My button

// Example: Pre-Populated SQLite Database in React Native

// https://aboutreact.com/example-of-pre-populated-sqlite-database-in-react-native

// Custom Button

import React from 'react';

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

const Mybutton = (props) => {

  return (

    <TouchableOpacity style={styles.button} onPress={props.customClick}>

      <Text style={styles.text}>{props.title}</Text>

    </TouchableOpacity>

  );

};

const styles = StyleSheet.create({

  button: {

    alignItems: 'center',

    backgroundColor: '#f05555',

    color: '#ffffff',

    padding: 10,

    marginTop: 16,

    marginLeft: 35,

    marginRight: 35,

  },

  text: {

    color: '#ffffff',

  },

});

export default Mybutton;

Mytext

// Example: Pre-Populated SQLite Database in React Native

// https://aboutreact.com/example-of-pre-populated-sqlite-database-in-react-native

// Custom Text

import React from 'react';

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

const Mytext = (props) => {

  return <Text style={styles.text}>{props.text}</Text>;

};

const styles = StyleSheet.create({

  text: {

    color: '#111825',

    fontSize: 18,

    marginTop: 16,

    marginLeft: 35,

    marginRight: 35,

  },

});

export default Mytext;

Mytextinput

// Example: Pre-Populated SQLite Database in React Native

// https://aboutreact.com/example-of-pre-populated-sqlite-database-in-react-native

// Custom TextInput

import React from 'react';

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

const Mytextinput = (props) => {

  return (

    <View

      style={{

        marginLeft: 35,

        marginRight: 35,

        marginTop: 10,

        borderColor: '#007FFF',

        borderWidth: 1,

      }}>

      <TextInput

        underlineColorAndroid="transparent"

        placeholder={props.placeholder}

        placeholderTextColor="#007FFF"

        keyboardType={props.keyboardType}

        onChangeText={props.onChangeText}

        returnKeyType={props.returnKeyType}

        numberOfLines={props.numberOfLines}

        multiline={props.multiline}

        onSubmitEditing={props.onSubmitEditing}

        style={props.style}

        blurOnSubmit={false}

        value={props.value}

      />

    </View>

  );

};

export default Mytextinput;

Deleteuser

// Example: Pre-Populated SQLite Database in React Native

// https://aboutreact.com/example-of-pre-populated-sqlite-database-in-react-native

// Screen to delete the user

import React, {useState} from 'react';

import { Button, Text, View, Alert, SafeAreaView } from 'react-native';

import Mytextinput from './components/Mytextinput';

import Mybutton from './components/Mybutton';

import { openDatabase } from 'react-native-sqlite-storage';

//Connction to access the pre-populated user\_db.db

var db = openDatabase({ name: 'user\_db.db', createFromLocation : 1});

const DeleteUser = ({navigation}) =>{

  let [inputUserId, setInputUserId] = useState('');

  let deleteUser = () => {

    db.transaction(tx => {

      tx.executeSql(

        'DELETE FROM table\_user where user\_id=?',

        [inputUserId],

        (tx, results) => {

          console.log('Results', results.rowsAffected);

          if (results.rowsAffected > 0) {

            Alert.alert(

              'Success',

              'User deleted successfully',

              [

                {

                  text: 'Ok',

                  onPress: () => navigation.navigate('HomeScreen'),

                },

              ],

              { cancelable: false }

            );

          } else {

            alert('Please insert a valid User Id');

          }

        }

      );

    });

  };

  return (

    <SafeAreaView style={{flex: 1}}>

      <View style={{flex: 1, backgroundColor: 'white'}}>

        <View style={{flex: 1}}>

          <Mytextinput

            placeholder="Enter User Id"

            onChangeText={inputUserId => setInputUserId(inputUserId)}

            style={{ padding:10 }}

          />

          <Mybutton

            title="Delete User"

            customClick={deleteUser}

          />

        </View>

        <Text style={{ fontSize: 18, textAlign: 'center', color: 'grey' }}>

          Example of SQLite Database in React Native

        </Text>

        <Text

          style={{ fontSize: 16, textAlign: 'center', color: 'grey' }}>

          www.aboutreact.com

        </Text>

      </View>

    </SafeAreaView>

  );

}

export default DeleteUser;

Homescreen

// Example: Pre-Populated SQLite Database in React Native

// https://aboutreact.com/example-of-pre-populated-sqlite-database-in-react-native

import React, { useEffect } from 'react';

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

import Mybutton from './components/Mybutton';

import Mytext from './components/Mytext';

import { openDatabase } from 'react-native-sqlite-storage';

