**BÁO CÁO 11: XÂY DỰNG APP VỚI AUTH0 VÀ REFRESH TOKEN**

1. **Ý tưởng**

* Ở phần đăng nhập, em sẽ tích hợp đăng nhập bằng facebook hoặc google nhờ Auth0. Ngoài ra, em sẽ sử dụng refresh token để xử lý khi token của JsonWebToken hết hạn

1. **Cách thực hiện**

* Em sẽ clone lại project sử dụng UmiJS và DvaJS và thêm Auth0 và refresh token trong project này

1. **Refresh token**

* Trong file ***sesstion.ts***

\*handleLogin(action: any, { call, put }: { call: any; put: any }) {

const user = action.user;

  yield axios.get(userData).then((response) => {

   for (const userInt of response.data) {

     if (user.username === userInt.username) {

       bcrypt

        .compare(user.password, userInt.password)

        .then((valid: any) => {

         if (valid) {

           const token = jwt.sign(

             {

             username: user.username,

             password: userInt.password,

             },

             'SECRET\_TOKEN',

             { expiresIn: '1h' },

         );

            const refreshToken = jwt.sign(

             {

               username: user.username,

                password: userInt.password,

              },

              'SECRET\_TOKEN',

              { expiresIn: '100d' },

            );

            window.localStorage.setItem('jwtToken', token);

            window.localStorage.setItem('refreshJwtToken', refreshToken);

          }      }

        });

        break;

      }

    }

  });

  yield put({

   type: 'login',

    user: user,

  });

  yield call(delay, 500);

  yield put({

   type: 'auth',

    token: window.localStorage.getItem('jwtToken'),

  });

},

* Như vậy em đã cấp phát token khi đăng nhập và nó có hiệu lực trong 1 giờ. Ngoài ra, em cấp phát thêm refresh token để cấp phát lại token khi nó hết hiệu lực.
* Trong file chứa component App:

function App(props: any) {

  const token: any = window.localStorage.getItem('jwtToken');

  let dec: any;

  if (token !== null) {

    jwt.verify(token, 'SECRET\_TOKEN', (err: any, decoded: any) => {

      console.log(err);

      dec = decoded;

      console.log(decoded);

    });

    if (dec === undefined) {

      const refreshToken: any = window.localStorage.getItem('refreshJwtToken');

      jwt.verify(refreshToken, 'SECRET\_TOKEN', (err: any, decoded: any) => {

        console.log(err);

        dec = decoded;

      });

      const newToken = jwt.sign(

        {

          username: dec.username,

          password: dec.password,

        },

        'SECRET\_TOKEN',

        { expiresIn: '1h' },

      );

      window.localStorage.setItem('jwtToken', newToken);

    }

  }

  const { isAuthenticated, isLoading, getAccessTokenSilently, user } =

    useAuth0();

  if (isLoading) {

    return <Spin></Spin>;

  }

  getAccessTokenSilently().then((token) => {

    window.localStorage.setItem('token', token);

  });

  console.log('is authenticated');

  console.log(isAuthenticated);

