

**COMSATS UNIVERSITY ISLAMABAD
(ATTOCK CAMPUS)**

Assignment No 4

Submitted By:

Remsha Haneef (Sp22-bse-056)

Course:

Mobile Application Development

Date:

24th Dec,2024

Submitted To:

Mr. Muhammad Kamran

Code:

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

import
{View,Text,TextInput,TouchableOpacity,Alert,StyleSheet,ScrollView,Image,
} from 'react-native';

import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/stack';
import { Provider, useDispatch, useSelector } from 'react-redux';
import { configureStore, createSlice } from '@reduxjs/toolkit';
import AsyncStorage from '@react-native-async-storage/async-storage';
import * as Location from 'expo-location';

const FIREBASE_API_ENDPOINT = 'https://mypriject-37990-default-
rtbd.firebaseio.com/';

const Stack = createStackNavigator();

const userSlice = createSlice({
  name: 'user',
  initialState: {
    name: "",
    email: "",
    phoneNumber: "",
    password: "",
    isAuthenticated: false,
    location: null,
```

```
    },  
    reducers: {  
      setName: (state, action) => {  
        state.name = action.payload;  
      },  
      setEmail: (state, action) => {  
        state.email = action.payload;  
      },  
      setPhoneNumber: (state, action) => {  
        state.phoneNumber = action.payload;  
      },  
      setPassword: (state, action) => {  
        state.password = action.payload;  
      },  
      setAuthentication: (state, action) => {  
        state.isAuthenticated = action.payload;  
      },  
      setLocation: (state, action) => {  
        state.location = action.payload;  
      },  
    },  
  });  
  
  const { setName, setEmail, setPhoneNumber, setPassword, setAuthentication,  
    setLocation } =  
    userSlice.actions;  
  
  const store = configureStore({
```

```
reducer: {
  user: userSlice.reducer,
},
});

// Signup Screen

const SignupScreen = ({ navigation }) => {

  const { name, email, phoneNumber, password } = useSelector((state) =>
state.user);

  const dispatch = useDispatch();

  const validateEmail = (email) => /^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(email);
  const validatePhoneNumber = (number) => /^[0-9]{10}$/.test(number);

  const handleSignup = async () => {
    if (!name || !email || !phoneNumber || !password) {
      Alert.alert('Error', 'Please fill in all fields');
    } else if (!validateEmail(email)) {
      Alert.alert('Error', 'Invalid email syntax');
    } else if (!validatePhoneNumber(phoneNumber)) {
      Alert.alert('Error', 'Invalid phone number. It should be 10 digits.');
```

Alert.alert('Error', 'Invalid phone number. It should be 10 digits.');

```
    } else if (password.length < 6) {
      Alert.alert('Error', 'Password must be at least 6 characters');
    } else {
      try {
        const response = await fetch(`${FIREBASE_API_ENDPOINT}/users.json`, {
          method: 'POST',
          body: JSON.stringify({ name, email, phoneNumber, password }),
          headers: { 'Content-Type': 'application/json' },
```

```
});  
if (response.ok) {  
  dispatch(setAuthentication(true));  
  Alert.alert('Success', 'Signup successful!');  
  navigation.navigate('Login');  
} else {  
  Alert.alert('Error', 'Failed to insert data.');
```



```
}  
} catch (error) {  
  Alert.alert('Error', 'Something went wrong.');
```



```
}  
}};  
return (  
  <View style={styles.container}>  
    <Text style={styles.title}>Signup</Text>  
    <TextInput  
      style={styles.input}  
      placeholder="Name"  
      value={name}  
      onChangeText={(text) => dispatch(setName(text))}  
    />  
    <TextInput  
      style={styles.input}  
      placeholder="Email"  
      keyboardType="email-address"
```

```

    value={email}
    onChangeText={{(text) => dispatch(setEmail(text))}}
  />
  <TextInput
    style={styles.input}
    placeholder="Phone Number"
    keyboardType="phone-pad"
    value={phoneNumber}
    onChangeText={{(text) => dispatch(setPhoneNumber(text))}}
  />
  <TextInput
    style={styles.input}
    placeholder="Password"
    secureTextEntry
    value={password}
    onChangeText={{(text) => dispatch(setPassword(text))}}
  />
  <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.user);
const dispatch = useDispatch();
const fetchLocation = async () => {
  const { status } = await Location.requestForegroundPermissionsAsync();
  if (status !== 'granted') {
    Alert.alert('Error', 'Permission to access location was denied.');
```

return null;

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

} else {

```
  try {
    const response = await fetch(`${FIREBASE_API_ENDPOINT}/users.json`);
    const data = await response.json();
    if (data) {
      const users = Object.values(data);
      const user = users.find(
        (user) => user.email === email && user.password === password
      );
      if (user) {
        const location = await fetchLocation();
```

