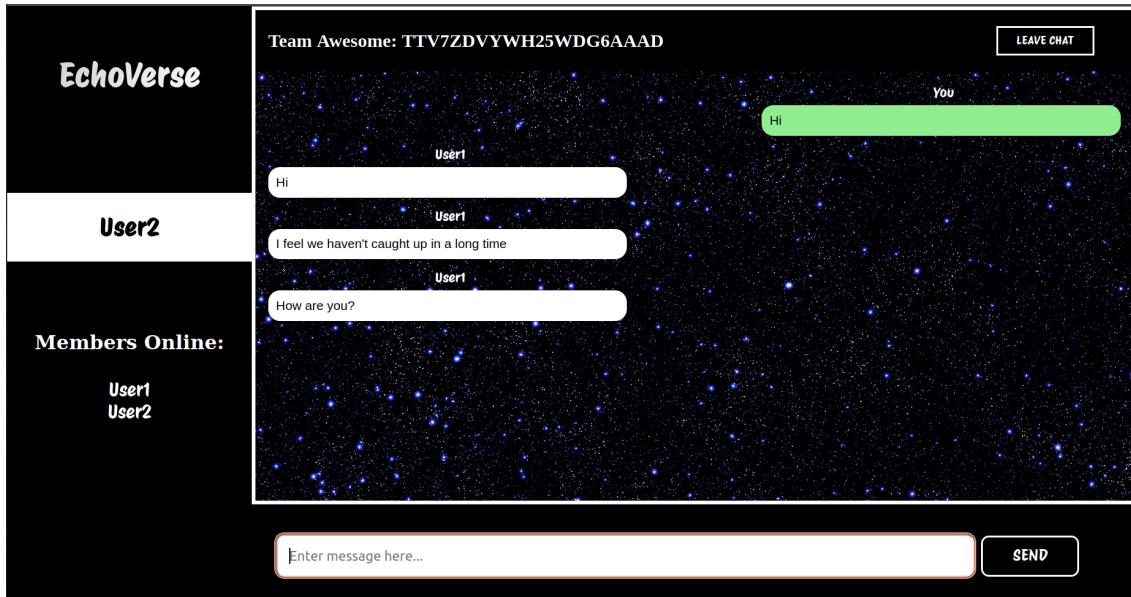
- Now click on the provided button to enter the EchoRoom.



- Send the EchoRoomID displayed in the top left corner to the people you wish to communicate with. If you are the one joining the EchoRoom, enter the ID into the mentioned field upon login.



- Press on the provided button and start chatting!



## 4 Considerations

- It was initially thought of allowing the users to upload photos for their avatars, but owing to security reasons was not implemented. A mechanism would be needed to ensure the file size remains within limit and to monitor the input for any malware.
- For the responsive design part, the application I developed requires quite a lot of modifications to be ready for mobile devices. Hoping to perform them in a future update.