**Lab Assignment 2**

**Input Screen**

A screenshot of a computer

Description automatically generated

Database Insertion:

A screenshot of a computer

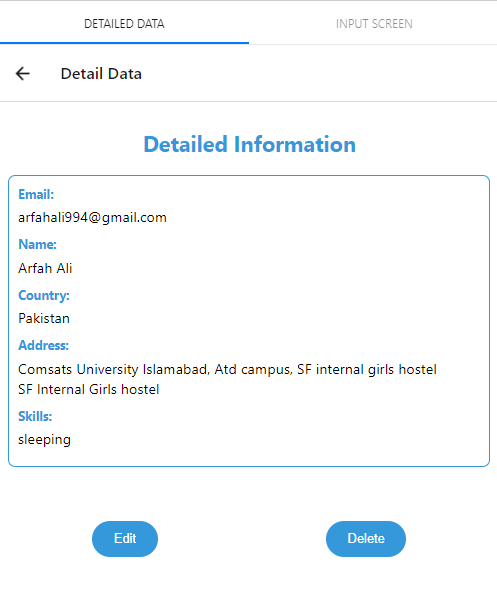
Description automatically generated

Show Data

A screenshot of a computer

Description automatically generated

Detailed Data



After Deletion  
  
  
A screenshot of a computer

Description automatically generated

**Code:**

**UseFirestore.js:**

import { useState, useEffect } from 'react';

import {

  collection,

  addDoc,

  getDocs,

  updateDoc,

  doc,

  setDoc,

  deleteDoc,

} from 'firebase/firestore';

import { db } from '../services/firebase.config';

const useFirestore = () => {

  const [isLoading, setIsLoading] = useState(false);

  const addData = async (data) => {

    setIsLoading(true);

    try {

      const docRef = doc(db, 'persons', data.name);

      await setDoc(docRef, data);

    } catch (error) {

      console.error('Error adding document: ', error);

    }

    setIsLoading(false);

  };

  const retrieveData = async () => {

    setIsLoading(true);

    const querySnapshot = await getDocs(collection(db, 'persons'));

    const data = [];

    querySnapshot.forEach((doc) => {

      data.push({ id: doc.id, ...doc.data() });

    });

    setIsLoading(false);

    return data;

  };

const updateData = async (originalName, updatedData) => {

  setIsLoading(true);

  const docRef = doc(db, 'persons', originalName);

  try {

    await updateDoc(docRef, updatedData);

  } catch (error) {

    console.error('Error updating document: ', error);

  }

  setIsLoading(false);

};

  const deleteData = async (name) => {

    setIsLoading(true);

    const docRef = doc(db, 'persons', name);

    try {

      await deleteDoc(docRef);

    } catch (error) {

      console.error('Error deleting document: ', error);

    }

    setIsLoading(false);

  };

  return { addData, retrieveData, updateData, isLoading, deleteData };

};

export default useFirestore;

**DetailedData.js:**

import React from 'react';

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

import { Button } from 'react-native-paper';

import useFirestore from '../hooks/useFirestore';

const DetailData = ({ navigation, route }) => {

  const { item } = route.params;

  const { deleteData } = useFirestore();

  const handleEdit = () => {

    navigation.navigate('Edit Data', { item });

  };

  const handleDelete = async (name) => {

    await deleteData(name);

  };

  return (

    <View style={styles.container}>

      <Text style={styles.header}>Detailed Information</Text>

      <View style={styles.dataContainer}>

        <Text style={styles.label}>Email:</Text>

        <Text style={styles.dataText}>{item.email}</Text>

        <Text style={styles.label}>Name:</Text>

        <Text style={styles.dataText}>{item.name}</Text>

        <Text style={styles.label}>Country:</Text>

        <Text style={styles.dataText}>{item.country}</Text>

        <Text style={styles.label}>Address:</Text>

        <Text style={styles.dataText}>{item.address}</Text>

        <Text style={styles.label}>Skills:</Text>

        <Text style={styles.dataText}>{item.skills}</Text>

        {/\* Add more details as needed \*/}

      </View>

      <View style={styles.buttonContainer}>

        <Button

          mode='contained'

          onPress={handleEdit}

          style={styles.button}

          color='#fff'>

          Edit

        </Button>

        <Button

          mode='contained'

          onPress={() => handleDelete(item.name)}

          style={styles.button}

          color='#fff'>

          Delete

        </Button>

      </View>

    </View>

  );

};

const styles = StyleSheet.create({

  container: {

    flex: 1,

    padding: 10,

    backgroundColor: '#fff',

  },

  button: {

    marginTop: 20,

    backgroundColor: '#3498db',

  },

  header: {

    fontSize: 24,

    textAlign: 'center',

    marginVertical: 20,

    color: '#3498db', // Updated color for the header

    fontWeight: 'bold',

  },

  dataContainer: {

    borderWidth: 1,

    borderColor: '#3498db', // Updated color for the border

    padding: 10,

    borderRadius: 8, // Added border radius for a rounded look

    marginBottom: 20,

  },

  label: {

    fontWeight: 'bold',

    marginBottom: 5,

    color: '#3498db', // Updated color for the label

  },

  dataText: {

    marginBottom: 10,

    fontSize: 16, // Adjusted font size for better readability

  },

  buttonContainer: {

    flexDirection: 'row',

    justifyContent: 'space-around',

    marginTop: 20,

  },

});

export default DetailData;

**ShowData.js**

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

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

import useFirestore from '../hooks/useFirestore';

import { ActivityIndicator, Button } from 'react-native-paper';

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

import DetailData from './DetailData';

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

import EditData from './EditData';

const ShowData = () => {

  const navigation = useNavigation();

  const { retrieveData } = useFirestore();

  const [data, setData] = useState([]);

  useEffect(() => {

    const fetchData = async () => {

      const retrievedData = await retrieveData();

      setData(retrievedData);

    };

    fetchData();

  }, []);

  return (

    <View style={styles.container}>

      {data.length === 0 ? (

        <ActivityIndicator style={styles.noDataText} size='large' />

      ) : (

        data.map((item, index) => (

          <View key={index} style={styles.dataRow}>

            <View style={styles.column}>

              <Text style={styles.dataLabel}>Email:</Text>

              <Text style={styles.dataText}>{item.email}</Text>

            </View>

            <View style={styles.column}>

              <Text style={styles.dataLabel}>Name:</Text>

              <Text style={styles.dataText}>{item.name}</Text>

            </View>

            {/\* Add more fields in separate columns as needed \*/}

            <Button

              mode='contained'

              onPress={() => navigation.navigate('Detail Data', { item })}

              style={styles.button}

              color='#3498db'>

              Select

            </Button>

          </View>

        ))

      )}

    </View>

  );

};

const styles = StyleSheet.create({

  container: {

    flex: 1,

    padding: 10,

  },

  button: {

    marginTop: 20,

    backgroundColor: '#3498db',

  },

  dataRow: {

    flexDirection: 'row', // Arrange items in a row

    justifyContent: 'space-between', // Add space between columns

    alignItems: 'center', // Align items vertically in the center

    marginBottom: 20,

    borderBottomWidth: 1,

    borderBottomColor: '#ccc',

    paddingBottom: 10,

  },

  column: {

    flex: 1, // Each column takes equal space

  },

  dataLabel: {

    fontSize: 16,

    fontWeight: 'bold',

    marginBottom: 5,

    color: '#3498db',

  },

  dataText: {

    fontSize: 16,

    marginBottom: 10,

  },

  noDataText: {

    flex: 1,

    justifyContent: 'center',

    alignItems: 'center',

  },

});

const Stack = createStackNavigator();

const DataNavigation = () => {

  return (

    <Stack.Navigator initialRouteName='Show Data'>

      <Stack.Screen name='Show Data' component={ShowData} />

      <Stack.Screen name='Detail Data' component={DetailData} />

      <Stack.Screen name='Edit Data' component={EditData} />

    </Stack.Navigator>

  );

};

export default DataNavigation;

**InputData.js**

import React, { useState } from 'react';

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

import { Button, RadioButton, TextInput } from 'react-native-paper';

import useFirestore from '../hooks/useFirestore';

const Input = () => {

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

  const [name, setName] = useState('');

  const [country, setCountry] = useState('');

  const [gender, setGender] = useState('Male');

  const [skills, setSkills] = useState('');

  const [address, setAddress] = useState('');

  const { addData } = useFirestore();

  const handleSubmit = async () => {

    console.log({ email, name, country, gender, skills, address });

    const formData = {

      email: email,

      name: name,

      country: country,

      gender: gender,

      skills: skills,

      address: address,

    };

    await addData(formData);

