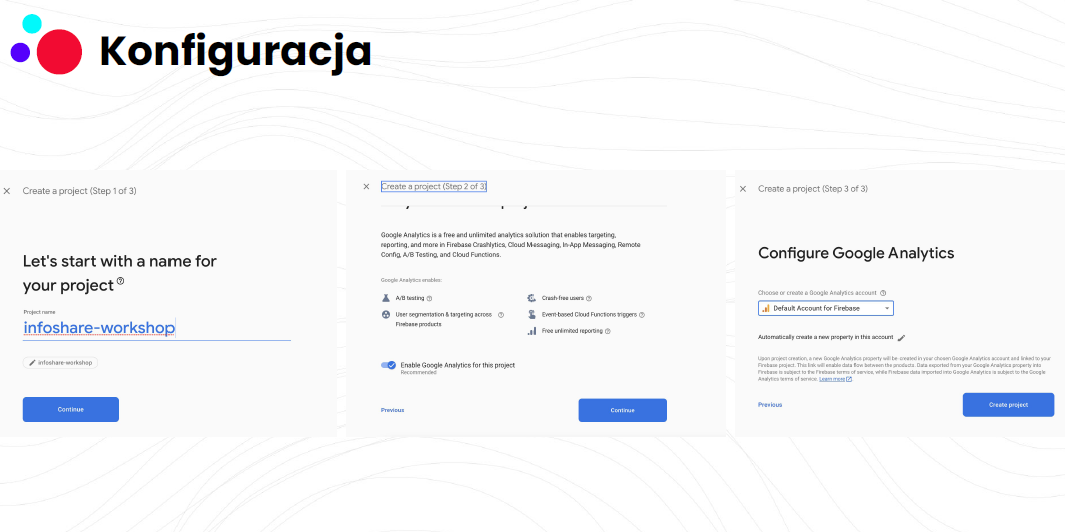
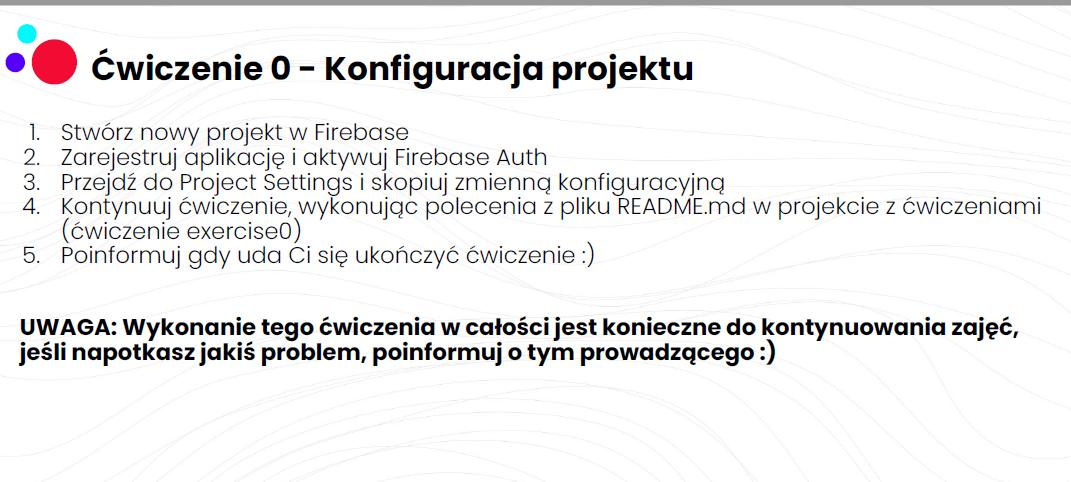
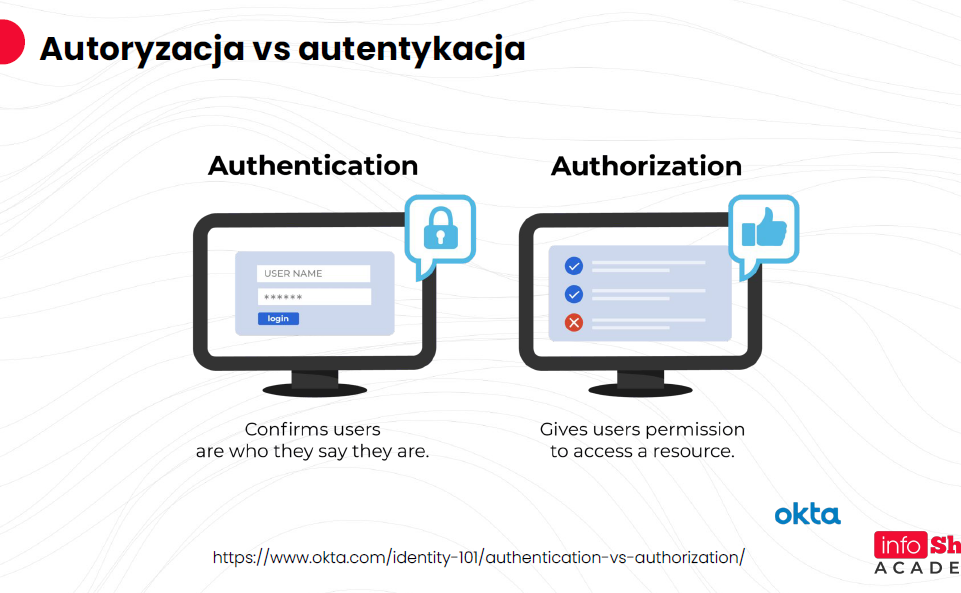
Firebase- rejestracja, logowanie i konto użytkownikan01.02.2023 Dawid Bulinski

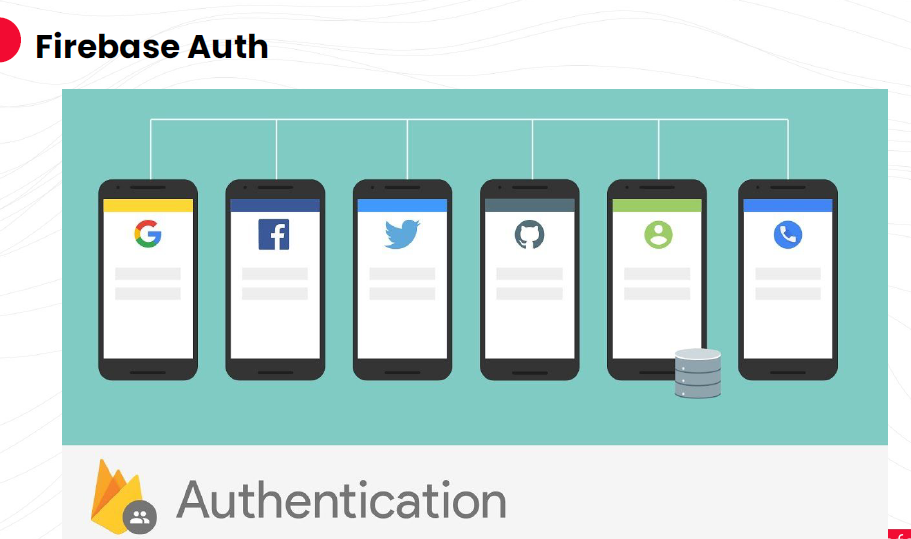
  

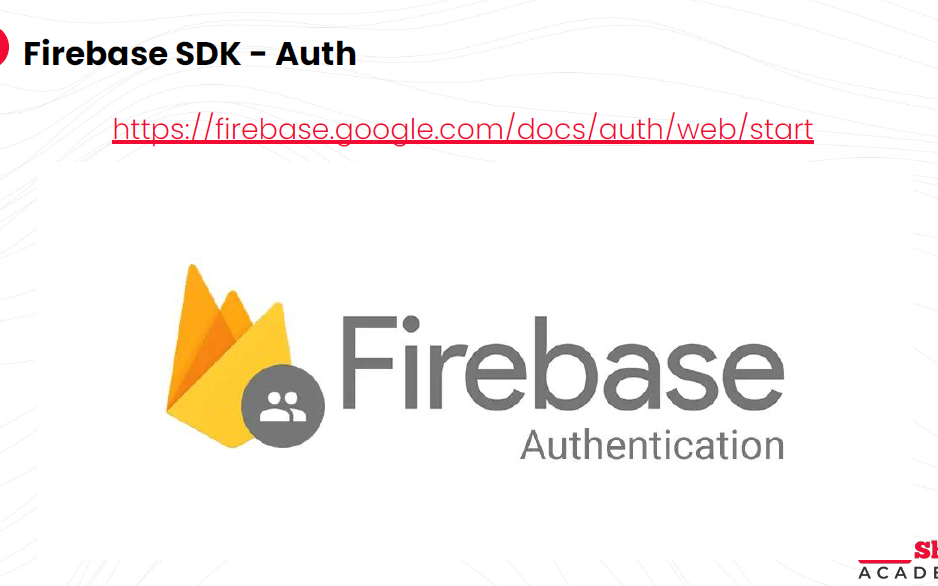


JWT – JSON web teken/ dla nas

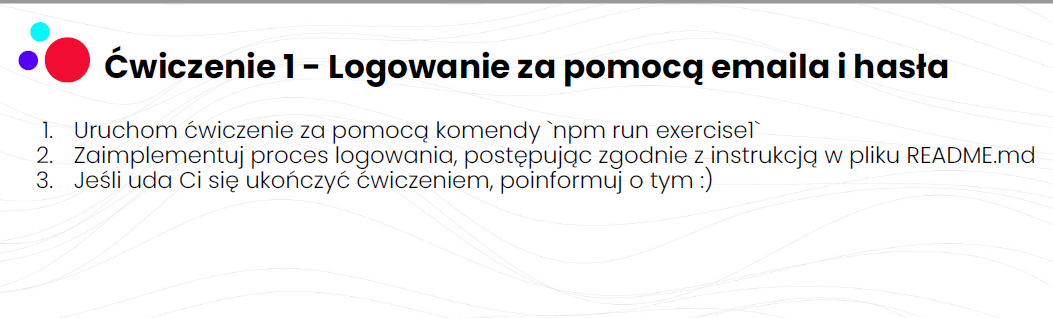
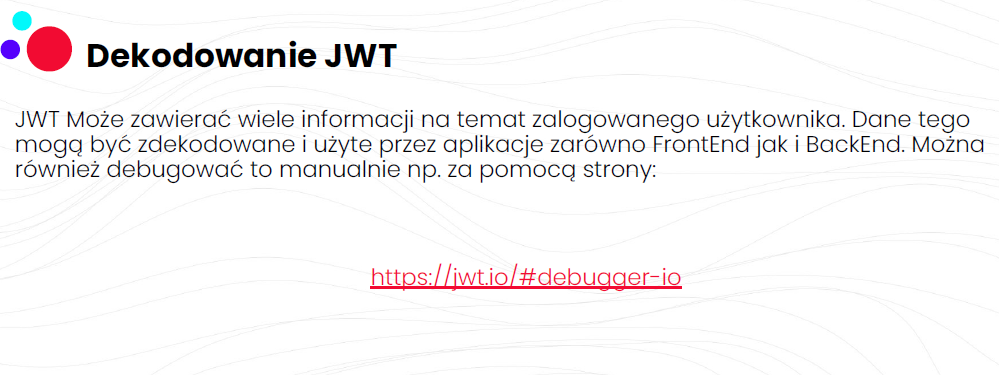
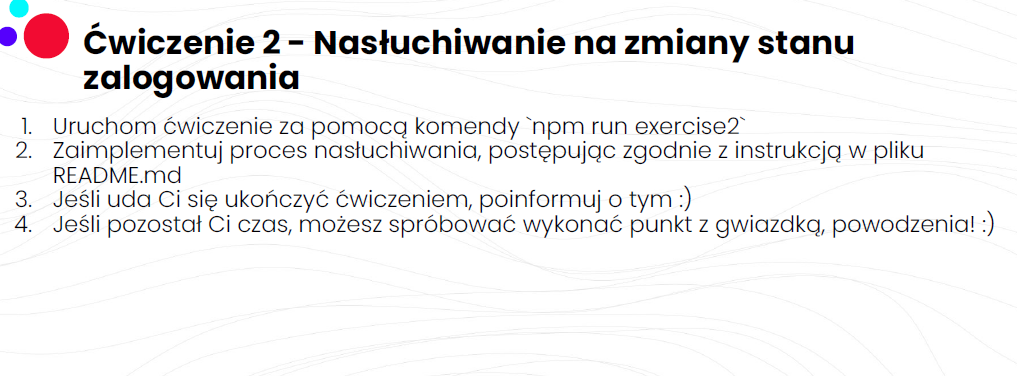
Cookie Based Auth- dla nas nie bardziej backend

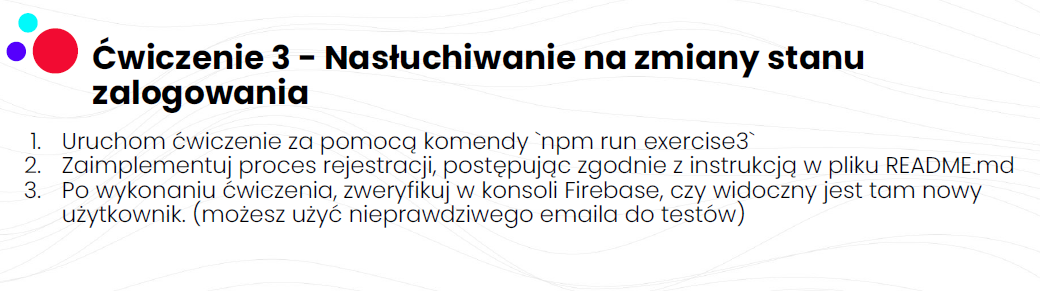


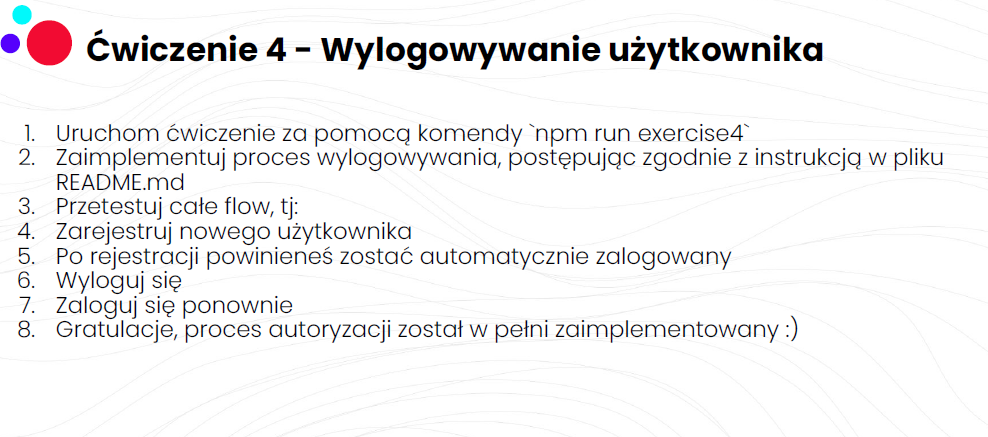


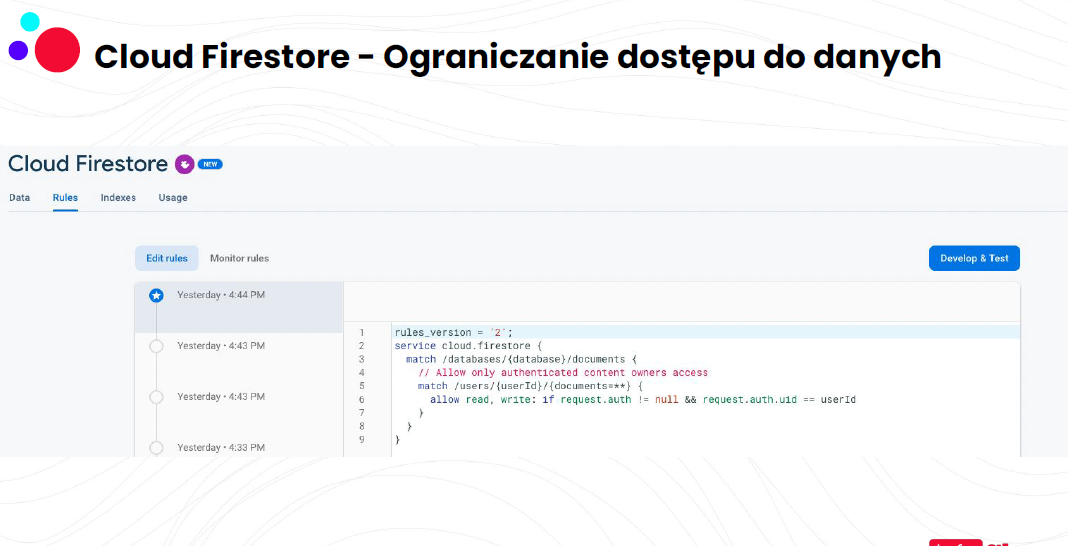


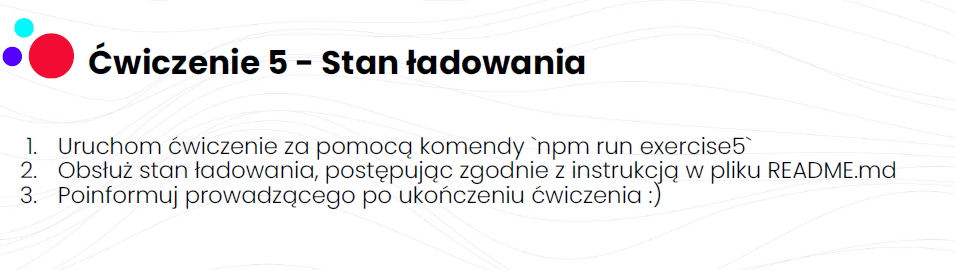


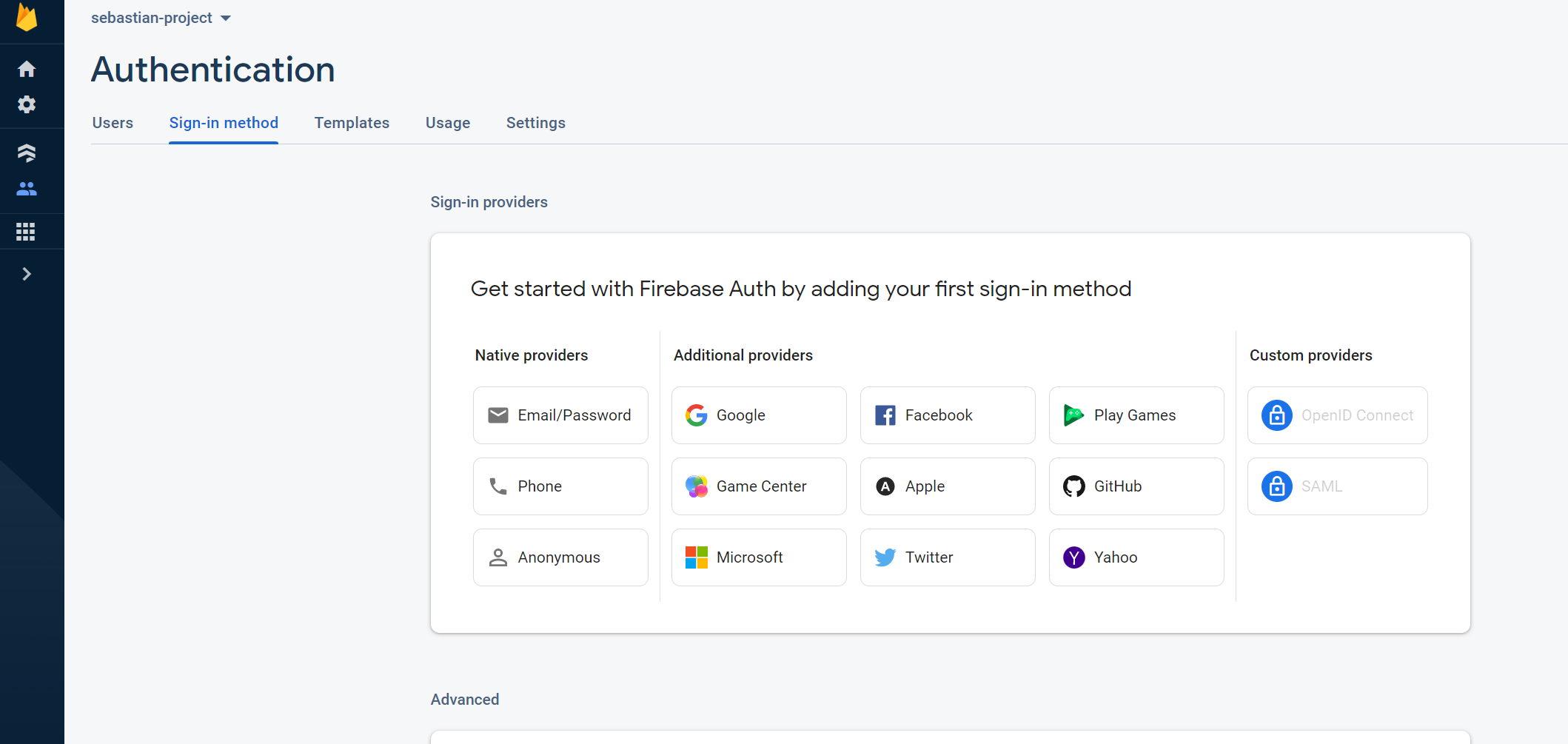
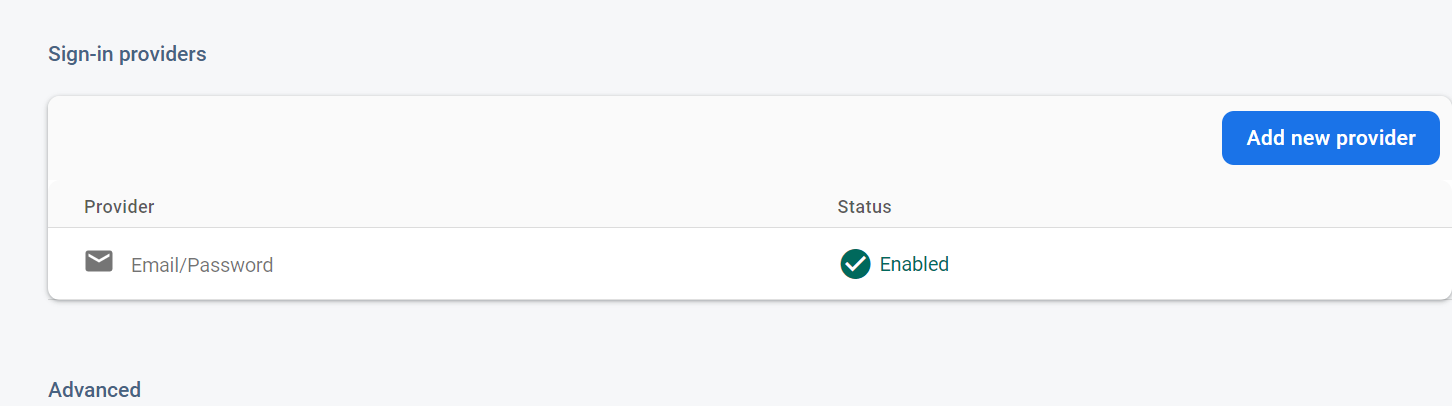


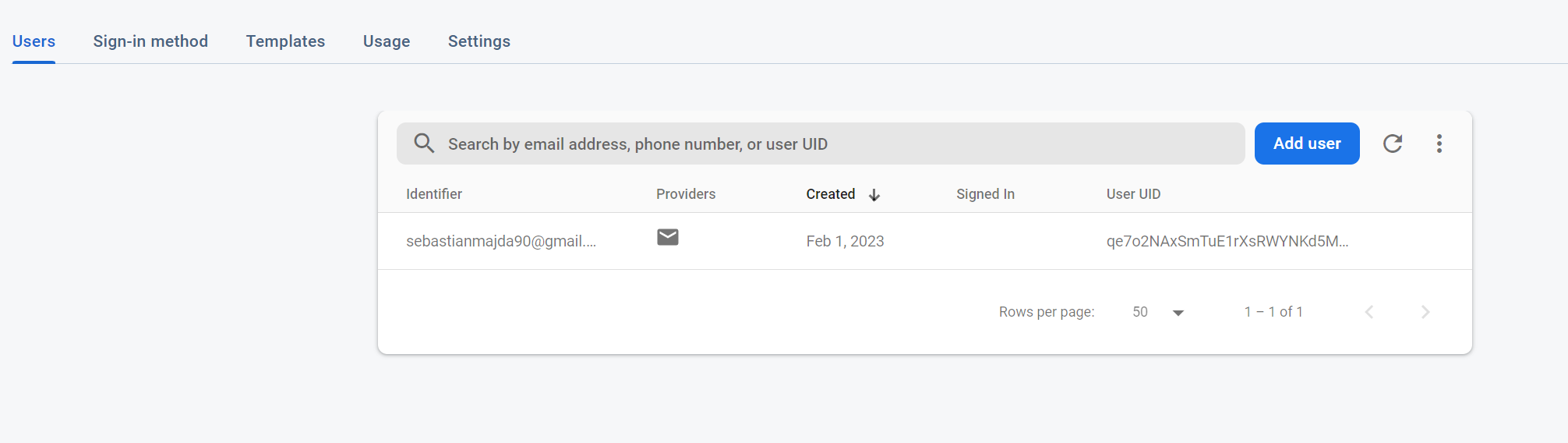






Zaczynamy…



Configuracja firebase.js

import { initializeApp } from "firebase/app";

import { getFirestore } from "firebase/firestore";

import { getAuth } from "firebase/auth";

const firebaseConfig = {

  apiKey: "AIzaSyBGIIA\_rWGRwDJ4Zxw0D8ApwkQu5knR3TU",

  authDomain: "sebastian-project-a24fd.firebaseapp.com",

  projectId: "sebastian-project-a24fd",

  storageBucket: "sebastian-project-a24fd.appspot.com",

  messagingSenderId: "375416779334",

  appId: "1:375416779334:web:5d618d283cd332819a1047",

};

const app = initializeApp(firebaseConfig);

export const db = getFirestore(app);

export const auth = getAuth(app);

zad1

import { Routes, Route } from "react-router-dom";

import { signInWithEmailAndPassword } from "firebase/auth";

import { GuestHomePage } from "../../guest/GuestHomePage";

import { RegisterForm } from "../../guest/RegisterForm";

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

// - Uruchom ćwiczenie za pomocą npm run exercise1

// - Stwórz nową funkcję obsługującą formularz logowania (prop: `onLogin` w komponencie `GuestHomePage`)

// - Przekaż dane logowania do `firebase` za pomocą funkcji: `signInWithEmailAndPassword(auth, email, password)`

// - Zaloguj w konsoli string "Logged" gdy Promise zwrócony przez funkcję rozwiąże się pomyślnie.

export default function App() {

  const handleLogin = async (data) => {

    await signInWithEmailAndPassword(auth, data.email, data.password);

    console.log("Logged");

  };

  return (

    <Routes>

      <Route path="/" element={<GuestHomePage />} /> podkładamy podstronę tylko na path / , żeby nie wyświetlało się zawsze tylko np. w danej sytuacji, tak jak poniżej rejestra ->

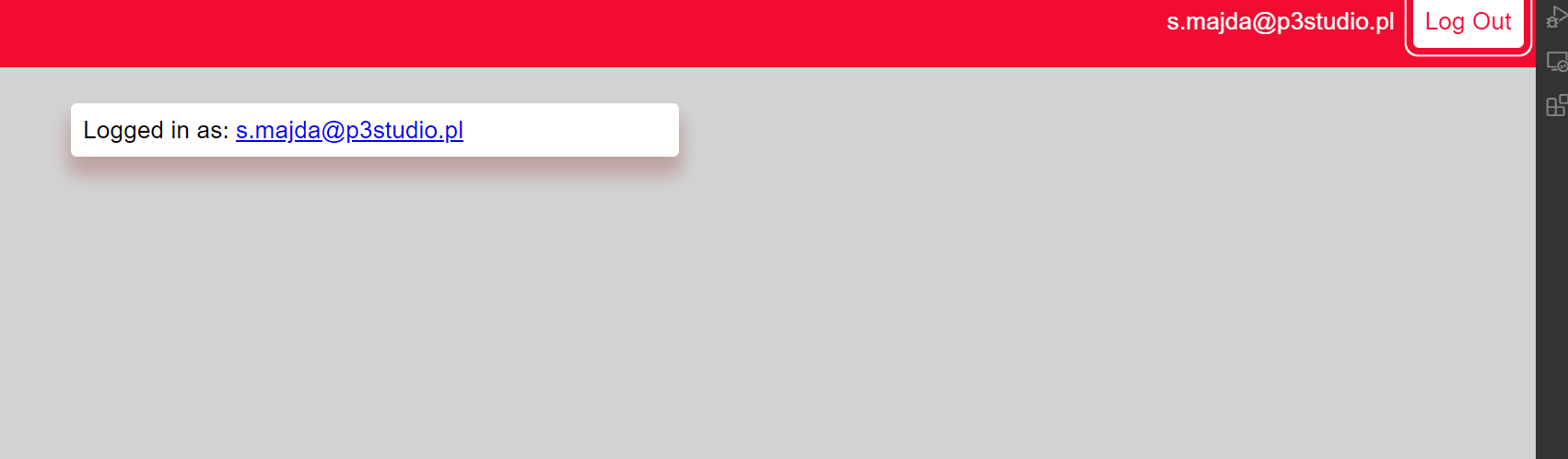
      <Route path="/register" element={<RegisterForm />} />

    </Routes>

  );

}

Żeby ładnie się wylogowało, i było bezpieczne najlepiej zrobić to tak:



import { Routes, Route } from "react-router-dom";

import { onAuthStateChanged, signInWithEmailAndPassword } from "firebase/auth";

import { AuthenticatedHomePage } from "../../authenticated/AuthenticatedHomePage";

import { GuestHomePage } from "../../guest/GuestHomePage";

import { RegisterForm } from "../../guest/RegisterForm";

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

import { useEffect, useState } from "react";

// - Dodaj state, w którym będziesz przechowywać dane o aktualnie zalogowanym użytkowniku

// - Stwórz nowy efekt, który będzie odpalany tylko raz w momencie zamontowania komponentu.

