

TITLE:
INFORMATION OF ORDERS

ASSIGNMENT NO 04
MOBILE APPLICATION DEVELOPMENT

PROGRAM: BSSE-6



SUBMITTED TO:
SIR MUHAMMAD KAMRAN

SUBMITTED BY:
ZARNAB IMRAN (sp22-bse-058)

DATE: 25TH, DEC 2024

PART#01:

CODE:

```
// App.js
import React from 'react';
import { SafeAreaView, StatusBar, StyleSheet, View, Text, Image, ScrollView,
TextInput, TouchableOpacity } from 'react-native';
import { Provider, useSelector, useDispatch } from 'react-redux';
import { configureStore, createSlice } from '@reduxjs/toolkit';
import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/stack';

// Redux slice for profile and order data
const profileSlice = createSlice({
  name: 'profile',
  initialState: {
    name: 'Zarnab Imran',
    stats: {
      productSells: '55.850',
      earning: '300bil+',
    },
    orders: [
      { description: 'Ice Cream', date: '1st, Monday 10.00', price: '150.00' },
      { description: 'Beef Burger', date: '2nd, Tuesday 11.00', price: '550.00' },
      { description: 'French Fries', date: '5th, Friday 10.00', price: '200.00' },
    ],
  },
  reducers: {},
});

const store = configureStore({
  reducer: {
    profile: profileSlice.reducer,
  },
});

const Stack = createStackNavigator();

const LoginScreen = ({ navigation }) => {
  return (
    <View style={styles.authContainer}>
      <Text style={styles.authHeading}>Login</Text>
      <TextInput style={styles.input} placeholder="Email" keyboardType="email-address" />
    </View>
  );
}
```

```

        <TextInput style={styles.input} placeholder="Password"
secureTextEntry={true} />
        <TouchableOpacity style={styles.authButton} onPress={() =>
navigation.navigate('Profile')}>
            <Text style={styles.authButtonText}>Login</Text>
        </TouchableOpacity>
        <TouchableOpacity onPress={() => navigation.navigate('Signup')}>
            <Text style={styles.authLink}>Don't have an account? Sign up</Text>
        </TouchableOpacity>
    </View>
  );
};

const SignupScreen = ({ navigation }) => {
  return (
    <View style={styles.authContainer}>
      <Text style={styles.authHeading}>Sign Up</Text>
      <TextInput style={styles.input} placeholder="Name" />
      <TextInput style={styles.input} placeholder="Email" keyboardType="email-
address" />
      <TextInput style={styles.input} placeholder="Password"
secureTextEntry={true} />
      <TouchableOpacity style={styles.authButton} onPress={() =>
navigation.navigate('Login')}>
        <Text style={styles.authButtonText}>Sign Up</Text>
      </TouchableOpacity>
      <TouchableOpacity onPress={() => navigation.navigate('Login')}>
        <Text style={styles.authLink}>Already have an account? Log in</Text>
      </TouchableOpacity>
    </View>
  );
};

const ProfileScreen = () => {
  const profile = useSelector((state) => state.profile);

  return (
    <View style={styles.container}>
      <Text style={styles.myProfileHeading}>My Profile</Text>

      <View style={styles.profileCard}>
        <View style={styles.profileInfo}>
          <Image
            source={require('./assets/images/profile.jpg')} // Replace with
actual image path
            style={styles.profileImage}
          />
          <View style={styles.profileDetails}>

```

```

        <Text style={styles.profileName}>{profile.name}</Text>
        <View style={styles.stats}>
          <View style={styles.statItem}>
            <Image source={require('./assets/images/box.jpg')}
style={styles.statIcon} />
            <View style={styles.statText}>
              <Text
style={styles.statValue}>{profile.stats.productSells}</Text>
              <Text style={styles.statLabel}>Product Sells</Text>
            </View>
          </View>
          <View style={styles.statItem}>
            <Image source={require('./assets/images/coin.jpg')}
style={styles.statIcon} />
            <View style={styles.statText}>
              <Text
style={styles.statValue}>{profile.stats.earning}</Text>
              <Text style={styles.statLabel}>Earning</Text>
            </View>
          </View>
        </View>
        <TouchableOpacity style={styles.settingsButton}>
          <Image source={require('./assets/images/setting.jpg')}
style={styles.settingsIcon} />
        </TouchableOpacity>
      </View>

      <View style={styles.recentOrdersHeader}>
        <Text style={styles.sectionTitle}>Recent Orders</Text>
        <TouchableOpacity>
          <Text style={styles.seeAll}>See All</Text>
        </TouchableOpacity>
      </View>

      <ScrollView style={styles.ordersList}>
        {profile.orders.map((order, index) => (
          <View key={index} style={styles.orderCard}>
            <Image
source={require('./assets/images/meal.jpg')} // Replace with
actual image path
style={styles.orderImage}
            />
            <View style={styles.orderDetails}>
              <Text style={styles.orderDescription}>{order.description}</Text>
              <Text style={styles.orderDate}>{order.date}</Text>
            </View>
          </View>
        ))}
      </ScrollView>
    </View>
  </View>
)