```
if (location) {
  const locationData = {
    latitude: location.coords.latitude,
    longitude: location.coords.longitude,
  };
  await AsyncStorage.setItem('userLocation', JSON.stringify(locationData));
  dispatch(setLocation(locationData));
}
dispatch(setAuthentication(true));
Alert.alert('Success', 'Login successful!');
navigation.navigate('Profile');
} else {
  Alert.alert('Error', 'Invalid email or password');
}
} else {
  Alert.alert('Error', 'No users found.');
```

```
}
} catch (error) {
  Alert.alert('Error', 'Something went wrong.');
```

```
}
}
};return (
<View style={styles.container}>
<Text style={styles.title}>Login</Text>
<TextInput
```



```

style={styles.input}
placeholder="Email"
keyboardType="email-address"
value={email}
onChangeText={(text) => dispatch(setEmail(text))}
/>

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

<TouchableOpacity style={styles.button} onPress={handleLogin}>
<Text style={styles.buttonText}>Login</Text>
</TouchableOpacity>

</View>

);

};

```

// Profile Screen

```

const ProfileScreen = () => {
const { name, email, phoneNumber, location } = useSelector((state) => state.user);
return (
<ScrollView contentContainerStyle={styles.scrollContent}>
<View style={styles.header}>

```

```
<Text style={styles.title}>My Profile</Text>
</View>
<View style={styles.profileSection}>
  <Image
    source={{ uri: 'https://tse1.mm.bing.net/th?id=OIP.DHVLtvV0hRBXlnOF-
    BTGAHaFD&pid=Api' }}
    style={styles.profileImage}
  />
  <Text style={styles.profileName}>Remsha Haneef</Text>
  <View style={styles.stats}>
    <View style={styles.statItem}>
      <Text style={styles.statValue}>14,850</Text>
      <Text style={styles.statLabel}>Product Sells</Text>
    </View>
    <View style={styles.statItem}>
      <Text style={styles.statValue}>30bil+</Text>
      <Text style={styles.statLabel}>Earning</Text>
    </View>
  </View>
  <View style={styles.ordersSection}>
    <View style={styles.orderHeader}>
      <Text style={styles.sectionTitle}>Recent Orders</Text>
      <TouchableOpacity>
        <Text style={styles.seeAllText}>See All</Text>
      </TouchableOpacity>
    </View>
  </View>
</View>
</View>
```

```
</View>
<View style={styles.orderItem}>
  <Image
    source={{ uri:
      'https://upload.wikimedia.org/wikipedia/commons/4/45/A_small_cup_of_coffee.JPG' }}
    style={styles.orderImage}
  />
  <View style={styles.orderDetails}>
    <Text style={styles.orderTitle}>Coffee, Sandwich, Muffin</Text>
    <Text style={styles.orderDate}>22 Thu 10.00</Text>
  </View>
  <Text style={styles.orderPrice}>$12.50</Text>
</View>
<View style={styles.orderItem}>
  <Image
    source={{ uri:
      'https://encryptedtbn0.gstatic.com/images?q=tbn:ANd9GcTjQFf31CNn3kmzrUBVZpxYUQBZ1rBzOv1jKA&s' }}
    style={styles.orderImage}
  />
  <View style={styles.orderDetails}>
    <Text style={styles.orderTitle}>Donut, Pancake, Juice</Text>
    <Text style={styles.orderDate}>15 Fri 18.00</Text>
  </View>
  <Text style={styles.orderPrice}>$20.30</Text>
```

```

</View>
<View style={styles.orderItem}>
  <Image
    source={{ uri:
      'https://encryptedtbn0.gstatic.com/images?q=tbn:ANd9GcQMrD6lWPfd_39zCAK
      mo0DatnyZf9VZXiNqvA&s' }}
    style={styles.orderImage}
  />
  <View style={styles.orderDetails}>
    <Text style={styles.orderTitle}>Beef Burger, Fries, Coke</Text>
    <Text style={styles.orderDate}>11 Mon 10.00</Text>
  </View>
  <Text style={styles.orderPrice}>$25.75</Text>
</View>
</View>
</ScrollView>
);
}
// App Navigation
const App = () => {
  return (
    <Provider store={store}>
      <NavigationContainer>
        <Stack.Navigator initialRouteName="Signup">
          <Stack.Screen name="Signup" component={SignupScreen} />
          <Stack.Screen name="Login" component={LoginScreen} />
        </Stack.Navigator>
      </NavigationContainer>
    </Provider>
  );
};

```

```
<Stack.Screen name="Profile" component={ProfileScreen} />
</Stack.Navigator>
</NavigationContainer>
</Provider>
);
};

