Comsats University, Islamabad, Attock Campus

Department of Computer Science

Assignment: 04

Subject: Mobile Application Development

Program: BSSE-6



Submitted To: Muhammad Kamran

Submitted By: Laveeza Bukhari

Registration Number: SP22-BSE-002

Part 1:

Code:

Store.js:

import { createStore, combineReducers } from 'redux';

*// Initial state*

const initialState = {

  signup: {

    username: '',

    email: '',

    phone: '',

    password: '',

  },

  login: {

    email: '',

    password: '',

  },

};

*// Reducers*

const signupReducer = (state = initialState.signup, action) => {

  switch (action.type) {

    case 'SET\_SIGNUP\_FIELD':

      return { ...state, [action.payload.field]: action.payload.value };

    default:

      return state;

  }

};

const loginReducer = (state = initialState.login, action) => {

  switch (action.type) {

    case 'SET\_LOGIN\_FIELD':

      return { ...state, [action.payload.field]: action.payload.value };

    default:

      return state;

  }

};

const rootReducer = combineReducers({

  signup: signupReducer,

  login: loginReducer,

});

*// Create store*

export const store = createStore(rootReducer);

App.js:

import React from 'react';

import { Provider } from 'react-redux';

import { store } from './store';

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

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

import SignupScreen from './SignUp';

import LoginScreen from './Login';

import ProfileScreen from './Profile';

const Stack = createStackNavigator();

export default function App() {

  return (

    <Provider store={store}>

      <NavigationContainer>

        <Stack.Navigator initialRouteName="Signup">

          <Stack.Screen name="Signup" component={SignupScreen} options={{ title: 'Sign Up' }} />

          <Stack.Screen name="Login" component={LoginScreen} options={{ title: 'Login' }} />

          <Stack.Screen

            name="Profile"

            component={ProfileScreen}

            options={{ headerShown: false }}

          />

        </Stack.Navigator>

      </NavigationContainer>

    </Provider>

  );

}

Signup.js:

import React from 'react';

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

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

export default function SignupScreen({ navigation }) {

  const dispatch = useDispatch();

  const { username, email, phone, password } = useSelector(state => state.signup);

  const handleInputChange = (field, value) => {

    dispatch({ type: 'SET\_SIGNUP\_FIELD', payload: { field, value } });

  };

  const validateAndProceed = () => {

    const usernameRegex = /^[A-Za-z]+$/;

    const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

    const phoneRegex = /^\+92-3\d{2}-\d{7}$/;

    if (!usernameRegex.test(username)) {

      Alert.alert('Error', 'Username must contain alphabets only.');

      return;

    }

    if (!emailRegex.test(email)) {

      Alert.alert('Error', 'Invalid email format.');

      return;

    }

    if (!phoneRegex.test(phone)) {

      Alert.alert('Error', 'Phone number must match +92-3xx-xxxxxxx format.');

      return;

    }

    if (password.length < 6) {

      Alert.alert('Error', 'Password must be at least 6 characters long.');

      return;

    }

    navigation.navigate('Login');

  };

  return (

    <View style={styles.container}>

      <Text style={styles.title}>Sign Up</Text>

      <TextInput

        style={styles.input}

        placeholder="Username"

        value={username}

        onChangeText={value => handleInputChange('username', value)}

      />

      <TextInput

        style={styles.input}

        placeholder="Email"

        value={email}

        keyboardType="email-address"

        onChangeText={value => handleInputChange('email', value)}

      />

      <TextInput

        style={styles.input}

        placeholder="Phone (+92-3xx-xxxxxxx)"

        value={phone}

        keyboardType="phone-pad"

        onChangeText={value => handleInputChange('phone', value)}

      />

      <TextInput

        style={styles.input}

        placeholder="Password"

        value={password}

        secureTextEntry

        onChangeText={value => handleInputChange('password', value)}

      />

      <TouchableOpacity style={styles.button} onPress={validateAndProceed}>

        <Text style={styles.buttonText}>Sign Up</Text>

      </TouchableOpacity>

    </View>

  );