//Connction to access the pre-populated user\_db.db

var db = openDatabase({ name: 'user\_db.db', createFromLocation : 1});

const HomeScreen = ({ navigation }) => {

  useEffect(() => {

    db.transaction(function (txn) {

      txn.executeSql(

        "SELECT name FROM sqlite\_master WHERE type='table' AND name='table\_user'",

        [],

        function (tx, res) {

          console.log('item:', res.rows.length);

          if (res.rows.length == 0) {

            txn.executeSql('DROP TABLE IF EXISTS table\_user', []);

            txn.executeSql(

              'CREATE TABLE IF NOT EXISTS table\_user(user\_id INTEGER PRIMARY KEY AUTOINCREMENT, user\_name VARCHAR(20), user\_contact INT(10), user\_address VARCHAR(255))',

              []

            );

          }

        }

      );

    });

  }, []);

  return (

    <SafeAreaView style={{ flex: 1 }}>

      <View style={{ flex: 1, backgroundColor: 'white' }}>

        <View style={{ flex: 1 }}>

          <Mytext text="WU Delivery" />

          <Mybutton

            title="Register"

            customClick={() => navigation.navigate('Register')}

          />

          <Mybutton

            title="Update"

            customClick={() => navigation.navigate('Update')}

          />

          <Mybutton

            title="View"

            customClick={() => navigation.navigate('View')}

          />

          <Mybutton

            title="View All"

            customClick={() => navigation.navigate('ViewAll')}

          />

          <Mybutton

            title="Delete"

            customClick={() => navigation.navigate('Delete')}

          />

        </View>

        <Text style={{ fontSize: 18, textAlign: 'center', color: 'grey' }}>

          Example of SQLite Database in React Native

        </Text>

        <Text style={{ fontSize: 16, textAlign: 'center', color: 'grey' }}>

          www.aboutreact.com

        </Text>

      </View>

    </SafeAreaView>

  );

};

export default HomeScreen;

registeruser

// Example: Pre-Populated SQLite Database in React Native

// https://aboutreact.com/example-of-pre-populated-sqlite-database-in-react-native

// Screen to register the user

import React, { useState } from 'react';

import {

  View,

  ScrollView,

  KeyboardAvoidingView,

  Alert,

  SafeAreaView,

  Text,

} from 'react-native';

import Mytextinput from './components/Mytextinput';

import Mybutton from './components/Mybutton';

import { openDatabase } from 'react-native-sqlite-storage';

//Connction to access the pre-populated user\_db.db

var db = openDatabase({ name: 'user\_db.db', createFromLocation : 1});

const RegisterUser = ({ navigation }) => {

  let [userName, setUserName] = useState('');

  let [userContact, setUserContact] = useState('');

  let [userAddress, setUserAddress] = useState('');

  let register\_user = () => {

    console.log(userName, userContact, userAddress);

    if (!userName) {

      alert('Please fill name');

      return;

    }

    if (!userContact) {

      alert('Please fill Contact Number');

      return;

    }

    if (!userAddress) {

      alert('Please fill Address');

      return;

    }

    db.transaction(function (tx) {

      tx.executeSql(

        'INSERT INTO table\_user (user\_name, user\_contact, user\_address) VALUES (?,?,?)',

        [userName, userContact, userAddress],

        (tx, results) => {

          console.log('Results', results.rowsAffected);

          if (results.rowsAffected > 0) {

            Alert.alert(

              'Success',

              'You are Registered Successfully',

              [

                {

                  text: 'Ok',

                  onPress: () => navigation.navigate('HomeScreen'),

                },

              ],

              { cancelable: false }

            );

          } else alert('Registration Failed');

        }

      );

    });

  };

  return (

    <SafeAreaView style={{ flex: 1 }}>

      <View style={{ flex: 1, backgroundColor: 'white' }}>

        <View style={{ flex: 1 }}>

          <ScrollView keyboardShouldPersistTaps="handled">

            <KeyboardAvoidingView

              behavior="padding"

              style={{ flex: 1, justifyContent: 'space-between' }}>

              <Mytextinput

                placeholder="Enter Name"

                onChangeText={(userName) => setUserName(userName)}

                style={{ padding: 10 }}

              />

              <Mytextinput

                placeholder="Enter Contact No"

                onChangeText={(userContact) => setUserContact(userContact)}

                maxLength={10}

                keyboardType="numeric"

                style={{ padding: 10 }}

              />

              <Mytextinput

                placeholder="Enter Address"

                onChangeText={(userAddress) => setUserAddress(userName)}

                maxLength={225}

                numberOfLines={5}

                multiline={true}

                style={{ textAlignVertical: 'top', padding: 10 }}

              />

              <Mybutton title="Submit" customClick={register\_user} />

            </KeyboardAvoidingView>

          </ScrollView>

        </View>

        <Text style={{ fontSize: 18, textAlign: 'center', color: 'grey' }}>

          Example of SQLite Database in React Native

        </Text>

        <Text style={{ fontSize: 16, textAlign: 'center', color: 'grey' }}>

          www.aboutreact.com

        </Text>

      </View>

    </SafeAreaView>

  );

