고급소프트웨어실습I

8주차 보고서

20171646 박태윤

8주차 실습에는 ChatLogItem, ChatLogTemplate, ChatTemplate, MessengerTemplate 총 4개의 주요 컴포넌트가 존재합니다.

**- ChatTemplate**

MessengerTemplate, ChatLogTemplate 두 컴포넌트를 감싸는 전체 컴포넌트입니다. 다음과 같이 구현하였습니다.

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

import styled from "styled-components";

import MessenagerTemplate from "./MessengerTemplate";

import ChatLogTemplate from "./ChatLogTemplate";

// 전체를 감싸는 컴포넌트

function ChatTemplate() {

const [ chats, setChats ] = useState([])

const nextId = useRef(0);

// chats 배열에 채팅 추가

const getChatLog = (username, message) => {

// 8주차 실습 구현

const chat = {

id: nextId.current,

username,

message,

};

setChats(chats.concat(chat))

nextId.current += 1;

}

return (

<Container>

<MessenagerTemplate getChatLog={getChatLog} />

<ChatLogTemplate chats={chats} />

</Container>

);

}

const Container = styled.div`

display: grid;

grid-template-columns: 1fr 1fr;

margin: 30px;

min-height: 30rem;

font-size: 20px;

position: relative;

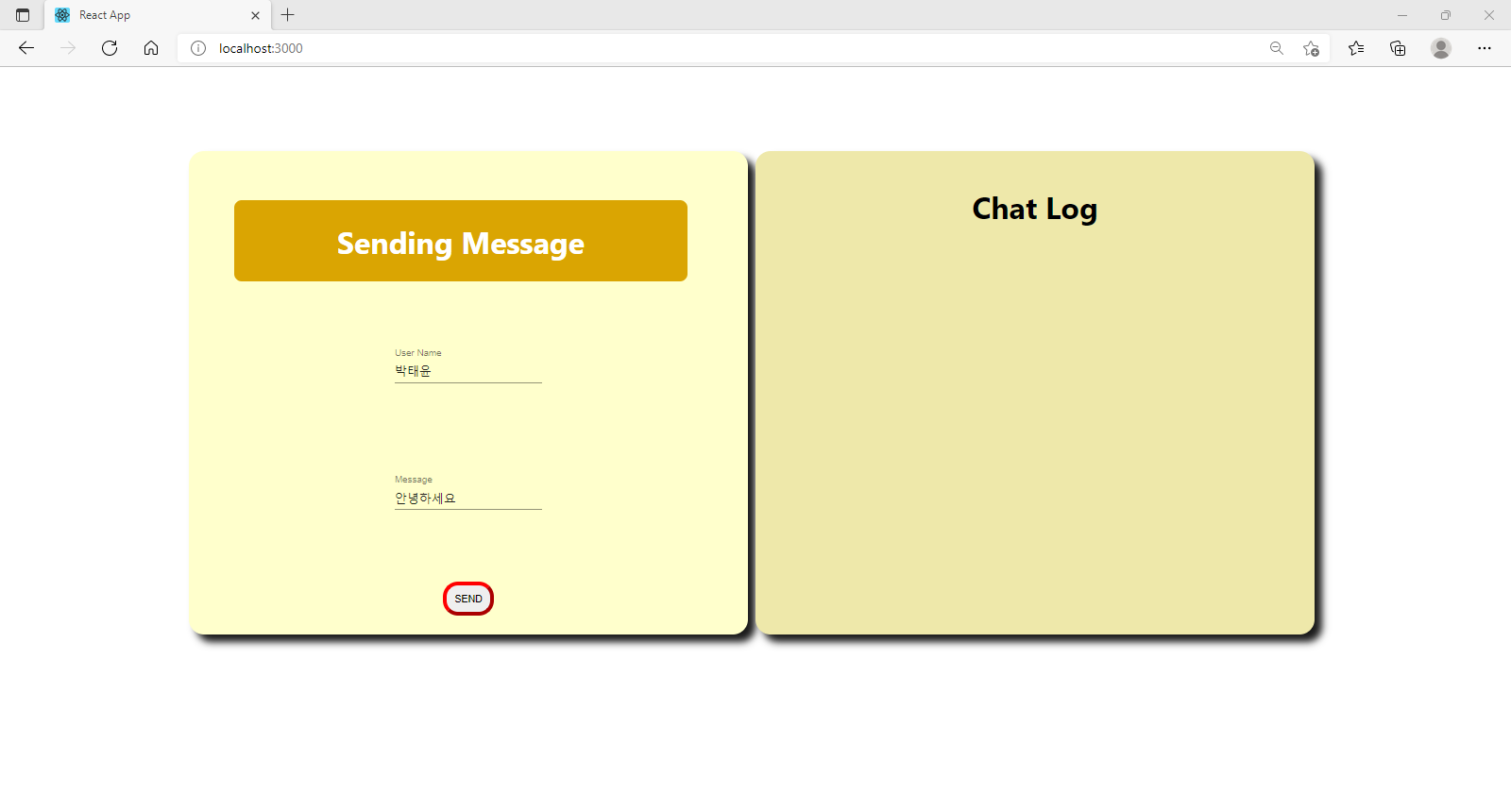
`;

export default ChatTemplate;