}const styles = StyleSheet.create({

  container: {

    flex: 1,

    padding: 20,

    justifyContent: 'center',

  },

  title: {

    fontSize: 24,

    fontWeight: 'bold',

    textAlign: 'center',

    marginBottom: 20,

  },

  input: {

    borderWidth: 1,

    borderColor: '#ccc',

    borderRadius: 10,

    padding: 10,

    marginBottom: 15,

  },

  button: {

    backgroundColor: '#007BFF',

    padding: 15,

    borderRadius: 10,

    alignItems: 'center',

  },

  buttonText: {

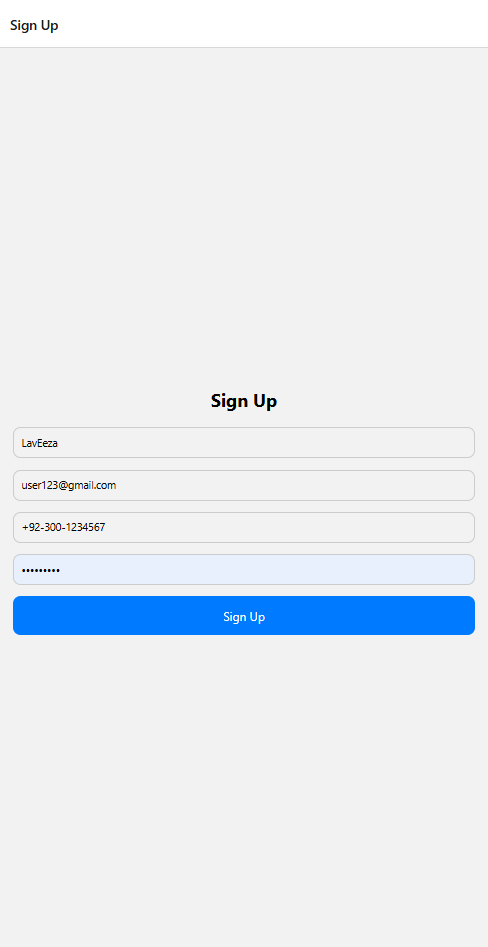
    color: '#fff',

    fontSize: 16,

  },

});

Output:



Login.js:

 import React from 'react';

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

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

export default function LoginScreen({ navigation }) {

  const dispatch = useDispatch();

  const { email, password } = useSelector(state => state.login);

  const handleInputChange = (field, value) => {

    dispatch({ type: 'SET\_LOGIN\_FIELD', payload: { field, value } });

  };

  const validateAndProceed = () => {

    const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

    if (!emailRegex.test(email)) {

      Alert.alert('Error', 'Invalid email format.');

      return;

    }

    if (password.length < 6) {

      Alert.alert('Error', 'Password must be at least 6 characters long.');

      return;

    }

    navigation.navigate('Profile');

  };

  return (

    <View style={styles.container}>

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

      <TextInput

        style={styles.input}

        placeholder="Email"

        value={email}

        keyboardType="email-address"

        onChangeText={value => handleInputChange('email', value)}

      />

      <TextInput

        style={styles.input}

        placeholder="Password"

        value={password}

        secureTextEntry

        onChangeText={value => handleInputChange('password', value)}

      />

      <TouchableOpacity style={styles.button} onPress={validateAndProceed}>

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

      </TouchableOpacity>

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    padding: 20,

    justifyContent: 'center',

  },

  title: {

    fontSize: 24,

    fontWeight: 'bold',

    textAlign: 'center',

    marginBottom: 20,

  },

  input: {

    borderWidth: 1,

    borderColor: '#ccc',

    borderRadius: 10,

    padding: 10,

    marginBottom: 15,

  },

  button: {

    backgroundColor: '#007BFF',

    padding: 15,

    borderRadius: 10,

  },

  buttonText: {

    color: '#fff',

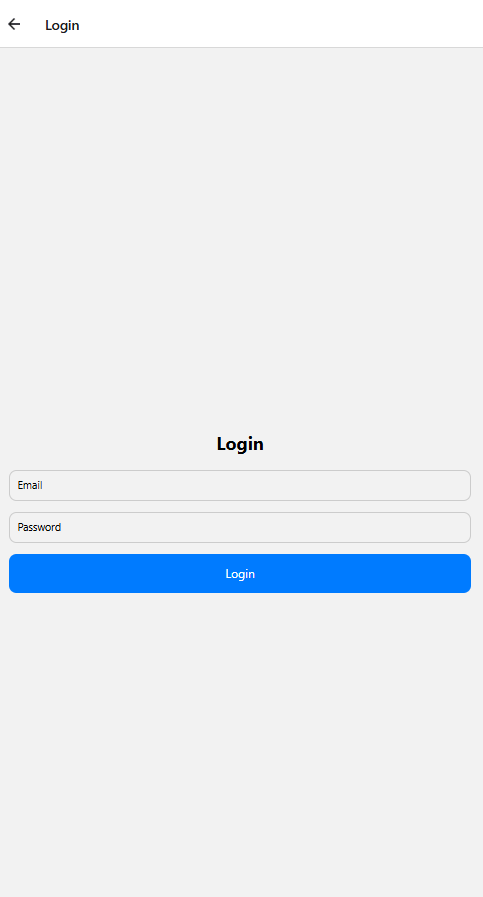
    fontSize: 16,

    textAlign: 'center',

  },

});

Output:



Profile.js:

import React from 'react';

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

const orders = [

{

id: '1',

items: 'Beef Burger, Ice Cream, Berries',

date: '21, Thu 10:00',

price: '$38.15',

image: require('./assets/d4.jpg')*// Replace with actual image paths*

},

{

id: '2',

items: 'Beef Burger, Pepperoni Moo, Cheese Bust',

date: '15, Fri 18:00',

price: '$38.15',

image: require('./assets/d5.jpg'), *// Replace with actual image paths*

},

{

id: '3',

items: 'Beef Burger, Ketchup, French Fries',

date: '11, Mon 10:00',

price: '$30.50',

image: require('./assets/d6.jpg'), *// Replace with actual image paths*

}

];

export default function ProfileScreen() {

*// Function to handle edit button press*

const handleEditPress = () => {

Alert.alert("Edit Profile", "This is where you can edit your profile.");