// - W efekcie dodaj listener za pomocą funkcji: `onAuthStateChanged(auth, (user) => {})`

// - Zapisuj obiekt user do stanu za każdym razem gdy funkcja zostanie wywołana

// - W przypadku gdy w state ustawiony jest user, wyrenderuj tylko komponent: `AuthenticatedHomePage`, oraz przekaż obiekt użytkownika jako prop: `currentUser`

// - Jeśli w state ustawiony jest `null`, wyrenderuj ścieżki do logowania / rejestracji

// \* Zaimplementuj proces `unsubscribe`. Jest to funkcja zwracana z wywołania `onAuthStateChanged`. Aby ją odsubskrybować, powinna zostać wywołana w cleanup function efektu.

export default function App() {

  const [currentUser, setCurrentUser] = useState(null);

  useEffect(() => {

    const unsubscribe = onAuthStateChanged(auth, (user) => {

      setCurrentUser(user); //user jako cały obiekt

    });

    return () => unsubscribe();

  }, []);

  const handleLogin = ({ email, password }) =>

    signInWithEmailAndPassword(auth, email, password).then((data) =>

      console.log(data.user)

    );

  if (currentUser) {

    //kiedy currentUSer nie jest null

    return <AuthenticatedHomePage currentUser={currentUser} />;

  }

  return (

    <Routes>

      <Route path="/" element={<GuestHomePage onLogin={handleLogin} />} />

      <Route path="/register" element={<RegisterForm />} />

    </Routes>

  );

}

Zad 3 REJESTRACJA

import { useEffect, useState } from "react";

import { Routes, Route } from "react-router-dom";

import {

  onAuthStateChanged,

  signInWithEmailAndPassword,

  createUserWithEmailAndPassword,

} from "firebase/auth";

import { AuthenticatedHomePage } from "../../authenticated/AuthenticatedHomePage";

import { GuestHomePage } from "../../guest/GuestHomePage";

import { RegisterForm } from "../../guest/RegisterForm";

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

// - Stwórz nową funkcję obsługującą formularz rejestracji (prop: `onSubmit` w komponencie `RegisterForm`)

// - Przekaż dane logowania do `firebase` za pomocą funkcji: `createUserWithEmailAndPassword(auth, email, password)`

// - Spróbuj dodać nowego użytkownika. Następnie przejdź do panelu Firebase w przeglądarce i zweryfikuj, czy użytkownik został dodany poprawnie

export default function App() {

  const [currentUser, setCurrentUser] = useState(null);

  useEffect(() => {

    const unsubscribe = onAuthStateChanged(auth, setCurrentUser);

    return unsubscribe;

  }, []);

  const handleLogin = ({ email, password }) =>

    createUserWithEmailAndPassword(auth, email, password).then(console.log);

    const handleRegister = ({ email, password }) =>

  createUserWithEmailAndPassword(auth, email, password);

  if (currentUser) {

    return <AuthenticatedHomePage currentUser={currentUser} />;

  }

  return (

    <Routes>

      <Route path="/" element={<GuestHomePage onLogin={handleLogin} />} />

      <Route path="/register" element={<RegisterForm onSubmit={handleRegister} />} />

    </Routes>

  );

}

ZAD 4 wylogowywanie użytkownika

import { useEffect, useState } from "react";

import { Routes, Route, useNavigate } from "react-router-dom";

import {

  signInWithEmailAndPassword,

  signOut,

  onAuthStateChanged,

  createUserWithEmailAndPassword,

} from "firebase/auth";

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

import { GuestHomePage } from "../../guest/GuestHomePage";

import { RegisterForm } from "../../guest/RegisterForm";

import { AuthenticatedHomePage } from "../../authenticated/AuthenticatedHomePage";

// - Stwórz nową funkcję obsługującą wylogowywanie (prop: `onLogOut` w komponencie `AuthenticatedHomePage`)

// - Wyloguj użytkownika za pomocą funkcji `signOut(auth)`

// - Jeśli operacja przebiegła poprawnie, powinieneś zostać przekierowany do strony logowania

export default function App() {

  const navigate = useNavigate();

  const [currentUser, setCurrentUser] = useState(null);

  useEffect(() => {

    const unsubscribe = onAuthStateChanged(auth, (user) => {

      if (user) {

        navigate("/");

      }

      setCurrentUser(user);//wazne !

    });

    return () => unsubscribe();

  }, [navigate]);

  const handleLogin = ({ email, password }) =>

    signInWithEmailAndPassword(auth, email, password);

  const handleRegister = ({ email, password }) =>

    createUserWithEmailAndPassword(auth, email, password);

  const handleLogOut = () => signOut(auth);

  if (currentUser) {

    return <AuthenticatedHomePage currentUser={currentUser} onLogOut={handleLogOut} />;

  }

  return (

    <Routes>

      <Route path="/" element={<GuestHomePage onLogin={handleLogin} />} />

      <Route

        path="/register"

        element={<RegisterForm onSubmit={handleRegister} />}

      />

    </Routes>

  );

}

PROWADZACY JUŻ WCZEŚNIEJ ZROBIŁ

