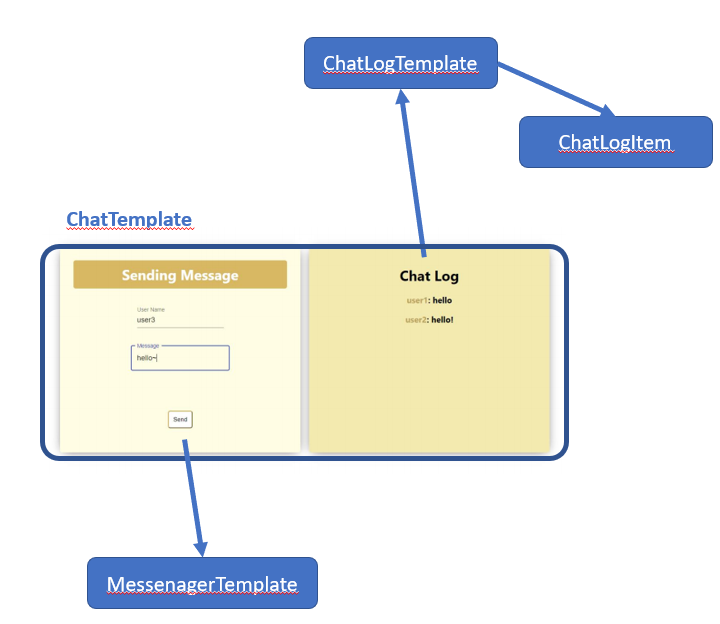
**고급소프트웨어실습 8주차 과제**

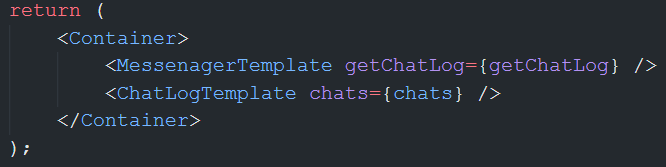
**20191657 최세은**



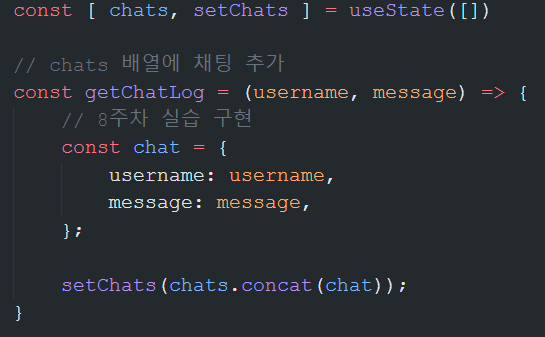
우선 이번 프로젝트는 react로 채팅 프로그램을 만들어보는 프로젝트이다. component에는 총 4개의 파일이 있으며 각각 ChatLogItem, ChatLogTemplate, ChatTemplate, MessenagerTemplate 이다.

1. ChatTemplate

두 개의 컴포넌트를 감싸는 템플릿 컴포넌트로 4개의 파일 중 가장 기본이 되는 파일이다.



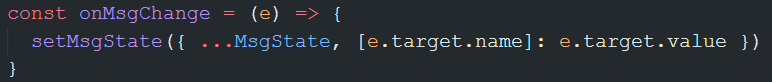
MessenagerTemplate, ChatLogTemplate 컴포넌트들을 return 하며 실행한다.



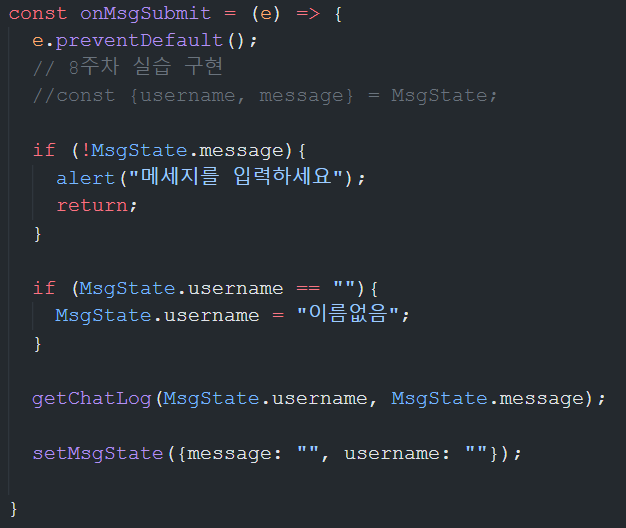
그리고 4개 중 가장 먼저 실행되는 파일이므로 chat를 여기서 채팅 보내는 사람(username)과 채팅 메시지(message)로 초기화 하고 getChatLog를 만들어두어 새로운 채팅이 setChats 함수로 chats 배열에 등록되게 하였다. 이 getChatLog는 채팅을 submit 하는 MessenagerTemplate 에서 쓰인다.