```

```

        <View style={styles.orderPrice}>
          <Text style={styles.priceText}>${order.price}</Text>
        </View>
      </View>
    )}}
  </ScrollView>
</View>
);
};

const App = () => (
  <Provider store={store}>
    <NavigationContainer>
      <Stack.Navigator screenOptions={{ headerShown: false }}>
        <Stack.Screen name="Login" component={LoginScreen} />
        <Stack.Screen name="Signup" component={SignupScreen} />
        <Stack.Screen name="Profile" component={ProfileScreen} />
      </Stack.Navigator>
    </NavigationContainer>
  </Provider>
);

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 20,
    backgroundColor: '#F6F2E9',
  },
  myProfileHeading: {
    fontSize: 24,
    fontWeight: 'bold',
    marginBottom: 20,
    textAlign: 'center',
  },
  profileCard: {
    backgroundColor: '#F9EED4',
    padding: 20,
    borderRadius: 20,
    flexDirection: 'row',
    alignItems: 'center',
    justifyContent: 'space-between',
  },
  profileInfo: {
    flexDirection: 'row',
    alignItems: 'center',
  },
  profileImage: {
    width: 50,

```

```
    height: 50,
    borderRadius: 25,
  },
  profileDetails: {
    marginLeft: 15,
  },
  profileName: {
    fontSize: 18,
    fontWeight: 'bold',
  },
  stats: {
    flexDirection: 'row',
    marginTop: 10,
  },
  statItem: {
    flexDirection: 'row',
    alignItems: 'center',
    marginRight: 20,
  },
  statIcon: {
    width: 20,
    height: 20,
    marginRight: 5,
  },
  statText: {
    flexDirection: 'column',
  },
  statValue: {
    fontSize: 16,
    fontWeight: 'bold',
  },
  statLabel: {
    fontSize: 12,
    color: '#888',
  },
  settingsButton: {
    padding: 10,
  },
  settingsIcon: {
    width: 20,
    height: 20,
  },
  recentOrdersHeader: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    marginTop: 30,
  },
  sectionTitle: {
```