  return (

    <div className="App">

      {(token !== null ) || isAuthenticated === true ? (

        <Contact />

      ) : (

        <Redirect to="/login" />

      )}

    </div>

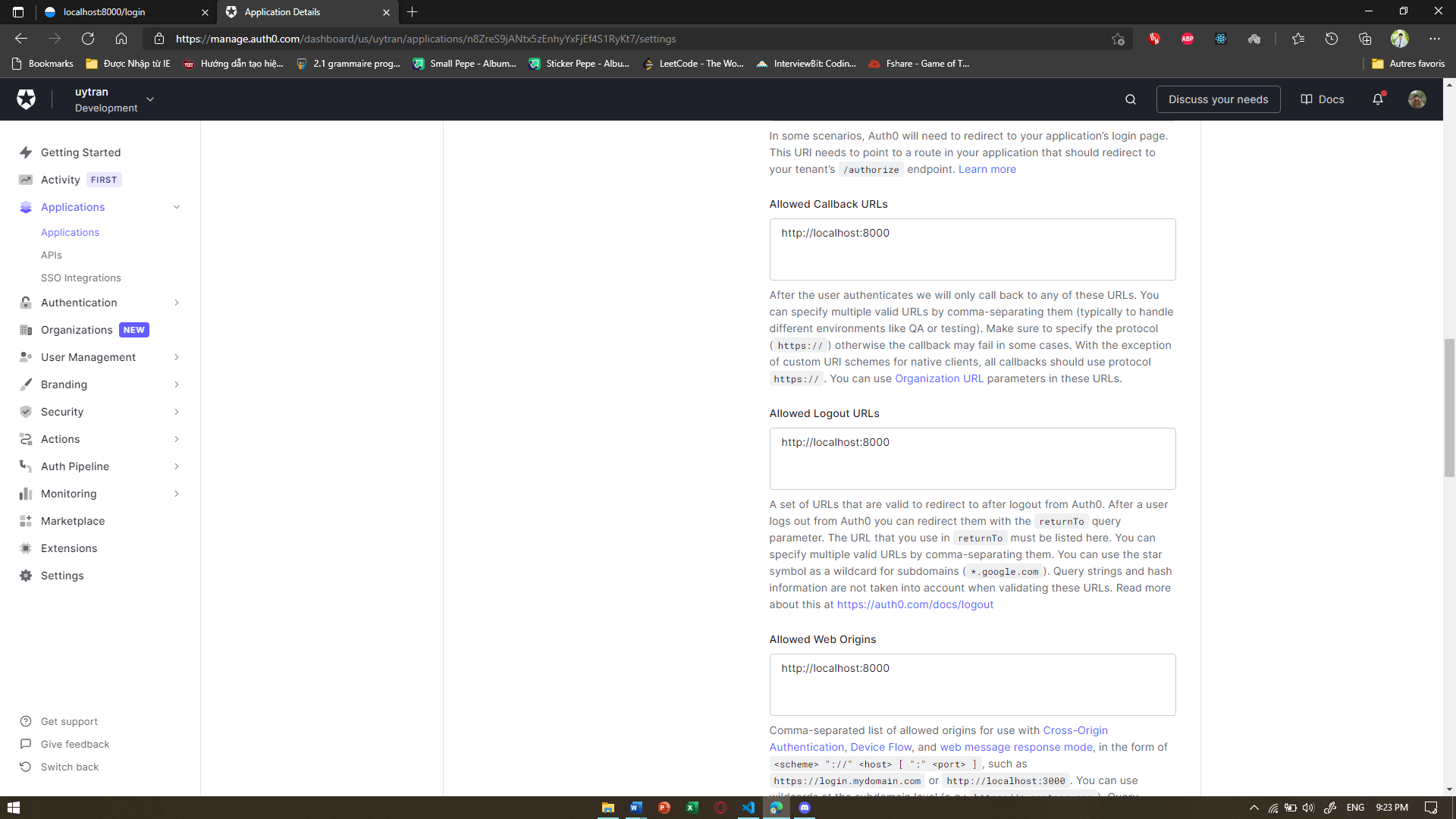
  );

}

* Em sẽ decode jwt và nếu như nó bị hết hạn thì sẽ sử dụng refresh token để tạo token mới.

1. **Auth0**

* Đầu tiên em tạo app trong Auth0 và thêm địa chỉ <http://localhost:8000> như sau:



* Để sử dụng được Auth0, em sẽ thêm folder wrapper như sau:



* Trong file ***Auth0.tsx***

import { Auth0Provider } from '@auth0/auth0-react';

export default (props: any) => {

  return (

    <Auth0Provider

      domain="uytran.us.auth0.com"

      clientId="n8ZreS9jANtx5zEnhyYxFjEf4S1RyKt7"

      redirectUri="http://localhost:8000"

      useRefreshTokens={true}

    >

      {props.children}

    </Auth0Provider>

  );

};

