```
// ================================================================================================ //
//                                          REACT NATIVE AUTH
// ================================================================================================ //

 "dependencies": {
    "@react-native-community/masked-view": "0.1.5",
    "@react-navigation/bottom-tabs": "^5.5.2",
    "@react-navigation/drawer": "^5.8.2",
    "@react-navigation/native": "^5.5.1",
    "@react-navigation/stack": "^5.5.1",
    "expo": "~36.0.0",
    "expo-linear-gradient": "~8.0.0",
    "react": "~16.9.0",
    "react-dom": "~16.9.0",
    "react-native": "https://github.com/expo/react-native/archive/sdk-36.0.0.tar.gz",
    "react-native-animatable": "^1.3.3",
    "react-native-gesture-handler": "~1.5.0",
    "react-native-paper": "^3.10.1",
    "react-native-reanimated": "~1.4.0",
    "react-native-safe-area-context": "0.6.0",
    "react-native-screens": "2.0.0-alpha.12",
    "react-native-web": "~0.11.7"
  },

// ================================================================================================ //
//                                                CONTEXT
// ================================================================================================ //

import React, { createContext } from "react";
export const AuthContext = createContext();

// ================================================================================================ //
//                                              SCREENS
// ================================================================================================ //

BOOKMARKS.JS
--------------------------------------------------------------------------------
import React from "react";
import { View, Text, Button, StyleSheet } from "react-native";

const BookmarkScreen = () => {
  return (
    <View style={styles.container}>
      <Text>Bookmark Screen</Text>
      <Button title="Click Here" onPress={() => alert("Button Clicked!")} />
    </View>
  );
};

export default BookmarkScreen;

DETAILSSCREEN.JS
--------------------------------------------------------------------------------
import React from "react";
import { View, Text, Button, StyleSheet, StatusBar } from "react-native";

const DetailScreen = ({ navigation }) => (
  <View style={{ flex: 1, justifyContent: "center", alignItems: "center" }}>
    <Text>Details screen</Text>
    <Button title="Click Here" onPress={() => alert("Button Clicked!")} />
  </View>
);

export default DetailScreen;
```

EXPLORESCREEN.JS

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

const ExploreScreen = () => {
  return (
    <View style={styles.container}>
      <Text>ExploreScreen</Text>
      <Button title="Click Here" onPress={() => alert("Button Clicked!")} />
    </View>
  );
};

export default ExploreScreen;
```

HOMESCREEN.JS

```
import React from "react";
import { View, Text, Button, StyleSheet, StatusBar } from "react-native";

const HomeScreen = ({ navigation }) => (
  <View style={{ flex: 1, justifyContent: "center", alignItems: "center" }}>
    <Text>Home screen</Text>
    <Button title="Click Here" onPress={() => alert("Button Clicked!")} />
  </View>
);

export default HomeScreen;
```

PROFILESCREEN.JS

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

const ProfileScreen = () => {
  return (
    <View style={styles.container}>
      <Text>Profile Screen</Text>
      <Button title="Click Here" onPress={() => alert("Button Clicked!")} />
    </View>
  );
};

export default ProfileScreen;
```

SETTINGSCREEN.JS

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

const SettingsScreen = () => {
  return (
    <View style={styles.container}>
      <Text>Settings Screen</Text>
      <Button title="Click Here" onPress={() => alert("Button Clicked!")} />
    </View>
  );
};

export default SettingsScreen;
```

SUPPORTSCREEN.JS

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

const SupportScreen = () => {
  return (
    <View style={styles.container}>
      <Text>Support Screen</Text>
      <Button title="Click Here" onPress={() => alert("Button Clicked!")} />
    </View>
  );
};

export default SupportScreen;
```