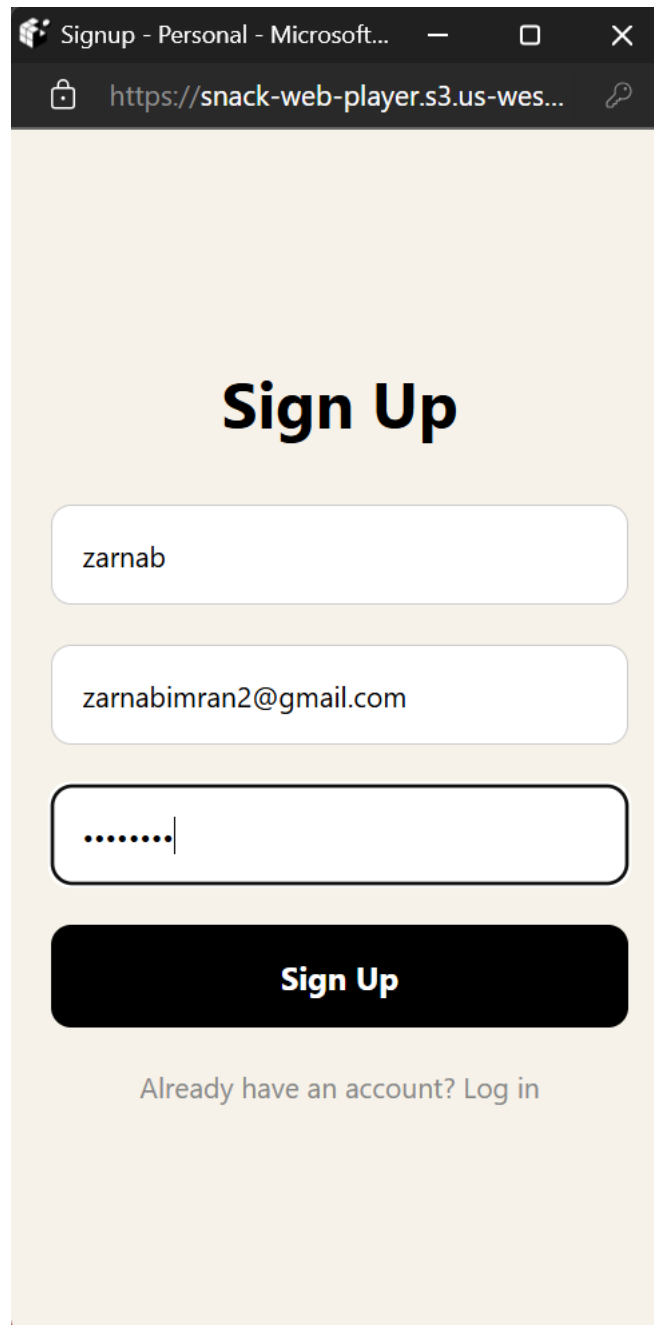
```
    fontSize: 20,
    fontWeight: 'bold',
  },
  seeAll: {
    fontSize: 16,
    color: '#888',
  },
  ordersList: {
    marginTop: 20,
  },
  orderCard: {
    flexDirection: 'row',
    alignItems: 'center',
    backgroundColor: '#000',
    padding: 15,
    borderRadius: 20,
    marginBottom: 10,
  },
  orderImage: {
    width: 50,
    height: 50,
    borderRadius: 10,
  },
  orderDetails: {
    marginLeft: 15,
    flex: 1,
  },
  orderDescription: {
    fontSize: 16,
    color: '#fff',
  },
  orderDate: {
    fontSize: 12,
    color: '#888',
    marginTop: 5,
  },
  orderPrice: {
    marginLeft: 'auto',
  },
  priceText: {
    fontSize: 18,
    fontWeight: 'bold',
    color: '#fff',
  },
  authContainer: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
```

```
padding: 20,
backgroundColor: '#F6F2E9',
},
authHeading: {
  fontSize: 32,
  fontWeight: 'bold',
  marginBottom: 20,
},
input: {
  width: '100%',
  padding: 15,
  marginVertical: 10,
  backgroundColor: 'fff',
  borderRadius: 10,
  borderWidth: 1,
  borderColor: 'ccc',
},
authButton: {
  width: '100%',
  padding: 15,
  backgroundColor: '#000',
  borderRadius: 10,
  alignItems: 'center',
  marginVertical: 10,
},
authButtonText: {
  color: 'fff',
  fontSize: 16,
  fontWeight: 'bold',
},
authLink: {
  color: '#888',
  fontSize: 14,
  marginTop: 10,
},
});

export default App;
```

OUTPUT:

Signup Screen:



The image shows a web browser window with the title "Signup - Personal - Microsoft...". The address bar displays "https://snack-web-player.s3.us-wes...". The main content area has a light beige background and features the heading "Sign Up" in bold black text. Below the heading are three input fields: the first contains "zarnab", the second contains "zarnabimran2@gmail.com", and the third contains seven dots followed by a vertical cursor. A black button with the text "Sign Up" in white is positioned below the input fields. At the bottom, there is a link that says "Already have an account? Log in".

Signup - Personal - Microsoft...

https://snack-web-player.s3.us-wes...

Sign Up

zarnab

zarnabimran2@gmail.com

.....|

Sign Up

Already have an account? [Log in](#)

Login Screen:

Login - Personal - Microsoft...

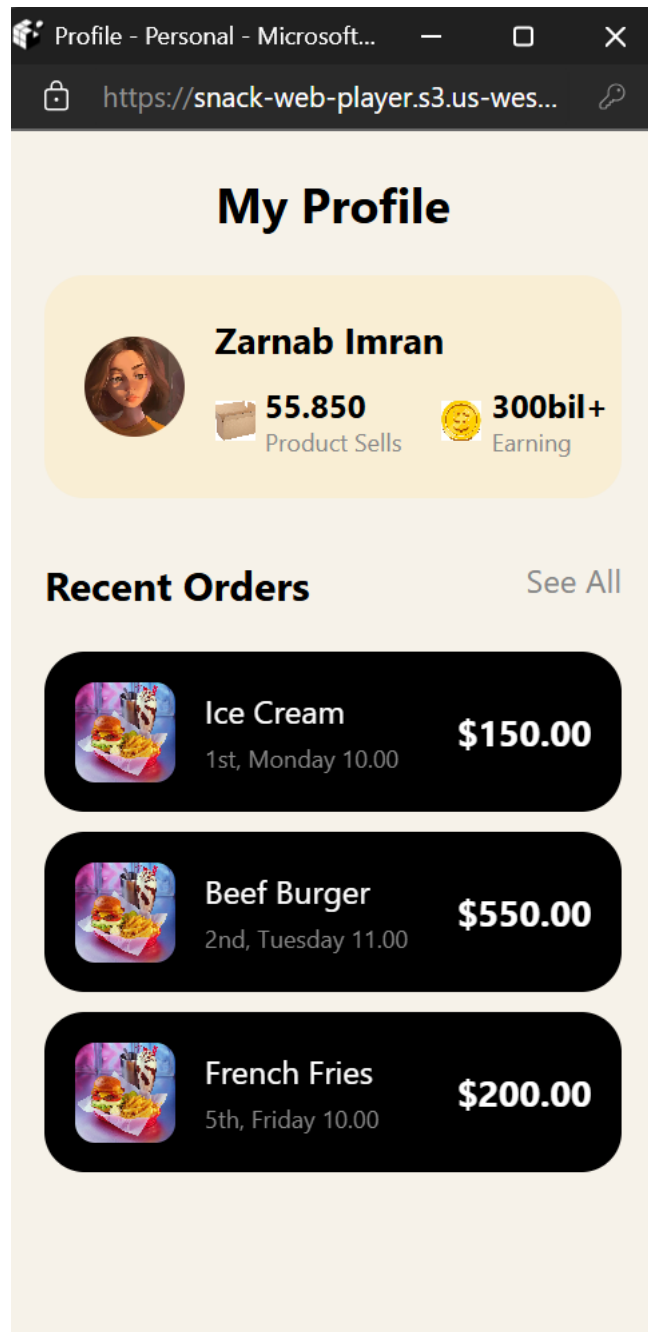
https://snack-web-player.s3.us-wes...

Login

Login

Don't have an account? Sign up

Profile Screen:



PART#02:

CODE:

```
// App.js
import React, { useEffect, useState } from 'react';
import {
  SafeAreaView,
  StatusBar,
  StyleSheet,
  View,
  Text,
  Image,
```

```

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

// Firebase configuration
const firebaseConfig = {
  apiKey: 'YOUR_API_KEY',
  authDomain: 'YOUR_AUTH_DOMAIN',
  projectId: 'YOUR_PROJECT_ID',
  storageBucket: 'YOUR_STORAGE_BUCKET',
  messagingSenderId: 'YOUR_MESSAGING_SENDER_ID',
  appId: 'YOUR_APP_ID',
};

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

// Redux slice for profile and user location data
const profileSlice = createSlice({
  name: 'profile',
  initialState: {
    name: '',
    stats: {
      productSells: '55.850',
      earning: '300bil+',
    },
    orders: [
      { description: 'Ice Cream', date: '1st, Monday 10.00', price: '150.00' },
      { description: 'Beef Burger', date: '2nd, Tuesday 11.00', price: '550.00' },
      { description: 'French Fries', date: '5th, Friday 10.00', price: '200.00' },
    ],
    location: '',
  },
});

```

```

    reducers: {
      setProfile: (state, action) => {
        state.name = action.payload.name;
      },
      setLocation: (state, action) => {
        state.location = action.payload;
      },
    },
  });

const { setProfile, setLocation } = profileSlice.actions;
const store = configureStore({
  reducer: {
    profile: profileSlice.reducer,
  },
});

const Stack = createStackNavigator();

const LoginScreen = ({ navigation }) => {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');

  const handleLogin = async () => {
    try {
      const userCredential = await
firebase.auth().signInWithEmailAndPassword(email, password);
      const userId = userCredential.user.uid;
      const userRef = firebase.database().ref(`/users/${userId}`);
      const snapshot = await userRef.once('value');
      const userData = snapshot.val();

      if (userData) {
        store.dispatch(setProfile({ name: userData.name }));
        Alert.alert('Login Successful', 'Welcome back!');
        navigation.navigate('Profile');
      } else {
        Alert.alert('Error', 'User data not found!');
      }
    } catch (error) {
      Alert.alert('Login Failed', error.message);
    }
  };

  return (
    <View style={styles.authContainer}>
      <Text style={styles.authHeading}>Login</Text>
      <TextInput

