

IBM TRAINING PROJECT PRESENTATION





DevTalk Hub

BUILDING SEAMLESS COMMUNICATION



REAL-TIME CHAT APPLICATION



INTRODUCTION

We delve into the world of instant connectivity through the lens of crafting a Real-Time Chat Application. As our digitally woven world seeks real-time experiences, we explore the architecture and features that power such communication. Join us to unravel development challenges, innovative solutions, and the significance of user experience, security, and scalability. Whether a developer or a curious mind, this journey offers insights into modern communication technology. Let's embark on this exploration of real-time interaction's transformative potential.



Goals



Objective 01

Create a real-time chat application to facilitate immediate and fluid communication between users, overcoming delays and enhancing engagement.



Objective 02

Develop the project to enable participants to gain hands-on experience in utilizing real-time communication technologies, fostering practical skills in modern app development.



METHODOLOGY



Planning and Design:

- Define project goals, features, and target users.
- Design UI/UX wireframes for the chat interface.
- Determine roles and responsibilities within the development team.



Frontend Development

- Develop the frontend using React.js for dynamic user interfaces.
- Create components for user authentication, chat messages, and settings.
- Focus on responsive design and intuitive user experience.



Backend Setup and Integration

- Set up Node.js and Express.js for the backend.
- Integrate the Stream Chat API for real-time messaging.
- Establish API endpoints for frontend-backend communication.

METHODOLOGY



User Authentication and Security

- Implement user registration and login using JWT.
- Apply security measures like encryption and hashing.
- Ensure validation and error handling for data security.



Real-Time Functionality

- Implement sending and receiving messages using Stream Chat API.
- Enable features like typing indicators and message synchronization.
- Handle real-time updates for notifications and user presence.