SIGNINGSCREEN.JS

```
import React, { useState, useContext } from "react";
import {View,Text, Alert, TouchableOpacity, TextInput, Dimensions, StyleSheet, StatusBar, Platform,
  Image} from "react-native";
import { LinearGradient } from "expo-linear-gradient";
import { FontAwesome, Feather } from "@expo/vector-icons";
import * as Animatable from "react-native-animatable";

import { AuthContext } from "../components/context";
import Users from "../models/users";

const SignInScreen = ({ navigation }) => {
  const [data, setData] = useState({
    username: "",
    password: "",
    check_TextInputChange: false,
    secureTextEntry: true,
    isValidUser: true,
    isValidPassword: true,
  });

  const { signIn } = useContext(AuthContext);

  const textInputChange = (val) => {
    if (val.trim().length >= 4) {
      setData({
        ...data,
        username: val,
        check_TextInputChange: true,
        isValidUser: true,
      });
    } else {
      setData({
        ...data,
        username: val,
        check_TextInputChange: false,
        isValidUser: false,
      });
    }
  };

  const handlePasswordChange = (val) => {
    if (val.trim().length >= 8) {
      setData({
        ...data,
        password: val,
        isValidPassword: true,
      });
    } else {
```

```
    setData({
      ...data,
      password: val,
      isValidPassword: false,
    });
  }
};
const updateSecureTextEntry = () => {
  setData({
    ...data,
    secureTextEntry: !data.secureTextEntry,
  });
};

const loginHandler = (userName, password) => {
  const foundUser = Users.filter((item) => {
    return userName === item.username && password === item.password;
  });
  if (data.username.length === 0 || data.password.length === 0) {
    Alert.alert("Wrong input!", "user o pass cant be empty", [
      { text: "Okay" },
    ]);
    return;
  }
  if (foundUser.length === 0) {
    Alert.alert("Wrong input!", "user o pass incorrect", [{ text: "Okay" }]);
    return;
  }
  signIn(foundUser);
};

const handleValidUser = (val) => {
  if (val.trim().length >= 4) {
    setData({
      ...data,
      isValidUser: true,
    });
  } else {
    setData({
      ...data,
      isValidUser: false,
    });
  }
};

return (
  <View style={styles.container}>
    {/* <StatusBar backgroundColor="@009387" barStyle="light-content" /> */}
    <View style={styles.header}>
      <Text style={styles.text_header}>Welcome</Text>
    </View>
    <Animatable.View animation="fadeInUpBig" style={styles.footer}>
      <Text style={styles.text_footer}>Username</Text>

      <View style={styles.action}>
        <FontAwesome name="user-o" color="#05375a" size={20} />
        <TextInput
          placeholder="Your username"
          style={styles.textInput}
          autoCapitalize="none"
          onChangeText={(val) => textInputChange(val)}
          onEndEditing={(e) => handleValidUser(e.nativeEvent.text)}
        />
        {data.check_TextInputChange ? (
          <Animatable.View animation="bounceIn">
            <Feather name="check-circle" color="green" size={20} />
          </Animatable.View>
        ) : null}
```

```
        </View>
        {data.isValidUser ? null : (
          <Animatable.View animation="fadeInLeft" duration={1400}>
            <Text style={styles.errorMsg}>Username must be 8 char long</Text>
          </Animatable.View>
        )}

        <Text style={[styles.text_footer, { marginTop: 20 }]}>Password</Text>
        <View style={styles.action}>
          <Feather name="lock" color="#05375a" size={20} />
          <TextInput
            placeholder="Your Password"
            secureTextEntry={data.secureTextEntry}
            style={styles.textInput}
            autoCapitalize="none"
            onChangeText={(val) => handlePasswordChange(val)}
          />
          <TouchableOpacity onPress={() => updateSecureTextEntry()}>
            {data.secureTextEntry ? (
              <Feather name="eye-off" color="grey" size={20} />
            ) : (
              <Feather name="eye" color="grey" size={20} />
            )}
          </TouchableOpacity>
        </View>
        {data.isValidPassword ? null : (
          <Animatable.View animation="fadeInLeft" duration={1400}>
            <Text style={styles.errorMsg}>Pass must be 4 char long</Text>
          </Animatable.View>
        )}

        <View style={styles.button}>
          <TouchableOpacity
            style={styles.signIn}
            onPress={() => loginHandler(data.username, data.password)}
          >
            <LinearGradient
              colors={["#08d4c4", "#01ab9d"]}
              style={styles.signIn}
            >
              <Text style={{ color: "#fff", fontWeight: "bold" }}>Sign In</Text>
            </LinearGradient>
          </TouchableOpacity>

          <TouchableOpacity
            style={{
              borderColor: "#009387",
              borderWidth: 1,
              marginTop: 15,
              width: "100%",
              alignItems: "center",
              justifyContent: "center",
              height: 50,
              borderRadius: 10,
            }}
            onPress={() => navigation.navigate("SignUpScreen")}
          >
            <Text style={{ fontWeight: "bold" }}>Sign Up!</Text>
          </TouchableOpacity>
        </View>
      </Animatable.View>
    </View>
  );
};

export default SignInScreen;
```