이 컴포넌트엔 getChatLog라는 함수가 존재합니다. 이 함수는 id와 채팅을 입력한 사람의 이름인 username, 해당 사용자가 입력한 메시지를 나타내는 message를 가지는 chat라는 객체를 생성해 채팅 목록을 나타내는 chats배열에 넣는 함수입니다.

**- MessengerTemplate**

사용자의 채팅을 입력 받는 부분을 나타내는 컴포넌트입니다. 프로그램 실행 화면에서 다음 부분을 나타냅니다.



붉은색 박스로 표시한 부분에서 사용자의 이름과 채팅을 입력 받아 SEND버튼을 누르면 오른쪽의 Chat Log로 보냅니다. 코드는 다음과 같이 구현하였습니다.

import React, { useState, useRef, useEffect } from "react";

import styled from "styled-components";

import TextField from "@material-ui/core/TextField";

import { width } from "../../node\_modules/@material-ui/system/index";

import ChatTemplate from "./ChatTemplate";

// Sending Message 부분

function MessenagerTemplate({ getChatLog }) {

const [MsgState, setMsgState] = useState({ message: "", username: "" })

const ref = useRef();

const onMsgChange = (e) => {

setMsgState({ ...MsgState, [e.target.name]: e.target.value })

}

// 메시지 입력 창에서 빈 경우에는 ‘메시지를 입력하세요‘라는 내용의 경고 창 출력 후 리턴(alert 함수 사용)

// 사용자의 이름 입력 창이 빈 경우에는 사용자 이름을 ‘이름없음’으로 저장

// MsgState의 속성(키)들의 값을 빈 문자열로 초기화

const onMsgSubmit = (e) => {

e.preventDefault();

// 8주차 실습 구현

if (!MsgState) return;

if (MsgState.message == "") {

alert("메시지를 입력하세요");

return;

}

if(MsgState.username == ""){

MsgState.username = "이름 없음";

}

//getChatLog.onSubmit(MsgState);

getChatLog(MsgState.username, MsgState.message);

setMsgState({ message: "", username: "" });

}

return (

<div>

<Form>

<UserName>

<TextField

label="User Name"

value={MsgState.username}

name="username"

onChange={onMsgChange}

/>

</UserName>

<MessageContents>

<div style = {{textAlign : 'center'}}>

<TextField

label="Message"

value={MsgState.message}

name="message"

onChange={onMsgChange}

/>

</div>

</MessageContents>

<Title>

<h1 style={{ color: '#ffffff' }}>Sending Message</h1>

</Title>

</Form>

<Button

type="submit"

onClick={onMsgSubmit}

style={{ color: '#000000' }}

>

SEND

</Button>

</div>

);

}

// Sending Message 글씨 박스

const Title = styled.h1`

width: 600px;

margin: 10px;

//padding: 1px;

display: flex;

flex-direction: column;

background-color: #DAA502;

font-size: 20px;

position: relative;

top : -200px;

right : -30px;

border-radius : 10px;

// 해당 영역 모서리를 둥글게

// 해당 영역의 배경색 변경

// 해당 영역 안 텍스트 폰트색 변경

// 8주차 실습 구현

`;

// 채팅 메시지 입력 form

const Form = styled.form`

width: 700px;

height: 600px;

margin: 10px;

padding: 20px;

display: flex;

flex-direction: column;

background-color: #FFFFCC;

position: relative;

top: -10px;

border-radius : 20px;

box-shadow : 10px 10px 10px rgba(0,0,0,0.9);

//overflow-y: scroll;

// 해당 영역 모서리를 둥글게

// 해당 영역 모서리에 그림자

// 해당 영역의 배경색 변경

// 8주차 실습 구현

`;

// username 표시

const UserName = styled.div`

margin: 40px;

position: relative;

top : 200px;

`;

// 채팅 메시지 부분

const MessageContents = styled.div`

margin: 30px;

position: relative;

font-size: 10px;

top : 250px;

