**TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC**



**KHOA CÔNG NGHỆ THÔNG TIN**

**🙜🕮🙞**

****

**BÁO CÁO MÔN HỌC**

**Lập trình di động 3**

**Giảng viên hướng dẫn:** Trương Bá Thái

**Nhóm sinh viên thực hiện**

* + - * Nguyễn Ngọc Sơn 16211TT0917
      * Lê Quốc Dũng 16211TT3128

*Tp. Hồ Chí Minh*, ngày 31 tháng 10 năm 2019

Menu

[CHƯƠNG I – TỔNG QUÁT 4](#_Toc23403244)

[1 Giới thiệu đề tài đồ án 4](#_Toc23403245)

[2 Màn hình mockup 4](#_Toc23403246)

[CHƯƠNG 2: ES6 12](#_Toc23403247)

[1 Kiến thức 12](#_Toc23403248)

[2 Ví dụ 15](#_Toc23403249)

[CHƯƠNG 3: COMPONENT 17](#_Toc23403250)

[1 Kiến thức 17](#_Toc23403251)

[2 Ví dụ 19](#_Toc23403252)

[CHƯƠNG 4: API 26](#_Toc23403253)

[1 Ví dụ 26](#_Toc23403254)

[2 Ví dụ 30](#_Toc23403255)

[CHƯƠNG 5: XÂY DỰNG ĐỒ ÁN 36](#_Toc23403256)

[1 Màn hình đăng nhập (Nguyễn Ngọc Sơn) 36](#_Toc23403257)

[2 Màn hình đăng ký (Nguễn Ngọc Sơn) 40](#_Toc23403258)

[3 Màn hình danh sách sản phẩm (Lê Quốc Dũng) 43](#_Toc23403259)

[4 Màn hình thêm sản phẩm (Lê Quốc Dũng) 48](#_Toc23403260)