  };

  return (

    <View style={styles.container}>

      <TextInput

        mode='outlined'

        label='Email'

        onChangeText={setEmail}

        style={styles.input}

        theme={{ colors: { primary: '#3498db' } }}

      />

      <TextInput

        mode='outlined'

        label='Name'

        onChangeText={setName}

        style={styles.input}

        theme={{ colors: { primary: '#3498db' } }}

      />

      <TextInput

        mode='outlined'

        label='Country'

        onChangeText={setCountry}

        style={styles.input}

        theme={{ colors: { primary: '#3498db' } }}

      />

      <RadioButton.Group onValueChange={setGender} value={gender}>

        <RadioButton.Item label='Male' value='Male' />

        <RadioButton.Item label='Female' value='Female' />

      </RadioButton.Group>

      <TextInput

        mode='outlined'

        label='Skills'

        multiline={true}

        maxLines={5}

        onChangeText={setSkills}

        style={styles.input}

        theme={{ colors: { primary: '#3498db' } }}

      />

      <TextInput

        mode='outlined'

        label='Address'

        multiline={true}

        maxLines={5}

        onChangeText={setAddress}

        style={styles.input}

        theme={{ colors: { primary: '#3498db' } }}

      />

      <Button

        mode='contained'

        onPress={handleSubmit}

        style={styles.button}

        labelStyle={styles.buttonText}

        color='#fff'

      >

        Submit

      </Button>

    </View>

  );

};

const styles = StyleSheet.create({

  container: {

    flex: 1,

    justifyContent: 'center',

    padding: 16,

    backgroundColor: '#ecf0f1',

  },

  input: {

    marginBottom: 10,

    fontFamily: 'Arial', // Change to your preferred font family

    fontSize: 16,

  },

  button: {

    marginTop: 20,

    backgroundColor: '#3498db',

  },

  buttonText: {

    fontSize: 18,

  },

});

export default Input;

**UpdateData.js:**

import React, { useState } from 'react';

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

import useFirestore from '../hooks/useFirestore';

const EditData = ({ route, navigation }) => {

  const { item } = route.params;

  // Create state for each field

  const [email, setEmail] = useState(item.email);

  const [name, setName] = useState(item.name);

  const [address, setAddress] = useState(item.address);

  const { updateData } = useFirestore();

  const handleSave = async () => {

    // Implement save logic here

    // For example, update the Firestore document with the new state values

    console.log('Data saved', {

      email,

      name,

      address,

    });

    await updateData(name, { email, name, address });

    // Then navigate back or show confirmation message

  };

  return (

    <View style={styles.container}>

      <TextInput

        style={styles.input}

        value={email}

        onChangeText={setEmail}

        placeholder='Email'

      />

      <TextInput

        style={styles.input}

        value={name}

        onChangeText={setName}

        placeholder='Name'

      />

      <TextInput

        style={styles.input}

        value={address}

        onChangeText={setAddress}

        placeholder='Address'

      />

      <Button title='Save Changes' onPress={handleSave} />

    </View>

  );

};

const styles = StyleSheet.create({

  container: {

    flex: 1,

    padding: 10,

  },

  input: {

    borderWidth: 1,

    borderColor: '#ddd',

    padding: 8,

    marginVertical: 6,

  },

});

export default EditData;

**App.js:**

import React from 'react';

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

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

import { createMaterialTopTabNavigator } from '@react-navigation/material-top-tabs';

import ShowData from './src/screens/ShowData';

import Input from './src/screens/Input';

import DetailData from './src/screens/DetailData';

const Stack = createStackNavigator();

const Tab = createMaterialTopTabNavigator();

const App = () => {

  return (

    <NavigationContainer>

      <Tab.Navigator>

        <Tab.Screen

          name='All Data'

          component={ShowData}

          options={{

            title: 'Detailed Data',

            headerShown: false,

          }}

        />

        <Tab.Screen

          name='Input Data'

          component={Input}

          options={{

            title: 'Input Screen',

            headerShown: false,

          }}

        />

      </Tab.Navigator>

    </NavigationContainer>

  );

};

export default App;