`;

// send 버튼

const Button = styled.button`

margin-top: 60px;

padding: 10px;

position: relative;

top : -150px;

border-radius : 20px;

border-color : red;

border-width : 5px;

// 해당 영역 모서리를 둥글게

// 해당 영역 모서리색 변경

// 해당 영역의 배경색 변경

// 8주차 실습 구현

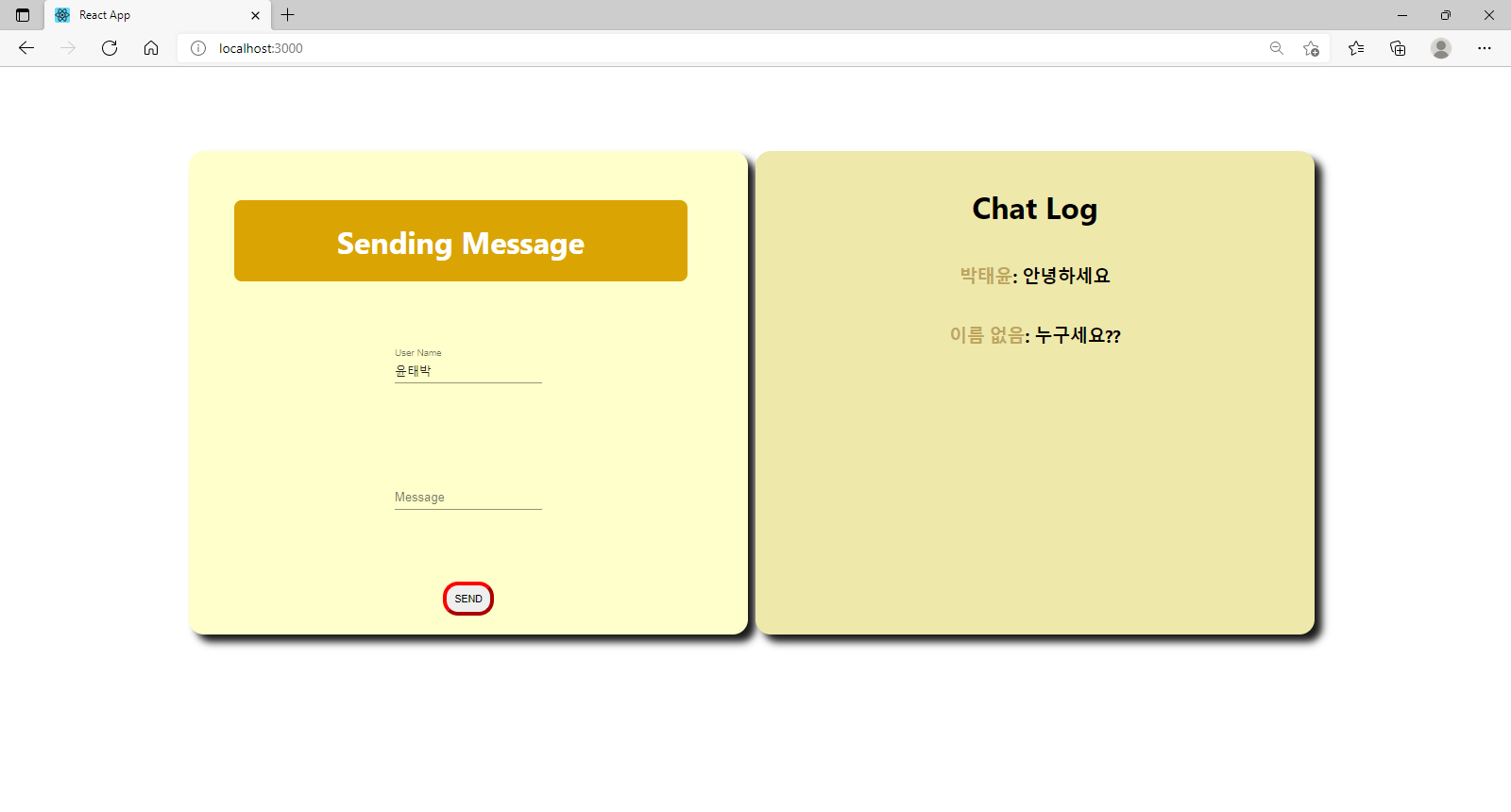
`;

export default MessenagerTemplate;