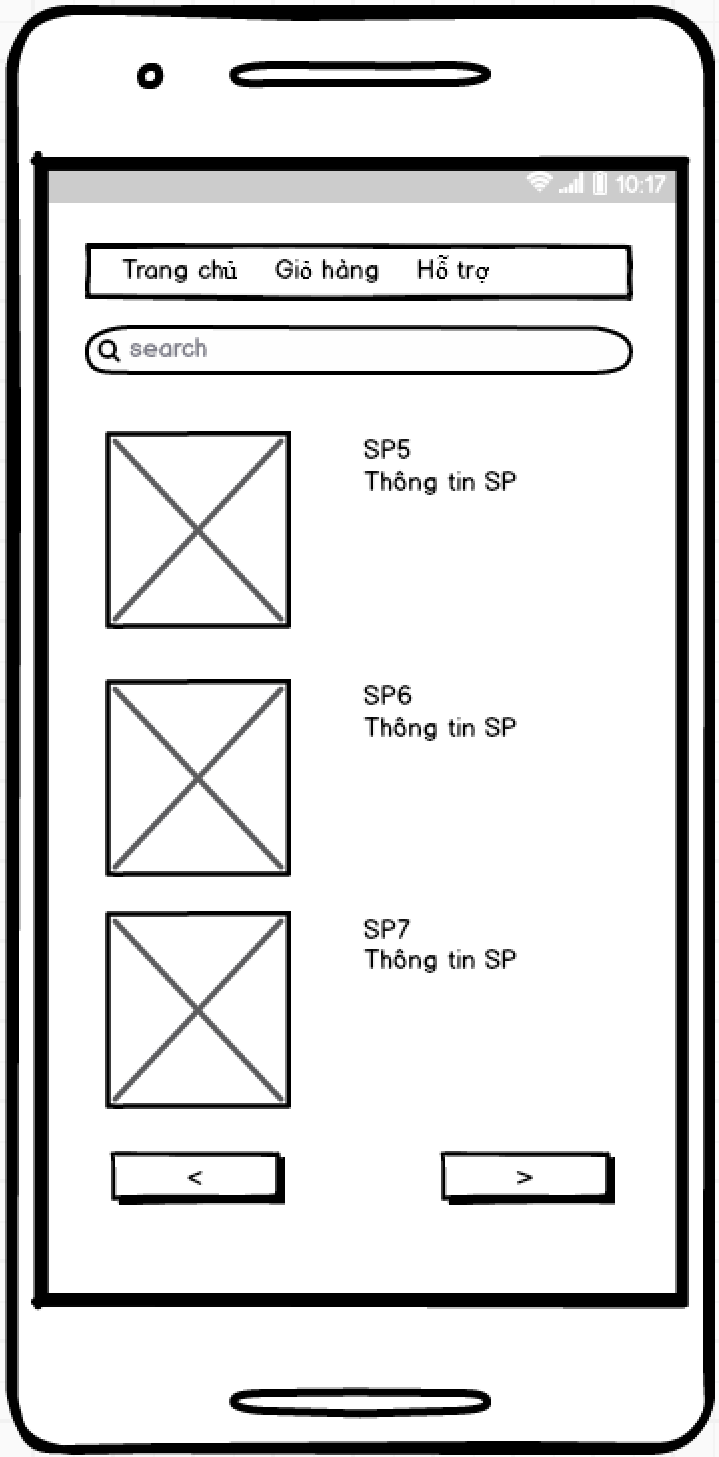
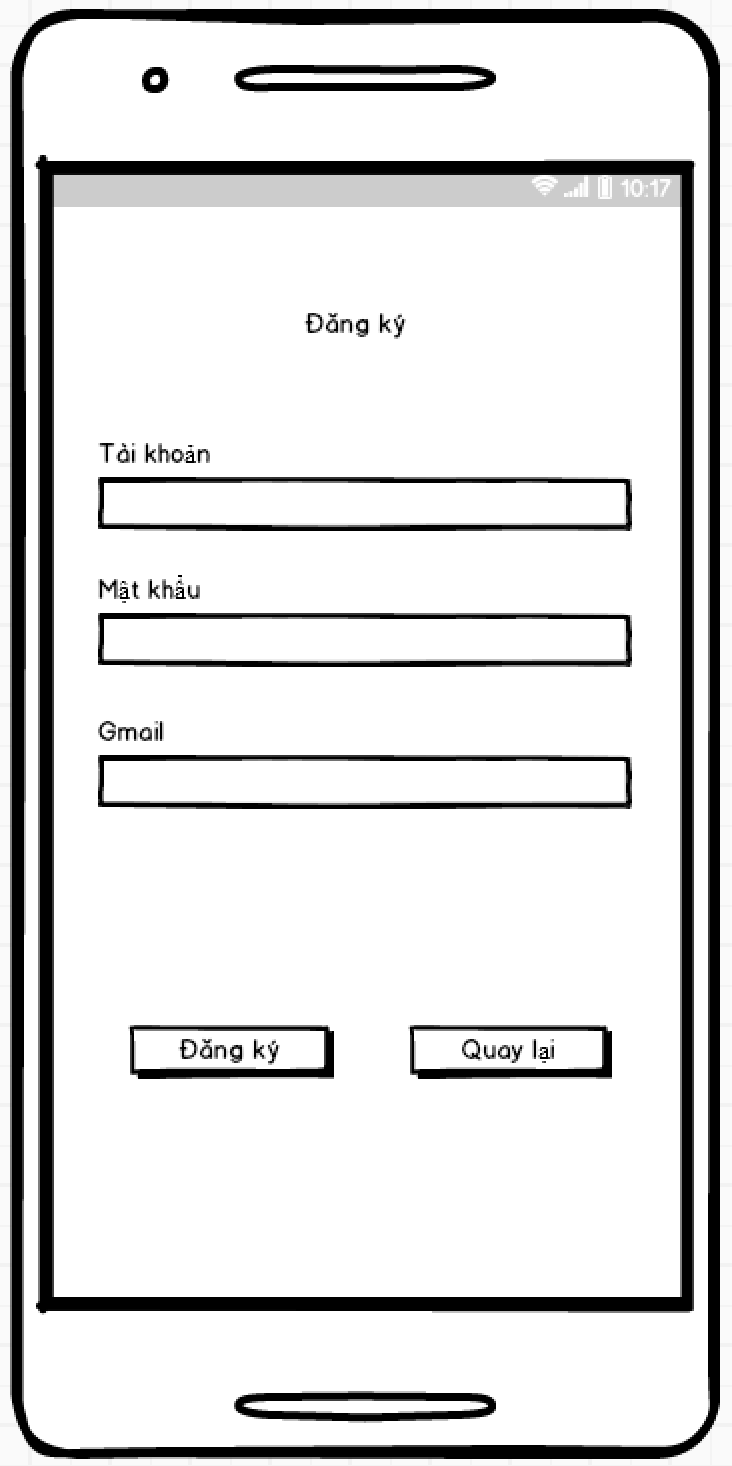
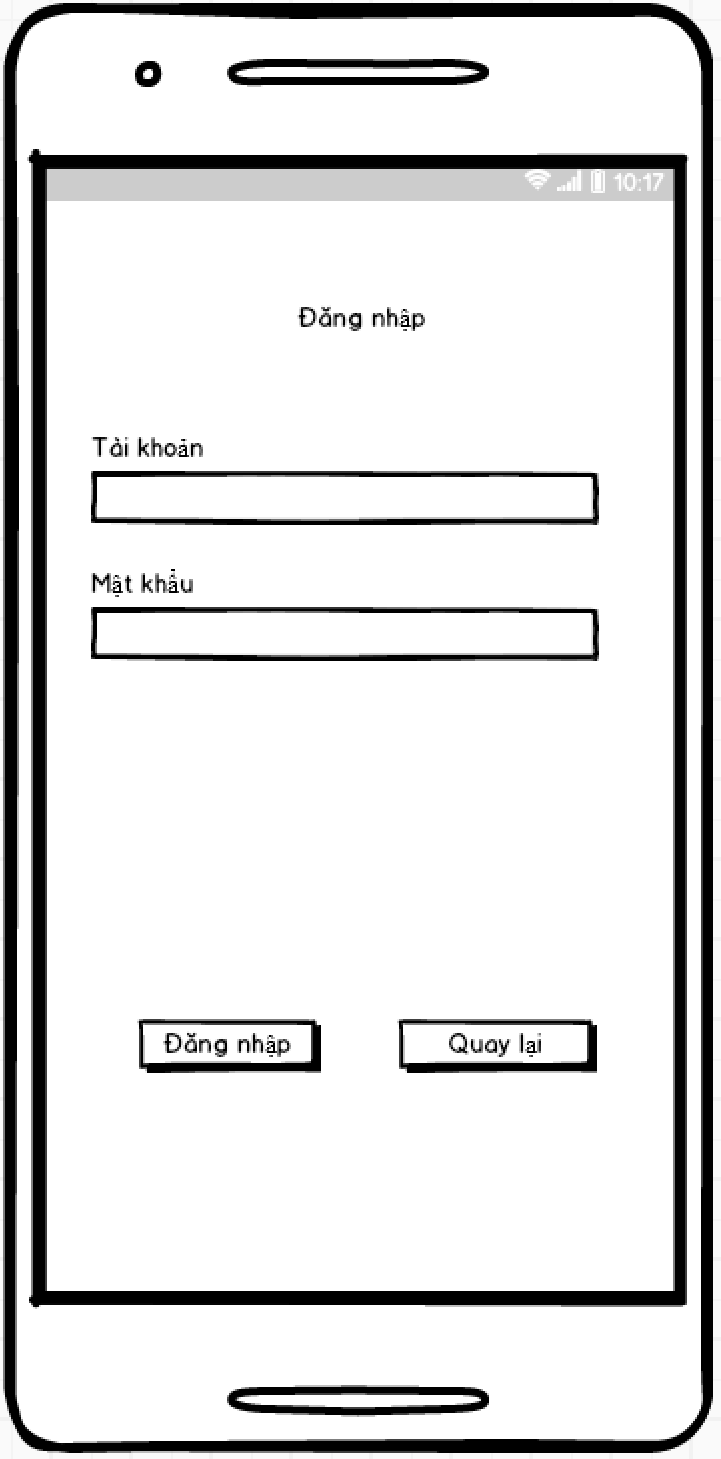
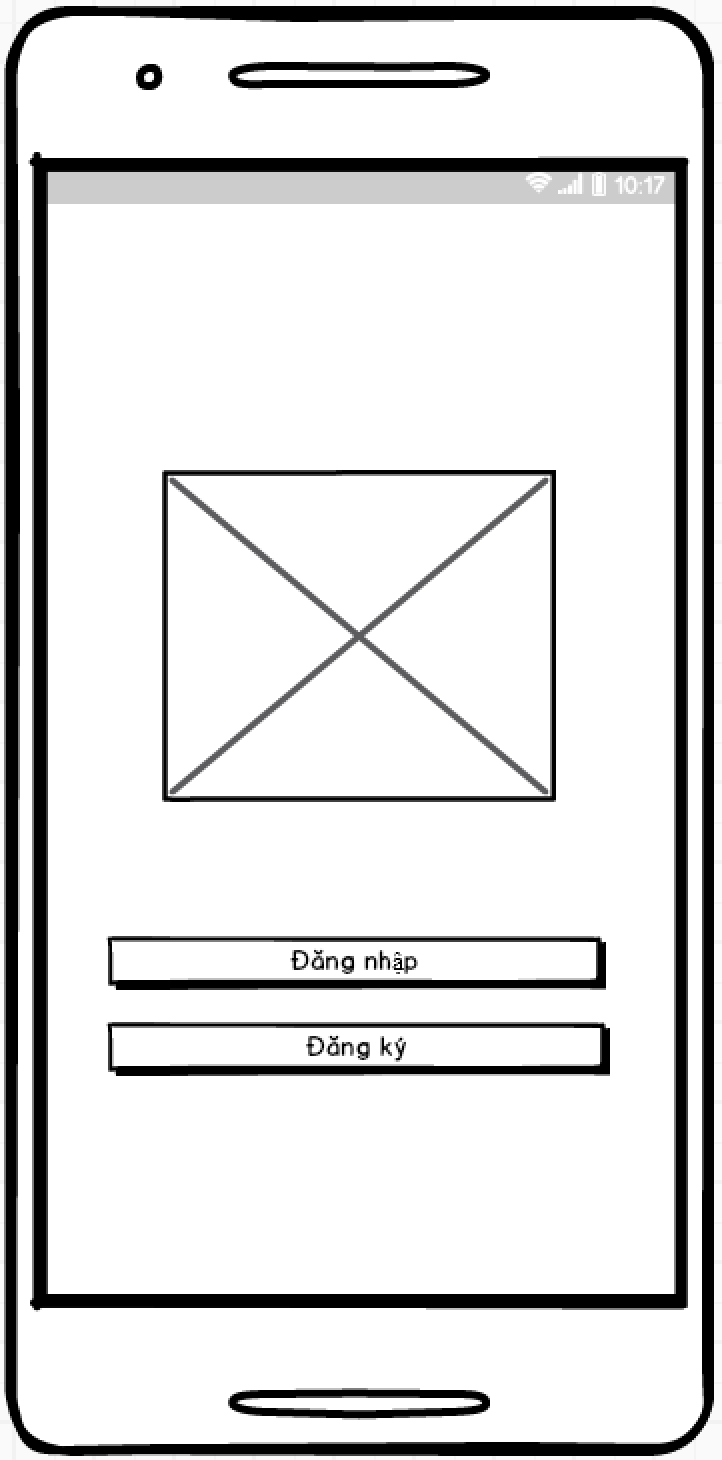
[CHƯƠNG 5: kết luận và kiến nghị 52](#_Toc23403261)