*// You can add navigation or modal logic here*

};

return (

<View style={styles.container}>

{*/\* Profile Section \*/*}

<View style={styles.profileContainer}>

<View style={styles.profileHeader}>

<Image

source={require('./assets/d3.jpg')} *// Replace with actual profile image*

style={styles.profilePic}

/>

<Text style={styles.profileName}>L^2 cafe</Text>

</View>

<View style={styles.earningsContainer}>

<View style={styles.profileStats}>

<View style={styles.statsItem}>

<Text style={styles.statsValue}>14,850</Text>

<Text style={styles.statsLabel}>Product Sells</Text>

</View>

<View style={styles.statsItem}>

<Text style={styles.statsValue}>30bil+</Text>

<Text style={styles.statsLabel}>Earnings</Text>

</View>

</View>

</View>

<TouchableOpacity style={styles.editButton} onPress={handleEditPress}>

<Text style={styles.editButtonText}>✎</Text>

</TouchableOpacity>

</View>

{*/\* Recent Orders Section \*/*}

<View style={styles.recentOrdersContainer}>

<View style={styles.ordersHeader}>

<Text style={styles.ordersTitle}>Recent Orders</Text>

<TouchableOpacity>

<Text style={styles.seeAll}>See All</Text>

</TouchableOpacity>

</View>

<FlatList

data={orders}

keyExtractor={item => item.id}

renderItem={({ item }) => (

<View style={styles.orderContainer}>

<View style={styles.orderItem}>

{*/\* Div for Order Details (image and items) \*/*}

<View style={styles.orderDetails}>

<Image source={item.image} style={styles.orderImage} />

<View>

<Text style={styles.orderItems}>{item.items}</Text>

<Text style={styles.orderDate}>{item.date}</Text>

</View>

</View>

{*/\* Div for Order Price \*/*}

<View style={styles.orderPriceContainer}>

<Text style={styles.orderPrice}>{item.price}</Text>

</View>

</View>

</View>

)}

/>

</View>

</View>

);

}

const styles = StyleSheet.create({

container: {

flex: 1,

backgroundColor: '#f9f9f9',

padding: 20,

paddingTop: 50,

},

profileContainer: {

marginBottom: 30,

backgroundColor: '#fff',

padding: 20,

borderRadius: 15,

shadowColor: '#000',

shadowOpacity: 0.1,

shadowOffset: { width: 0, height: 2 },

shadowRadius: 10,

elevation: 5,

},

profileHeader: {

alignItems: 'center',

marginBottom: 10,

},

profilePic: {

width: 80,

height: 80,

borderRadius: 40,

},

profileName: {

fontSize: 20,

fontWeight: 'bold',

marginTop: 10, *// Added margin for spacing*

},

earningsContainer: {

borderWidth: 1, *// Set border width*

borderColor: 'black', *// Set border color to black*

padding: 10,

borderRadius: 10,

marginTop: 10, *// Added margin for spacing*

},

profileStats: {

flexDirection: 'row',

justifyContent: 'space-between',

},

statsItem: {

marginRight: 20,

},

statsValue: {

fontSize: 18,

fontWeight: 'bold',

},

statsLabel: {

fontSize: 12,

color: '#666',

},

editButton: {

position: 'absolute',

top: 10,

right: 0,

backgroundColor: '#e0e0e0',

padding: 5,

borderRadius: 20,

},

editButtonText: {

fontSize: 18,

},

recentOrdersContainer: {

flex: 1,

},

ordersHeader: {

flexDirection: 'row',

justifyContent: 'space-between',

marginBottom: 20,

},

ordersTitle: {

fontSize: 18,

fontWeight: 'bold',

},

seeAll: {

color: '#007BFF',

fontSize: 14,

},

orderContainer: {

backgroundColor: '#fff',

padding: 15,

borderRadius: 10,

shadowColor: '#000',

shadowOpacity: 0.05,

shadowOffset: { width: 0, height: 2 },

shadowRadius: 5,

elevation: 3,

marginBottom: 15,

borderWidth: 1, *// Add border width*

borderColor: 'black', *// Set border color to black*

},

orderItem: {

flexDirection: 'row',

justifyContent: 'space-between',

alignItems: 'center',

},

orderDetails: {

flexDirection: 'row',

alignItems: 'center',

},

orderImage: {

width: 40,

height: 40,

marginRight: 10,

},

orderItems: {

fontSize: 14,

color: '#333',

},

orderDate: {

fontSize: 12,

color: '#999',

},

orderPriceContainer: {

justifyContent: 'center',

alignItems: 'flex-end',

},

orderPrice: {

fontSize: 14,

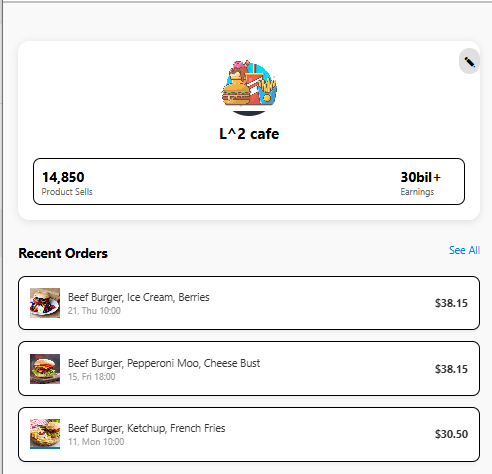
fontWeight: 'bold',

color: '#333',

},

});

Output:



LINK:

https://snack.expo.dev/@lbukhari30/af0d44

PART 2:

Code:

App.js:

import React, { useState } from 'react';

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

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

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

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

const FIREBASE\_API\_ENDPOINT = 'https://laveeza-8f472-default-rtdb.firebaseio.com';

*// Set up navigation stack*

const Stack = createStackNavigator();

const App = ({ navigation }) => {

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

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

  const [confirmPassword, setConfirmPassword] = useState('');

  const [isLogin, setIsLogin] = useState(false);

  const [userData, setUserData] = useState(null);

  const register = async () => {

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

      Alert.alert('Error', 'All fields are required.');

      return;

    }

    if (password !== confirmPassword) {

      Alert.alert('Error', 'Passwords do not match.');

      return;

    }

    try {

      await AsyncStorage.setItem('user', JSON.stringify({ email, password }));

      Alert.alert('Success', 'User registered successfully!');

    } catch (error) {

      Alert.alert('Error', 'An error occurred during registration.');

    }

  };

  const login = async () => {

    if (!email || !password) {

      Alert.alert('Error', 'Both fields are required.');

      return;

    }

    try {

      const storedUser = await AsyncStorage.getItem('user');

      const user = storedUser ? JSON.parse(storedUser) : null;

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

        Alert.alert('Success', 'Login successful!');

*// Navigate to Profile screen after login*

        navigation.navigate('Profile', { email });

      } else {

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

      }

    } catch (error) {

      Alert.alert('Error', 'An error occurred during login.');

    }

  };

  const retrieveData = async () => {

    try {

      const response = await fetch(`${FIREBASE\_API\_ENDPOINT}/users.json`);

      const data = await response.json();

      const userArray = Object.values(data);

      setUserData(userArray);

    } catch (error) {

      Alert.alert('Error', 'Failed to retrieve data.');

    }

  };

  return (

    <ScrollView contentContainerStyle={styles.container}>

      <View style={styles.header}>

        <Image

          source={{

            uri: 'https://www.bing.com/images/search?view=detailV2&ccid=HV0w4p9U&id=033B8983ABEBA64C1BA26A9CE0EE48FF4321D2C7&thid=OIP.HV0w4p9UfSW0iI36Xs41ggHaD7&mediaurl=https%3a%2f%2fstatic.vecteezy.com%2fsystem%2fresources%2fpreviews%2f025%2f337%2f736%2foriginal%2fpeople-avatar-smiling-human-circle-portrait-female-and-male-person-round-avatars-flat-icon-illustration-collection-vector.jpg&cdnurl=https%3a%2f%2fth.bing.com%2fth%2fid%2fR.1d5d30e29f547d25b4888dfa5ece3582%3frik%3dx9IhQ%252f9I7uCcag%26pid%3dImgRaw%26r%3d0&exph=1017&expw=1920&q=Human+Avatar&simid=607996456152694274&FORM=IRPRST&ck=EE9818E2145D2B61445BC28F406E4BA8&selectedIndex=8&itb=0',

          }}

          style={styles.headerIcon}

        />

        <Text style={styles.headerText}>

          {isLogin ? 'Welcome Back!' : 'Create Your Account'}

        </Text>

      </View>

      <View style={styles.formContainer}>

        <TextInput

          style={styles.input}

          placeholder="Email"

          value={email}

          onChangeText={setEmail}

          keyboardType="email-address"

        />

        <TextInput

          style={styles.input}

          placeholder="Password"

          value={password}

          secureTextEntry

          onChangeText={setPassword}

        />

        {!isLogin && (

          <TextInput

            style={styles.input}

            placeholder="Confirm Password"

            value={confirmPassword}

            secureTextEntry

            onChangeText={setConfirmPassword}

          />

        )}

        <TouchableOpacity

          style={[styles.button, styles.primaryButton]}

          onPress={isLogin ? login : register}

        >

          <Text style={styles.buttonText}>{isLogin ? 'Login' : 'Register'}</Text>

        </TouchableOpacity>

        <TouchableOpacity onPress={() => setIsLogin(!isLogin)} style={styles.switchButton}>

          <Text style={styles.switchText}>

            {isLogin ? "Don't have an account? Register" : 'Already have an account? Login'}

          </Text>

        </TouchableOpacity>

        {*/\* Button to fetch user data from Firebase \*/*}

        <TouchableOpacity onPress={retrieveData} style={[styles.button, styles.secondaryButton]}>

          <Text style={styles.buttonText}>Retrieve Data</Text>

        </TouchableOpacity>

        {*/\* Display fetched data \*/*}

        {userData && (

          <View style={styles.dataContainer}>

            <Text style={styles.dataText}>Fetched Users:</Text>

            {userData.map((user, index) => (

              <Text key={index} style={styles.userText}>

                {user.email}

              </Text>

            ))}

          </View>

        )}

      </View>

    </ScrollView>

  );