SIGNIUPSCREEN.JS

```
import React, { useState } from "react";
import {View, Text, TouchableOpacity, TextInput, Dimensions, StyleSheet, StatusBar, Platform, Image}
from "react-native";
import { LinearGradient } from "expo-linear-gradient";
import { FontAwesome, Feather } from "@expo/vector-icons";
import * as Animatable from "react-native-animatable";

const SignUpScreen = ({ navigation }) => {
  const [data, setData] = useState({
    email: "",
    password: "",
    confirm_password: "",
    check_TextInputChange: false,
    secureTextEntry: true,
    confirm_secureTextEntry: true,
  });

  const textInputChange = (val) => {
    if (val.length !== 0) {
      setData({
        ...data,
        email: val,
        check_TextInputChange: true,
      });
    } else {
      setData({
        ...data,
        email: val,
        check_TextInputChange: false,
      });
    }
  };

  const handlePasswordChange = (val) => {
    setData({
      ...data,
      password: val,
    });
  };

  const handleConfirmPasswordChange = (val) => {
    setData({
      ...data,
      confirm_password: val,
    });
  };

  const updateSecureTextEntry = () => {
    setData({
      ...data,
      secureTextEntry: !data.secureTextEntry,
    });
  };

  const updateConfirmSecureTextEntry = () => {
    setData({
      ...data,
      confirm_secureTextEntry: !data.confirm_secureTextEntry,
    });
  };
```

```
return (
  <View style={styles.container}>
    <StatusBar backgroundColor="@009387" barStyle="light-content" />
    <View style={styles.header}>
      <Text style={styles.text_header}>Register now!</Text>
    </View>
    <Animatable.View animation="fadeInUpBig" style={styles.footer}>
      <Text style={styles.text_footer}>Email</Text>

      <View style={styles.action}>
        <FontAwesome name="user-o" color="#05375a" size={20} />
        <TextInput
          placeholder="Your Email"
          style={styles.textInput}
          autoCapitalize="none"
          onChangeText={(val) => textInputChange(val)}
        />
        {data.check_TextInputChange ? (
          <Animatable.View animation="bounceIn">
            <Feather name="check-circle" color="green" size={20} />
          </Animatable.View>
        ) : null}
      </View>

      <Text style={[styles.text_footer, { marginTop: 20 }]}>Password</Text>
      <View style={styles.action}>
        <Feather name="lock" color="#05375a" size={20} />
        <TextInput
          placeholder="Your Password"
          secureTextEntry={data.confirm_secureTextEntry}
          style={styles.textInput}
          autoCapitalize="none"
          onChangeText={(val) => handlePasswordChange(val)}
        />
        <TouchableOpacity onPress={() => updateSecureTextEntry()}>
          {data.secureTextEntry ? (
            <Feather name="eye-off" color="grey" size={20} />
          ) : (
            <Feather name="eye" color="grey" size={20} />
          )}
        </TouchableOpacity>
      </View>

      <Text style={[styles.text_footer, { marginTop: 20 }]}>
        Confirm Password
      </Text>
      <View style={styles.action}>
        <Feather name="lock" color="#05375a" size={20} />
        <TextInput
          placeholder="Confirm Password"
          secureTextEntry={data.secureTextEntry}
          style={styles.textInput}
          autoCapitalize="none"
          onChangeText={(val) => handleConfirmPasswordChange(val)}
        />
        <TouchableOpacity onPress={() => updateConfirmSecureTextEntry()}>
          {data.secureTextEntry ? (
            <Feather name="eye-off" color="grey" size={20} />
          ) : (
            <Feather name="eye" color="grey" size={20} />
          )}
        </TouchableOpacity>
      </View>

      <View style={styles.button}>
        <LinearGradient colors={["#08d4c4", "#01ab9d"]} style={styles.signIn}>
          <Text style={{ color: "#fff", fontWeight: "bold" }}>Sign Up</Text>
        </LinearGradient>
```

```
          <TouchableOpacity
            style={{
              borderColor: "#009387",
              borderWidth: 1,
              marginTop: 15,
              width: "100%",
              alignItems: "center",
              justifyContent: "center",
              height: 50,
              borderRadius: 10,
            }}
            onPress={() => navigation.goBack()}
          >
            <Text style={{ fontWeight: "bold" }}>Sign In!</Text>
          </TouchableOpacity>
        </View>
      </Animatable.View>
    </View>
  );
};

export default SignUpScreen;

// ================================================================================================ //
//                                       ROOT STACK NAVIGATOR
// ================================================================================================ //

import React from "react";

import { createStackNavigator } from "@react-navigation/stack";

import SplashScreen from "./SplashScreen";
import SignInScreen from "./SignInScreen";
import SignUpScreen from "./SignUpScreen";

const RootStack = createStackNavigator();

const RootStackScreen = ({ navigation }) => (
  <RootStack.Navigator headerMode="none">
    <RootStack.Screen name="SplashScreen" component={SplashScreen} />
    <RootStack.Screen name="SignInScreen" component={SignInScreen} />
    <RootStack.Screen name="SignUpScreen" component={SignUpScreen} />
  </RootStack.Navigator>
);

export default RootStackScreen;

// ================================================================================================ //
//                                         MAIN TAB SCREEN
//                HOMESTACKNAVIGATOR – DETAILSSTACKNAVIGATOR - TABNAVIGATOR
// ================================================================================================ //

import React from "react";
import { createStackNavigator } from "@react-navigation/stack";
import { createBottomTabNavigator } from "@react-navigation/bottom-tabs";
import { Ionicons } from "@expo/vector-icons";

import DetailScreen from "./DetailsScreen";
import HomeScreen from "./HomeScreen";
import Profile from "./ProfileScreen";
import Explore from "./ExploreScreen";
```