# CHƯƠNG I – TỔNG QUÁT

## 1 Giới thiệu đề tài đồ án

* Vận dụng những kiến thức trong quá trình học, nhóm 19 lựa chọn đề tài app bán bánh, trong app có các màn hình chính như màn hình trang chủ, màn hình hiện thị danh sách sản phẩm, màn hình chi tiết sản phẩm, màn hình thêm sản phẩm, màn hình đăng ký, đăng nhập.... Dữ liệu được lấy từ API của web service. Do kiến thức của chúng em còn hạn hẹp nên nhóm em chưa hoàn thiện được hết tất cả các chức năng của app cũng như không tránh được những sai xót trong quá trình xây dựng đồ án. Mong thầy thông cảm và đóng góp ý kiến cho đồ án nhóm em.

## 2 Màn hình mockup



Màn hình chính(Ngọc Sơn) (Sẽ có 2 chức năng chính để hiển thị đến người dùng):

+ Chức năng đăng nhập(dành cho những người đã có tài khoản).

+ Chức năng đang ký(dành cho những người chưa có tài khoản), buộc người dung phải đăng ký để có tài khoản mới sử dụng được app.

Màn hình đăng nhập (Ngọc Sơn)(Hiển thị 2 text box là tài khoản và mật khẩu):

+ Nút đăng nhập(Người dùng sẽ ấn vào nếu đã nhập tài khoản và mật khẩu).

+ Nút quay lại(Người dùng sẽ ấn vào nếu không muốn đăng nhập nữa để quay lại màn hình chính).

Màn hình đăng ký (Ngọc Sơn)(Hiển thị 3 text box là tài khoản, mật khẩu và gmail):

+ Nút đăng ký(Người dùng sẽ ấn vào nếu đã nhập tài khoản, mật khẩu và gmail).

+ Nút quay lại(Người dùng sẽ ấn vào nếu không muốn đăng nhập nữa để quay lại màn hình chính).

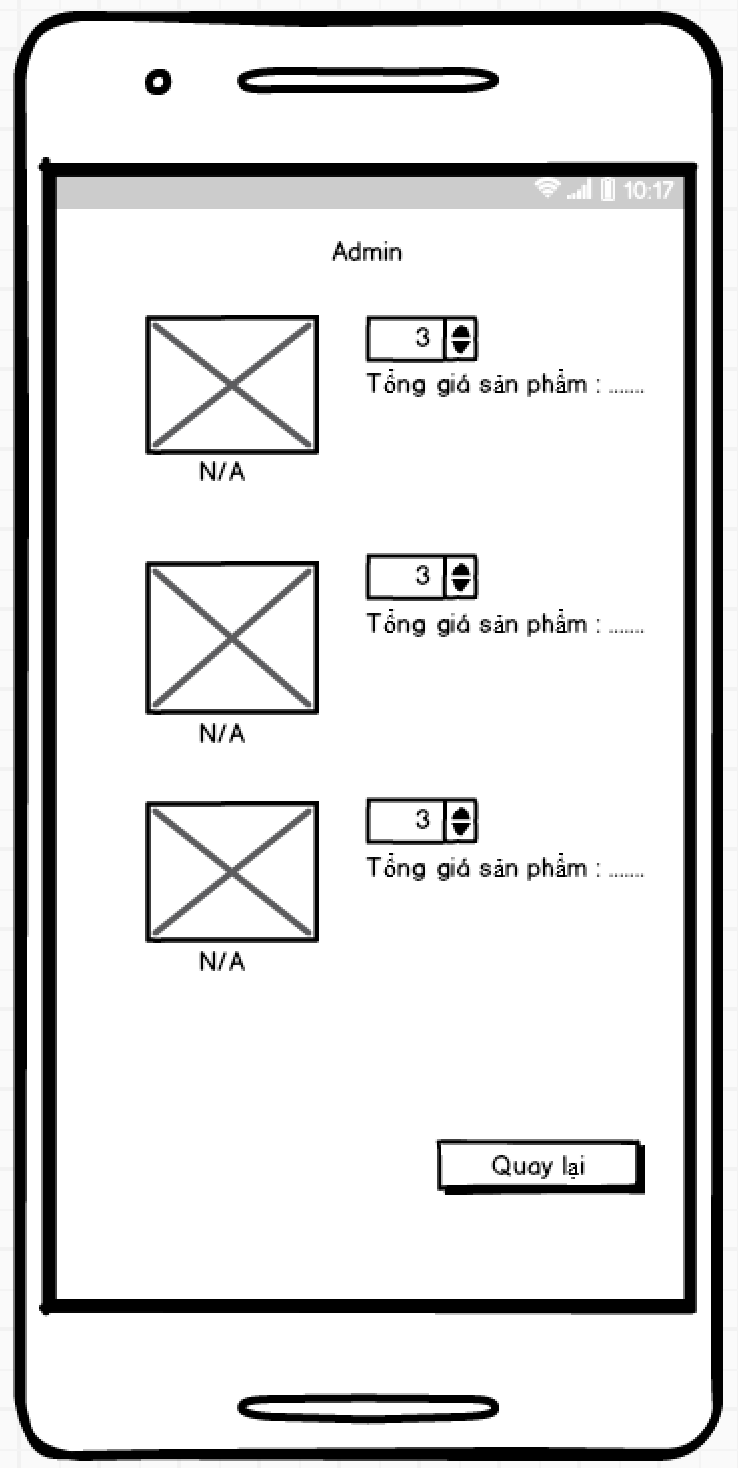
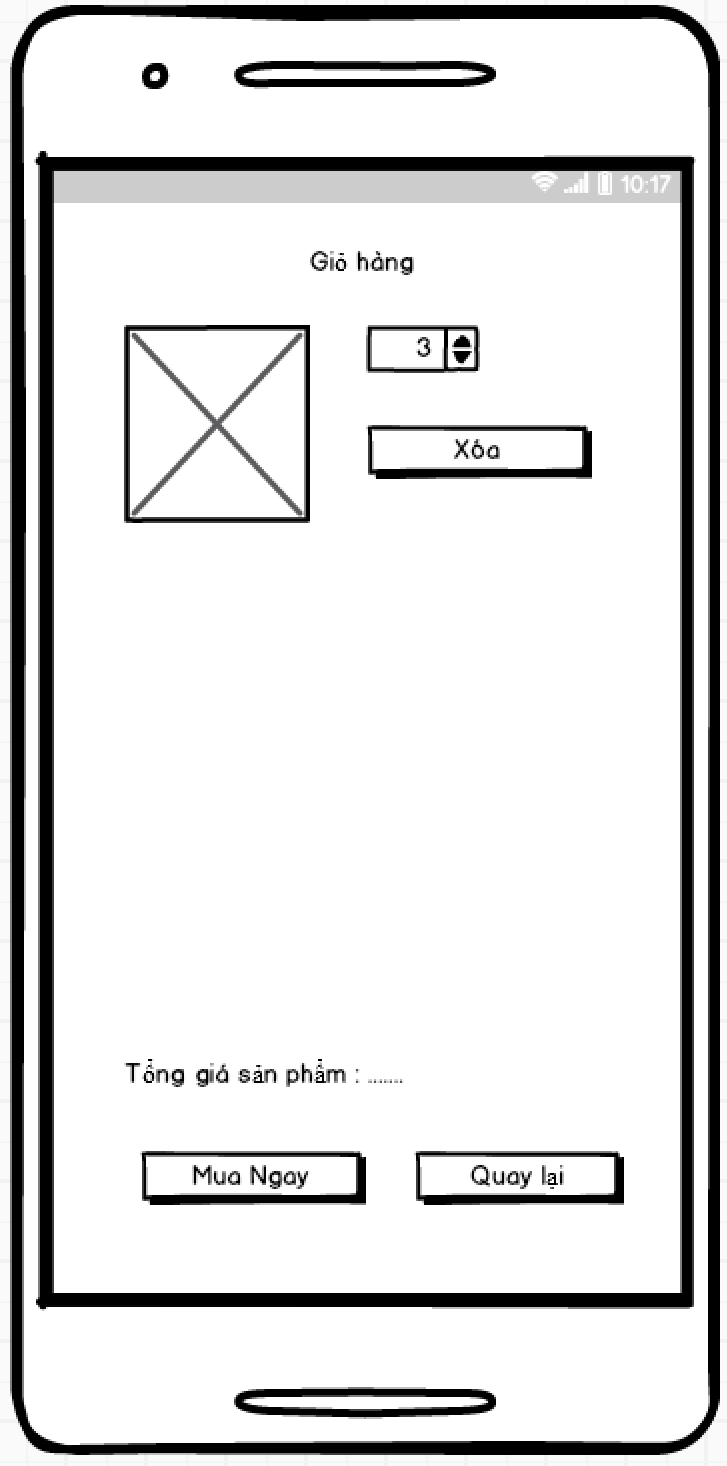
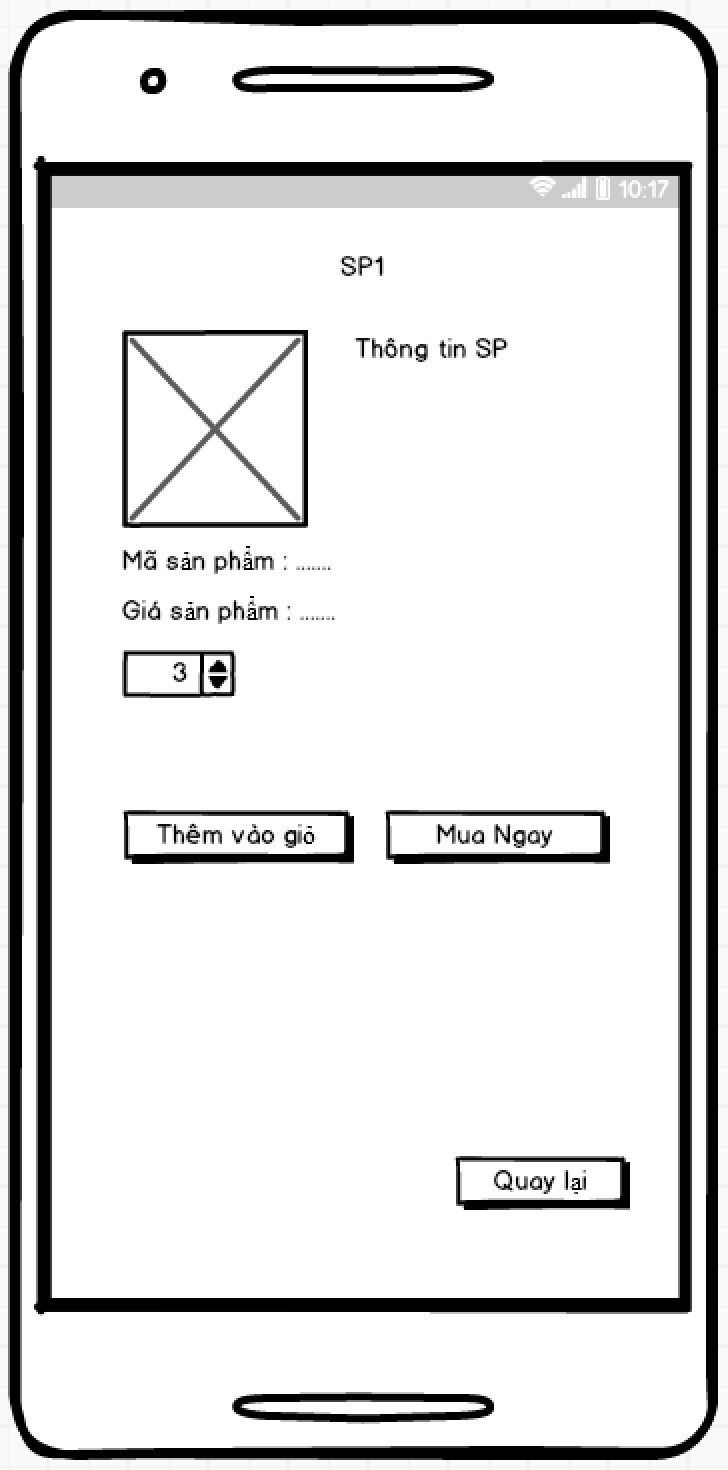
Màn hình trang chủ (Quốc Dũng)(Hiển thị các chức năng như: Giỏ hàng, hỗ trợ, tìm kiếm và một số hình ảnh cũng như là thông tin của sản phẩm):

+ Khi ấn vào chức năng giỏ hàng thì màn hình sẽ chuyển đến trang giỏ hàng.

+ Khi ấn vào chức năng hỗ trợ thì màn hình sẽ chuyển đến trang hỗ trợ khách hàng.

+ Nút quay lại(Khi người dùng ấn vào thì màn hình chuyển về màn hình chính).

+ Nút trang kế(Khi người dùng ấn vào sẽ chuyển đến trang sản phẩm tiếp theo).



Màn hình thông tin sản phẩm (Quốc Dũng)(Hiển thị các thông tin của sản phẩm như là: Mã sản phẩm, giá của sản phẩm):

+ Khi ấn vào chức năng thêm vào giỏ hàng thì màn hình xuất hiện thông báo (đã thêm thành công).

+ Khi ấn vào chức năng mua ngay(Màn hình chuyển đến trang giỏ hàng để hiển thị tổng giá tiền mà người mua hàng phải trả).

+ Nút quay lại(Khi người dùng ấn vào thì màn hình chuyển về màn hình trang chủ để hiển thị các sản phẩm khác).

Màn hình giỏ hàng (Quốc Dũng)(Hiển thị các sản phẩm mà người dùng đã chọn):

+ Khi ấn vào chức năng xóa(Sản phẩm sẽ bị xóa khỏi giỏ hàng).

+ Khi ấn vào chức năng mua ngay(Màn hình sẽ xuất hiện thông báo xác nhận đơn hàng).

+ Nút quay lại(Khi người dùng ấn vào thì màn hình chuyển về màn hình trang chủ để hiển thị các sản phẩm khác).

Màn hình Amin (Quốc Dũng)(Hiển thị các sản phẩm mà người quản lý app muốn chỉnh sửa các thông tin của nó):

+ Nút quay lại(Khi người dùng ấn vào thì màn hình chuyển về màn hình trang chủ để hiển thị các sản phẩm khác).

|  |  |
| --- | --- |
| User:  + Đăng nhập, đăng ký.  + Đặt hang sản phẩm.  + Xem giỏ hàng. | Admin:  + Đăng nhập.  + Quản lý thông tin sản phẩm. |

# CHƯƠNG 2: ES6

## 1 Kiến thức

* **Variable**
* **var:** với từ khóa *var* chúng ta có thể khai báo đa dạng các kiểu biến như number, string, boolean, etc. Biến var sẽ có scope là globally scoped trừ trường hợp được khai báo bên trong 1 function (khi đó biến var sẽ có scope là local). Đặc biệt, biến var còn có thêm tính chất hoisting: nghĩa là dù khai báo ở đâu thì biến đều sẽ được đem lên đầu scope trước khi code được thực hiện. Hạn chế của biến var là sự không rõ ràng, dễ nhầm lẫn cho lập trình viên vì thế với bản update ES6 ngôn ngữ Javascript đã được cộng đồng lập trình viên trên thế giới đánh giá cao vể nhiều mặt. Một trong những thay đổi đầu tiên và được hoan nghênh nhất trong phiên bản cập nhật ES6 là bổ sung thêm từ khóa let/const trong khai bao biến. Với let/const giờ đây các lập trình viên có thêm tùy chọn hợp lý hơn so với việc phải dùng var trong tất cả các trường hợp.
* **let:**là từ khóa định nghĩa 1 biến có phạm vi truy cập trong 1 block – khối code. Biến let có phạm vi trong dấu 1 cặp dấu {} bao quanh nó
* **const:** là từ khóa định nghĩa 1 biến sẽ là hằng số. Biến const lưu trữ giá trị không thể thay đổi được trong suốt vòng đời của biến.
* **Array**
* Mảng là một tập hợp các phần tử lại và mỗi phần tử sẽ được đánh dấu một vị trí trong tập hợp đó. Trong javascript nếu mảng có 10 phần tử thì các phần tử sẽ được đánh dấu từ 0 -9. Array giúp chúng ta quản lý dữ liệu thống nhất, không rời rạc.
* Cấu trúc khai báo:
* *Khai báo theo kiểu new Array()*

var name\_array = new **Array**();

*// Hoặc*

var name\_array = new **Array**(1,2,3);

* *Khai báo với cặp dấu ngoặc vuông []*

var name\_array = [];

*// Hoặc*

var name\_array = [1,2,3];

* Truy cập phần tử trong mảng theo cú pháp:

var firstItem = name\_array[0];

var secondItem = name\_array[1];

* Duyệt mảng:

name\_array.**forEach**(function(item, index, array){

    console.**log**(item, index);

});

*//1 0*

*//2 1*

*//3 2*

*//hoặc nếu không xử lý mảng thì có thể duyệt*

*theo cách dưới*

console.**log**(`${name\_array}`); //1,2,3

* Thêm phần tử vào cuối:

var newLength = name\_array**.push**(4); // 1,2,3,4

* Thêm phần tử vào đầu:

 var newLength = name\_array**.unshift**(0); //0,1,2,3

* Xóa phần tử cuối:

 var newLength = name\_array**.pop**(); //1,2

* Xóa phần tử đầu:

 var newLength = name\_array**.shift**(); //2,3

* Array.map():

Phương thức map () tạo ra một mảng mới với kết quả gọi hàm được cung cấp trên mọi phần tử trong mảng gọi. Với cú pháp:

var new\_array = arr.**map**(function **callback**(currentValue[, index[, array]]) {

*// Return element for new\_array*

}

* **Arrow function**

Arrow functon trong ES6 là kiểu cú pháp giúp đơn giản hoá việc việc định nghĩa hàm. Cách định nghĩa một hàm JavaScript theo cách thông thường trước đây:

**var** sumNumbers = **function** (a, b) {

**return** a + b;

}

Với tính năng arrow function trong ES6 chúng ta có thể viết lại đoạn code trên như sau:

**const** sumNumbers = (a, b) => {

**return** a + b;

}

* Từ khoá function được bỏ đi
* Thêm ký tự => (gần giống với mũi tên) đặt giữa dấu ngoặc kết thúc danh sách tham số ) và dấu ngoặc bắt đầu logic của hàm {

Ngoài ra chúng ta thay từ khoá var bằng const để quy định rằng giá trị của sumNumbers sẽ không thay đổi (tất nhiên bạn có thể dùng let nếu muốn).

Đối với các hàm được định nghĩa chỉ với một tham số thì bạn có thể bỏ qua cặp dấu ngoặc ():

**let** doubleNumber = a => {

**return** a \* 2;

}

Tuy nhiên nếu hàm không có đối số nào thì chúng ta vẫn phải sử dụng ():

**let** sayHello => () {

console.log("Hello everyone!");

}

## 2 Ví dụ

* **Ví dụ về es6 (Lê Quốc Dũng)**

let  **xuLyMang** = num =>

{

*//nhân đôi các phần tử mảng*

  const myNewDoubleArray = num.**map**(x => x \* 2);

  console.**log**(`Mảng sau khi nhân đôi: ${myNewDoubleArray}`);

*//Mảng sau khi nhân đôi: 2,4,6,8,10,12,14,16,18,20*

  var lastItem = myNewDoubleArray.length - 1;*//20*

  if(myNewDoubleArray[lastItem] != 0)

  {

*//phan tử cuối khác không thì thêm số 0 vào cuối mảng*

    var newLength = myNewDoubleArray**.push**(0);

  }

  else

  {

*//nếu phần tử cuối bằng 0 thì xóa phần tử ở cuối mảng*

    var newLength = myNewDoubleArray**.pop**();

  }

  console.**log**(`Mảng sau khi xử lý: ${myNewDoubleArray}`);

*// Mảng sau khi xử lý: 2,4,6,8,10,12,14,16,18,20,0*

}

let arr = [1,2,3,4,5,6,7,8,9,10];

**xuLyMang**(arr);

* **Ví dụ es6 (Nguyễn Ngọc Sơn)**

var set = new Set();

// Add key

set.add({x: 1});

set.add(2);

set.add("text");

console.log(set.has("text")); // kiếm tra xem có key == 'text' hay không.

set.delete(1); // xóa một key

var set\_1 = new Set([1, 2, 3, 4]); // khởi tạo một set bằng dữ liệu là một mảng

console.log(set\_1.size); // in ra size của 1 set "5"

* var set\_2 = new Set(set\_1.values()); // sao chép các giá trị key từ một set khác.

# CHƯƠNG 3: COMPONENT

## 1 Kiến thức

* **View:**

Là thành phần cơ bản trong xây dựng giao diện, View là một khung chứa được tùy chỉnh bởi các thông số flexbox, style, xử lý cảm ứng và kiểm soát khả năng truy cập. Khi ứng dụng được chạy trên nền React-native trên thiết bị iOS hay Android, View sẽ được biên dịch tương ứng sao cho phù hợp với hệ điều hành trên thiết bị đang chạy.

View được thiết kế có thể lồng nhiều View khác hoặc nhiều Component khác.

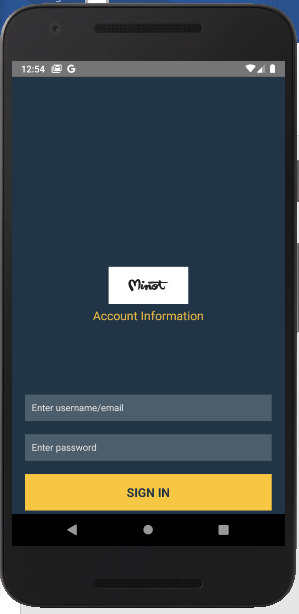
* **Text:**

Một thành phần React để hiển thị văn bản.Văn bản hỗ trợ lồng, tạo kiểu và xử lý.

* **TouchableHighlight:**
* Cung cấp các xử lý đa dạng hơn 1 Button bình thường như nó có thể lồng các Component khác như Image hoặc Text vào .Nền màn hình sẽ bị tối khi người dùng nhấn xuống nút.
* **Image:**
* Component hiển thị hình ảnh và cung cấp 1 số xử lý tương tự như Text hoặc View
* **TextInput:**
* Là UI nhận dữ liệu do người dùng nhập vào, textInput cũng cung cấp 1 số xử lý cơ bản. Trường hợp sử dụng đơn giản nhất là sự kiện onChangeText để đọc dữ liệu nhập của người dùng. Ngoài ra còn có các sự kiện khác, chẳng hạn như onSubmitEditing và onFocus.
* **ImageBackground:** giúp hiển thị nền giao diện là một hình ảnh từ local hoặc network.
* **Flatlist**: là một cách dễ dàng để tạo một list của data. Không chỉ hiệu quả mà còn có một API cực kỳ đơn giản để làm việc. Flatlist không còn phải định dạng dữ liệu – mà là truyền cho nó một array data và hiển thị lại giúp nâng cao hiệu xuất so với ScrollView thông thường. Flatlist hỗ trợ các tính năng tiện dụng nhất:
* Đa nền tảng.
* Chế độ ngang tùy chọn.
* Cấu hình khả năng callback.
* Hỗ trợ tiêu đề.
* Hỗ trợ chân trang.
* Hỗ trợ tách.
* Kéo để làm mới.
* Cuộn tải.
* Hỗ trợ ScrollToIndex.

## 2 Ví dụ

Nguyễn Ngọc Sơn



Code:

import React, {Component} from 'react';

import {

  StyleSheet,

  Text,

  View,

  Image,

  TouchableWithoutFeedback,

  StatusBar,

  TextInput,

  SafeAreaView,

  Keyboard,

  TouchableOpacity,

  KeyboardAvoidingView,

} from 'react-native';

import {red} from 'ansi-colors';

export default class Login extends Component {

  render() {

    return (

      // container

      <SafeAreaView style={styles.container}>

        <StatusBar barStyle="light-content" />

        <KeyboardAvoidingView behavior="padding" style={styles.container}>

          <TouchableWithoutFeedback

            style={styles.container}

            onPress={Keyboard.dismiss}>

            <View style={styles.logoContainer}>

              <View style={styles.logoContainer}>

                <Image

                  style={styles.logo}

                  source={require('../Images/logo.png')}></Image>

                <Text style={styles.title}>Account Information</Text>

              </View>

            </View>

          </TouchableWithoutFeedback>

        </KeyboardAvoidingView>

        <TouchableWithoutFeedback

          style={styles.container}

          onPress={Keyboard.dismiss}>

          <View style={styles.infoContainer}>

            <TextInput style={styles.input}

              placeholder="Enter username/email"

              placeholderTextColor="rgba(255,255,255,0.8)"

              keyboardType="email-address"

              returnKeyType="next"

              autoCorrect={false}

              onSubmitEditing={()=> this.refs.txtPassword.focus()}

            />

            <TextInput style={styles.input}

              placeholder="Enter password"

              placeholderTextColor="rgba(255,255,255,0.8)"

              returnKeyType="go"

              secureTextEntry

              autoCorrect={false}

              ref={"txtPassword"}

            />

            <TouchableOpacity style={styles.buttonContainer}>

                <Text style={styles.buttonText}>SIGN IN</Text>

            </TouchableOpacity>

          </View>

        </TouchableWithoutFeedback>

      </SafeAreaView>

    );

  }

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: 'rgb(32,53,70)',

    flexDirection: 'column',

  },

  logoContainer: {

    alignItems: 'center',

    justifyContent: 'center',

    flex: 1,

  },

  logo: {

    width: 120,

    height: 56,

  },

  title: {

    color: '#f7c744',

    fontSize: 18,

    textAlign: 'center',

    marginTop: 5,

    opacity: 0.9,

  },

  infoContainer: {

    position: 'absolute',

    left: 0,

    right: 0,

    bottom: 0,

    height: 200,

    padding: 20,

  },

  input: {

    height: 40,

    backgroundColor: 'rgba(255,255,255,0.2)',

    color: '#FFF',

    marginBottom: 20,

    paddingHorizontal: 10,

  },

  buttonContainer:{

      backgroundColor:'#f7c744',

      paddingVertical:15

  },

  buttonText:{

      textAlign:'center',

      color: 'rgb(32, 53,70)',

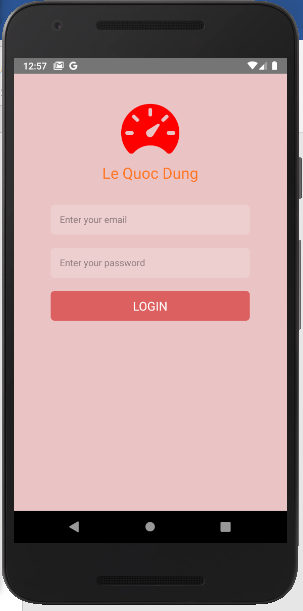
      fontWeight: 'bold',

      fontSize: 18

  }

});

Lê Quốc Dũng



Code:

import React, { Component } from 'react'

import {

    StyleSheet,

    Text,

    View,

    Image,

    TouchableWithoutFeedback,

    StatusBar,

    TextInput,

    Keyboard,

    TouchableOpacity,

  } from 'react-native';

import { red } from 'ansi-colors';

import IOnicons from 'react-native-vector-icons/Ionicons'

import FontAwesome from 'react-native-vector-icons/FontAwesome'

export default class Loginfb extends Component {

    render() {

        return (

          <TouchableWithoutFeedback onPress={Keyboard.dismiss}>

            <View style={styles.container}>

            <StatusBar barStyle="light-content" />

              <View style={styles.up}>

                <IOnicons

                  name="ios-speedometer"

                  size={100}

                  color={'red'}></IOnicons>

                <Text style={styles.title}>Le Quoc Dung</Text>

              </View>

              <View style={styles.down}>

                <View style={styles.textInputContainer}>

                  <TextInput

                    style={styles.textInput}

                    textContentType="emailAddress"

                    keyboardType="email-address"

                    placeholder="Enter your email"></TextInput>

                </View>

                <View style={styles.textInputContainer}>

                  <TextInput

                    style={styles.textInput}

                    placeholder="Enter your password"

                    secureTextEntry={true}></TextInput>

                </View>

                <TouchableOpacity style={styles.loginButton}>

                  <Text style={styles.loginButtonTitle}>LOGIN</Text>

                </TouchableOpacity>

              </View>

            </View>

          </TouchableWithoutFeedback>

        );

