import React, { useEffect } from "react";

import { View, Text, Button, StyleSheet, ScrollView } from "react-native";

import { NavigationContainer } from "@react-navigation/native";

import { createDrawerNavigator } from "@react-navigation/drawer";

import "react-native-gesture-handler";

import { DrawerContent } from "./Screens/DrawerContent";

import MainTabScreen from "./Screens/MainTabScreen";

import SettingsScreen from "./Screens/SettingsScreen";

import RootStackScreen from "./Screens/RootStackScreen";

//import { View } from 'react-native-animatable';

import { ActivityIndicator, Switch } from "react-native-paper";

import { AuthContext } from "./Components/Context";

import AsyncStorage from "@react-native-community/async-storage";

import { firebaseConfig } from "./Firebase";

import \* as firebase from "firebase";

if (!firebase.apps.length) firebase.initializeApp(firebaseConfig);

const Drawer = createDrawerNavigator();

const App = () => {

// const [isLoading,setIsLoading] =React.useState(true);

// const [userToken,setUserToken] =React.useState(null);

const initialLoginState = {

isLoading: true,

userName: null,

userToken: null,

};

const loginReducer = (prevState, action) => {

switch (action.type) {

case "RETRIEVE\_TOKEN":

return {

...prevState,

userToken: action.token,

isLoading: false,

};

case "LOGIN":

return {

...prevState,

userName: action.id,

userToken: action.token,

isLoading: false,

};

case "LOGOUT":

return {

...prevState,

userName: null,

userToken: null,

isLoading: false,

};

case "REGISTER":

return {

...prevState,

userName: action.id,

userToken: action.token,

isLoading: false,

};

}

};

const [loginState, dispatch] = React.useReducer(

loginReducer,

initialLoginState

);

useEffect(() => {

setTimeout(async () => {

//setIsLoading(false);

let userToken;

userToken = null;

//console.log('user token:', userToken);

try {

userToken = await AsyncStorage.getItem("userToken");

} catch (e) {

console.log(e);

}

dispatch({ type: "RETRIEVE\_TOKEN", token: userToken });

}, 1000);

}, []);

if (loginState.isLoading) {

return (

<View style={{ flex: 1, justifyContent: "center", alignItems: "center" }}>

<ActivityIndicator size="large" />

</View>

);

}

return (

// <AuthContext.Provider value={authContext}>

<NavigationContainer>

{loginState.userToken !== null ? (

<Drawer.Navigator

drawerContent={(props) => <DrawerContent {...props} />}

>

<Drawer.Screen name="HomeDrawer" component={MainTabScreen} />

<Drawer.Screen name="SettingsScreen" component={SettingsScreen} />

</Drawer.Navigator>

) : (

<RootStackScreen />

)}

</NavigationContainer>

// </AuthContext.Provider>

);

};

export default App;