// Styles
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#F7F7F7',
    padding: 20,
  },
  scrollContent: {
    paddingBottom: 20,
  },
  header: {
    alignItems: 'center',
    marginBottom: 20,
  },
  title: {
    fontSize: 24,
    fontWeight: 'bold',
    marginBottom: 20,
    textAlign: 'center',
```

```
color: '#333',
},
input: {
borderWidth: 1,
borderColor: '#ccc',
padding: 10,
marginBottom: 15,
borderRadius: 5,
backgroundColor: '#FFF',
fontSize: 16,
color: '#333',
},
button: {
backgroundColor: '#007BFF',
padding: 15,
borderRadius: 5,
alignItems: 'center',
marginTop: 10,
},
buttonText: {
color: '#FFF',
fontSize: 16,
fontWeight: 'bold',
},
profileSection: {
```

```
backgroundColor: '#FFF',
padding: 20,
borderRadius: 10,
alignItems: 'center',
marginBottom: 20,
shadowColor: '#000',
shadowOffset: { width: 0, height: 2 },
shadowOpacity: 0.1,
shadowRadius: 5,
elevation: 2,
},
profileImage: {
width: 80,
height: 80,
borderRadius: 40,
marginBottom: 10,
},
profileName: {
fontSize: 18,
fontWeight: 'bold',
marginBottom: 5,
color: '#333',
},profileDetails: {
fontSize: 16,
color: '#666',
```

```
marginBottom: 5,
},
ordersSection: {
marginTop: 20,
},
sectionTitle: {
fontSize: 20,
fontWeight: 'bold',
marginBottom: 10,
color: '#333',
},
orderItem: {
flexDirection: 'row',
backgroundColor: '#FFF',
padding: 15,
borderRadius: 5,
marginBottom: 10,
alignItems: 'center',
shadowColor: '#000',
shadowOffset: { width: 0, height: 2 },
shadowOpacity: 0.1,
shadowRadius: 5,
elevation: 2,
},
orderImage: {
```



```
width: 60,
height: 60,
borderRadius: 5,
marginRight: 15,
},
orderDetails: {
flex: 1,
},
orderTitle: {
fontSize: 16,
fontWeight: 'bold',
color: '#333',
marginBottom: 5,
},
orderDate: {
fontSize: 12,
color: '#666',
},
orderPrice: {
fontSize: 16,
fontWeight: 'bold',
color: '#333',
marginLeft: 10,
},
// Recent Orders Header
```

