File - D:\react_project\nomadCoder\nomad_nwitter_Study\nf_nwitter36_nweet_CRUD_done\src\fbase.js

```
import {initializeApp} from "firebase/app";
import {getAuth} from "firebase/auth";
import {getFirestore} from "firebase/firestore";

const firebaseConfig = {
    apiKey: process.env.REACT_APP_API_KEY,
    authDomain: process.env.REACT_APP_AUTH_DOMAIN,
    projectId: process.env.REACT_APP_PROJECT_ID,
    storageBucket: process.env.REACT_APP_STORAGE_BUCKET,
    messagingSenderId: process.env.REACT_APP_MESSAGING_SENDER_ID,
    appId: process.env.REACT_APP_APP_ID,
};

//export const app = initializeApp(firebaseConfig);
const app = initializeApp(firebaseConfig);

export const db = getFirestore(app);

export const auth = getAuth(app);

```

File - D:\react_project\nomadCoder\nomad_nwitter_Study\nf_nwitter36_nweet_CRUD_done\src\index.js

```
import React from "react";
import ReactDOM from "react-dom";
import App from "./components/App";

ReactDOM.render(
    <React.StrictMode>
        <App />
    </React.StrictMode>,
    document.getElementById("root")
);

```

```
import {
  createUserWithEmailAndPassword,
  GithubAuthProvider,
  GoogleAuthProvider,
  signInWithEmailAndPassword,
  signInWithPopup,
} from "firebase/auth";
import React, { useState } from "react";
import { auth } from "../fbase";


const Auth = () => {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [newAccount, setNewAccount] = useState(true);
  const [error, setError] = useState("");

  const onChange = (e) => {
    console.log("e.target.value:", e.target.value);
    const {
      target: { name, value },
    } = e;
    if (name === "email") setEmail(value);
    if (name === "password") setPassword(value);
  };

  const onSubmit = async (e) => {
    e.preventDefault();
    console.log("email, password ...", email, password);
    try {
      let data;
      if (newAccount) {
        data = await createUserWithEmailAndPassword(auth, email, password);
      } else {
        data = await signInWithEmailAndPassword(auth, email, password);
      }
      console.log("data:", data);
    } catch (error) {
      setError(error.message);
      console.log("login error", error.message);
    }
  };

  const toggleAccount = () => setNewAccount((prev) => !prev);

  const onSocialClick = async (event) => {
    const {
      target: { name },
    } = event;
    let provider;
    try {
      if (name === "google") {
        provider = new GoogleAuthProvider();
      }
      if (name === "github") {
        provider = new GithubAuthProvider();
      }
      console.log("000000======>", provider);
      //
      const result = await signInWithPopup(auth, provider);
      //
      console.log("1111111======>", provider);

      console.log("result:", result);

    } catch (error) {
      console.log("ERROR:", error);
    }
  };

  return (
    <div>
      <form onSubmit={onSubmit}>
        <input
          name="email"
          type="text"
          placeholder="Email"
          required
          value={email}
          onChange={onChange}
        />
        <input
          name="password"
          type="password"
          placeholder="Pathword"
          required
          value={password}
          onChange={onChange}
        />
        <input type="submit" value={newAccount ? "Create Account" : "Log In"} />
        {error}
      </form>
      <span onClick={toggleAccount}>
        {newAccount ? "Sign In..." : "Create Account"}
      </span>
      <div>
        <button onClick={onSocialClick} name="google">
          Continue with Google
        </button>
        <button onClick={onSocialClick} name="github">
          Continue with Github
        </button>
      </div>
    </div>
  );
};

export default Auth;