```

```

        style={styles.input}
        placeholder="Email"
        keyboardType="email-address"
        value={email}
        onChangeText={setEmail}
      />
      <TextInput
        style={styles.input}
        placeholder="Password"
        secureTextEntry={true}
        value={password}
        onChangeText={setPassword}
      />
      <TouchableOpacity style={styles.authButton} onPress={handleLogin}>
        <Text style={styles.authButtonText}>Login</Text>
      </TouchableOpacity>
      <TouchableOpacity onPress={() => navigation.navigate('Signup')}>
        <Text style={styles.authLink}>Don't have an account? Sign up</Text>
      </TouchableOpacity>
    </View>
  );
};

const SignupScreen = ({ navigation }) => {
  const [name, setName] = useState('');
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');

  const handleSignup = async () => {
    try {
      const userCredential = await
firebase.auth().createUserWithEmailAndPassword(email, password);
      const userId = userCredential.user.uid;

      await firebase.database().ref(`/users/${userId}`).set({
        name,
        email,
      });

      Alert.alert('Signup Successful', 'Account created successfully!');
      navigation.navigate('Login');
    } catch (error) {
      Alert.alert('Signup Failed', error.message);
    }
  };

  return (
    <View style={styles.authContainer}>

```

```

<Text style={styles.authHeading}>Sign Up</Text>
<TextInput
  style={styles.input}
  placeholder="Name"
  value={name}
  onChangeText={setName}
/>
<TextInput
  style={styles.input}
  placeholder="Email"
  keyboardType="email-address"
  value={email}
  onChangeText={setEmail}
/>
<TextInput
  style={styles.input}
  placeholder="Password"
  secureTextEntry={true}
  value={password}
  onChangeText={setPassword}
/>
<TouchableOpacity style={styles.authButton} onPress={handleSignup}>
  <Text style={styles.authButtonText}>Sign Up</Text>
</TouchableOpacity>
<TouchableOpacity onPress={() => navigation.navigate('Login')}>
  <Text style={styles.authLink}>Already have an account? Log in</Text>
</TouchableOpacity>
</View>
);
};

const ProfileScreen = () => {
  const profile = useSelector((state) => state.profile);
  const dispatch = useDispatch();

  useEffect(() => {
    const fetchLocation = async () => {
      const { status } = await Location.requestForegroundPermissionsAsync();
      if (status !== 'granted') {
        Alert.alert('Permission Denied', 'Unable to access location');
        return;
      }
      const location = await Location.getCurrentPositionAsync();
      const coords = `${location.coords.latitude},
${location.coords.longitude}`;

      dispatch(setLocation(coords));
      await AsyncStorage.setItem('userLocation', coords);
    }
  }, []);
};

```

```

    };
    fetchLocation();
  }, [dispatch]));

  return (
    <View style={styles.container}>
      <Text style={styles.myProfileHeading}>My Profile</Text>

      <View style={styles.profileCard}>
        <View style={styles.profileInfo}>
          <Image
            source={require('./assets/images/profile.jpg')}
            style={styles.profileImage}
          />
          <View style={styles.profileDetails}>
            <Text style={styles.profileName}>{profile.name}</Text>
            <View style={styles.stats}>
              <Text>Location: {profile.location || 'Fetching...'}</Text>
            </View>
          </View>
        </View>
      </View>
    </View>
  );
};

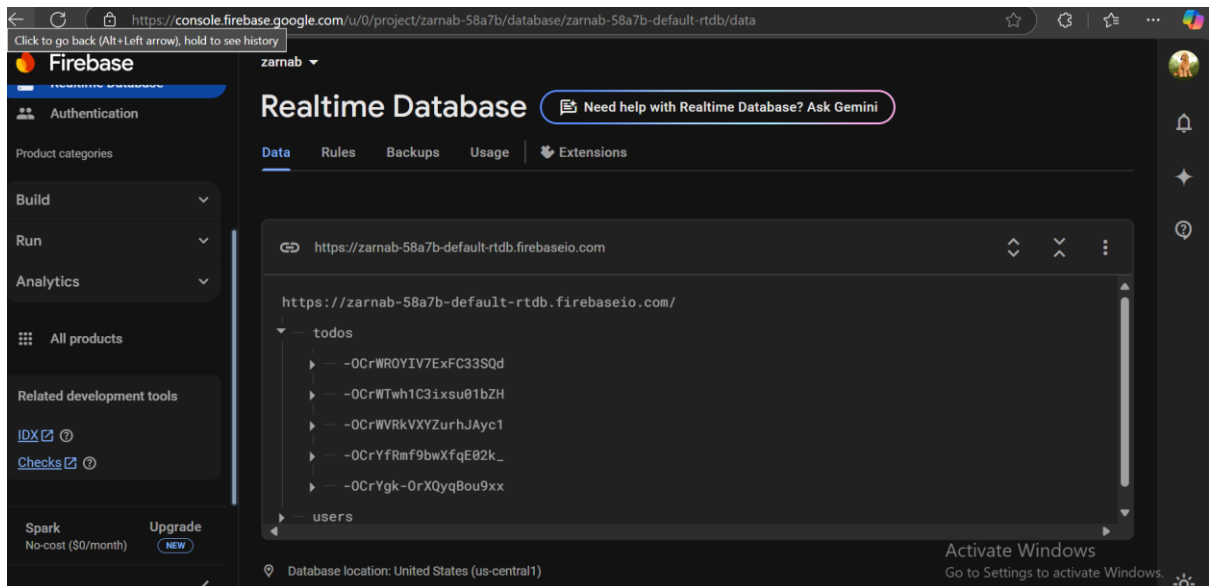
const App = () => (
  <Provider store={store}>
    <NavigationContainer>
      <Stack.Navigator screenOptions={{ headerShown: false }}>
        <Stack.Screen name="Login" component={LoginScreen} />
        <Stack.Screen name="Signup" component={SignupScreen} />
        <Stack.Screen name="Profile" component={ProfileScreen} />
      </Stack.Navigator>
    </NavigationContainer>
  </Provider>
);

const styles = StyleSheet.create({
  // Add styles here (from previous sections)
});

export default App;

```

OUTPUT:



PART#03:

CODE:

```
// App.js
import React, { useEffect, useState } from 'react';
import {
  SafeAreaView,
  StatusBar,
  StyleSheet,
  View,
  Text,
  Image,
  ScrollView,
  TextInput,
  TouchableOpacity
} from 'react-native';
import { Provider, useSelector, useDispatch } from 'react-redux';
import { configureStore, createSlice } from '@reduxjs/toolkit';
import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/stack';
import AsyncStorage from '@react-native-async-storage/async-storage';
import * as Location from 'expo-location';

// Redux slice for profile and order data
const profileSlice = createSlice({
  name: 'profile',
  initialState: {
    name: 'Zarnab Imran',
    stats: {
      productSells: '55.850',
    },
  },
});
```

```

    earning: '300bil+',
  },
  orders: [
    { description: 'Ice Cream', date: '1st, Monday 10.00', price: '150.00'
  },
    { description: 'Beef Burger', date: '2nd, Tuesday 11.00', price:
'550.00' },
    { description: 'French Fries', date: '5th, Friday 10.00', price:
'200.00' },
  ],
  location: null,
},
reducers: {
  setLocation: (state, action) => {
    state.location = action.payload;
  },
},
});

const { setLocation } = profileSlice.actions;
const store = configureStore({
  reducer: {
    profile: profileSlice.reducer,
  },
});

const Stack = createStackNavigator();

const LoginScreen = ({ navigation }) => {
  const dispatch = useDispatch();

  const fetchLocation = async () => {
    let { status } = await Location.requestForegroundPermissionsAsync();
    if (status !== 'granted') {
      console.log('Permission to access location was denied');
      return;
    }

    let location = await Location.getCurrentPositionAsync({});
    const locationString = `${location.coords.latitude},
${location.coords.longitude}`;
    await AsyncStorage.setItem('userLocation', locationString);
    dispatch(setLocation(locationString));
  };

  const handleLogin = async () => {
    await fetchLocation();
    navigation.navigate('Profile');
  };

