**삭제하기**

1. firestore 내부 doc 내부 delete 함수를 사용. Promise를 반환한다.

<https://firebase.google.com/docs/reference/js/firebase.firestore.DocumentReference#delete>

1. 삭제 작업을 수행하는 onDeleteClick을 정의

* window.confirm을 통해 진짜 삭제할 것인지 물어본자
* 삭제한다고 하면 dbService.doc(`nweet/${userObj.id}`).delete() 로 삭제
* **이때, home.js에서 nweet를 불러올 때, doc.id를 이미 nweet에 넣었어서 위처럼 그대로 쓰면 됨**
* 이때 doc() 내부의 ``은 역 따옴표로 안에 다른 변수를 사용할 때 쓰는듯.(${userObj.id} 처럼)

const onDeleteClick = async() => {

        const ok = window.confirm("Are you sure you want to delete this nweet?");

        if(ok) {

            //delete nweet

            await dbService.doc(`nweets/${nweetObj.id}`).delete();

        }

    }

1. 삭제 버튼에다가 onClick={onDeleteClick} 을 추가하면 끝.

**수정하기**

1. 사용자가 버튼을 누르면 edit 창이 뜨도록 수정

* 먼저 edit모드인지 아닌지를 기록하는 상태 추가
* 그 상태를 바꾸는 toggle 함수 추가
* edit 모드시 nweet.text가 아닌 editform이 뜨도록 함
* edit 모드시 edit을 취소할 수 있도록 Cancel 버튼 추가

const [editing, setEditing] = useState(false);

…

    const toggleEditing = () => setEditing((prev) => !prev);

    …

    return (

        <div>

            {editing ? (

                <>

                    <form onSubmit={onSubmit}>

                        <input

                            text="text"

                            placeholder="Edit your nweet"

                            value={newNweet}

                            required

                        />

                    </form>

                    <button onClick={toggleEditing}>Cancel</button>

                </>

            ) : (

                <>

                    <h4>{nweetObj.text}</h4>

                    {isOwner && (

                        <>

                            <button onClick={onDeleteClick}>

                                Delete Nweet

                            </button>

                            <button onClick={toggleEditing}>Edit Nweet</button>

                        </>

                    )}

                </>

            )}

        </div>

    );

1. 수정하기 위한 onChange 함수와, 업데이트하기 위한 onSubmit함수 추가

* onChange에선 newNweet를 바꾼다.
* onSubmit에선 dbService.doc().update() 함수를 사용한다.
* .doc()함수는 마찬가지로 필드위치를 넣고, update함수엔 그 필드의 수정 함수를 집어넣는다.
* 이 예제에선, 필드의 text를 newNweet로 바꾼다.
* 수정후 onSubmit 마지막에서, edit form을 닫기 위해 editing을 false로 바꾸자.
* 두 함수를 만들면, form에 추가

const onSubmit = async (event) => {

        event.preventDefault();

        await dbService.doc(`nweets/${nweetObj.id}`).update({

            text: newNweet,

        });

        setEditing(false);

    };

    const onChange = (event) => {

        const {

            target: { value },

        } = event;

        setNewNweet(value);

    };

    return (

        <div>

            {editing ? (

                <>

                    <form onSubmit={onSubmit}>

                        <input

                            text="text"

                            placeholder="Edit your nweet"

                            value={newNweet}

                            required

                            onChange={onChange}

                        />

                        <input type="submit" value="Update Nweet" />

                    </form>

                    <button onClick={toggleEditing}> Cancel</button>

                </>

            )