=============App.js==============

import React from 'react';

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

import Navegacao from './Componentes/Navegacao';

import EnviaProduto from './Componentes/EnviaProduto';

export default function App() {

return (

<NavigationContainer>

<EnviaProduto>

<Navegacao/>

</EnviaProduto>

</NavigationContainer>

);

}

=============Componentes/EnviaProduto.js==============

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

export const ProdutoContext = createContext({});

function EnviaProduto({ children }) {

const [produto, setProduto] = useState('');

function editaProduto(

inputId,

inputNome,

inputPreco,

inputQtd

) {

setProduto({

id: inputId,

nome: inputNome,

preco: inputPreco,

quantidade: inputQtd

});

return produto;

}

return (

<ProdutoContext.Provider value={{ produto, editaProduto }}>

{children}

</ProdutoContext.Provider>

);

}

export default EnviaProduto;

=============Componentes/Navegacao.js==============

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

import React from 'react';

import Home from '../Telas/Home';

import Formulario from '../Telas/Formulario';

const Stack = createNativeStackNavigator();

function Navegacao() {

return (

<Stack.Navigator>

<Stack.Screen name='Home' component={Home} />

<Stack.Screen name='Formulario' component={Formulario} />

</Stack.Navigator>

);

}

export default Navegacao;

=============Telas/Formulario.js==============

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

import {

StyleSheet,

TextInput,

Text,

View,

TouchableOpacity,

Alert,

} from 'react-native';

import { ProdutoContext } from '../Componentes/EnviaProduto';

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

import \* as SQLite from 'expo-sqlite';

export default function Formulario() {

const navigation = useNavigation();

const { produto } = useContext(ProdutoContext);

const db = SQLite.openDatabase('db.MainDB');

const [id, setId] = useState('');

const [inputNome, setNome] = useState('');

const [inputPreco, setPreco] = useState('');

const [inputQtd, setQtd] = useState('');

useEffect(() => {

setId(produto.id);

if (produto.id != '') {

let preco = produto.preco.toString();

let qtd = produto.quantidade.toString();

setNome(produto.nome);

setPreco(preco);

setQtd(qtd);

} else {

setNome('');

setPreco('');

setQtd('');

}

}, []);

const setData = async () => {

if (inputNome == '') {

Alert.alert(

'ERRO AO CADASTRAR',

'Todos os campos precisam ser preenchidos'

);

} else {

await db.transaction(async (tx) => {

await tx.executeSql(

'INSERT INTO PRODUTO (NOME, PRECO, QUANTIDADE) VALUES (?,?,?)',

[inputNome, inputPreco, inputQtd],

(tx, results) => {

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

if (results.rowsAffected > 0) {

Alert.alert('PARABÉNS', 'Cadastro realizado com sucesso');

} else Alert.alert('ERRO DE CADASTRO', 'Ops, ocorreu um erro.');

}

);

});

navigation.navigate('Home');

}

};

const deleteData = () => {

db.transaction((tx) => {

tx.executeSql('DELETE FROM PRODUTO WHERE ID=?', [id], (tx, results) => {

console.log('Resultado: ', results.rowsAffected);

if (results.rowsAffected > 0) {

Alert.alert('CADASTRO APAGADO', 'Cadastro apagado com sucesso.');

navigation.navigate('Home');

} else Alert.alert('ERRO AO APAGAR', 'Ops, ocorreu um erro.');

});

});

};

const updateData = () => {

db.transaction((tx) => {

tx.executeSql(

'UPDATE PRODUTO SET NOME=?, PRECO=?, QUANTIDADE=? WHERE ID=?',

[inputNome, Number(inputPreco), Number(inputQtd), id],

(tx, results) => {

console.log('Resultado: ', results.rowsAffected);

if (results.rowsAffected > 0) {

Alert.alert('CADASTRO EDITADO', 'Cadastro editado com sucesso.');

navigation.navigate('Home');

} else Alert.alert('ERRO AO EDITAR', 'Ops, ocorreu um erro.');

}

);

});

};

return (

<View style={styles.container}>

<View style={styles.inputContainer}>

<TextInput

style={styles.entrada}

placeholder="Descrição"

value={inputNome}

onChangeText={(text) => setNome(text)}

/>

<TextInput

style={styles.entrada}

placeholder="Preço"

value={inputPreco}

onChangeText={(num) => setPreco(num)}

/>

<TextInput

style={styles.entrada}

placeholder="Quantidade"

value={inputQtd}

onChangeText={(num) => setQtd(num)}

/>

{id == '' ? (

<TouchableOpacity style={styles.botao} onPress={setData}>

<Text style={styles.textoBotao}>Novo Produto</Text>

</TouchableOpacity>

) : null}

{id != '' ? (

<TouchableOpacity style={styles.botao} onPress={updateData}>

<Text style={styles.textoBotao}>Editar Produto</Text>

</TouchableOpacity>

) : null}

{id != '' ? (

<TouchableOpacity style={styles.botao} onPress={deleteData}>

<Text style={styles.textoBotao}>Apagar Produto</Text>

</TouchableOpacity>

) : null}

</View>

</View>

);

}

const styles = StyleSheet.create({

container: {

flex: 1,

backgroundColor: 'white',

alignItems: 'center',

marginTop: 30,

},

inputContainer: {

width: 300,

},

entrada: {

textAlign: 'center',

borderWidth: 2,

marginBottom: 3,

fontSize: 20,

},

botao: {

alignItems: 'center',

backgroundColor: '#FFA500',

padding: 10,

marginTop: 10,

marginBottom: 5,

borderRadius: 20,

fontSize: 20,

},

textoBotao: {

color: 'white',

fontWeight: 'bold',

textTransform: 'uppercase',

},

});

=============Telas/Home.js==============

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

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

import {

StyleSheet,

View,

Text,

FlatList,

TouchableOpacity,

} from 'react-native';

import \* as SQLite from 'expo-sqlite';

import { ProdutoContext } from '../Componentes/EnviaProduto';

export default function Home() {

const { editaProduto } = useContext(ProdutoContext);

const navigation = useNavigation();

const db = SQLite.openDatabase('db.MainDB');

const isFocused = useIsFocused();

const [produto, setProduto] = useState([]);

const [empty, setEmpty] = useState([]);

useEffect(() => {

getData();

createTable();

resetProduto();

}, [isFocused]);

const createTable = () => {

db.transaction((tx) => {

tx.executeSql(

'CREATE TABLE IF NOT EXISTS ' +

'PRODUTO ' +

'(ID INTEGER PRIMARY KEY AUTOINCREMENT, NOME TEXT, PRECO NUMERIC, QUANTIDADE NUMERIC);'

);

});

};

const getData = () => {

db.transaction((tx) => {

tx.executeSql(

'SELECT ID, NOME, PRECO, QUANTIDADE FROM PRODUTO',

[],

(tx, results) => {

var temp = [];

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

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

setProduto(temp);

if (results.rows.length >= 1) {

setEmpty(false);

} else {

setEmpty(true);

}

}

);

});

};

const enviaProdutoForm = (id, nome, preco, quantidade) => {

navigation.navigate('Formulario');

editaProduto(id, nome, preco, quantidade);

};

const resetProduto = () => {

editaProduto('', '', '', '');

};

return (

<View style={styles.container}>

<FlatList

style={styles.lista}

data={produto}

showsVerticalScrollIndicator={false}

renderItem={({ item }) => (

<TouchableOpacity

style={styles.item}

onPress={() =>

enviaProdutoForm(

item.ID,

item.NOME,

item.PRECO,

item.QUANTIDADE

)

}>

<Text> ID: {item.ID} </Text>

<Text> DESCRICAO: {item.NOME} </Text>

<Text> PRECO: {item.PRECO.toFixed(2)} </Text>

<Text> QUANTIDADE: {item.QUANTIDADE} </Text>

</TouchableOpacity>

)}

/>

<TouchableOpacity

style={styles.floatbutton}

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

<Text style={{ fontSize: 30, color: 'white' }}>+</Text>

</TouchableOpacity>

</View>

);

}

const styles = StyleSheet.create({

container: {

flex: 1,

backgroundColor: 'white',

alignItems: 'center',

justifyContent: 'center',

},

item: {

borderBottomWidth: 2,

padding: 3,

},

floatbutton: {

borderWidth: 1,

alignItems: 'center',

justifyContent: 'center',

width: 50,

position: 'absolute',

bottom: 10,

right: 10,

height: 50,

backgroundColor: '#FFA500',

borderRadius: 100,

},

lista: {

marginTop: 25,

},

});