```
orderHeader: {  
  flexDirection: 'row',  
  justifyContent: 'space-between',  
  alignItems: 'center',  
  marginBottom: 10,  
},  
seeAllText: {  
  fontSize: 14,  
  color: '#007BFF',  
},  
});  
export default App;
```

OUTPUT:

SignUp:

2:14 PM

Vo LTE Vo LTE 93% 9/8 24%

Signup

Signup

Remsha

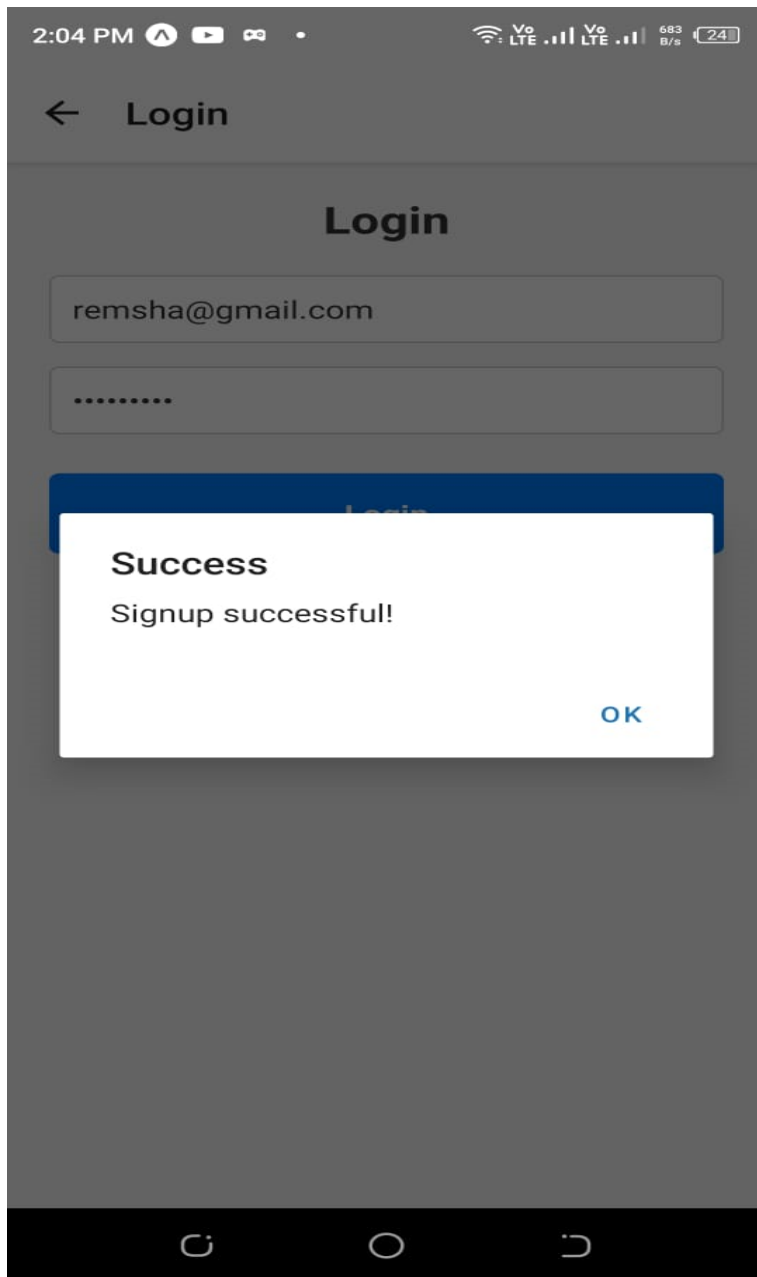
remsha@gmail.com

0345632786

.....

Signup





Login:

2:04 PM

Vs LTE Vs LTE 2 K/s 24

← Login

Login

remsha@gmail.com

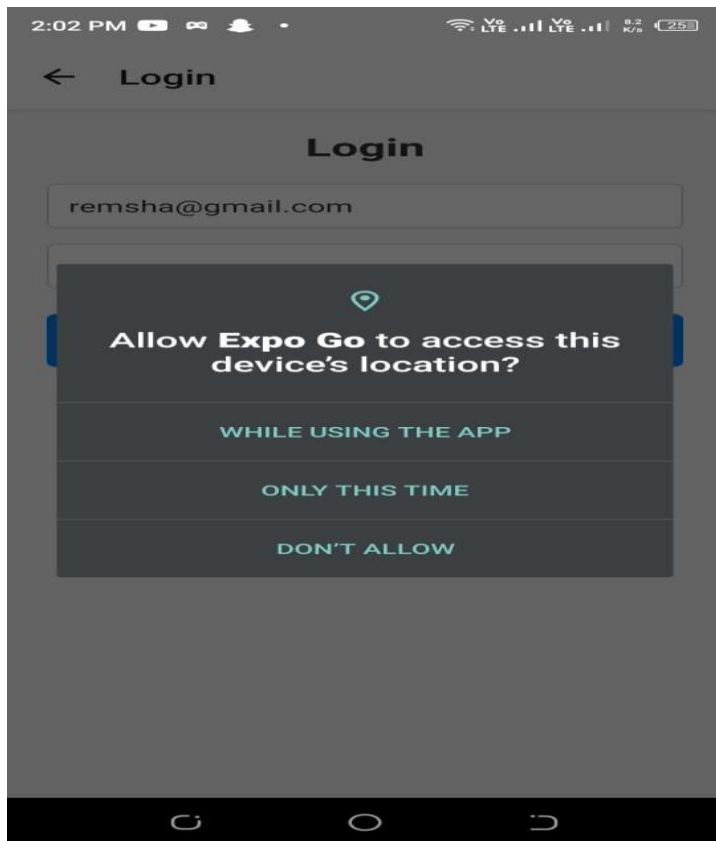
.....

Login

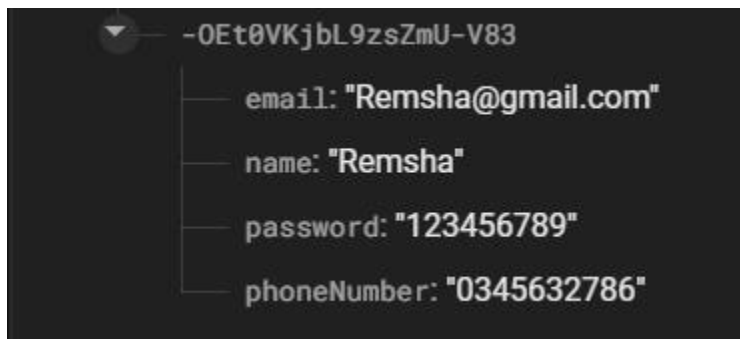
Screenshot has been saved to Pictures/
Screenshot



Location:



Firestore:



Profile:

