**소셜 서비스로 로그인하는 법을 추가하는 챕터**

1. 인증하는 화면은 auth.js 이므로 auth.js 추가
2. auth.js에서 예전에 추가한 continue with google, continue with github 버튼에 이름을 지어줌

<button name=”google”> …

<button name=”gitgub”>….

1. 소셜 로그인을 위해선 firebase 기능을 사용해야함. 두가지가 있음
   1. <https://firebase.google.com/docs/reference/js/firebase.auth.Auth?authuser=2#signinwithpopup>
      1. signInWithPopup : redirect보다 더 간결함
   2. <https://firebase.google.com/docs/reference/js/firebase.auth.Auth?authuser=2#signinwithredirect>
      1. signInWithRedirect
2. 우리는 popup을 사용함. popup을 쓰기 위해선 provider를 만들어줘야함. 따라서 google일때와 github일 때에 맞춰 각각 다른 provider를 생성해주기 위해 onSocialClick함수를 추가하고, 버튼에 추가
   1. 만든 provider는 auth.signInWithPopup(provider)로, 소셜 로그인 창을 띄울 수 있다.

const onSocialClick = async(event) => {

const {target:{name}} = event;

let provider;

if(name == “google”) {

provider = new firebaseInstance.auth.GoogleAuthProvider();

} else if (name == “github”) {

provider = new firebaseInstance.auth.GithubAuthProcider();

}

const data = await authService.signInWithPopup(provider);

console.log(data);

}

…

<button onClick={onSocialClick} name=”google”>…

<button onClick={onSocialClick} name=”github”>…

1. 문서에 보면 각 소셜 AuthProvider를 생성하기 위해 firebase.auth로 접근하는 걸 알 수 있다.
   1. 따라서 그냥 firebase를 firebaseInstance라고 export하면 굳이 또 firebase를 import하지 않아도 된다.

//firebase.js

…

export const firebaseInstance = firebase;