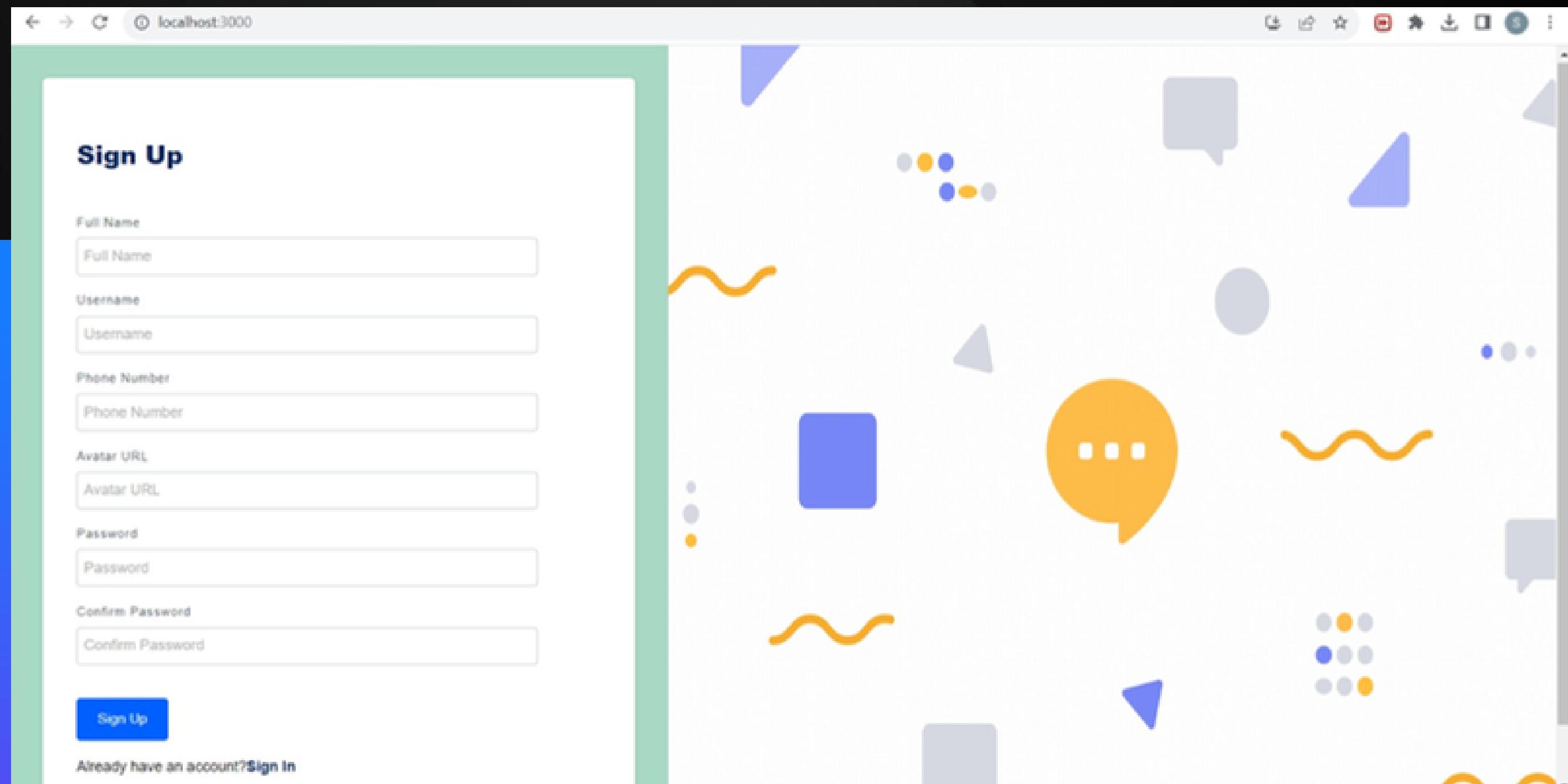


Testing, Deployment, and Documentation

- Conduct thorough testing of both frontend and backend components.
- Deploy the application using suitable hosting platforms.
- Create comprehensive documentation for usage and maintenance.