};

export default RegisterUser;

updateuser

// Example: Pre-Populated SQLite Database in React Native

// https://aboutreact.com/example-of-pre-populated-sqlite-database-in-react-native

// Screen to update the user

import React, { useState } from 'react';

import {

  View,

  ScrollView,

  KeyboardAvoidingView,

  Alert,

  SafeAreaView,

  Text,

} from 'react-native';

import Mytextinput from './components/Mytextinput';

import Mybutton from './components/Mybutton';

import { openDatabase } from 'react-native-sqlite-storage';

//Connction to access the pre-populated user\_db.db

var db = openDatabase({ name: 'user\_db.db', createFromLocation : 1});

const UpdateUser = ({ navigation }) => {

  let [inputUserId, setInputUserId] = useState('');

  let [userName, setUserName] = useState('');

  let [userContact, setUserContact] = useState('');

  let [userAddress, setUserAddress] = useState('');

  let updateAllStates = (name, contact, address) => {

    setUserName(name);

    setUserContact(contact);

    setUserAddress(address);

  };

  let searchUser = () => {

    console.log(inputUserId);

    db.transaction((tx) => {

      tx.executeSql(

        'SELECT \* FROM table\_user where user\_id = ?',

        [inputUserId],

        (tx, results) => {

          var len = results.rows.length;

          if (len > 0) {

            let res = results.rows.item(0);

            updateAllStates(res.user\_name, res.user\_contact, res.user\_address);

          } else {

            alert('No user found');

            updateAllStates('', '', '');

          }

        }

      );

    });

  };

  let updateUser = () => {

    console.log(inputUserId, userName, userContact, userAddress);

    if (!inputUserId) {

      alert('Please fill User id');

      return;

    }

    if (!userName) {

      alert('Please fill name');

      return;

    }

    if (!userContact) {

      alert('Please fill Contact Number');

      return;

    }

    if (!userAddress) {

      alert('Please fill Address');

      return;

    }

    db.transaction((tx) => {

      tx.executeSql(

        'UPDATE table\_user set user\_name=?, user\_contact=? , user\_address=? where user\_id=?',

        [userName, userContact, userAddress, inputUserId],

        (tx, results) => {

          console.log('Results', results.rowsAffected);

          if (results.rowsAffected > 0) {

            Alert.alert(

              'Success',

              'User updated successfully',

              [

                {

                  text: 'Ok',

                  onPress: () => navigation.navigate('HomeScreen'),

                },

              ],

              { cancelable: false }

            );

          } else alert('Updation Failed');

        }

      );

    });

  };

  return (

    <SafeAreaView style={{ flex: 1 }}>

      <View style={{ flex: 1, backgroundColor: 'white' }}>

        <View style={{ flex: 1 }}>

          <ScrollView keyboardShouldPersistTaps="handled">

            <KeyboardAvoidingView

              behavior="padding"

              style={{ flex: 1, justifyContent: 'space-between' }}>

              <Mytextinput

                placeholder="Enter User Id"

                style={{ padding: 10 }}

                onChangeText={(inputUserId) => setInputUserId(inputUserId)}

              />

              <Mybutton title="Search User" customClick={searchUser} />

              <Mytextinput

                placeholder="Enter Name"

                value={userName}

                style={{ padding: 10 }}

                onChangeText={(userName) => setUserName(userName)}

              />

              <Mytextinput

                placeholder="Enter Contact No"

                value={'' + userContact}

                onChangeText={(userContact) => setUserContact(userContact)}

                maxLength={10}

                style={{ padding: 10 }}

                keyboardType="numeric"

              />

              <Mytextinput

                value={userAddress}

                placeholder="Enter Address"

                onChangeText={(userAddress) => setUserAddress(userAddress)}

                maxLength={225}

                numberOfLines={5}

                multiline={true}

                style={{ textAlignVertical: 'top', padding: 10 }}

              />

              <Mybutton title="Update User" customClick={updateUser} />

            </KeyboardAvoidingView>

          </ScrollView>

        </View>

        <Text style={{ fontSize: 18, textAlign: 'center', color: 'grey' }}>

          Example of SQLite Database in React Native

        </Text>

        <Text style={{ fontSize: 16, textAlign: 'center', color: 'grey' }}>

          www.aboutreact.com

        </Text>

      </View>

    </SafeAreaView>

  );