};

const ProfileScreen = ({ route }) => {

  const { email } = route.params; *// Receive email passed from the login screen*

  return (

    <View style={styles.container}>

      <View style={styles.header}>

        <Image

          source={{

            uri: 'https://www.bing.com/images/search?view=detailV2&ccid=HV0w4p9U&id=033B8983ABEBA64C1BA26A9CE0EE48FF4321D2C7&thid=OIP.HV0w4p9UfSW0iI36Xs41ggHaD7&mediaurl=https%3a%2f%2fstatic.vecteezy.com%2fsystem%2fresources%2fpreviews%2f025%2f337%2f736%2foriginal%2fpeople-avatar-smiling-human-circle-portrait-female-and-male-person-round-avatars-flat-icon-illustration-collection-vector.jpg&cdnurl=https%3a%2f%2fth.bing.com%2fth%2fid%2fR.1d5d30e29f547d25b4888dfa5ece3582%3frik%3dx9IhQ%252f9I7uCcag%26pid%3dImgRaw%26r%3d0&exph=1017&expw=1920&q=Human+Avatar&simid=607996456152694274&FORM=IRPRST&ck=EE9818E2145D2B61445BC28F406E4BA8&selectedIndex=8&itb=0',

          }}

          style={styles.headerIcon}

        />

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

      </View>

      <View style={styles.formContainer}>

        <Text style={styles.dataText}>User Email: {email}</Text>

      </View>

    </View>

  );

};

*// Main App Component with Navigation*

export default function Main() {

  return (

    <NavigationContainer>

      <Stack.Navigator initialRouteName="App">

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

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

      </Stack.Navigator>

    </NavigationContainer>

  );

}

const styles = StyleSheet.create({

  container: {

    flexGrow: 1,

    justifyContent: 'center',

    alignItems: 'center',

    backgroundColor: '#f9f9f9',

    paddingVertical: 20,

  },

  header: {

    width: '100%',

    padding: 20,

    flexDirection: 'row',

    alignItems: 'center',

    justifyContent: 'center',

    backgroundColor: '#4CAF50',

    borderBottomLeftRadius: 50,

    borderBottomRightRadius: 50,

    elevation: 5,

  },

  headerIcon: {

    width: 40,

    height: 40,

    marginRight: 10,

    borderRadius: 20,

  },

  headerText: {

    fontSize: 24,

    color: '#fff',

    fontWeight: 'bold',

  },

  formContainer: {

    width: '90%',

    marginTop: 20,

  },

  input: {

    height: 50,

    borderColor: '#ccc',

    borderWidth: 1,

    marginBottom: 15,

    paddingHorizontal: 10,

    borderRadius: 8,

    backgroundColor: '#fff',

  },

  button: {

    height: 50,

    justifyContent: 'center',

    alignItems: 'center',

    borderRadius: 8,

    marginBottom: 10,

    width: '100%',

  },

  primaryButton: {

    backgroundColor: '#4CAF50',

  },

  secondaryButton: {

    backgroundColor: '#888',

  },

  buttonText: {

    color: '#fff',

    fontSize: 16,

  },

  switchButton: {

    alignItems: 'center',

  },

  switchText: {

    color: '#4CAF50',

    fontSize: 16,

  },

  dataContainer: {

    marginTop: 20,

  },

  dataText: {

    fontSize: 16,

  },

  userText: {

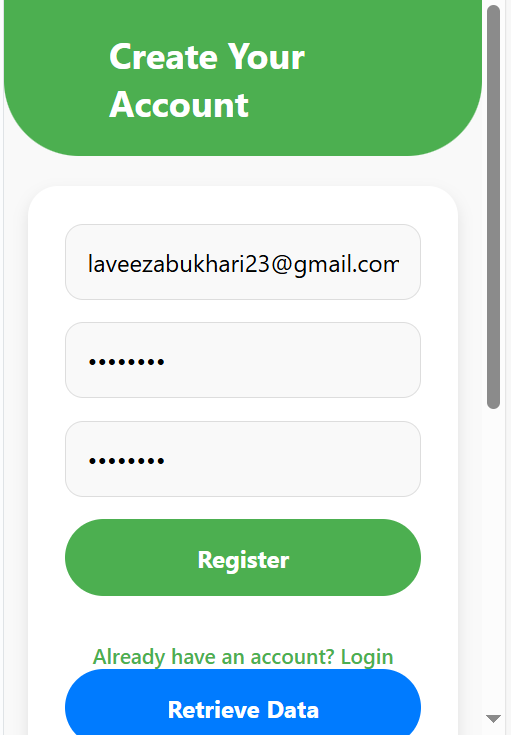
    fontSize: 14,

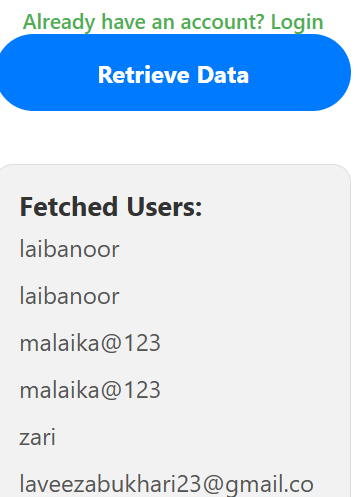
    color: '#555',

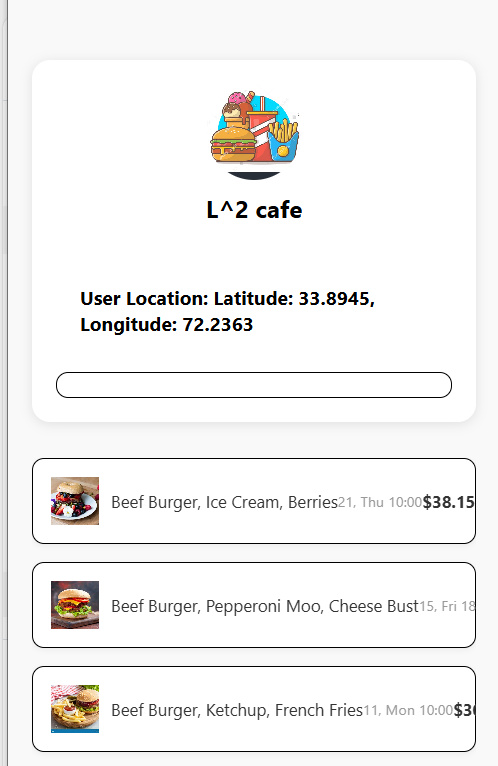
  },

});