```

```
import React, {useEffect, useState} from "react";
import {db} from "../fbase";
import {
    collection,
    addDoc,
    query,
    // getDocs,
    // getFirestore,
    orderBy,
    onSnapshot,
} from "firebase/firestore";
import Nweet from "../components/Nweet";

// from Router.js
const Home = ({userObj}) => {
    console.log("-----------> user", userObj.uid);
    const [nweet, setNweet] = useState("");
    const [nweets, setNweets] = useState([]);

    // const getNweets = async () => {
    //   console.log("====>Start");
    //   const q = query(collection(db, "nweets"));
    //   const querySnapshot = await getDocs(q);
    //   querySnapshot.forEach((doc) => {
    //     const nweetObj = {
    //       ...doc.data(),
    //       id: doc.id,
    //     };
    //     setNweets((prev) => [nweetObj, ...prev]);
    //   });
    //   console.log(nweets);
    //   console.log("====>END");
    // };

    useEffect(() => {
        // 실시간으로 데이타를 데이타베이스에서 가져오기
        const q = query(
            //collection(getFirestore(), "nweets"), //db=getFirestore()가 fbase.js에서 export 되었다
            collection(db, "nweets"),
            // where('text', '==', 'hehe') // where뿐만아니라 각종 조건 이 영역에 때려부우면 됨
            orderBy("createdAt")
        );
        const unsubscribe = onSnapshot(q, (querySnapshot) => {
            const newArray = querySnapshot.docs.map((doc) => {
                return {
                    id: doc.id,
                    ...doc.data(),
                };
            });

            setNweets(newArray);
            console.log("Current tweets in CA: ", newArray);
        });

        return () => {
            unsubscribe();
        };
    }, []);

    //

    const onSubmit = async (e) => {
        e.preventDefault();

        try {
            const docRef = await addDoc(collection(db, "nweets"), {
                text: nweet,
                createdAt: Date.now(),
                creatorId: userObj.uid,
            });
            setNweet("");
            console.log("Document - Succeed: ", docRef);
        } catch (error) {
            console.error("Error adding document: ", error);
        }
    };

    const onChange = ({target: {value}}) => {
        setNweet(value);
    };

    //   const onChange = (e) => {
    //     const {
    //       target: { value },
    //     } = e;
    //     setNweet(value);
    //   };

    return (
        <div>
            {/* 로그인-계정 */}
            <form onSubmit={onSubmit}>
                Home
                <input
                    onChange={onChange}
                    value={nweet}
                    type="text"
                    placeholder="What is on your mind?"
                    maxLength={120}
                />
                <input type="submit" value="Nweet" />
            </form>

            {/* 리스팅 */}
            <div>
                {nweets.map((e) => (
                    <Nweet key={e.id} nweetObj={e} isOwner={e.creatorId === userObj.uid} />
                ))}
            </div>
        </div>
    );
};

export default Home;

```

File - D:\react_project\nomadCoder\nomad_nwitter_Study\nf_nwitter36_nweet_CRUD_done\src\routes\Profile.js

```
import React from "react";
import { useNavigate } from "react-router-dom";
//import { useNavigate } from "react-router"; //useHistory, push 사용불가

import { auth } from "../fbase";

const Profile = () => {
  const navigate = useNavigate();
  const onLogOutClick = () => {
    auth.signOut();
    navigate("/");
  };

  return (
    <>
      <button onClick={onLogOutClick}>Log Out</button>
    </>
  );
};

export default Profile;

```

File - D:\react_project\nomadCoder\nomad_nwitter_Study\nf_nwitter36_nweet_CRUD_done\src\routes\EditProfile.js

```
import React from "react";

export default () => <span>Edit Profile</span>;

```

File - D:\react_project\nomadCoder\nomad_nwitter_Study\nf_nwitter36_nweet_CRUD_done\src\components\App.js

```
import React, {useEffect, useState} from "react";
import AppRouter from "./Router";
import {auth} from "../fbase";
import {onAuthStateChanged} from "firebase/auth";

function App() {
    console.log("auth.r..", auth);
    console.log("auth.currentUser..", auth.currentUser);

    const [init, setInit] = useState(false);
    const [isLoggedIn, setIsLoggedIn] = useState(false); //이것 대신 AppROuter 속성을 조정할 수 있음...
    const [userObj, setUserObj] = useState(null);

    useEffect(() => {
        onAuthStateChanged(auth, (user) => {
            if (user) {
                setIsLoggedIn(true);
                setUserObj(user);
                console.log("userObj ...", userObj);
            } else {
                setIsLoggedIn(false);
            }
            setInit(true);
            console.log("useEffect changed ...", user);
        });
    }, []);

    return (
        <>
            {init ? <AppRouter isLoggedIn={isLoggedIn} userObj={userObj} /> : "Initializing"}
            {/* {init ? <AppRouter isLoggedIn={Boolean(userObj)} userObj={userObj}/> : "Initializing" } */}
            <footer>&copy; {new Date().getFullYear()} Nwitter</footer>
        </>
    );
}

export default App;

