



**COMSATS UNIVERSITY
ISLAMABAD ATTOCK CAMPUS**

THEORY ASSIGNMENT NO.4

SUBMITTED TO: SIR MUHAMMAD KAMRAN

SUBMITTED BY: MOMINA TARIQ

REGISTRATION NO: SP22-BSE-043

SUBJECT: MOBILE APPLICATION

DATED: 25-DECEMBER-2024

```
import React, { useEffect } from 'react';
import {
  View,
  Text,
  TextInput,
  TouchableOpacity,
  StyleSheet,
  Alert,
  ScrollView,
} from 'react-native';
import { Provider, useDispatch, useSelector } from 'react-redux';
import { createStore } from 'redux';
import * as Location from 'expo-location';
import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/stack';

// Firebase API Endpoint
const FIREBASE_API_ENDPOINT =
  'https://myproj-a348a-default-rtdb.firebaseio.com/';

// Redux initial state
const initialState = {
  name: '',
  email: '',
  phone: '',
  password: '',
  location: '',
};

// Redux actions
const UPDATE_USER = 'UPDATE_USER';
const SET_LOCATION = 'SET_LOCATION';

const updateUser = (payload) => ({
  type: UPDATE_USER,
  payload,
});

const setLocation = (location) => ({
  type: SET_LOCATION,
  location,
});
```

```

});

// Redux reducer
const reducer = (state = initialState, action) => {
  switch (action.type) {
    case UPDATE_USER:
      return { ...state, ...action.payload };
    case SET_LOCATION:
      return { ...state, location: action.location };
    default:
      return state;
  }
};

// Redux store
const store = createStore(reducer);

// Stack Navigator
const Stack = createStackNavigator();

// Signup Screen
const SignupScreen = ({ navigation }) => {
  const dispatch = useDispatch();
  const { name, email, phone, password } = useSelector((state) => state);

  const handleSignup = async () => {
    if (!name || !email || !phone || !password) {
      Alert.alert('Error', 'Please fill in all fields.');
```

return;

```
    }

    // Store user data in Firebase
    const userData = { name, email, phone, password };
    try {
      await fetch(`${FIREBASE_API_ENDPOINT}/users.json`, {
        method: 'POST',
        body: JSON.stringify(userData),
        headers: { 'Content-Type': 'application/json' },
      });
      Alert.alert('Success', 'Signup successful!');
      navigation.navigate('Login');
    } catch (error) {
      Alert.alert('Error', 'Could not store user data.');
```

```

    }
  };

  return (
    <View style={styles.container}>
      <Text style={styles.title}>Signup</Text>
      <TextInput
        placeholder="Name"
        style={styles.input}
        value={name}
        onChangeText={(value) => dispatch(updateUser({ name: value })))}
      />
      <TextInput
        placeholder="Email"
        style={styles.input}
        value={email}
        onChangeText={(value) => dispatch(updateUser({ email: value })))}
      />
      <TextInput
        placeholder="Phone"
        style={styles.input}
        value={phone}
        onChangeText={(value) => dispatch(updateUser({ phone: value })))}
      />
      <TextInput
        placeholder="Password"
        secureTextEntry
        style={styles.input}
        value={password}
        onChangeText={(value) => dispatch(updateUser({ password: value })))}
      />
      <TouchableOpacity style={styles.button} onPress={handleSignup}>
        <Text style={styles.buttonText}>Signup</Text>
      </TouchableOpacity>
    </View>
  );
};

// Login Screen
const LoginScreen = ({ navigation }) => {
  const { email, password } = useSelector((state) => state);
  const dispatch = useDispatch();

```

```

const handleLogin = async () => {
  if (!email || !password) {
    Alert.alert('Error', 'Please fill in all fields.');
```

return;

```
  }

  try {
    const { status } = await Location.requestForegroundPermissionsAsync();
    if (status !== 'granted') {
      Alert.alert('Permission denied', 'Location access is required.');
```

return;

```
    }

    const location = await Location.getCurrentPositionAsync({});
    const locationString = `${location.coords.latitude},
    ${location.coords.longitude}`;
    dispatch(setLocation(locationString));
    Alert.alert('Success', 'Login successful!');
```

navigation.navigate('Profile');

```
  } catch (error) {
    Alert.alert('Error', 'Could not fetch location.');
```

}

```
};

return (
  <View style={styles.container}>
    <Text style={styles.title}>Login</Text>
    <TextInput
      placeholder="Email"
      style={styles.input}
      value={email}
      onChangeText={(value) => dispatch(updateUser({ email: value })))}
    />
    <TextInput
      placeholder="Password"
      secureTextEntry
      style={styles.input}
      value={password}
      onChangeText={(value) => dispatch(updateUser({ password: value })))}
    />
    <TouchableOpacity style={styles.button} onPress={handleLogin}>
      <Text style={styles.buttonText}>Login</Text>
    </TouchableOpacity>
  </View>
);

```

```

    </View>
  );
};

// Profile Screen
const ProfileScreen = () => {
  const { name, email, phone, location } = useSelector((state) => state);

  return (
    <ScrollView style={styles.container}>
      <Text style={styles.title}>My Profile</Text>
      <Text>Name: {name}</Text>
      <Text>Email: {email}</Text>
      <Text>Phone: {phone}</Text>
      <Text>Location: {location}</Text>
    </ScrollView>
  );
};

// Main App Component
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, backgroundColor: '#fff' },
  title: { fontSize: 24, fontWeight: 'bold', marginBottom: 20 },
  input: { borderWidth: 1, borderColor: '#ddd', marginBottom: 15, padding: 10 },
  button: { backgroundColor: '#007BFF', padding: 15, alignItems: 'center' },
  buttonText: { color: '#fff', fontSize: 16 },
});

```

```
export default App;
```

Signup

Signup

[← Login](#)

Login

← Profile

My Profile

Name: Momina Tariq

Email: Momitariq512@gmail.com

Phone: 03123456789

Location: 33.6188, 73.0569