HOME STACK NAVIGATOR

```
const Stack = createStackNavigator();
const HomeStackScreen = ({ navigation }) => (
  <Stack.Navigator
    screenOptions={{
      headerStyle: { backgroundColor: "#009387" },
      headerTintColor: "white",
      headerTitleStyle: { fontWeight: "bold" },
      headerRight: () => (
        <Ionicons
          name="ios-menu"
          size={25}
          color="white"
          onPress={() => navigation.toggleDrawer()}
        />
      ),
    }}
  >
    <Stack.Screen
      name="Home"
      component={HomeScreen}
      options={{ title: "Home Screen" }}
    />
  </Stack.Navigator>
);
```

DETAILS STACK NAVIGATOR

```
const DetailsStack = createStackNavigator();
const DetailsStackScreen = ({ navigation }) => (
  <DetailsStack.Navigator
    screenOptions={{
      headerStyle: { backgroundColor: "#009387" },
      headerTintColor: "white",
      headerTitleStyle: { fontWeight: "bold" },
      headerRight: () => (
        <Ionicons
          name="ios-menu"
          size={25}
          color="#009387"
          onPress={() => navigation.toggleDrawer()}
        />
      ),
    }}
  >
    <DetailsStack.Screen name="Details" component={DetailScreen} />
  </DetailsStack.Navigator>
);
```