```

```

    };

    return (
      <View style={styles.authContainer}>
        <Text style={styles.authHeading}>Login</Text>
        <TextInput style={styles.input} placeholder="Email" keyboardType="email-
address" />
        <TextInput style={styles.input} placeholder="Password"
secureTextEntry={true} />
        <TouchableOpacity style={styles.authButton} onPress={handleLogin}>
          <Text style={styles.authButtonText}>Login</Text>
        </TouchableOpacity>
        <TouchableOpacity onPress={() => navigation.navigate('Signup')}>
          <Text style={styles.authLink}>Don't have an account? Sign up</Text>
        </TouchableOpacity>
      </View>
    );
  };
};

const SignupScreen = ({ navigation }) => {
  return (
    <View style={styles.authContainer}>
      <Text style={styles.authHeading}>Sign Up</Text>
      <TextInput style={styles.input} placeholder="Name" />
      <TextInput style={styles.input} placeholder="Email" keyboardType="email-
address" />
      <TextInput style={styles.input} placeholder="Password"
secureTextEntry={true} />
      <TouchableOpacity style={styles.authButton} onPress={() =>
navigation.navigate('Login')}>
        <Text style={styles.authButtonText}>Sign Up</Text>
      </TouchableOpacity>
      <TouchableOpacity onPress={() => navigation.navigate('Login')}>
        <Text style={styles.authLink}>Already have an account? Log in</Text>
      </TouchableOpacity>
    </View>
  );
};

const ProfileScreen = () => {
  const profile = useSelector((state) => state.profile);

  return (
    <View style={styles.container}>
      <Text style={styles.myProfileHeading}>My Profile</Text>

      <View style={styles.profileCard}>
        <View style={styles.profileInfo}>

```

```

        <Image
            source={require('./assets/images/profile.jpg')} // Replace with
actual image path
            style={styles.profileImage}
        />
        <View style={styles.profileDetails}>
            <Text style={styles.profileName}>{profile.name}</Text>
            <View style={styles.stats}>
                <View style={styles.statItem}>
                    <Image source={require('./assets/images/box.jpg')}
style={styles.statIcon} />
                    <View style={styles.statText}>
                        <Text
style={styles.statValue}>{profile.stats.productSells}</Text>
                        <Text style={styles.statLabel}>Product Sells</Text>
                    </View>
                </View>
                <View style={styles.statItem}>
                    <Image source={require('./assets/images/coin.jpg')}
style={styles.statIcon} />
                    <View style={styles.statText}>
                        <Text
style={styles.statValue}>{profile.stats.earning}</Text>
                        <Text style={styles.statLabel}>Earning</Text>
                    </View>
                </View>
            </View>
            <TouchableOpacity style={styles.settingsButton}>
                <Image source={require('./assets/images/setting.jpg')}
style={styles.settingsIcon} />
            </TouchableOpacity>
        </View>

        <View style={styles.locationContainer}>
            <Text style={styles.locationText}>Location: {profile.location ||
'Fetching location...'}</Text>
        </View>

        <View style={styles.recentOrdersHeader}>
            <Text style={styles.sectionTitle}>Recent Orders</Text>
            <TouchableOpacity>
                <Text style={styles.seeAll}>See All</Text>
            </TouchableOpacity>
        </View>

        <ScrollView style={styles.ordersList}>

```

```

        {profile.orders.map((order, index) => (
          <View key={index} style={styles.orderCard}>
            <Image
              source={require('./assets/images/meal.jpg')} // Replace with
actual image path
              style={styles.orderImage}
            />
            <View style={styles.orderDetails}>
              <Text style={styles.orderDescription}>{order.description}</Text>
              <Text style={styles.orderDate}>{order.date}</Text>
            </View>
            <View style={styles.orderPrice}>
              <Text style={styles.priceText}>${order.price}</Text>
            </View>
          </View>
        ))}
      </ScrollView>
    </View>
  );
};

const App = () => (
  <Provider store={store}>
    <NavigationContainer>
      <Stack.Navigator screenOptions={{ headerShown: false }}>
        <Stack.Screen name="Login" component={LoginScreen} />
        <Stack.Screen name="Signup" component={SignupScreen} />
        <Stack.Screen name="Profile" component={ProfileScreen} />
      </Stack.Navigator>
    </NavigationContainer>
  </Provider>
);

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 20,
    backgroundColor: '#F6F2E9',
  },
  myProfileHeading: {
    fontSize: 24,
    fontWeight: 'bold',
    marginBottom: 20,
    textAlign: 'center',
  },
  profileCard: {
    backgroundColor: '#F9EED4',
    padding: 20,
  },
});

