

TASK-28

28. Create a Stateful Component:

- Store chat messages in the state and update them as new messages are sent.

1. Create a Stateful ChatRoom Component:

- We'll modify the ChatRoom component to handle the state of the messages. We'll use React's `useState` hook to store and update the messages.

2. Updated ChatRoom.jsx with State:

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

const ChatRoom = ({ chatId }) => {
  // Initializing state to store messages
  const [messages, setMessages] = useState([]);

  // Function to handle new messages
  const handleSendMessage = (message) => {
    // Update the messages state with the new message
    setMessages((prevMessages) => [...prevMessages, { id: prevMessages.length,
    text: message }]);
  };

  return (
    <div className="chat-room">
      <h2>{chatId} Chat Room</h2>
      /* Render the list of messages */
      <MessageList messages={messages} />
      /* Message input component */
      <MessageInput onSendMessage={handleSendMessage} />
    </div>
  );
};

export default ChatRoom;
```

3. **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;
```

4. **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); // Send the message to the parent (ChatRoom)
      setMessage(""); // Clear the input after sending
    }
  };

  return (
    <div className="message-input">
      <input
        type="text"
        value={message}
      />
    </div>
  );
};
```

```

        onChange={handleInputChange}
        placeholder="Type your message here..."
      />
      <button onClick={handleSendClick}>Send</button>
    </div>
  );
};

export default MessageInput;

```

5. ChatApp.jsx:

```

import React, { useState } from "react";
import ChatHeader from "../ChatHeader";
import ChatRoom from "../ChatRoom";
import "../ChatApp.css";

const ChatApp = () => {
  const [activeChat, setActiveChat] = useState("general");

  const chats = [
    { id: "general", name: "General Chat" },
    { id: "random", name: "Random Chat" },
    { id: "tech", name: "Tech Chat" }
  ];

  const handleChatChange = (chatId) => {
    setActiveChat(chatId);
  };

  return (
    <div className="chat-app">
      <ChatHeader chats={chats} activeChat={activeChat}
        onChatChange={handleChatChange} />
      <ChatRoom chatId={activeChat} />
    </div>
  );
};

export default ChatApp;

```

Output:

