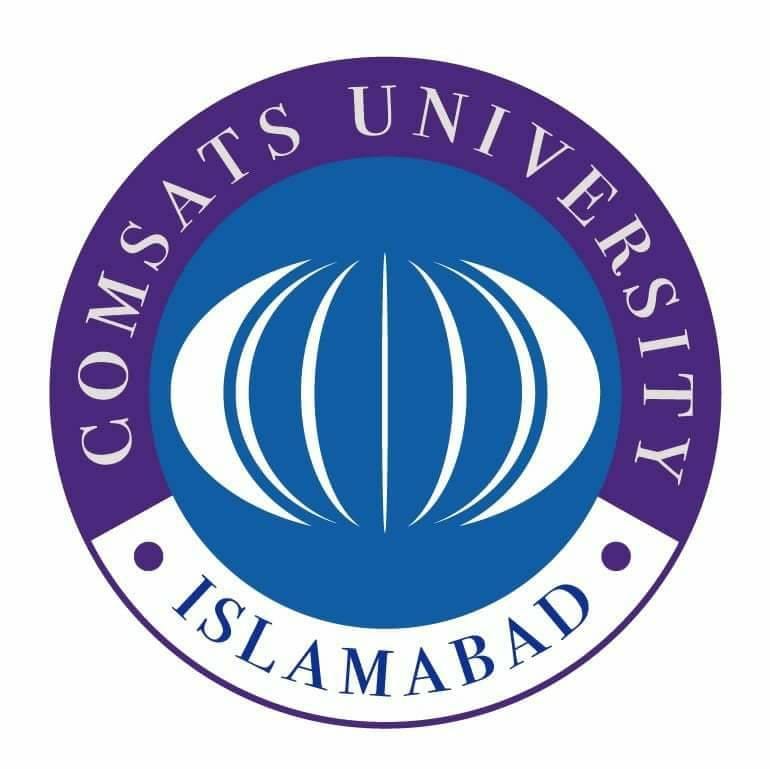
**COMSATS UNIVERSITY ISLAMABAD**

**ATTOCK CAMPUS**



**DEPARTMENT OF COMPUTER SCIENCES**

**SUBMITTED TO:**

**Sir Muhammad Kamran**

**SUBMITTED BY:**

**Malaika**

**(SP22-BSE-003)**

**COURSE NAME:**

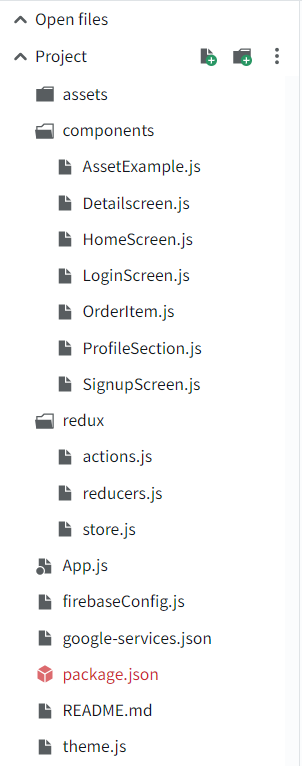
**Mobile Application Development.**

**DATED:**

**24-DEC-2024**

**Assignment:**

**04.**

****

import React, { useState } from 'react';

import {

  View,

  Text,

  StyleSheet,

  TextInput,

  TouchableOpacity,

  Alert,

} from 'react-native';

import { createUserWithEmailAndPassword } from 'firebase/auth';

import { ref, set } from 'firebase/database';

import { useDispatch } from 'react-redux';

import { setUser } from '../redux/actions';

import \* as SecureStore from 'expo-secure-store';

import { auth, database } from '../firebase';

import { MyTheme } from '../theme';

const SignupScreen = ({ navigation }) => {

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

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

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

  const dispatch = useDispatch();

  const handleSignup = async () => {

    if (!email || !password) {

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

      return;

    }

    if (password !== confirmPassword) {

      Alert.alert('Passwords do not match!');

      return;

    }

    try {

      const userCredential = await createUserWithEmailAndPassword(auth, email, password);

      const user = userCredential.user;

