import React, { useState } from 'react';

import {

  View,

  Text,

  TextInput,

  SectionList,

  TouchableOpacity,

  Modal,

  StyleSheet,

} from 'react-native';

const contactsData = [

  { name: 'Ahmed', number: '03001234567', group: 'Family' },

  { name: 'Fatima', number: '03012345678', group: 'Friends' },

  { name: 'Usman', number: '03123456789', group: 'Work' },

  { name: 'Ayesha', number: '03234567890', group: 'Family' },

  { name: 'Ali', number: '03345678901', group: 'Friends' },

  { name: 'Zara', number: '03456789012', group: 'Work' },

  { name: 'Bilal', number: '03567890123', group: 'Family' },

  { name: 'Hina', number: '03678901234', group: 'Friends' },

  { name: 'Omar', number: '03789012345', group: 'Work' },

  { name: 'Noor', number: '03890123456', group: 'Family' },

];

export default function App() {

  const [search, setSearch] = useState('');

  const [selectedContact, setSelectedContact] = useState(null);

  const filtered = contactsData.filter(

    c =>

      c.name.toLowerCase().includes(search.toLowerCase()) ||

      c.number.includes(search)

  );

  const grouped = ['Family', 'Friends', 'Work'].map(group => ({

    title: group,

    data: filtered.filter(contact => contact.group === group),

  }));

  return (

    <View style={styles.container}>

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

      <TextInput

        style={styles.input}

        placeholder="Search by name or number"

        value={search}

        onChangeText={setSearch}

      />

      <SectionList

        sections={grouped}

        keyExtractor={(item, index) => item.name + index}

        renderItem={({ item }) => (

          <TouchableOpacity

            style={styles.item}

            onPress={() => setSelectedContact(item)}>

            <Text>{item.name}</Text>

            <Text>{item.number}</Text>

          </TouchableOpacity>

        )}

        renderSectionHeader={({ section: { title } }) => (

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

        )}

      />

      <Modal

        visible={!!selectedContact}

        transparent={true}

        animationType="slide"

        onRequestClose={() => setSelectedContact(null)}>

        <View style={styles.modalContainer}>

          <View style={styles.modalContent}>

            {selectedContact && (

              <>

                <Text style={styles.modalText}>Name: {selectedContact.name}</Text>

                <Text style={styles.modalText}>Number: {selectedContact.number}</Text>

                <Text style={styles.modalText}>Group: {selectedContact.group}</Text>

              </>

            )}

            <TouchableOpacity onPress={() => setSelectedContact(null)}>

              <Text style={styles.closeButton}>Close</Text>

            </TouchableOpacity>

          </View>

        </View>

      </Modal>

    </View>

  );

}

const styles = StyleSheet.create({

  container: { flex: 1, padding: 20, backgroundColor: '#f2f2f2' },

  title: { fontSize: 24, fontWeight: 'bold', marginBottom: 10 },

  input: {

    padding: 10,

    backgroundColor: '#fff',

    borderRadius: 5,

    marginBottom: 10,

  },

  header: { fontSize: 18, fontWeight: 'bold', backgroundColor: '#ddd', padding: 5 },

  item: {

    backgroundColor: '#fff',

    padding: 10,

    borderBottomWidth: 1,

    borderBottomColor: '#eee',

  },

  modalContainer: {

    flex: 1,

    justifyContent: 'center',

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

  },

  modalContent: {

    backgroundColor: 'white',

    margin: 20,

    padding: 20,

    borderRadius: 10,

  },

  modalText: { fontSize: 16, marginBottom: 10 },

  closeButton: {

    marginTop: 10,

    color: 'blue',

    textAlign: 'right',

  },

});