};

export default UpdateUser;

viewalluser

// Example: Pre-Populated SQLite Database in React Native

// https://aboutreact.com/example-of-pre-populated-sqlite-database-in-react-native

// Screen to view all the user

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

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

import { openDatabase } from 'react-native-sqlite-storage';

//Connction to access the pre-populated user\_db.db

var db = openDatabase({ name: 'user\_db.db', createFromLocation : 1});

const ViewAllUser = () => {

  let [flatListItems, setFlatListItems] = useState([]);

  useEffect(() => {

    db.transaction((tx) => {

      tx.executeSql('SELECT \* FROM table\_user', [], (tx, results) => {

        var temp = [];

        for (let i = 0; i < results.rows.length; ++i)

          temp.push(results.rows.item(i));

        setFlatListItems(temp);

      });

    });

  }, []);

  let listViewItemSeparator = () => {

    return (

      <View

        style={{ height: 0.2, width: '100%', backgroundColor: '#808080' }}

      />

    );

  };

  let listItemView = (item) => {

    return (

      <View

        key={item.user\_id}

        style={{ backgroundColor: 'white', padding: 20 }}>

        <Text>Id: {item.user\_id}</Text>

        <Text>Name: {item.user\_name}</Text>

        <Text>Contact: {item.user\_contact}</Text>

        <Text>Address: {item.user\_address}</Text>

      </View>

    );

  };

  return (

    <SafeAreaView style={{ flex: 1 }}>

      <View style={{ flex: 1, backgroundColor: 'white' }}>

        <View style={{ flex: 1 }}>

          <FlatList

            data={flatListItems}

            ItemSeparatorComponent={listViewItemSeparator}

            keyExtractor={(item, index) => index.toString()}

            renderItem={({ item }) => listItemView(item)}

          />

        </View>

        <Text style={{ fontSize: 18, textAlign: 'center', color: 'grey' }}>

          Example of SQLite Database in React Native

        </Text>

        <Text style={{ fontSize: 16, textAlign: 'center', color: 'grey' }}>

          www.aboutreact.com

        </Text>

      </View>

    </SafeAreaView>

  );

};

export default ViewAllUser;

// Example: Pre-Populated SQLite Database in React Native

// https://aboutreact.com/example-of-pre-populated-sqlite-database-in-react-native

// Screen to view single user

import React, { useState } from 'react';

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

import Mytextinput from './components/Mytextinput';

import Mybutton from './components/Mybutton';

import { openDatabase } from 'react-native-sqlite-storage';

//Connction to access the pre-populated user\_db.db

var db = openDatabase({ name: 'user\_db.db', createFromLocation : 1});

const ViewUser = () => {

  let [inputUserId, setInputUserId] = useState('');

  let [userData, setUserData] = useState({});

  let searchUser = () => {

    console.log(inputUserId);

    setUserData({});

    db.transaction((tx) => {

      tx.executeSql(

        'SELECT \* FROM table\_user where user\_id = ?',

ViewUser

        [inputUserId],

        (tx, results) => {

          var len = results.rows.length;

          console.log('len', len);

          if (len > 0) {

            setUserData(results.rows.item(0));

          } else {

            alert('No user found');

          }

        }

      );

    });

  };

  return (

    <SafeAreaView style={{ flex: 1 }}>

      <View style={{ flex: 1, backgroundColor: 'white' }}>

        <View style={{ flex: 1 }}>

          <Mytextinput

            placeholder="Enter User Id"

            onChangeText={(inputUserId) => setInputUserId(inputUserId)}

            style={{ padding: 10 }}

          />

          <Mybutton title="Search User" customClick={searchUser} />

          <View style={{ marginLeft: 35, marginRight: 35, marginTop: 10 }}>

            <Text>User Id: {userData.user\_id}</Text>

            <Text>User Name: {userData.user\_name}</Text>

            <Text>User Contact: {userData.user\_contact}</Text>

            <Text>User Address: {userData.user\_address}</Text>

          </View>

        </View>

        <Text style={{ fontSize: 18, textAlign: 'center', color: 'grey' }}>

          Example of SQLite Database in React Native

        </Text>

        <Text style={{ fontSize: 16, textAlign: 'center', color: 'grey' }}>

          www.aboutreact.com

        </Text>

      </View>

    </SafeAreaView>

  );

};

export default ViewUser;