*// Store user data in Firebase Realtime Database*

      set(ref(database, 'users/' + user.uid), {

        email: user.email,

        uid: user.uid,

      });

*// Store user data securely using SecureStore*

      await SecureStore.setItemAsync('user', JSON.stringify({ email: user.email, uid: user.uid }));

      dispatch(setUser({ email: user.email, uid: user.uid }));

      Alert.alert('Registration Successful', `Welcome, ${user.email}!`);

      navigation.navigate('Login');

    } catch (error) {

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

    }

  };

  return (

    <View style={styles.container}>

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

      <TextInput

        style={styles.input}

        placeholder="Email"

        placeholderTextColor="#BDBDBD"

        value={email}

        onChangeText={setEmail}

        keyboardType="email-address"

      />

      <TextInput

        style={styles.input}

        placeholder="Password"

        placeholderTextColor="#BDBDBD"

        value={password}

        onChangeText={setPassword}

        secureTextEntry

      />

      <TextInput

        style={styles.input}

        placeholder="Confirm Password"

        placeholderTextColor="#BDBDBD"

        value={confirmPassword}

        onChangeText={setConfirmPassword}

        secureTextEntry

      />

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

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

      </TouchableOpacity>

      <Text style={styles.footerText}>

        Already have an account?{' '}

        <Text

          style={styles.link}

          onPress={() => navigation.navigate('Login')}

        >

          Log In

        </Text>

      </Text>

    </View>

  );

};

const styles = StyleSheet.create({

  container: {

    flex: 1,

    justifyContent: 'center',

    alignItems: 'center',

    backgroundColor: '#F8F9FA',

    padding: 20,

  },

  title: {

    fontSize: 24,

    fontWeight: 'bold',

    color: '#343A40',

    marginBottom: 20,

  },

  input: {

    width: '100%',

    height: 50,

    backgroundColor: '#FFFFFF',

    borderRadius: 12,

    paddingHorizontal: 15,

    fontSize: 16,

    color: '#495057',

    marginBottom: 15,

    borderWidth: 1,

    borderColor: '#CED4DA',

  },

  button: {

    width: '100%',

    height: 50,

    backgroundColor: '#FF5722',

    justifyContent: 'center',

    alignItems: 'center',

    borderRadius: 12,

    marginTop: 10,

  },

  buttonText: {

    color: '#FFFFFF',

    fontSize: 18,

    fontWeight: 'bold',

  },

  footerText: {

    marginTop: 20,

    fontSize: 14,

    color: '#6C757D',

  },

  link: {

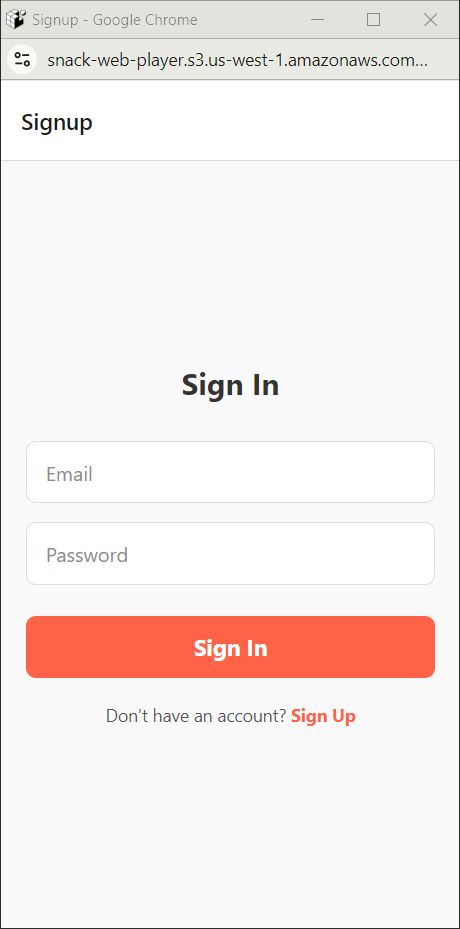
    color: '#FF5722',

    fontWeight: 'bold',

  },

});

export default SignupScreen;



import React, { useState } from 'react';

import {

  View,

  Text,

  StyleSheet,

  TextInput,

  TouchableOpacity,

  Alert,

} from 'react-native';

import { signInWithEmailAndPassword } from '@react-native-firebase/auth';