TAB NAVIGATOR

```
const Tabs = createBottomTabNavigator();
const TabsScreen = ({ color }) => (
  <Tabs.Navigator
    shifting={true}
    initialRouteName="Home"
    tabBarOptions={{
      activeTintColor: "white",
      style: {
        backgroundColor: "#009387",
      },
    }}
  >
```

```
    <Tabs.Screen
      name="Home"
      component={HomeStackScreen}
      options={{
        tabBarLabel: "Home",
        tabBarIcon: ({ color }) => (
          <Ionicons name="ios-home" color={color} size={26} />
        ),
      }}
    />

    <Tabs.Screen
      name="Details"
      component={DetailsStackScreen}
      options={{
        tabBarLabel: "Details",
        tabBarIcon: (props) => (
          <Ionicons
            name="ios-notifications"
            size={props.size}
            color={props.color}
          />
        ),
      }}
    />

    <Tabs.Screen
      name="Profile"
      component={Profile}
      options={{
        tabBarLabel: "Profile",
        tabBarIcon: (props) => (
          <Ionicons name="ios-person" size={props.size} color={props.color} />
        ),
      }}
    />

    <Tabs.Screen
      name="Explore"
      component={Explore}
      options={{
        tabBarLabel: "Explore",
        tabBarIcon: (props) => (
          <Ionicons name="ios-settings" size={props.size} color={props.color} />
        ),
      }}
    />
  </Tabs.Navigator>
);

export default TabsScreen;

// ================================================================================== //
//                                    DRAWER NAVIGATOR
// ================================================================================== //

import React, { useState, useContext } from "react";
import { View, StyleSheet } from "react-native";
import {Avatar, Title, Caption, Paragraph, Drawer, Text, TouchableRipple, Switch} from "react-
native-paper";
import { DrawerContentScrollView, DrawerItem } from "@react-navigation/drawer";
import { Ionicons } from "@expo/vector-icons";

import { AuthContext } from "../components/context";

export function DrawerContent(props) {
  const [isDarkTheme, setIsDarkTheme] = useState(false);
  const { signOut } = useContext(AuthContext);
```

```
const toggleTheme = () => {
  setIsDarkTheme(!isDarkTheme);
  console.log(isDarkTheme);
};

return (
  <View style={{ flex: 1 }}>
    <DrawerContentScrollView {...props}>
      <View style={styles.drawerContent}>
        {/* ----------------------- USER INFO START------------------ */}
        <View style={styles.userInfoSection}>
          <View style={{ flexDirection: "row", marginTop: 15 }}>
            <Avatar.Image
              source={{
                uri: "https://randomuser.me/api/portraits/men/46.jpg",
              }}
              size={50}
            />
            <View style={{ marginLeft: 25, flexDirection: "column" }}>
              <Title style={styles.title}>John Frusciante</Title>
              <Caption style={styles.caption}>@rhcp</Caption>
            </View>
          </View>
          <View style={styles.row}>
            <View style={styles.section}>
              <Paragraph style={[styles.paragraph, styles.caption]}>
                RHCP
              </Paragraph>
              <Caption style={styles.caption}>Band</Caption>
            </View>
            <View style={styles.section}>
              <Paragraph style={[styles.paragraph, styles.caption]}>
                Guitar
              </Paragraph>
              <Caption style={styles.caption}>Instrument</Caption>
            </View>
          </View>
        </View>
        {/* ----------------------- USER INFO END ------------------ */}

        <Drawer.Section style={styles.drawerSection}>
          <DrawerItem
            icon={({ color, size }) => (
              <Ionicons name="ios-home" color={color} size={size} />
            )}
            label="Home"
            onPress={() => {
              props.navigation.navigate("Home");
            }}
          />
          <DrawerItem
            icon={({ color, size }) => (
              <Ionicons name="ios-person" color={color} size={size} />
            )}
            label="Profile"
            onPress={() => {
              props.navigation.navigate("Profile");
            }}
          />
          <DrawerItem
            icon={({ color, size }) => (
              <Ionicons name="ios-bookmarks" color={color} size={size} />
            )}
            label="Bookmarks"
            onPress={() => {
              props.navigation.navigate("Bookmarks");
            }}
          />
```

```
            <DrawerItem
              icon={({ color, size }) => (
                <Ionicons name="ios-settings" color={color} size={size} />
              )}
              label="Settings"
              onPress={() => {
                props.navigation.navigate("Settings");
              }}
            />
            <DrawerItem
              icon={({ color, size }) => (
                <Ionicons name="md-help-buoy" color={color} size={size} />
              )}
              label="Support"
              onPress={() => {
                props.navigation.navigate("Support");
              }}
            />
          </Drawer.Section>
          <Drawer.Section title="Preferences">
            <TouchableRipple
              onPress={() => {
                toggleTheme();
              }}
            >
              <View style={styles.preference}>
                <Text>Dark Theme</Text>
                <View pointerEvents="none">
                  <Switch value={isDarkTheme} />
                </View>
              </View>
            </TouchableRipple>
          </Drawer.Section>
        </View>
      </DrawerContentScrollView>
      <Drawer.Section style={styles.bottomDrawerSection}>
        <DrawerItem
          icon={({ color, size }) => (
            <Ionicons name="ios-log-out" color={color} size={size} />
          )}
          label="Sign Out"
          onPress={() => signOut()}
        />
      </Drawer.Section>
    </View>
  );
}


// ================================================================================ //
//                                      APP.JS
// ================================================================================ //

import React, { useEffect, useReducer, useMemo } from "react";
import { NavigationContainer } from "@react-navigation/native";

import { AuthContext } from "./components/context";

import { createDrawerNavigator } from "@react-navigation/drawer";
import MainTabScreen from "./screens/MainTabScreen";
import SupportScreen from "./screens/SupportScreen";
import SettingsScreen from "./screens/SettingsScreen";
import BookmarkScreen from "./screens/BookmarkScreen";
import { DrawerContent } from "./screens/DrawerContent";

import { AsyncStorage } from "react-native";
import RootStackScreen from "./screens/RooStackScreen";
import { View, ActivityIndicator } from "react-native";
```