* Domain và clientId được lấy từ cài đặt app trong Auth0
* Sau đó em config wrapper như sau:

export default defineConfig({

  dva: {},

  nodeModulesTransform: {

    type: 'none',

  },

  routes: [

    {

      exact: true,

      path: '/',

      component: '@/pages/common/index',

      wrappers: ['@/wrappers/Auth0'],

    },

    {

      exact: true,

      path: '/login',

      component: '@/pages/session/FormLogin',

      wrappers: ['@/wrappers/Auth0'],

    },

    {

      exact: true,

      path: '/register',

      component: '@/pages/session/FormRegister',

      wrappers: ['@/wrappers/Auth0'],

    },

  ],

  fastRefresh: {},

});

* Em thêm button đăng nhập vào Auth0 trong app của mình:

const { loginWithRedirect} = useAuth0();

<button onClick={() => loginWithRedirect()}>Login with Auth0</button>

* Tương tự với button logout:

const { logout } = useAuth0();

  //when click logout button

  const onClickLogout = () => {

    window.localStorage.removeItem('jwtToken');

    window.localStorage.removeItem('refreshJwtToken');

    window.localStorage.removeItem('token');

    logout();

  };

<Button

type="primary"

  size="large"

  onClick={props.onClickLogout}

>

Logout

</Button>

* Khi login thành công bằng Auth0, em sẽ lưu token của Auth0 vào local storage

getAccessTokenSilently().then((token) => {

window.localStorage.setItem('token', token);

});

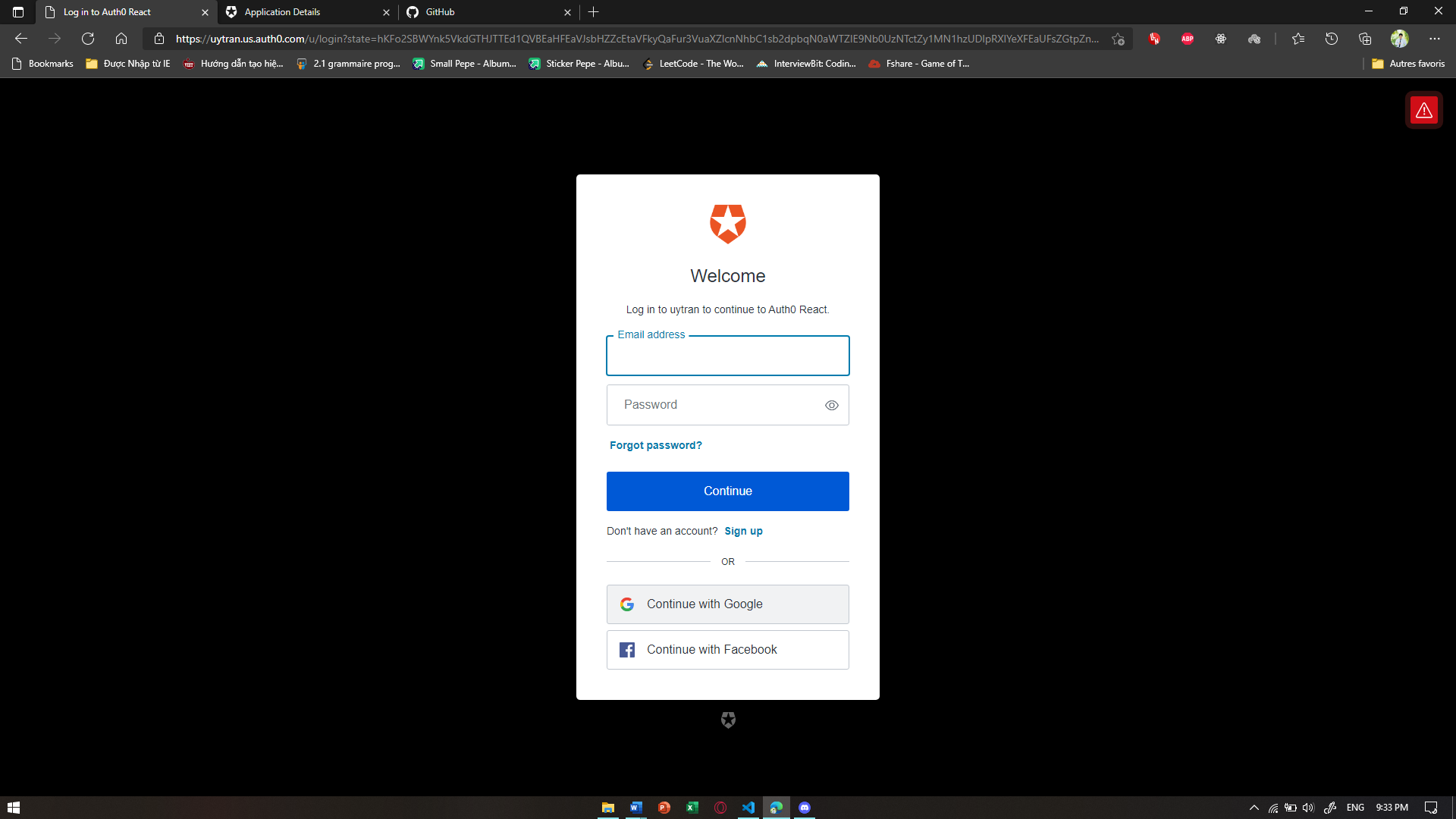
* Như vậy là em có thể sử dụng được Auth0 trong app của mình.