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

import { useDispatch } from 'react-redux';

import { setUser } from '../redux/actions';

import { MyTheme } from '../theme';

const LoginScreen = ({ navigation }) => {

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

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

  const dispatch = useDispatch();

  const handleLogin = async () => {

    if (!email || !password) {

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

      return;

    }

    try {

      await signInWithEmailAndPassword(email, password);

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

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

        dispatch(setUser(storedUser));

        Alert.alert('Success', 'Logged in successfully!');

        navigation.navigate('Home');

      } else {

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

      }

    } catch (error) {

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

    }

  };

  return (

    <View style={styles.container}>

      <Text style={styles.title}>Log In</Text>

      <TextInput

        style={styles.input}

        placeholder="Email"

        placeholderTextColor="#BDBDBD"

        value={email}

        onChangeText={setEmail}

        keyboardType="email-address"

      />

      <TextInput

        style={styles.input}

        placeholder="Password"

        placeholderTextColor="#BDBDBD"

        value={password}

        onChangeText={setPassword}

        secureTextEntry

      />

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

        <Text style={styles.buttonText}>Log In</Text>

      </TouchableOpacity>

      <Text style={styles.footerText}>

        Don't have an account?{' '}

        <Text

          style={styles.link}

          onPress={() => navigation.navigate('Signup')}

        >

          Sign Up

        </Text>

      </Text>

    </View>

  );

};

const styles = StyleSheet.create({

  container: {

    flex: 1,

    justifyContent: 'center',

    alignItems: 'center',

    backgroundColor: '#F8F9FA',

    padding: 20,

  },

  title: {

    fontSize: 24,

    fontWeight: 'bold',

    color: '#343A40',

    marginBottom: 20,

  },

  input: {

    width: '100%',

    height: 50,

    backgroundColor: '#FFFFFF',

    borderRadius: 12,

    paddingHorizontal: 15,

    fontSize: 16,

    color: '#495057',

    marginBottom: 15,

    borderWidth: 1,

    borderColor: '#CED4DA',

  },

  button: {

    width: '100%',

    height: 50,

    backgroundColor: '#FF5722',

    justifyContent: 'center',

    alignItems: 'center',

    borderRadius: 12,

    marginTop: 10,

  },

  buttonText: {

    color: '#FFFFFF',

    fontSize: 18,

    fontWeight: 'bold',

  },

  footerText: {

    marginTop: 20,

    fontSize: 14,

    color: '#6C757D',

  },

  link: {

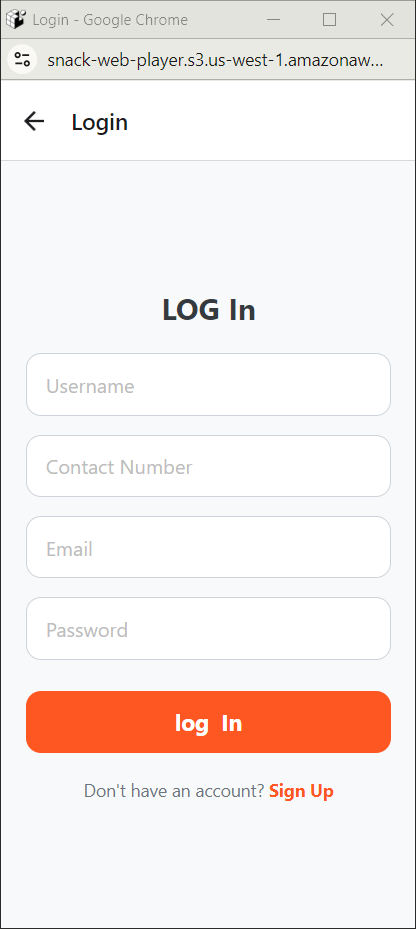
    color: '#FF5722',

    fontWeight: 'bold',

  },

});

export default LoginScreen;