해당 컴포넌트에는 onMsgChange, onMsgSubmit함수가 존재합니다. onMsgChange는 입력창이 바뀔 때 마다 update를 해주는 함수이며 onMsgSubmit은 ChatTemplate에서 구현된 getChatLog를 호출하여 입력받은 MsgState.username, MsgState.message를 chats배열에 추가하는 함수입니다. 이 때, 입력 받은 MsgState.message의 내용이 없는 경우 즉 메시지를 입력 받지 않은 경우에는 “메시지를 입력하세요”라는 내용을 가지는 경고창을 alert함수를 통해 부른 뒤 chats배열에 추가하지 않고 return하며 username을 입력 받지 않은 경우에는 default로 “이름 없음”을 MsgState.username에 assign해줍니다. 해당 컴포넌트에서는 Sending Message라고 적혀져 있는 글씨 박스와 전체 채팅 메시지 입력 폼, username을 입력 받는 부분과 채팅 메시지를 입력 받는 부분 그리고 SEND버튼을 css를 통해 화면에 나타나도록 합니다. 특히 사용자 이름, 메시지를 입력 받는 부분은 TextField태그를 이용하여 onMsgChange함수를 호출하여 입력창 state를 업데이트 하는 식으로 입력을 받게 구현하였습니다. 또한 SEND버튼을 누르면 입력 받은 내용을 제출해야 하기 때문에 onMsgSubmit함수를 호출하여 리턴하였습니다. Sending Message 글씨 박스를 나타내는 css는 position을 relative로 설정한 뒤 top, right에 적절하게 값을 넣어 위치를 조정하였으며 border-radius를 통해 모서리를 둥글게 만들었습니다. 또한 전체 채팅 메시지 입력 폼을 나타내는 Form css역시 position을 relative로 설정하여 다른 css와 겹쳐서 보일 수 있도록 구현하였으며 box-shadow를 사용하여 배경 그림자를 나타냈습니다. 그 외 나머지 css도 position을 relative로 설정하여 다른 css와 겹쳐서 표현이 될 수 있도록 하였으며 top등을 이용하여 적절하게 위치를 조정하였습니다.

**- ChatLogTemplate**

사용자가 입력한 채팅 기록을 나타내는 컴포넌트입니다. 실행 화면에선 다음 부분에 표현이 됩니다.



앞서 언급한 MessengerTemplate에서 채팅을 입력 받으면 그 내용을 출력합니다. 코드는 다음과 같이 구현하였습니다.

import React from "react";

import styled from "styled-components";

import ChatLogItem from "./ChatLogItem";

import MessengerTemplate from "./MessengerTemplate";

function ChatLogTemplate({ chats }) {

// chats 배열을 받아 저장해 둔 채팅 로그를 보여줌

// ChatLogItem 컴포넌트 사용

return (

<ChatContainer>

<h1 style={{ color: '000000'}}>Chat Log</h1>

{chats.map((chat) => {

return <ChatLogItem chat={chat} username={chat.username} message={chat.message} />

})}

</ChatContainer>

);

}

const ChatContainer = styled.div`

width: 700px;

height: 600px;

margin: 0px;

padding: 20px;

display: flex;

flex-direction: column;

background-color: #EEE8AA;

position: relative;

border-radius : 20px;

box-shadow : 10px 10px 10px rgba(0,0,0,0.9);

// 해당 영역 모서리를 둥글게

// 해당 영역 모서리에 그림자

// 해당 영역의 배경색 변경

// 8주차 실습 구현

`;

export default ChatLogTemplate;

ChatLogTemplate에서는 채팅 기록을 담고 있는 chats배열을 인자로 받아 map함수를 통해 채팅 로그를 보여줍니다. chat객체를 한 줄로 표현하는 ChatLogItem을 사용하여 이를 나타냅니다. ChatContainer css는 전체 Chat Log폼을 나타내며 이 역시 box-shadow를 이용하여 배경 그림자를 나타냈습니다.

**- ChatLogItem**

사용자 이름과 메시지를 담고 있는 chat객체 하나를 한 줄로 표현하는 컴포넌트입니다. 코드는 다음과 같이 구현되어있습니다.

import React from "react";

import styled from "styled-components";

// chat 객체 하나를 인자로 받아 한 줄로 표현하기 위한 컴포넌트

function ChatLogItem({chat}) {

return (

<div>

<UsernameContainer>

{chat.username}<ChatContainer>: {chat.message}</ChatContainer>

</UsernameContainer>

</div>

);

}

const UsernameContainer = styled.h3`

color: #bda55d;

`;

const ChatContainer = styled.span`

color: black;

`;

export default ChatLogItem;

각 주요 컴포넌트들의 전체적인 구조는 다음과 같이 표현할 수 있습니다.

MessengerTemplate

ChatLogItem

ChatLogTemplate

ChatTemplate