```

File - D:\react_project\nomadCoder\nomad_nwitter_Study\nf_nwitter36_nweet_CRUD_done\src\components\Nweet.js

```
import {deleteDoc, doc, updateDoc} from "@firebase/firestore";
import React, {useState} from "react";
import {db} from "../fbase";

// called each nweet message from Home.js... map
const Nweet = ({nweetObj, isOwner}) => {
    console.log("nweetObj.creatorId called from Home.js ...", nweetObj.creatorId);
    console.log("nweetObj.text called from Home.js ...", nweetObj.text);
    console.log("isOwner ???", isOwner);

    const [isEditing, setIsEditing] = useState(false);
    const [newNweet, setNewNweet] = useState(nweetObj.text);

    const onDeleteClick = async () => {
        const ok = window.confirm("Are you sure you want to delete this nweet?");
        console.log("window.confirm OK???", ok);
        if (ok) {
            // Delete literal
            const nweetTextRef = doc(db, "nweets", `${nweetObj.id}`);
            await deleteDoc(nweetTextRef);
        }
    };

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

    const onChangeClick = (e) => {
        const {
            target: {value},
        } = e;
        setNewNweet(value);
    };

    const onSubmit = async (e) => {
        e.preventDefault();
        console.log("onSubmit -nweetobj - newNweet", nweetObj, newNweet);
        const nweetTextRef = doc(db, "nweets", `${nweetObj.id}`);
        await updateDoc(nweetTextRef, {text: newNweet});
        setIsEditing(false);
    };

    return (
        <div>
            {isEditing ? (
                <>
                    <form onSubmit={onSubmit}>
                        <input
                            type="text"
                            placeholder="Edit your nweet"
                            value={newNweet}
                            required
                            onChange={onChangeClick}
                        />
                        <input type="submit" value="Update Nweet" />
                    </form>
                    <button onClick={toggleEditing}>Cancel</button>
                </>
            ) : (
                <>
                    <h4>{nweetObj.text}</h4>
                    {isOwner && (
                        <>
                            <button onClick={onDeleteClick}>Delete Nweet!!</button>
                            <button onClick={toggleEditing}>Edit Nweet...</button>
                        </>
                    )}
                </>
            )}
        </div>
    );
};

export default Nweet;
```

File - D:\react_project\nomadCoder\nomad_nwitter_Study\nf_nwitter36_nweet_CRUD_done\src\components\Router.js

```
import { BrowserRouter as Router, Route, Routes } from "react-router-dom";

import Auth from "../routes/Auth";
import Home from "../routes/Home";
import Navigation from "./Navigation";
import Profile from "../routes/Profile";
// import Auth from 'routes/Auth';
// import Home from 'routes/Home';

const AppRouter = ({ isLoggedIn, userObj }) => {
  return (
    <Router>
      {isLoggedIn && <Navigation />}
      <Routes>
        {isLoggedIn ? (
          <>
            <Route exact path="/" element={<Home userObj={userObj} />} />
            <Route exact path="/profile" element={<Profile />} />
          </>
        ) : (
          <Route exact path="/" element={<Auth />}></Route>
        )}
      </Routes>
    </Router>
  );
};

export default AppRouter;

```

File - D:\react_project\nomadCoder\nomad_nwitter_Study\nf_nwitter36_nweet_CRUD_done\src\components\Navigation.js

```
import React from "react";
import { Link } from "react-router-dom";

const Navigation = () => {
  return (
    <ul>
      <li>
        <Link to="/">Home</Link>
      </li>
      <li>
        <Link to="/profile">Profile</Link>
      </li>
    </ul>
  );
};

export default Navigation;

```