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

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

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

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

import ProfileSection from './ProfileSection';

import OrderItem from './OrderItem';

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

import { setUser } from '../redux/actions';

const initialData = {

  name: 'Malaika Khattak',

  profilePic: 'https://via.placeholder.com/150',

  productSells: '14,850 Product Sells',

  earning: '30bil+ Earning',

  recentOrders: [

    {

      id: '1',

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

      date: '21, Thu 10:00',

      price: '$38.15',

      image: 'https://via.placeholder.com/80',

    },

    {

      id: '2',

      items: 'Pepperoni Pizza, Cheese Burst',

      date: '15, Fri 18:00',

      price: '$38.15',

      image: 'https://via.placeholder.com/80',

    },

    {

      id: '3',

      items: 'Beef Burger, Fries, Ketchup',

      date: '11, Mon 10:00',

      price: '$30.50',

      image: 'https://via.placeholder.com/80',

    },

  ],

};

const HomeScreen = () => {

  const dispatch = useDispatch();

  const user = useSelector((state) => state.user);

  const [profileData, setProfileData] = useState(initialData);

  const [modalVisible, setModalVisible] = useState(false);

  const [newOrder, setNewOrder] = useState({

    items: '',

    date: '',

    price: '',

    image: '',

  });

  useEffect(() => {

    const fetchLocation = async () => {

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

      if (status !== 'granted') {

        Alert.alert('Permission to access location was denied');

        return;

      }

      let loc = await Location.getCurrentPositionAsync({});

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

    };

    fetchLocation();

    const loadUser = async () => {

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

      if (storedUser) {

        dispatch(setUser(storedUser));

      }

    };

    loadUser();

  }, [dispatch]);

*// Add new order*

  const handleAddOrder = () => {

    setProfileData((prevData) => ({

      ...prevData,

      recentOrders: [

        ...prevData.recentOrders,

        { id: Date.now().toString(), ...newOrder },

      ],

    }));

    setModalVisible(false);

    setNewOrder({ items: '', date: '', price: '', image: '' });

  };

  return (

    <View style={styles.container}>

      {*/\* Profile Section \*/*}

      <ProfileSection profile={profileData} />

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

      <View style={styles.ordersSection}>

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

        <FlatList

          data={profileData.recentOrders}

          keyExtractor={(item) => item.id}

          renderItem={({ item }) => <OrderItem order={item} />}

          showsVerticalScrollIndicator={false}

        />

        {*/\* Add Order Button \*/*}

        <TouchableOpacity

          style={styles.addOrderButton}

          onPress={() => setModalVisible(true)}

        >

          <Text style={styles.addOrderButtonText}>+ Add Order</Text>

        </TouchableOpacity>

      </View>

      {*/\* Modal for Adding Order \*/*}

      <Modal

        visible={modalVisible}

        animationType="slide"

        transparent={true}

        onRequestClose={() => setModalVisible(false)}

      >

        <View style={styles.modalContainer}>

          <View style={styles.modalContent}>

            <Text style={styles.modalTitle}>Add New Order</Text>

            <TextInput

              style={styles.input}

              placeholder="Items"

              value={newOrder.items}

              onChangeText={(text) => setNewOrder({ ...newOrder, items: text })}

            />

            <TextInput

              style={styles.input}

              placeholder="Date"

              value={newOrder.date}

              onChangeText={(text) => setNewOrder({ ...newOrder, date: text })}

            />

            <TextInput

              style={styles.input}

              placeholder="Price"

              value={newOrder.price}

              onChangeText={(text) => setNewOrder({ ...newOrder, price: text })}

            />

            <TextInput

              style={styles.input}

              placeholder="Image URL"

              value={newOrder.image}

              onChangeText={(text) => setNewOrder({ ...newOrder, image: text })}

            />

            <Button title="Add Order" onPress={handleAddOrder} />

            <Button

              title="Cancel"

              color="red"

              onPress={() => setModalVisible(false)}

            />

          </View>

        </View>

      </Modal>

    </View>

  );

};