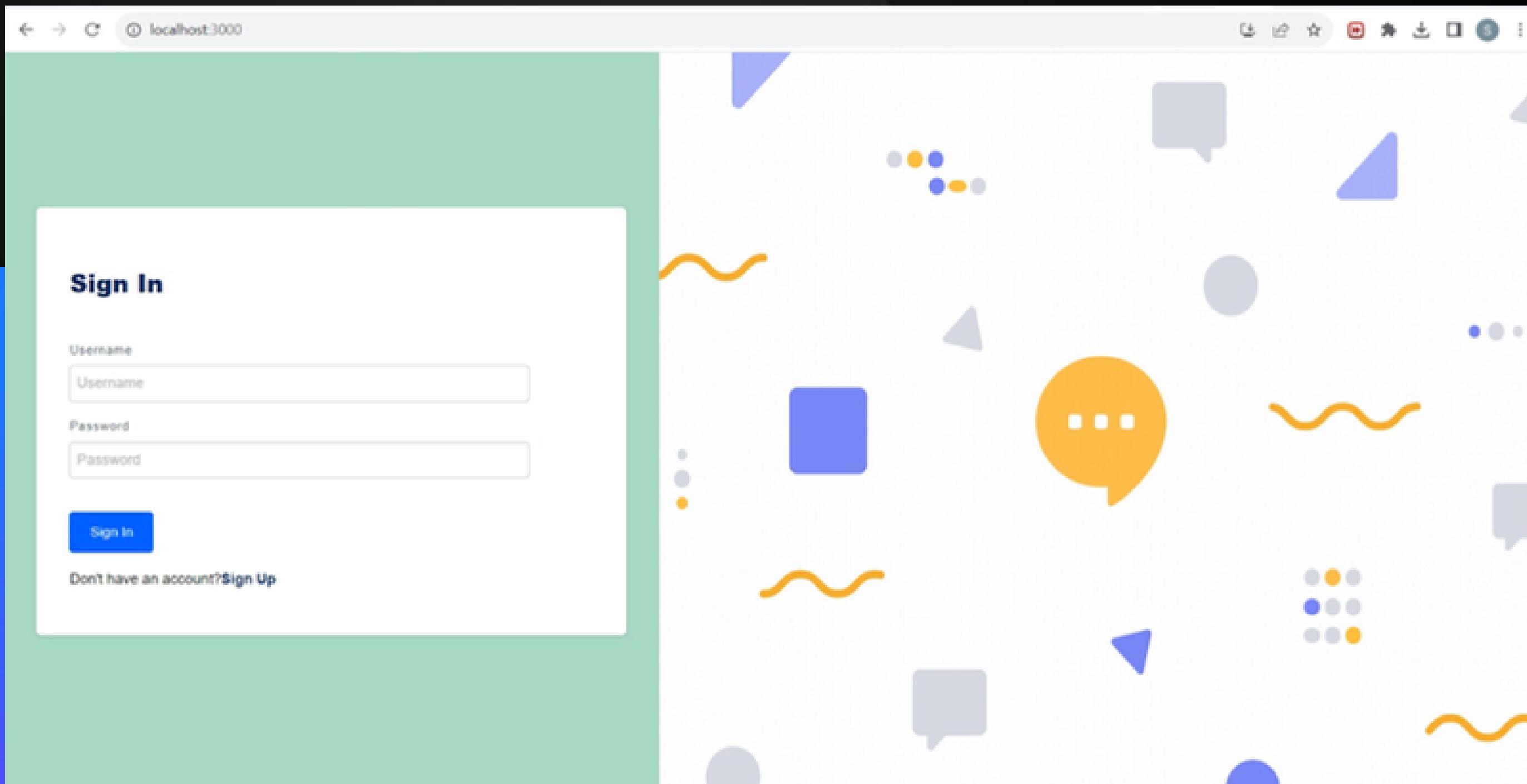
RESULTS & INTERPRETATION

SIGN UP



RESULTS & INTERPRETATION

SIGN IN



RESULTS & INTERPRETATION

GROUP MESSAGES

The screenshot shows a web-based messaging application interface titled "DevTalk Hub". The left sidebar includes a profile icon, a search bar, and sections for "Channels" (# Chit-Chat) and "Direct Messages" (sachin, alpha). The main area displays a message from "sachin" at 1:12 AM: "Hello". Below it, another message from "sachin" at 1:22 AM shows two yellow thumbs-up emojis. The top right corner indicates "2 users online". A large red circular icon in the bottom right corner features a hand holding a smartphone displaying a messaging interface.

