INTERNSHIP PROJECT PRESENTATION

ON

Web Chat Application

For

Multiple Users

PRESENTED TO

SNTI

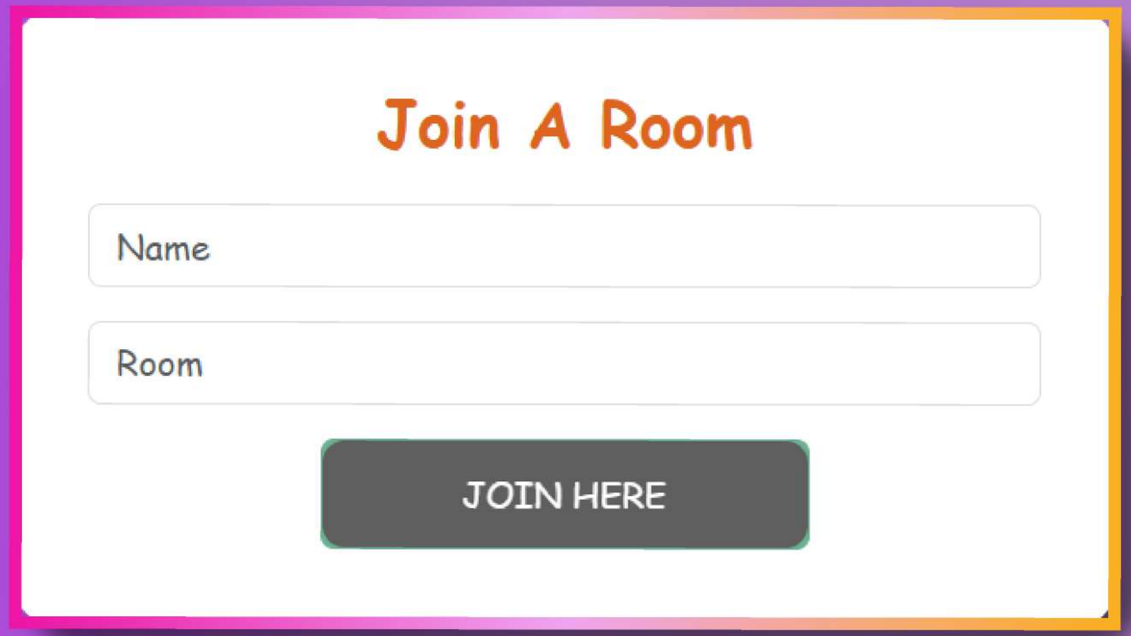
PRESENTED BY

Kriti Verma

VT0524206945

Overview

We chat Application



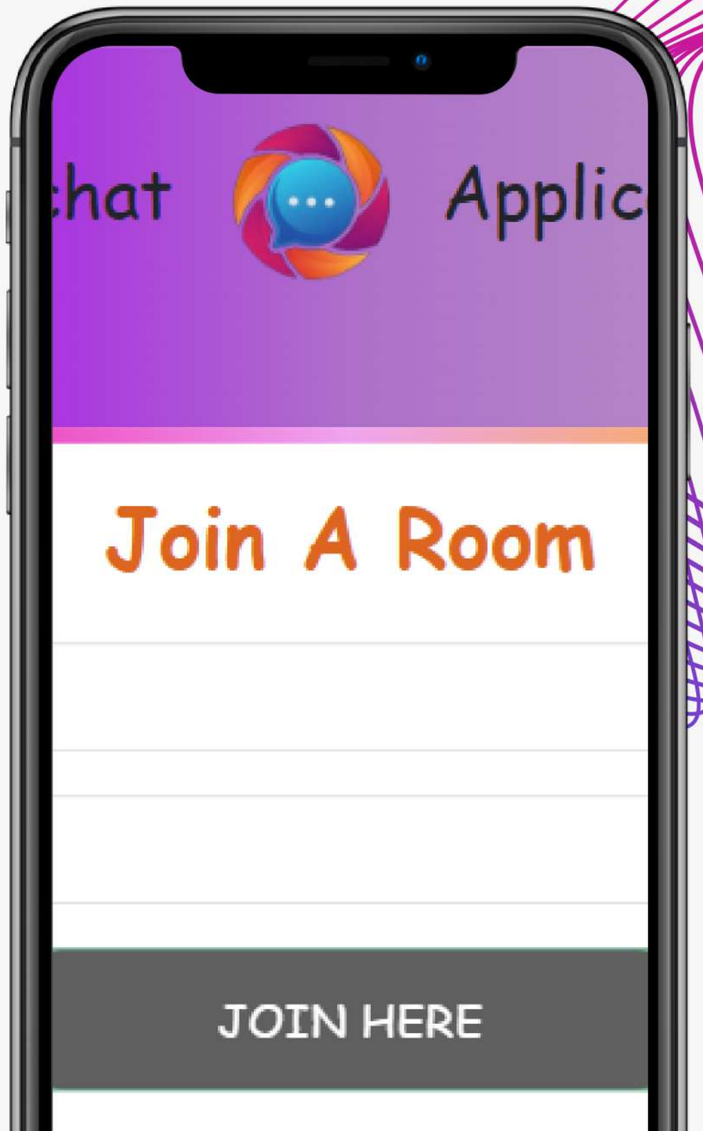
Introduction

The Real-time Web Chat Application is designed to facilitate seamless communication between users in various chat rooms. Built using modern web technologies, this application supports real-time message exchange, user presence indication, and robust backend infrastructure to ensure data persistence and reliability.

The frontend is developed using React, while the backend leverages ASP.NET Core with SignalR for real-time functionalities. The chat messages are stored in a SQL Server database to maintain a history and support future data retrieval.

Objectives

- Develop a real-time chat application that allows multiple users to communicate simultaneously in various chat rooms.
- Ensure a user-friendly and responsive interface for both desktop and mobile users.
- Implement backend services to handle real-time data exchange and persist chat messages.
- Provide a reliable data storage solution to maintain chat history and user information.



Outcomes

- Successful development of a real-time chat application supporting multiple chat rooms.
- Responsive UI that works seamlessly across different devices.
- Real-time message exchange and user presence indication using SignalR.
- Persistent storage of chat messages in a SQL Server database.
- Enhanced user experience with added features like emoji support and smooth UI animations.

Technologies Used

Frontend:

- React
- React-Bootstrap
- CSS (including animations and media queries for responsiveness)

Backend:

- ASP.NET Core
- SignalR
- Entity Framework Core

Database:

- SQL Server

Software and Hardware Required

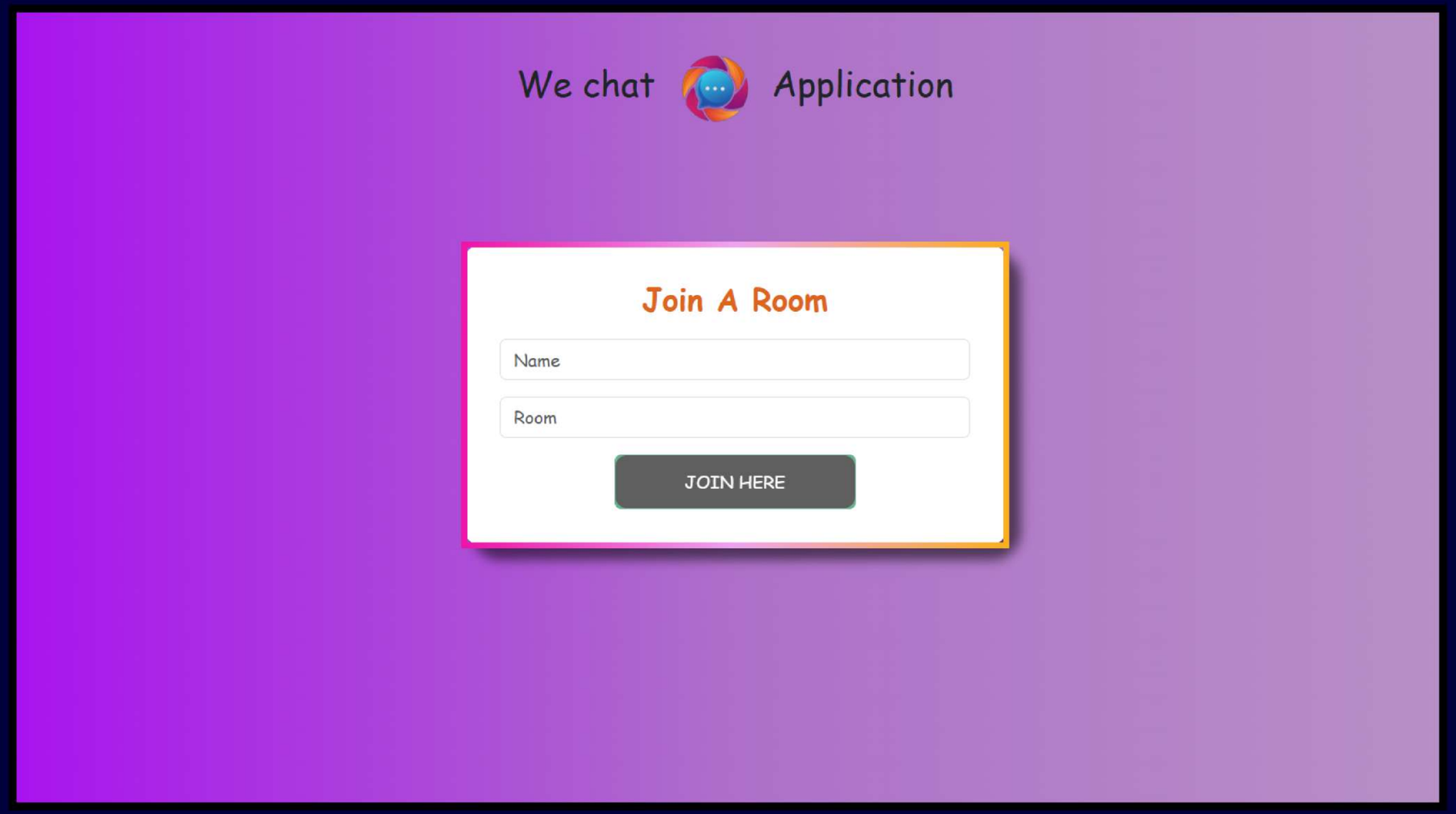
Software:

- Visual Studio 2022
- Node.js and npm (for managing frontend dependencies)
- SQL Server Management Studio
- Web Browser (Chrome, Firefox, etc.)
- Visual Studio Code

Hardware:

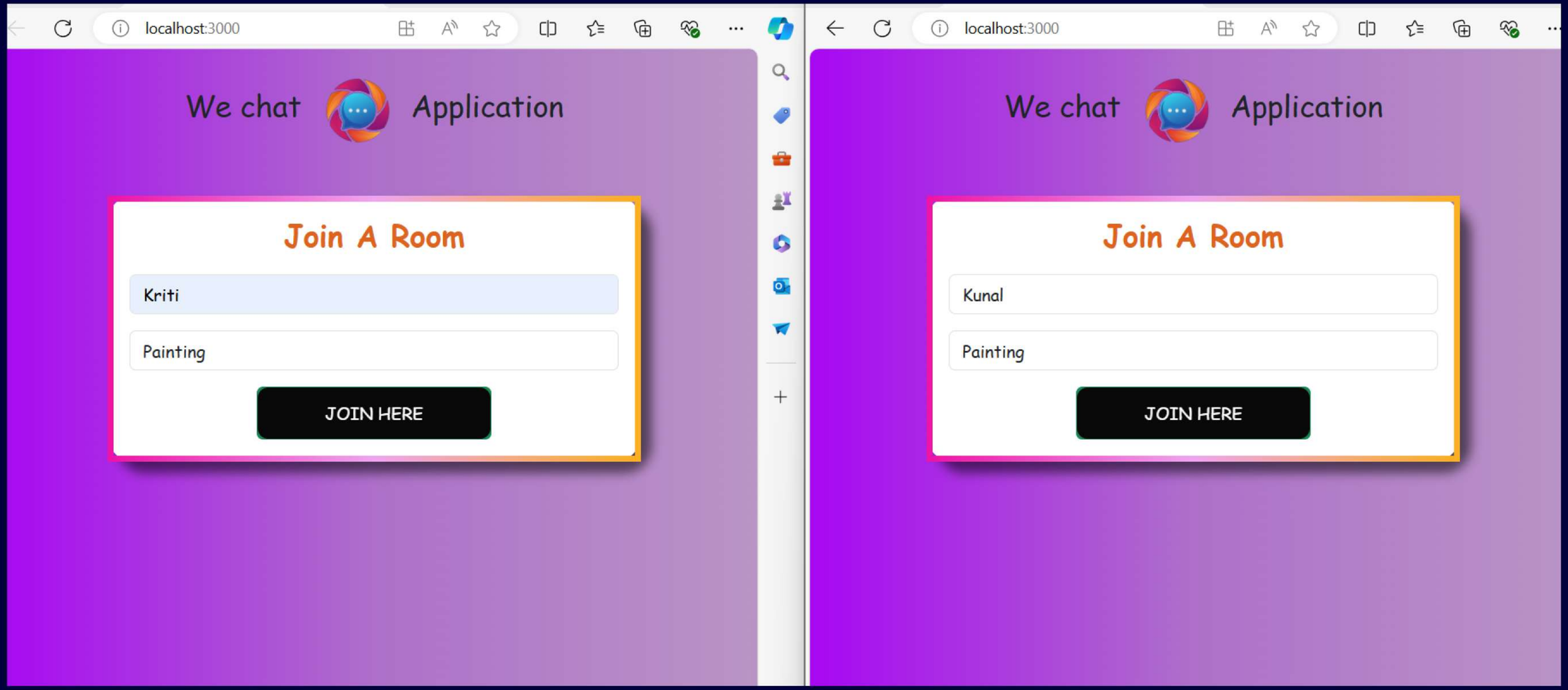
- Development Machine with minimum 8GB RAM
- 5th Generation