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#F8F9FA',

    padding: 20,

  },

  ordersSection: {

    marginTop: 20,

    backgroundColor: '#FFFFFF',

    borderRadius: 12,

    padding: 15,

    elevation: 3,

    shadowColor: '#000',

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

    shadowOpacity: 0.2,

    shadowRadius: 4,

  },

  sectionTitle: {

    fontSize: 20,

    fontWeight: 'bold',

    color: '#343A40',

    marginBottom: 10,

  },

  addOrderButton: {

    marginTop: 10,

    backgroundColor: '#28A745',

    padding: 12,

    borderRadius: 10,

    alignItems: 'center',

  },

  addOrderButtonText: {

    color: '#FFFFFF',

    fontWeight: 'bold',

    fontSize: 16,

  },

  modalContainer: {

    flex: 1,

    justifyContent: 'center',

    alignItems: 'center',

    backgroundColor: 'rgba(0,0,0,0.5)',

  },

  modalContent: {

    width: '80%',

    backgroundColor: '#FFFFFF',

    padding: 20,

    borderRadius: 10,

    elevation: 5,

  },

  modalTitle: {

    fontSize: 18,

    fontWeight: 'bold',

    marginBottom: 15,

  },

  input: {

    borderWidth: 1,

    borderColor: '#CED4DA',

    borderRadius: 8,

    padding: 10,

    marginBottom: 10,

  },

});

export default HomeScreen;

Profile Screen :

import React from 'react';

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

const ProfileSection = ({ profile }) => {

  return (

    <View style={styles.container}>

      {*/\* Profile Picture \*/*}

      <Image source={{ uri: profile.profilePic }} style={styles.profilePic} />

      {*/\* Profile Details \*/*}

      <Text style={styles.name}>{profile.name}</Text>

      <Text style={styles.details}>{profile.productSells}</Text>

      <Text style={styles.details}>{profile.earning}</Text>

    </View>

  );

};

const styles = StyleSheet.create({

  container: {

    backgroundColor: '#FFFFFF',

    borderRadius: 12,

    padding: 20,

    alignItems: 'center',

    marginBottom: 20,

    elevation: 3,

    shadowColor: '#000',

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

    shadowOpacity: 0.2,

    shadowRadius: 4,

  },

  profilePic: {

    width: 100,

    height: 100,

    borderRadius: 50,

    marginBottom: 15,

    borderColor: '#FF5722',

    borderWidth: 2,

  },

  name: {

    fontSize: 24,

    fontWeight: 'bold',

    color: '#343A40',

    marginBottom: 10,

  },

  details: {

    fontSize: 16,

    color: '#868E96',

  },

});

export default ProfileSection;

order items screen:

import React from 'react';

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

const OrderItem = ({ order }) => {

  return (

    <View style={styles.container}>

      {*/\* Food Image \*/*}

      <Image source={{ uri: order.image }} style={styles.foodImage} />

      {*/\* Order Details \*/*}

      <View style={styles.detailsContainer}>

        <Text style={styles.items}>{order.items}</Text>

        <Text style={styles.date}>{order.date}</Text>

      </View>

      {*/\* Order Price \*/*}

      <Text style={styles.price}>{order.price}</Text>

    </View>

  );

};