Output:







PART 3:

Code:

LocationDisplay.js:

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

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

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

const LocationDisplay = () => {

  const [location, setLocation] = useState('');

  useEffect(() => {

    const getLocation = async () => {

      const storedLocation = await AsyncStorage.getItem('userLocation');

      if (storedLocation) {

        setLocation(storedLocation);

      } else {

        setLocation('Location not available');

      }

    };

    getLocation();

  }, []);

  return (

    <View style={styles.container}>

      <Text style={styles.text}>User Location: {location}</Text>

    </View>

  );

};

const styles = StyleSheet.create({

  container: {

    padding: 20,

    backgroundColor: '#fff',

    borderRadius: 10,

    marginTop: 20,

    alignItems: 'center',

  },

  text: {

    fontSize: 16,

    fontWeight: 'bold',

  },

});

export default LocationDisplay;

App.js:

import React, { useEffect } from 'react';

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

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

import Geolocation from 'react-native-geolocation-service';

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

export default function LoginScreen({ navigation }) {

  const dispatch = useDispatch();

  const { email, password } = useSelector(state => state.login);

  useEffect(() => {

*// Check if location is already saved*

    const getLocation = async () => {

      const storedLocation = await AsyncStorage.getItem('userLocation');

      if (storedLocation) {

        console.log('Stored Location: ', storedLocation);

      }

    };

    getLocation();

  }, []);

  const handleInputChange = (field, value) => {

    dispatch({ type: 'SET\_LOGIN\_FIELD', payload: { field, value } });

  };

  const validateAndProceed = () => {

    const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

    if (!emailRegex.test(email)) {

      Alert.alert('Error', 'Invalid email format.');

      return;

    }

    if (password.length < 6) {

      Alert.alert('Error', 'Password must be at least 6 characters long.');

      return;

    }

*// Fetch user location*

    Geolocation.getCurrentPosition(

      async (position) => {

        const { latitude, longitude } = position.coords;

        const location = `Latitude: ${latitude}, Longitude: ${longitude}`;

        await AsyncStorage.setItem('userLocation', location);

        console.log('User Location:', location);

*// Navigate to the Profile screen*

        navigation.navigate('Profile');

      },

      (error) => {

        console.log('Location Error:', error);

        Alert.alert('Error', 'Unable to fetch location.');

      },

      { enableHighAccuracy: true, timeout: 15000, maximumAge: 10000 }

    );

  };

  return (

    <View style={styles.container}>

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

      <TextInput

        style={styles.input}

        placeholder="Email"

        value={email}

        keyboardType="email-address"

        onChangeText={value => handleInputChange('email', value)}

      />

      <TextInput

        style={styles.input}

        placeholder="Password"

        value={password}

        secureTextEntry

        onChangeText={value => handleInputChange('password', value)}

      />

      <TouchableOpacity style={styles.button} onPress={validateAndProceed}>

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

      </TouchableOpacity>

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    padding: 20,

    justifyContent: 'center',

  },

  title: {

    fontSize: 24,

    fontWeight: 'bold',

    textAlign: 'center',

    marginBottom: 20,

  },

  input: {

    borderWidth: 1,

    borderColor: '#ccc',

    borderRadius: 10,

    padding: 10,

    marginBottom: 15,

  },

  button: {

    backgroundColor: '#007BFF',

    padding: 15,

    borderRadius: 10,

  },

  buttonText: {

    color: '#fff',

    fontSize: 16,

    textAlign: 'center',

  },

});

Signup.js:

import React from 'react';

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

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

export default function SignupScreen({ navigation }) {

  const dispatch = useDispatch();

  const { username, email, phone, password } = useSelector(state => state.signup);

  const handleInputChange = (field, value) => {

    dispatch({ type: 'SET\_SIGNUP\_FIELD', payload: { field, value } });

  };

  const validateAndProceed = () => {

    const usernameRegex = /^[A-Za-z]+$/;

    const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

    const phoneRegex = /^\+92-3\d{2}-\d{7}$/;

    if (!usernameRegex.test(username)) {

      Alert.alert('Error', 'Username must contain alphabets only.');

      return;

    }

    if (!emailRegex.test(email)) {

      Alert.alert('Error', 'Invalid email format.');

      return;

    }

    if (!phoneRegex.test(phone)) {

      Alert.alert('Error', 'Phone number must match +92-3xx-xxxxxxx format.');

      return;

    }

    if (password.length < 6) {

      Alert.alert('Error', 'Password must be at least 6 characters long.');

      return;

    }

    navigation.navigate('Login');