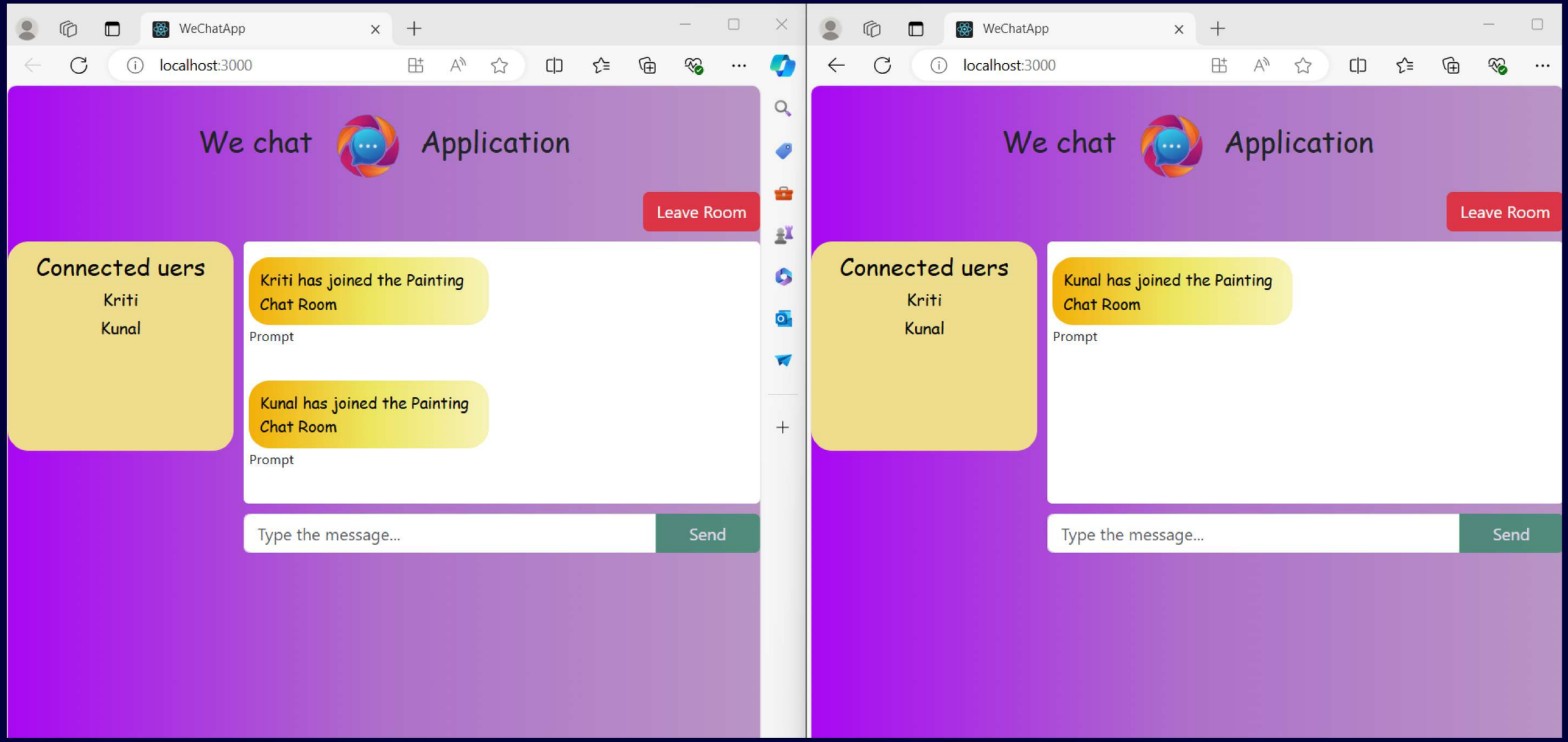
Outputs



LOBBY PAGE

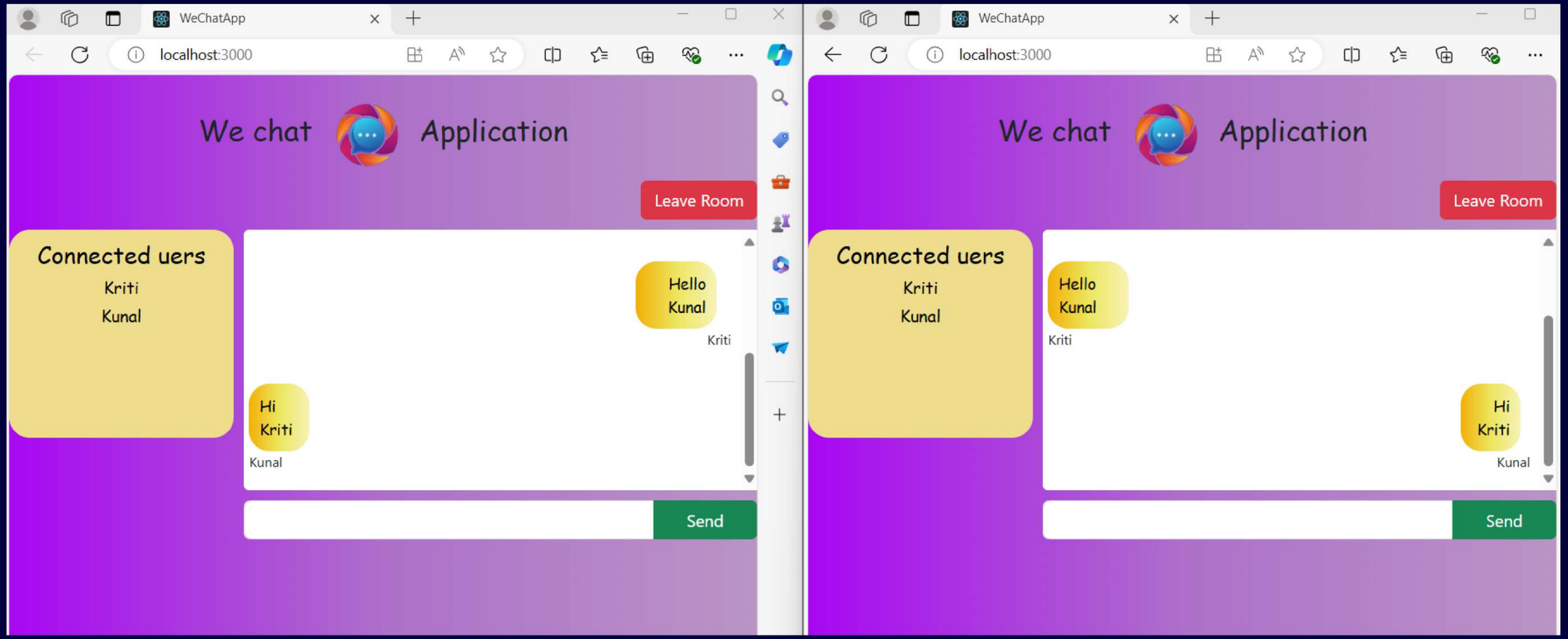
Outputs


localhost:3000
We chat Application
Join A Room
Kriti
Painting
JOIN HERE
localhost:3000
We chat Application
Join A Room
Kunal
Painting
JOIN HERE

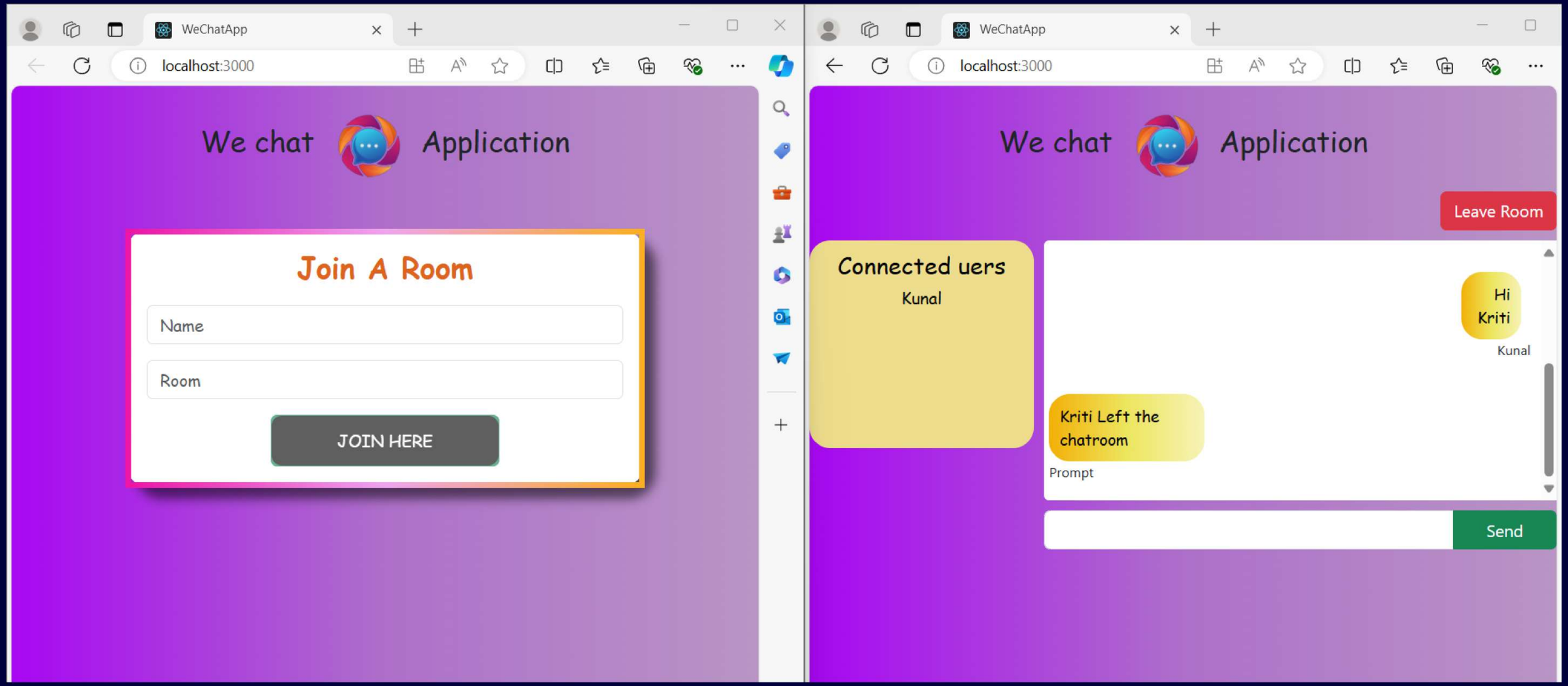


Chat Room Design

When the Users gets added to the Chat Room then prompt is generated



Users Are able to send and Receive Message



When One User Leaves the room then Prompt is generated and name gets deleted from connected users list

DropAndCreat...ChatMessages | **dbo.ChatMessages [Data]** | ChatHub.cs | Program.cs

Max Rows: 1000

Id	User	Room	Message	Timestamp
20	Kriti	painting	hii	7/4/2024 11:01:...
21	Kv	painting	hello	7/4/2024 11:01:...
22	Kv	painting	what's going on	7/4/2024 11:01:...
23	Kriti	painting	nothing much!!....	7/4/2024 11:02:...
24	Kunal	painting	Heloo	7/4/2024 2:14:1...
25	Kunal	painting	hi	7/4/2024 2:16:0...
26	Kunal	painting	hi	7/4/2024 2:28:2...
27	Kriti	coding	hiii	7/9/2024 1:18:3...
28	Kunal	coding	Hi	7/9/2024 1:19:1...
29	Kriti	coding	Hello	7/9/2024 1:19:3...
30	KV	coding	Hey	7/9/2024 1:19:4...
31	KV	coding	hi	7/9/2024 5:14:5...
32	Kriti	coding	hello	7/9/2024 5:15:0...
33	Kriti	coding	What's Up	7/9/2024 5:19:4...
34	Kriti	coding	All	7/9/2024 5:20:3...
35	Kriti	coding	Hi all	7/9/2024 5:21:2...
36	Kriti	Painting	Hello Kunal	7/11/2024 12:0...
37	Kunal	Painting	Hi Kriti	7/11/2024 12:0...
NULL	*NULL*	*NULL*	*NULL*	*NULL*

Database that stores the Users name, the chat room name, Messages send by Users and the Time and Date

Limitations

1

Limited to text and emoji-based communication; no support for multimedia messages.

2

Scalability might be a concern with a large number of concurrent users.

3

As the project is Running on one Device only so now it cannot be accessed from other device

4

Lack of advanced search functionality to find specific messages or users within the chat history.

Future Scope

- Implement multimedia support, including image and video sharing.
- Enhance scalability to support a larger user base.
- Add more advanced features like user profiles, direct messaging, and notification systems.
- Incorporate AI-based moderation tools to ensure safe and appropriate communication.
- Improve security features, including end-to-end encryption.

Conclusion

The Real-time Web Chat Application successfully meets its objectives of providing a seamless and responsive communication platform. Leveraging modern web technologies and robust backend services, the application offers real-time messaging and user presence features. While there are some limitations in terms of multimedia support and scalability, the application lays a solid foundation for future enhancements and scalability improvements. The project demonstrates a comprehensive understanding of both frontend and backend development, ensuring a high-quality user experience and reliable data management.