const styles = StyleSheet.create({

  container: {

    flexDirection: 'row',

    alignItems: 'center',

    backgroundColor: '#F1F3F5',

    borderRadius: 10,

    padding: 10,

    marginBottom: 10,

  },

  foodImage: {

    width: 50,

    height: 50,

    borderRadius: 10,

    marginRight: 10,

  },

  detailsContainer: {

    flex: 2,

    justifyContent: 'center',

  },

  items: {

    fontSize: 16,

    fontWeight: 'bold',

    color: '#495057',

  },

  date: {

    fontSize: 14,

    color: '#868E96',

  },

  price: {

    fontSize: 16,

    fontWeight: 'bold',

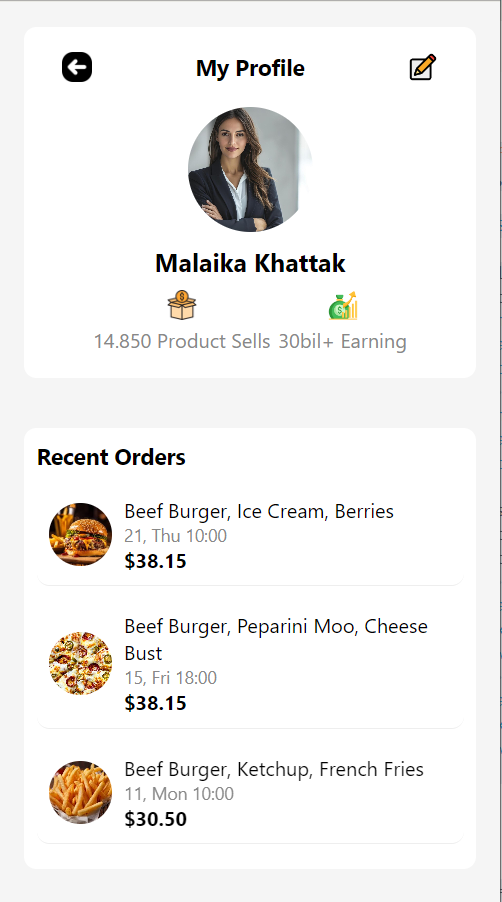
    color: '#FF5722',

    textAlign: 'right',

  },

});

export default OrderItem;

****

Fire Base Configuration:

*// firebaseConfig.js*

import { initializeApp } from 'firebase/app';

import { getAuth } from 'firebase/auth';

import { getDatabase } from 'firebase/database';

const firebaseConfig = {

  apiKey: 'AIzaSyCOxzi\_MVG39EqxJ0\_EWY6Ptm9iomyB2cA',

  authDomain: 'myproject-4a357.firebaseapp.com',

  databaseURL: 'https://myproject-4a357-default-rtdb.firebaseio.com/',

  projectId: 'myproject-4a357',

  storageBucket: 'myproject-4a357.appspot.com',

  messagingSenderId: '1053630772457',

  appId: '1:1053630772457:web:e67477186ab1f325296c78',

};