APP DRAWER NAVIGATOR

```
================================================================================
```

```
const AppDrawer = createDrawerNavigator();
const AppDrawerScreen = () => (
  <AppDrawer.Navigator drawerContent={(props) => <DrawerContent {...props} />}>
    <AppDrawer.Screen
      component={MainTabScreen}
      name="Drawer"
      options={{ title: "Home" }}
    />
    <AppDrawer.Screen name="Support" component={SupportScreen} />
    <AppDrawer.Screen name="Settings" component={SettingsScreen} />
    <AppDrawer.Screen name="Bookmarks" component={BookmarkScreen} />
  </AppDrawer.Navigator>
);
```

APP COMPONENT

```
================================================================================
```

```
const App = () => {
  const initialState = {
    isLoading: true,
    userName: null,
    userToken: null,
  };


  const loginReducer = (state, action) => {
    switch (action.type) {
      case "RETRIEVE_TOKEN":
        return {
          ...state,
          userToken: action.token,
          isLoading: false,
        };
      case "LOGIN":
        return {
          ...state,
          userName: action.id,
          userToken: action.token,
          isLoading: false,
        };
      case "LOGOUT":
        return {
          ...state,
          userName: null,
          userToken: null,
          isLoading: false,
        };
      case "REGISTER":
        return {
          ...state,
          userName: action.id,
          userToken: action.token,
          isLoading: false,
        };
      default:
        return state;
    }
  };
```

```
const [state, dispatch] = useReducer(loginReducer, initialState);

  const authContext = useMemo(
    () => ({
      signIn: async (foundUser) => {
        const userToken = String(foundUser[0].userToken);
        const userName = String(foundUser[0].username);
        // if (userName === "user" && password === "pass") {
        try {
          userToken = "abababab";
          await AsyncStorage.setItem("userToken", userToken);
        } catch (error) {
          console.log(error);
        }

        dispatch({
          type: "LOGIN",
          id: userName,
          token: userToken,
        });
      },
      signOut: async () => {
        try {
          await AsyncStorage.removeItem("userToken");
        } catch (error) {
          console.log(error);
        }
        dispatch({
          type: "LOGOUT",
        });
      },
      signUp: () => {},
    }),
    []
  );

  useEffect(() => {
    setTimeout(async () => {
      let userToken;
      userToken = null;
      try {
        userToken = await AsyncStorage.getItem("userToken");
      } catch (error) {
        console.log(error);
      }

      dispatch({ type: "RETRIEVE_TOKEN", token: userToken });
    }, 1000);
  }, []);

  if (state.isLoading) {
    return (
      <View style={{ flex: 1, justifyContent: "center", alignItems: "center" }}>
        <ActivityIndicator size="large" />
      </View>
    );
  }

  return (
    <AuthContext.Provider value={authContext}>
      <NavigationContainer>
        {state.userToken !== null ? <AppDrawerScreen /> : <RootStackScreen />}
      </NavigationContainer>
    </AuthContext.Provider>
  );
};

export default App;
```