2. MessenagerTemplate

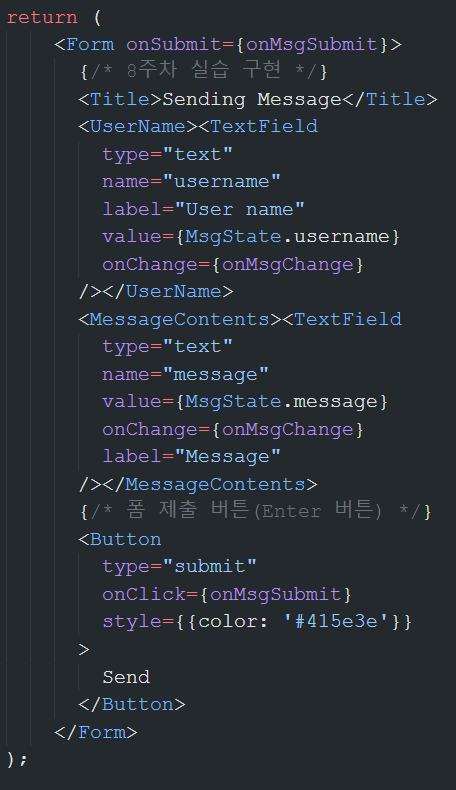
사용자의 채팅을 입력받는 컴포넌트이다.



onMsgChange는 채팅 메시지에 변화가 있을 때마다 바로 MsgState에 반영시켜준다. 채팅을 send 하기 전 메시지나 이름 칸이 변할 때마다 실행된다.



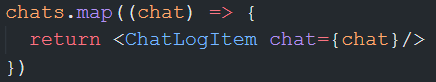
onMsgSubmit는 채팅을 send할 때 실행되는 함수로, 채팅 메시지가 비어있으면 ‘메시지를 입력하세요’라는 알림이 뜬다. 그리고 사용자 이름이 비어있으면 ‘이름없음’으로 자동 등록된다. 그리고 채팅이 send 되었다는 것은 chat log에 해당 채팅이 추가되었음을 의미하므로 getChatLog로 chats 배열에 채팅을 추가해주며, send 된 후 사용자 이름과 메시지 칸은 비워져야 하므로 setMsgState 함수로 비워준다.

이 때 화면의 입력 칸에 입력되는 값들이 바뀔 때마다 onMsgChange 함수가 실행되어 실시간으로 반영할 수 있었던 이유는 왼쪽의 코드 때문이다.

우선 각 field, 즉 입력 칸에 onChange로 변화가 있을 때마다 해당 함수(onMsgChange)가 실행되게 하였다.

그리고 채팅을 send 할 때, 즉 send 버튼을 누를 때에는 onClick을 이용해 onMsgSubmit 함수가 실행되게 하였다.