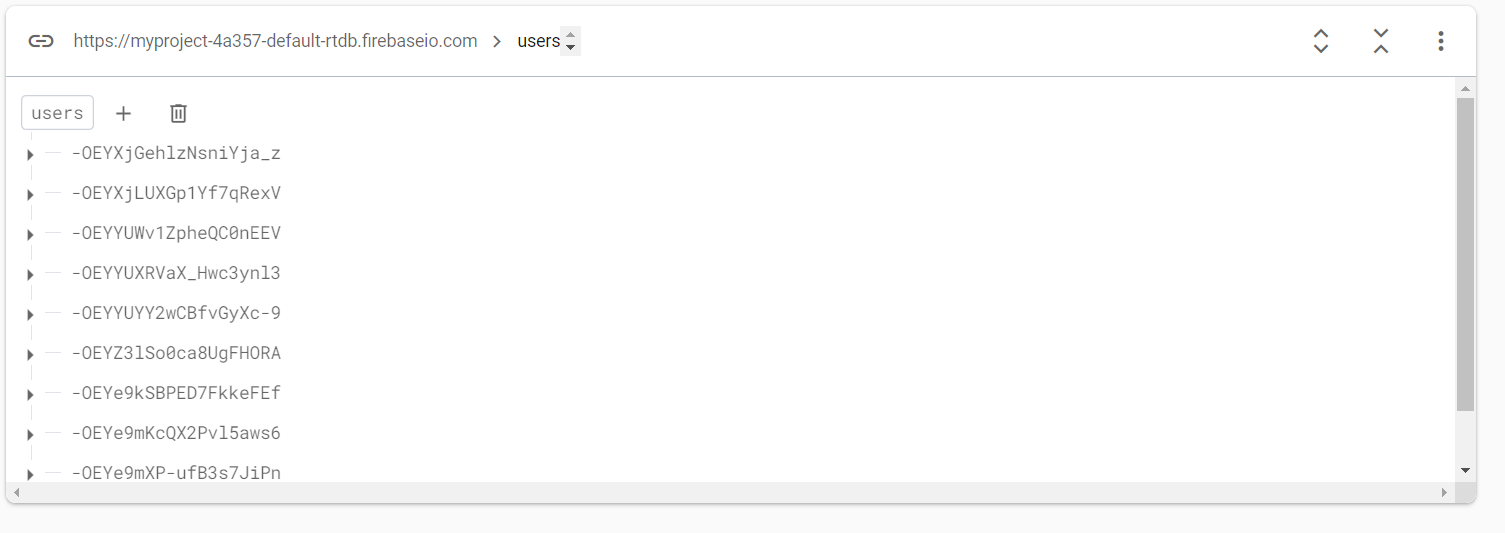
*// Initialize Firebase*

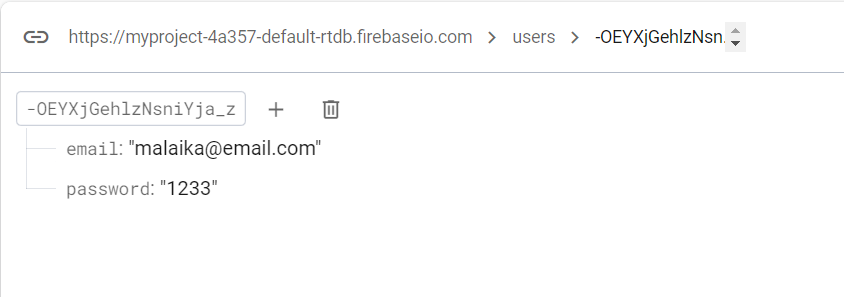
const app = initializeApp(firebaseConfig);

const auth = getAuth(app);

const database = getDatabase(app);

export { app, auth, database };





REDUX:

Action.js:

*// redux/actions.js*

export const setUser = (user) => ({

  type: 'SET\_USER',

  payload: user,

});

Reducer.js:

*// redux/reducers.js*

const initialState = {

  user: null,

};

const userReducer = (state = initialState, action) => {

  switch (action.type) {

    case 'SET\_USER':

      return {

        ...state,

        user: action.payload,

      };

    default:

      return state;

  }

};

export default userReducer;

store.js:

import { createStore, combineReducers } from 'redux';

import userReducer from './reducers';

const rootReducer = combineReducers({

  user: userReducer,

});

const store = createStore(rootReducer);

export default store;

App.js:

import React from 'react';

import { Provider } from 'react-redux';

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

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

import SignupScreen from './components/SignupScreen';

import LoginScreen from './components/LoginScreen';

import HomeScreen from './components/HomeScreen';

import store from './redux/store';

import { MyTheme } from './theme';

const Stack = createStackNavigator();

*// Firebase Realtime Database URL*

const FIREBASE\_API\_ENDPOINT = 'https://myproject-4a357-default-rtdb.firebaseio.com';

const App = () => {

  return (

    <Provider store={store}>

      <NavigationContainer theme={MyTheme}>

        <Stack.Navigator

          initialRouteName="Signup"

          screenOptions={{

            headerStyle: {

              backgroundColor: MyTheme.colors.primary, *// Customize the header background color*

            },

            headerTintColor: '#fff', *// Customize the header text color*

            headerTitleStyle: {

              fontWeight: 'bold', *// Add bold styling to the header text*

            },

          }}

        >

          <Stack.Screen

            name="Signup"

            component={SignupScreen}

            options={{ title: 'Sign Up' }} *// Custom title for Signup screen*

          />

          <Stack.Screen

            name="Login"

            component={LoginScreen}

            options={{ title: 'Log In' }} *// Custom title for Login screen*

          />

          <Stack.Screen

            name="Home"

            component={HomeScreen}

            options={{ title: 'Home' }} *// Custom title for Home screen*

          />

        </Stack.Navigator>

      </NavigationContainer>

    </Provider>

  );

};

export default App;

snack expo link:

<https://snack.expo.dev/@malaikakhattak/mad-assignment-3>