```

```
borderRadius: 20,
flexDirection: 'row',
alignItems: 'center',
justifyContent: 'space-between',
},
profileInfo: {
  flexDirection: 'row',
  alignItems: 'center',
},
profileImage: {
  width: 50,
  height: 50,
  borderRadius: 25,
},
profileDetails: {
  marginLeft: 15,
},
profileName: {
  fontSize: 18,
  fontWeight: 'bold',
},
stats: {
  flexDirection: 'row',
  marginTop: 10,
},
statItem: {
  flexDirection: 'row',
  alignItems: 'center',
  marginRight: 20,
},
statIcon: {
  width: 20,
  height: 20,
  marginRight: 5,
},
statText: {
  flexDirection: 'column',
},
statValue: {
  fontSize: 16,
  fontWeight: 'bold',
},
statLabel: {
  fontSize: 12,
  color: '#888',
},
settingsButton: {
  padding: 10,
```

```
},
settingsIcon: {
  width: 20,
  height: 20,
},
recentOrdersHeader: {
  flexDirection: 'row',
  justifyContent: 'space-between',
  marginTop: 30,
},
sectionTitle: {
  fontSize: 20,
  fontWeight: 'bold',
},
seeAll: {
  fontSize: 16,
  color: '#888',
},
ordersList: {
  marginTop: 20,
},
orderCard: {
  flexDirection: 'row',
  alignItems: 'center',
  backgroundColor: '#000',
  padding: 15,
  borderRadius: 20,
  marginBottom: 10,
},
orderImage: {
  width: 50,
  height: 50,
  borderRadius: 10,
},
orderDetails: {
  marginLeft: 15,
  flex: 1,
},
orderDescription: {
  fontSize: 16,
  color: '#fff',
},
orderDate: {
  fontSize: 12,
  color: '#888',
  marginTop: 5,
},
orderPrice: {
```

```
    marginLeft: 'auto',
  },
  priceText: {
    fontSize: 18,
    fontWeight: 'bold',
    color: '#fff',
  },
  authContainer: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    padding: 20,
    backgroundColor: '#F6F2E9',
  },
  authHeading: {
    fontSize: 32,
    fontWeight: 'bold',
    marginBottom: 20,
    color: '#000',
  },
  input: {
    width: '100%',
    padding: 15,
    marginVertical: 10,
    backgroundColor: '#fff',
    borderRadius: 10,
    borderWidth: 1,
    borderColor: '#ccc',
  },
  authButton: {
    width: '100%',
    padding: 15,
    backgroundColor: '#000',
    borderRadius: 10,
    alignItems: 'center',
    marginVertical: 10,
  },
  authButtonText: {
    color: '#fff',
    fontSize: 16,
    fontWeight: 'bold',
  },
  authLink: {
    color: '#888',
    fontSize: 14,
    marginTop: 10,
    textDecorationLine: 'underline',
  },
}
```



```
locationContainer: {  
  marginTop: 20,  
  backgroundColor: '#F9EED4',  
  padding: 15,  
  borderRadius: 10,  
},  
locationText: {  
  fontSize: 16,  
  color: '#000',  
},  
});
```

```
export default App;
```


OUTPUT:

Location is displayed on the profile screen as it can be seen in the screenshot given below:


Profile - Personal - Microsoft...


https://snack-web-player.s...

My Profile



Zarnab Imran


**55.850**
Product Sells

**300bil+**
Earning

Location: 33.810578, 72.34986


Recent Orders

See All




Ice Cream
1st, Monday 10.00

\$150.00



Beef Burger
2nd, Tuesday 11.00

\$550.00



French Fries
5th, Friday 10.00

\$200.00

About 13 results

Higher Education Commission



Who can see school results? ⓘ



Hi ZARNAB,

Find answers about school—personalized just for you.



Recent messages

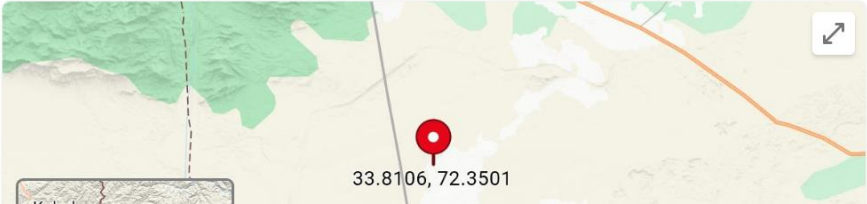


My profile



My calendar

[Send feedback](#)



Punjab -, Pakistan

bing.com/maps