

**COMSATS UNIVERSITY ISLAMABAD,**

**ATTOCK CAMPUS**

**PREPARED BY:**

Roshan Ali Khattak (SP22-BSE-014)

**SUBJECT:**

Mobile Application Development

**SUBMITTED TO:**

MR Muhammad Kamran

**DATE:**  24thOctober,2024

**ASSIGNMENT #04**

**CODE:**

import React, { useEffect, useState } from 'react';

import { View, Text, TextInput, Button, StyleSheet, Alert } from 'react-native';

import AsyncStorage from '@react-native-async-storage/async-storage';

import { Provider, useDispatch, useSelector } from 'react-redux';

import { configureStore, createSlice } from '@reduxjs/toolkit';

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

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

import \* as Location from 'expo-location';

*// Redux Setup*

const userSlice = createSlice({

  name: 'user',

  initialState: {

    userData: null,

    location: null,

  },

  reducers: {

    setUserData: (state, action) => {

      state.userData = action.payload;

    },

    setLocation: (state, action) => {

      state.location = action.payload;

    },

  },

});

const { setUserData, setLocation } = userSlice.actions;

const store = configureStore({ reducer: { user: userSlice.reducer } });

*// Screens*

const SignupScreen = ({ navigation }) => {

  const [email, setEmail] = useState('');

  const [password, setPassword] = useState('');

  const [username, setUsername] = useState('');

  const handleSignup = async () => {

    if (!email || !password || !username) {

      Alert.alert('Error', 'Please fill all fields');

      return;

    }

*// Mock signup (replace with Firebase or API integration)*

    try {

      const user = { email, username };

      await AsyncStorage.setItem('user', JSON.stringify(user));

      Alert.alert('Signup Successful', 'You can now login.');

      navigation.navigate('Login');

    } catch (error) {

      Alert.alert('Error', error.message);

    }

  };

  return (

    <View style={styles.container}>

      <Text style={styles.title}>Signup</Text>

      <TextInput placeholder="Username" value={username} onChangeText={setUsername} style={styles.input} />

      <TextInput placeholder="Email" value={email} onChangeText={setEmail} style={styles.input} keyboardType="email-address" />

      <TextInput placeholder="Password" value={password} onChangeText={setPassword} style={styles.input} secureTextEntry />

      <Button title="Signup" onPress={handleSignup} />

    </View>

  );

};

const LoginScreen = ({ navigation }) => {

  const [email, setEmail] = useState('');

  const [password, setPassword] = useState('');

  const dispatch = useDispatch();

  const handleLogin = async () => {

    try {

      const user = JSON.parse(await AsyncStorage.getItem('user'));

      if (user && user.email === email) {

        dispatch(setUserData(user));

*// Fetch location (mock or real using expo-location)*

        const { status } = await Location.requestForegroundPermissionsAsync();

        if (status !== 'granted') {

          Alert.alert('Permission Denied', 'Location permission is required.');

          return;

        }

        const location = await Location.getCurrentPositionAsync({});

        const locationData = { lat: location.coords.latitude, lng: location.coords.longitude };

        await AsyncStorage.setItem('location', JSON.stringify(locationData));

        dispatch(setLocation(locationData));

        navigation.navigate('Profile');

      } else {

        Alert.alert('Error', 'Invalid email or password');

      }

    } catch (error) {

      Alert.alert('Error', error.message);

    }

  };

  return (

    <View style={styles.container}>

      <Text style={styles.title}>Login</Text>

      <TextInput placeholder="Email" value={email} onChangeText={setEmail} style={styles.input} keyboardType="email-address" />

      <TextInput placeholder="Password" value={password} onChangeText={setPassword} style={styles.input} secureTextEntry />

      <Button title="Login" onPress={handleLogin} />

    </View>

  );

};

const ProfileScreen = () => {

  const { userData, location } = useSelector((state) => state.user);

  return (

    <View style={styles.container}>

      <Text style={styles.title}>Profile</Text>

      <Text>Username: {userData?.username}</Text>

      <Text>Email: {userData?.email}</Text>

      <Text>Location: {location ? `${location.lat}, ${location.lng}` : 'Fetching...'}</Text>

    </View>

  );

};

*// App Navigation*

const Stack = createStackNavigator();

const App = () => {

  return (

    <Provider store={store}>

      <NavigationContainer>

        <Stack.Navigator initialRouteName="Signup">

          <Stack.Screen name="Signup" component={SignupScreen} />

          <Stack.Screen name="Login" component={LoginScreen} />

          <Stack.Screen name="Profile" component={ProfileScreen} />

        </Stack.Navigator>

      </NavigationContainer>

    </Provider>

  );

};

*// Styles*

const styles = StyleSheet.create({

  container: { flex: 1, padding: 20, justifyContent: 'center' },

  title: { fontSize: 24, marginBottom: 20, fontWeight: 'bold' },

  input: { borderBottomWidth: 1, marginBottom: 15, padding: 10 },

});

export default App;

**SCREENSHOT:**

A screen shot of a login form

Description automatically generated

A screenshot of a login screen

Description automatically generated

A screenshot of a login screen

Description automatically generated