  };

  return (

    <View style={styles.container}>

      <Text style={styles.title}>Sign Up</Text>

      <TextInput

        style={styles.input}

        placeholder="Username"

        value={username}

        onChangeText={value => handleInputChange('username', value)}

      />

      <TextInput

        style={styles.input}

        placeholder="Email"

        value={email}

        keyboardType="email-address"

        onChangeText={value => handleInputChange('email', value)}

      />

      <TextInput

        style={styles.input}

        placeholder="Phone (+92-3xx-xxxxxxx)"

        value={phone}

        keyboardType="phone-pad"

        onChangeText={value => handleInputChange('phone', value)}

      />

      <TextInput

        style={styles.input}

        placeholder="Password"

        value={password}

        secureTextEntry

        onChangeText={value => handleInputChange('password', value)}

      />

      <TouchableOpacity style={styles.button} onPress={validateAndProceed}>

        <Text style={styles.buttonText}>Sign Up</Text>

      </TouchableOpacity>

    </View>

  );

}const styles = StyleSheet.create({

  container: {

    flex: 1,

    padding: 20,

    justifyContent: 'center',

  },

  title: {

    fontSize: 24,

    fontWeight: 'bold',

    textAlign: 'center',

    marginBottom: 20,

  },

  input: {

    borderWidth: 1,

    borderColor: '#ccc',

    borderRadius: 10,

    padding: 10,

    marginBottom: 15,

  },

  button: {

    backgroundColor: '#007BFF',

    padding: 15,

    borderRadius: 10,

    alignItems: 'center',

  },

  buttonText: {

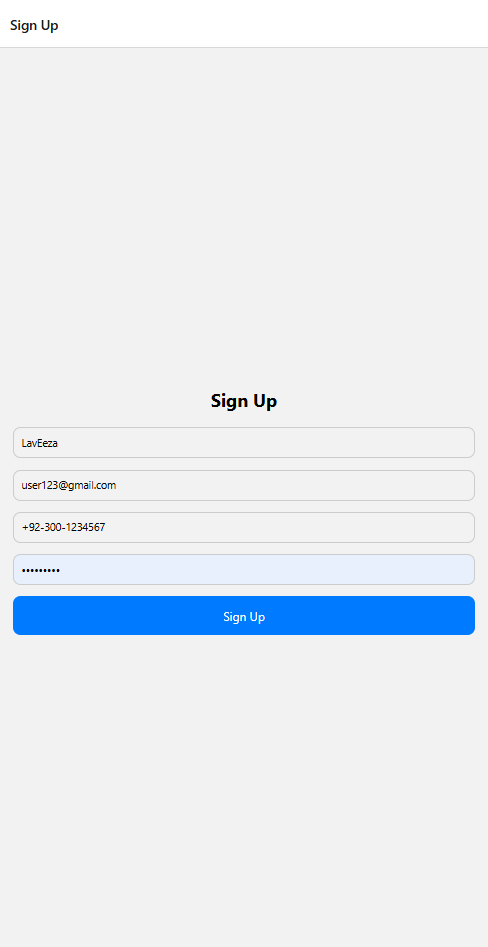
    color: '#fff',

    fontSize: 16,

  },

});

Output:



Login.js:

 import React from 'react';

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

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

export default function LoginScreen({ navigation }) {

  const dispatch = useDispatch();

  const { email, password } = useSelector(state => state.login);

  const handleInputChange = (field, value) => {

    dispatch({ type: 'SET\_LOGIN\_FIELD', payload: { field, value } });

  };

  const validateAndProceed = () => {

    const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

    if (!emailRegex.test(email)) {

      Alert.alert('Error', 'Invalid email format.');

      return;

    }

    if (password.length < 6) {

      Alert.alert('Error', 'Password must be at least 6 characters long.');

      return;

    }

    navigation.navigate('Profile');

  };

  return (

    <View style={styles.container}>

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

      <TextInput

        style={styles.input}

        placeholder="Email"

        value={email}

        keyboardType="email-address"

        onChangeText={value => handleInputChange('email', value)}

      />

      <TextInput

        style={styles.input}

        placeholder="Password"

        value={password}

        secureTextEntry

        onChangeText={value => handleInputChange('password', value)}

      />

      <TouchableOpacity style={styles.button} onPress={validateAndProceed}>

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

      </TouchableOpacity>

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    padding: 20,

    justifyContent: 'center',

  },

  title: {

    fontSize: 24,

    fontWeight: 'bold',

    textAlign: 'center',

    marginBottom: 20,

  },

  input: {

    borderWidth: 1,

    borderColor: '#ccc',

    borderRadius: 10,

    padding: 10,

    marginBottom: 15,

  },

  button: {

    backgroundColor: '#007BFF',

    padding: 15,

    borderRadius: 10,

  },

  buttonText: {

    color: '#fff',

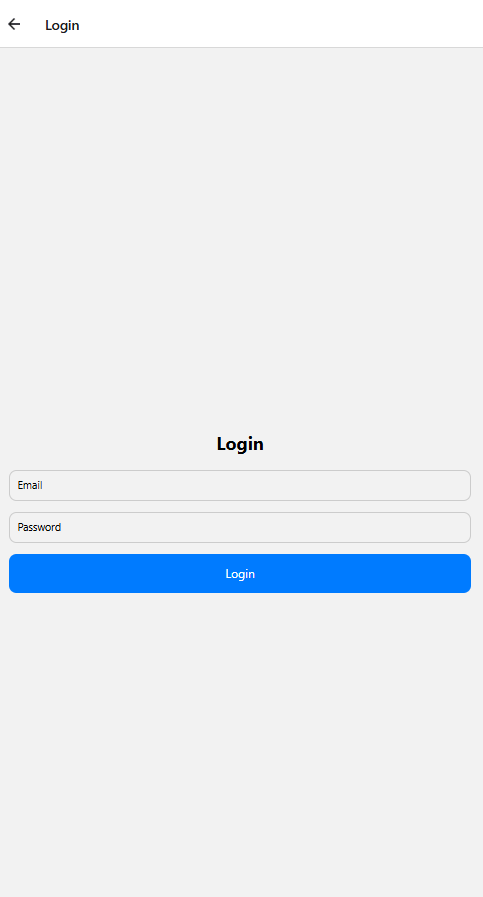
    fontSize: 16,

    textAlign: 'center',

  },

});

Output:



Profile.js:

import React from 'react';

import { StyleSheet, View, Text, FlatList, TouchableOpacity, Image } from 'react-native';

import LocationDisplay from './LocationDisplay';

const orders = [

  {

    id: '1',

    items: 'Beef Burger, Ice Cream, Berries',

    date: '21, Thu 10:00',

    price: '$38.15',

    image: require('./assets/d4.jpg'), *// Replace with actual image paths*

  },

  {

    id: '2',

    items: 'Beef Burger, Pepperoni Moo, Cheese Bust',

    date: '15, Fri 18:00',

    price: '$38.15',

    image: require('./assets/d5.jpg'), *// Replace with actual image paths*

  },

  {

    id: '3',

    items: 'Beef Burger, Ketchup, French Fries',

    date: '11, Mon 10:00',

    price: '$30.50',

    image: require('./assets/d6.jpg'), *// Replace with actual image paths*

  },

];

export default function ProfileScreen() {

  return (

    <View style={styles.container}>

      {*/\* Profile Section \*/*}

      <View style={styles.profileContainer}>

        <View style={styles.profileHeader}>

          <Image

            source={require('./assets/d3.jpg')} *// Replace with actual profile image*

            style={styles.profilePic}

          />

          <Text style={styles.profileName}>L^2 cafe</Text>

        </View>

        <LocationDisplay />

        <View style={styles.earningsContainer}>

          {*/\* Earnings and Stats \*/*}

        </View>

      </View>

      {*/\* Recent Orders Section \*/*}

      <FlatList

        data={orders}

        keyExtractor={item => item.id}

        renderItem={({ item }) => (

          <View style={styles.orderContainer}>

            <Text>{item.items}</Text>

          </View>

        )}

      />

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#f9f9f9',

    padding: 20,

    paddingTop: 50,

  },

  profileContainer: {

    marginBottom: 30,

    backgroundColor: '#fff',

    padding: 20,

    borderRadius: 15,

    shadowColor: '#000',

    shadowOpacity: 0.1,

    shadowOffset: { width: 0, height: 2 },

    shadowRadius: 10,

    elevation: 5,

  },

  profileHeader: {

    alignItems: 'center',

    marginBottom: 10,

  },

  profilePic: {

    width: 80,

    height: 80,

    borderRadius: 40,

  },

  profileName: {

    fontSize: 20,

    fontWeight: 'bold',

    marginTop: 10,  },

  earningsContainer: {

    borderWidth: 1,    borderColor: 'black',

    padding: 10,

    borderRadius: 10,

    marginTop: 10,

  },

  orderContainer: {

    backgroundColor: '#fff',

    padding: 15,

    borderRadius: 10,

    marginBottom: 15,

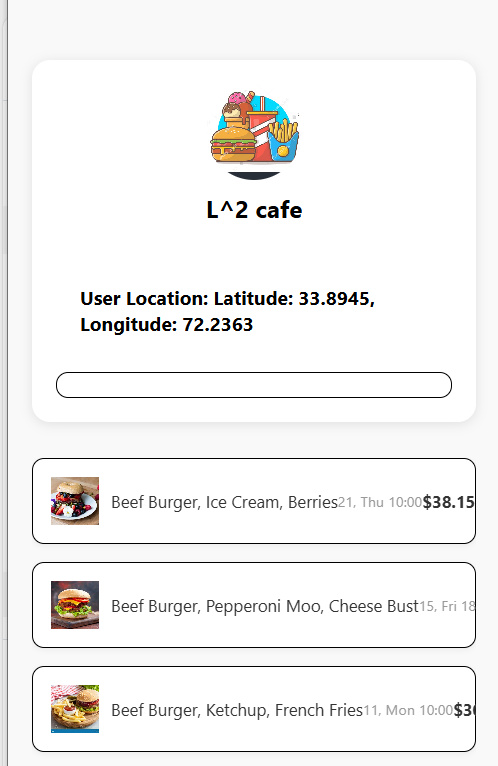
    borderWidth: 1, *// Add border width*

    borderColor: 'black', *// Set border color to black*

  },

});

Output:



LINK:

<https://snack.expo.dev/@lbukhari30/as3-3>