localhost:3000

DevTalk Hub

sachin

Search

Channels

Chit-Chat

Direct Messages

sachin

alpha

Hello

sachin Today at 1:12 AM

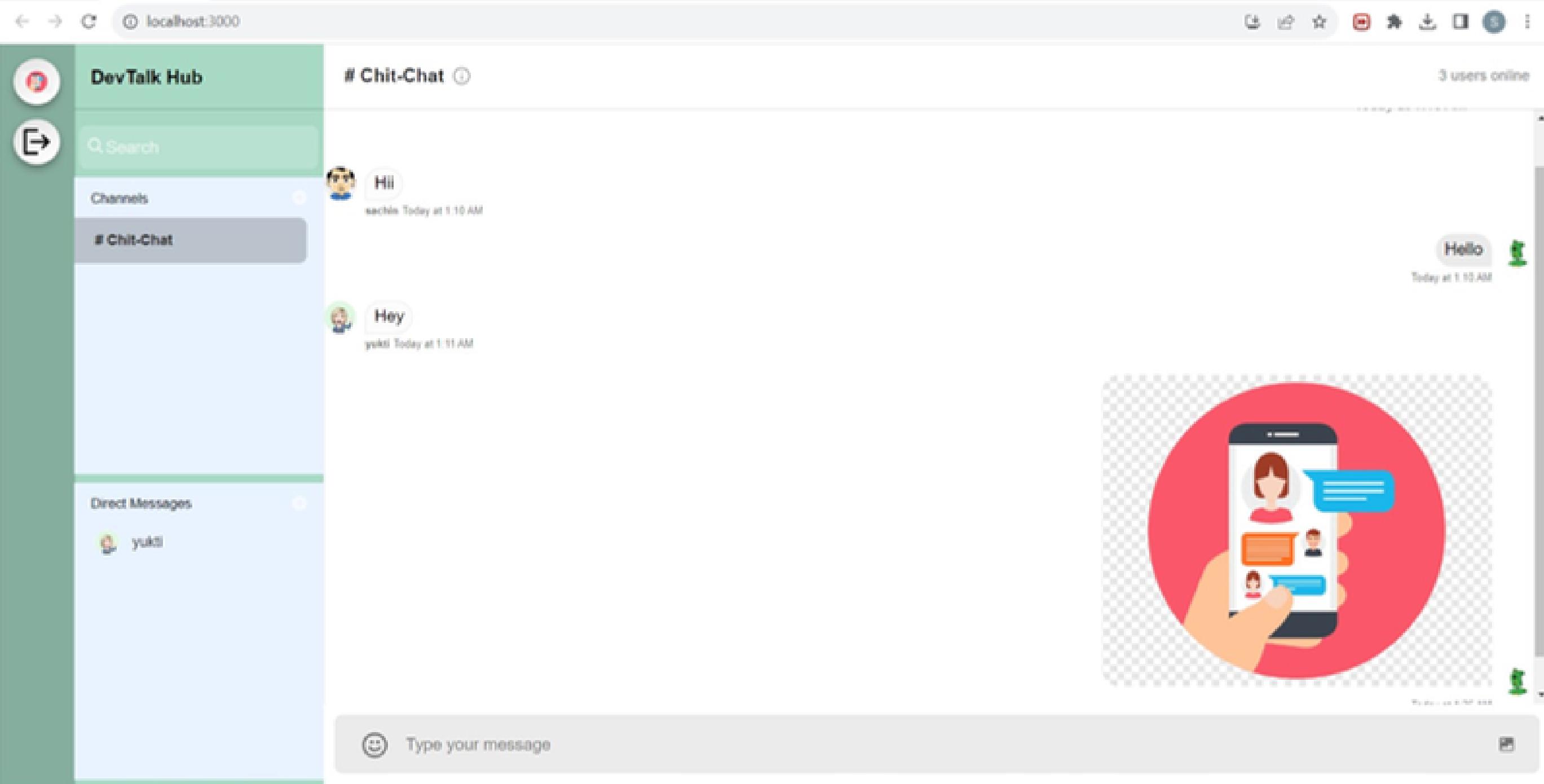
Now - Today at 1:12 AM

Today at 1:22 AM

Type your message

RESULTS & INTERPRETATION

DIRECT MESSAGES



The screenshot shows a web-based messaging application interface. On the left, there's a sidebar with a green header "DevTalk Hub" containing icons for user profile and direct message. Below it are sections for "Search" and "Channels", with "# Chit-Chat" highlighted. Under "Direct Messages", there's a list with one item: "yukt". The main area is titled "# Chit-Chat" and shows a conversation between two users:

- sachin: Hi! Today at 1:10 AM
- yukt: Hey Today at 1:11 AM
- (Your message placeholder)

A large circular icon in the bottom right corner features a hand holding a smartphone displaying a messaging interface.

CONCLUSION

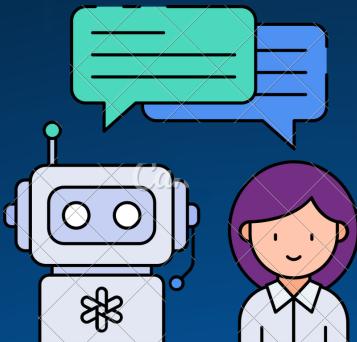
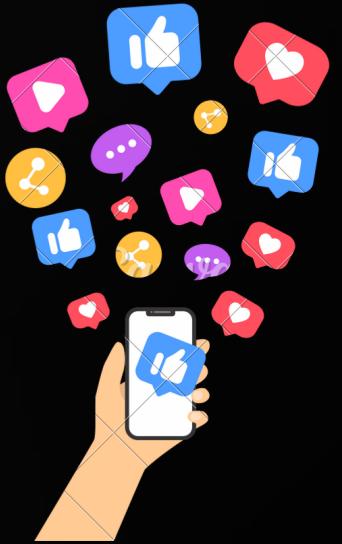
In the culmination of our Real-Time Chat Application journey, we've ventured into the heart of instant communication and technology fusion. The collaborative synergy of React.js, Node.js, Express.js, and the Stream Chat API has ushered in a new era of connectedness.

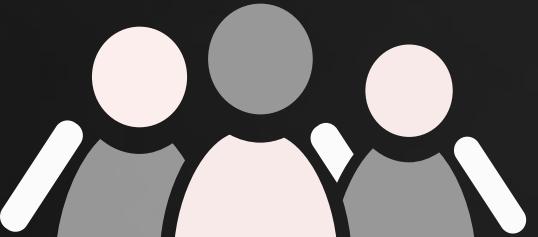
As we draw the curtain on this project, we celebrate the successful creation of a platform that breaks down geographical barriers, uniting individuals through real-time interaction. Beyond the technical achievement, this application symbolizes the evolving landscape of modern communication.

Our real-time messaging functionality, driven by the seamless interplay of frontend and backend technologies, underscores the potential of converging cutting-edge tools to bring practical solutions to life. The incorporation of the Stream Chat API elevates our application into a realm of instantaneous communication.

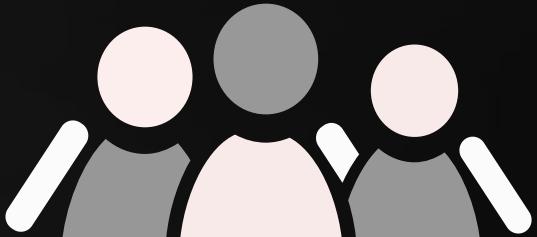
As this chapter closes, let's extend our gratitude to the dedicated team whose efforts have manifested this vision. Our project is not merely lines of code; it's a testament to innovation, resilience, and the transformative power of technology.

However, our journey continues. User feedback will guide us toward enhancements, ensuring our application remains dynamic and relevant in the face of ever-evolving digital landscapes.





Our team



Sachin Kumar Ray

00215611621

AI&ML

Yukti Batra

02115611621

AI&ML

Abhay Arya

01515611621

AI&ML

Nikita Kasotiya

14115602721

CSE

Neha

14415602721

CSE





THANK YOU

For watching this presentation