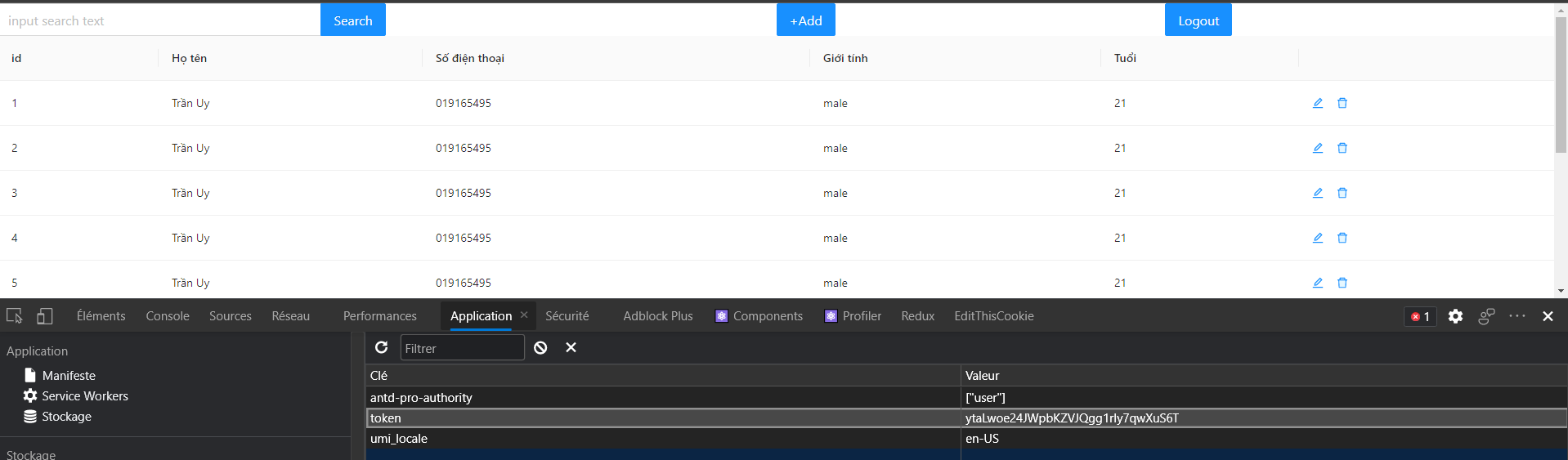
1. **Source**

* Tất cả source code nằm trong link github sau:

[internSCC/React/auth0-umijs at main · uytran36/internSCC (github.com)](https://github.com/uytran36/internSCC/tree/main/React/auth0-umijs)

1. **Demo**



****