TASK-29

29. Render State in the User Interface Another Way:

- *Use a `map` function to dynamically render messages in the `MessageList` component.*

---

1. MessageList.jsx: *Using map to Render Messages Dynamically:*

```
import React from "react";

const MessageList = ({ messages }) => {
 return (
  <div className="message-list">
   {messages.length === 0 ? (
    <p>No messages yet. Start chatting!</p>
   ) : (
    messages.map((message) => (
     <div key={message.id} className="message">
      {message.text}
     </div>
    ))
   )}
  </div>
 );
};

export default MessageList;
```

---

2. How This Works:

- *Whenever the messages state in the parent component (like ChatRoom) updates, the MessageList component will automatically re-render with the new messages.*
- *The map function is a powerful way to dynamically render lists based on an array of data (in this case, the messages).*

---

3. Full Example:

ChatRoom.jsx:

```
import React, { useState } from "react";
import MessageList from "./MessageList";
import MessageInput from "./MessageInput";

const ChatRoom = ({ chatId }) => {
 const [messages, setMessages] = useState([]);

 const handleSendMessage = (message) => {
  setMessages((prevMessages) => [
   ...prevMessages,
   { id: prevMessages.length, text: message },
  ]);
 };

 return (
  <div className="chat-room">
   <h2>{chatId} Chat Room</h2>
   <MessageList messages={messages} />
   <MessageInput onSendMessage={handleSendMessage} />
  </div>
 );
};

export default ChatRoom;
```

MessageList.jsx:

```
import React from "react";

const MessageList = ({ messages }) => {
 return (
  <div className="message-list">
   {messages.length === 0 ? (
    <p>No messages yet. Start chatting!</p>
   ) : (
    messages.map((message) => (
     <div key={message.id} className="message">
      {message.text}
     </div>
    ))
   )}
  </div>
 );
};
export default MessageList;
```

MessageInput.jsx:

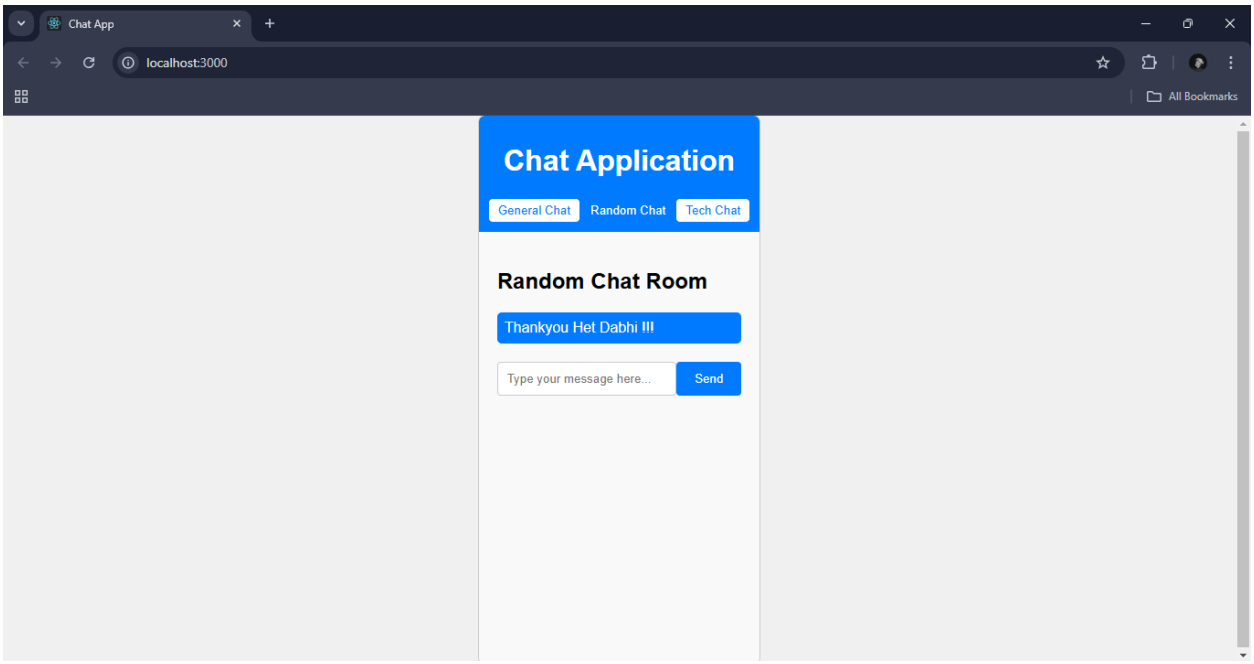
```
import React, { useState } from "react";

const MessageInput = ({ onSendMessage }) => {
 const [message, setMessage] = useState("");

 const handleInputChange = (e) => {
  setMessage(e.target.value);
 };

 const handleSendClick = () => {
  if (message.trim()) {
   onSendMessage(message);
   setMessage(""); // Clear the input after sending
  }
 };

 return (
  <div className="message-input">
   <input
    type="text"
    value={message}
    onChange={handleInputChange}
    placeholder="Type your message here..."
   />
   <button onClick={handleSendClick}>Send</button>
  </div>
 );
};

export default MessageInput;
```

---

4. *Final Outcome*:

- *Messages are added to the state when the user sends them.*
- ***map** dynamically renders these messages in the **MessageList**.*
- *The interface updates instantly as new messages are added.*

Output:


Chat App
localhost:3000
All Bookmarks
Chat Application
General Chat
Random Chat
Tech Chat
Random Chat Room
Thankyou Het Dabhi !!!
Type your message here...
Send