3. ChatLogTemplate

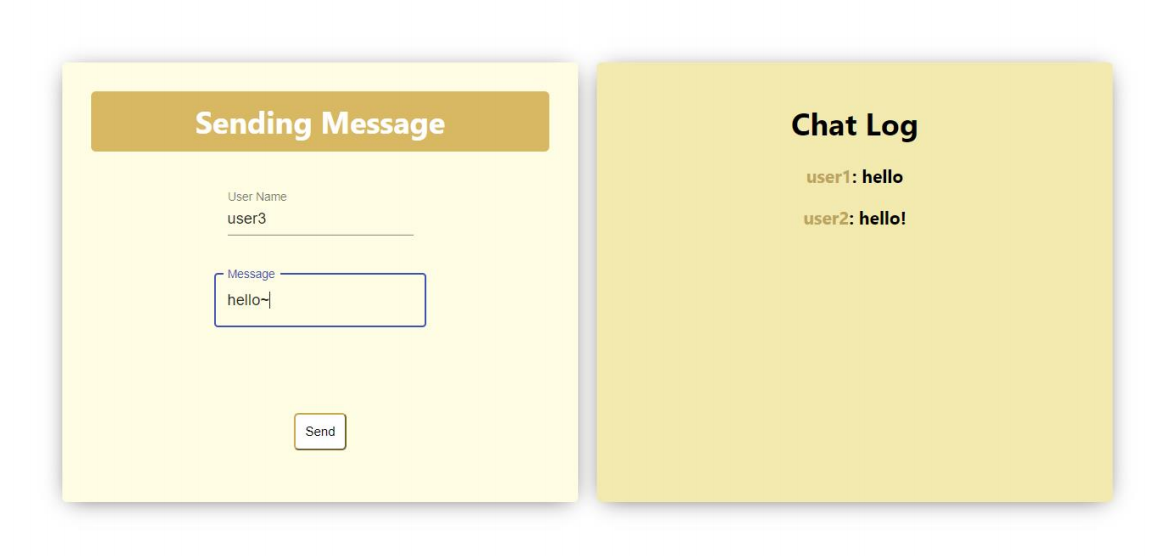


채팅 기록이 출력되는 컴포넌트이다. 이 컴포넌트는 위 코드로 채팅 한 개마다 ChatLogItem 컴포넌트를 이용해 출력한다. 따라서 map 함수를 사용하였다. map 함수란 배열 내 각각의 요소에 대해 주어진 함수를 수행하고 그 결과를 모아 새로운 배열 생성하는 함수이다.

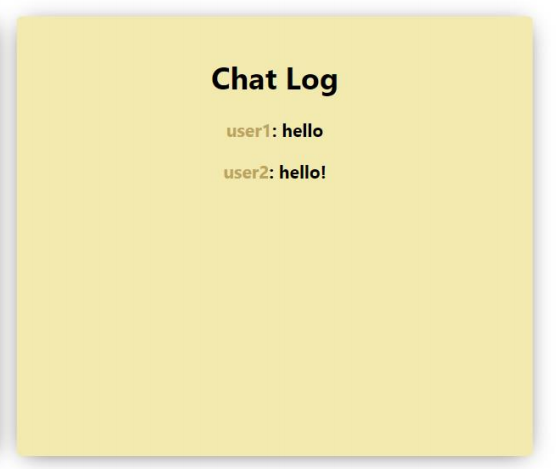
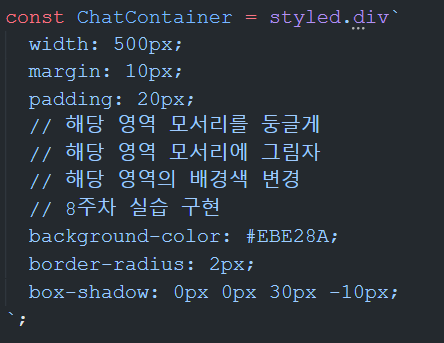
4. ChatLogItem



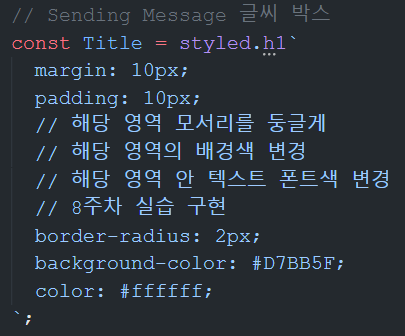
채팅 하나가 출력되는 컴포넌트이다. html 형식으로 출력시켜 준다.



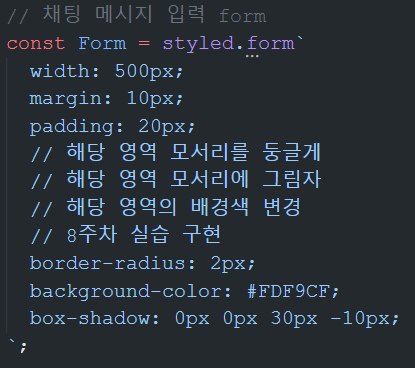
또한 위 화면을 그대로 구현하기 위해 알맞은 css를 작성하였는데, 다음과 같이 작성하였다.

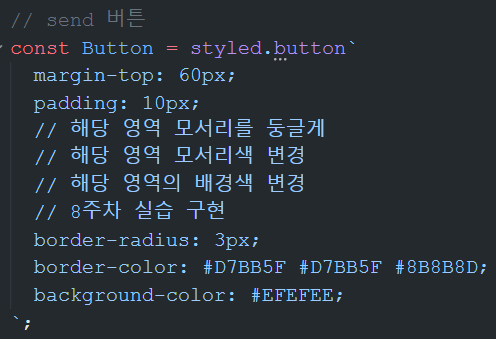


우선 큰 박스(컨테이너)의 모서리와 배경색을 위처럼 css를 추가하여 구현하였다.

그리고 왼쪽 박스의 title 박스에서 모서리, 배경색 변경 및 텍스트 스타일 변경도 css를 추가해 구현해 주었다.

왼쪽 박스의 큰 틀과 아래 send 버튼도 위와 같은 css 추가를 통해 스타일을 알맞게 변화시켰다. 이 때 ‘import TextField from "@material-ui/core/TextField";’ 코드를 통해 TextField의 css 스타일을 입력하는 박스에 적용하였다.