    }

}

const styles = StyleSheet.create({

    container:{

        flex: 1,

        flexDirection:'column',

        justifyContent:'center',

        alignItems:'stretch',

        backgroundColor:'rgb(234, 195, 196)',

    },

    up:{

        flex:3,

        flexDirection:'column',

        justifyContent:'center',

        alignItems:'center',

    },

    down:{

        flex:7,

        flexDirection:'column',

        justifyContent:'flex-start',

        alignItems:'center'

    },

    title:{

        color:'white',

        textAlign:'center',

        color:'rgb(255,119,34)',

        width:400,

        fontSize:23

    },

    textInputContainer:{

        paddingHorizontal:10,

        borderRadius:6,

        marginBottom:20,

        backgroundColor:'rgba(255,255,255,0.2)'

    },

    textInput:{

        width:280,

        height:45,

    },

    loginButton:{

        width:300,

        height:45,

        borderRadius:6,

        justifyContent:'center',

        alignItems:'center',

        backgroundColor:'rgb(221, 97,97)'

    },

    loginButtonTitle:{

        fontSize:18,

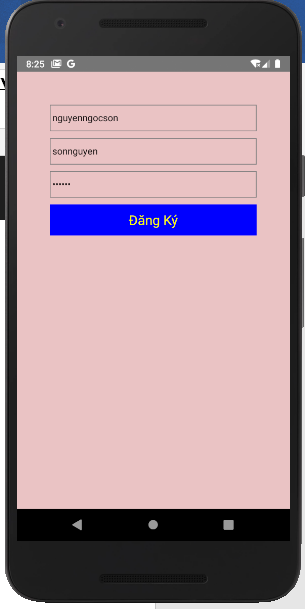
        color:'white'

    }

})

# CHƯƠNG 4: API

## 1 Ví dụ (Nguyễn Ngọc Sơn)



Code:

Kết nối sql

<?php

$json = file\_get\_contents("php://input");

$obj = json\_decode($json, TRUE);

$hoten = $obj["HOTEN"];

$username = $obj["USERNAME"];

$password = $obj["PASSWORD"];

$dbhandle = mysqli\_connect("localhost", "root", "");

$selected = mysqli\_select\_db($dbhandle, "project");

$result = mysqli\_query($dbhandle, "SET NAMES 'utf8");

$qr = "INSERT INTO taikhoan VALUES(

    null, '$hoten', '$username', '$password'

    )";

    mysqli\_query($dbhandle,$qr);

?>

Code chương trình

/\*Nguyễn Ngọc Sơn \*/

import React, {Component} from 'react';

import {

View, Text,StyleSheet,TextInput, TouchableOpacity

} from 'react-native';

import {red} from 'ansi-colors';

export default class DangKy extends Component {

    constructor(props){

        super(props);

        this.state={

            HOTEN:"",

            USERNAME:"",

            PASSWORD:"",

            resulf:"...",

        }

    }

    clickPlus(){

        fetch('http://192.168.129.180:81/Project\_dd3/dangky.php',{

            method:"POST",

            headers: {

                Accept: 'application/json',

                'Content-Type': 'application/json',

              },

            body: JSON.stringify({

                "HOTEN":this.state.HOTEN,

                "USERNAME":this.state.USERNAME,

                "PASSWORD":this.state.PASSWORD,

            })

        })

        .then((response)=>response.json())

        .then((reponseJson)=>{

            this.setState({resulf:reponseJson.id})

        })

        .catch((error)=>{console.log(error)});

    }

  render() {

    return (

        <View style={styles.wrapper}>

            <TextInput

            style={{height:40, borderColor:'gray', borderWidth:1}}

            onChangeText={(HOTEN) => this.setState({HOTEN})}

            placeholder="  Nhập họ tên"

            value={this.state.HOTEN}

            />

            <TextInput

            style={{height:40, borderColor:'gray', borderWidth:1,marginTop:10}}

            onChangeText={(USERNAME) => this.setState({USERNAME})}

            placeholder="  Nhập username"

            value={this.state.USERNAME}

            />

            <TextInput

            style={{height:40, borderColor:'gray', borderWidth:1,marginTop:10,marginBottom:10}}

            onChangeText={(PASSWORD) => this.setState({PASSWORD})}

            placeholder="  Nhập password"

            secureTextEntry={true}

            value={this.state.PASSWORD}

            />

            <TouchableOpacity style={styles.plus} onPress={()=>{this.clickPlus()}}>

                <Text style={styles.plusText,{fontSize:20, color:'yellow'}}> Đăng Ký </Text>

            </TouchableOpacity>

        </View>

    )

  }

}

var styles = StyleSheet.create({

    wrapper:{flex:1, backgroundColor:'rgb(234, 195, 196)', padding:50},

    plus:{padding:10, backgroundColor:"blue",justifyContent:"center",alignItems:"center"},

    plusText:{color:"white"},

    resulf:{padding:30,backgroundColor:"white",justifyContent:"center",alignItems:"center",}

});

## 2 Ví dụ (Lê Quốc Dũng)



Kết nối sql

<?php

//Define your host here.

$HostName ="localhost";

//Define your database name here.

$DatabaseName = "project";

//Define your database username here.

$HostUser = "root";

//Define your database password here.

$HostPass = "";

// mysql\_query('SET NAMES utf8');

?>

<?php

include 'DBConfig.php';

$conn = new mysqli($HostName, $HostUser, $HostPass, $DatabaseName);

if ($conn->connect\_error) {

 die("Connection failed: " . $conn->connect\_error);

}

$sql = "SELECT \* FROM tb\_sanpham";

$result = $conn->query($sql);

if ($result->num\_rows >0) {

 while($row[] = $result->fetch\_assoc()) {

 $tempt = $row;

 $json = json\_encode($tempt);

 }

} else {

 echo "No Results Found.";

}

 echo $json;

$conn->close();

?>

Code chương trình

/\*\*

 \*Name: Lê Quốc Dũng

 \* Sample React Native App

 \* https://github.com/facebook/react-native

 \* @flow

 \*/

import React, { Component } from 'react';

import {

  TouchableOpacity,

  StyleSheet,

  Text,

  View,

  Image,

} from 'react-native';

var icon;

export default class SanPham extends Component{

  constructor(props){

    super(props);

    this.state={

      SL : 1

    }

  }

  tangSLuong(){

      this.setState({

        SL: this.state.SL + 1

      });

  }

  giamSLuong(){

   if(this.state.SL > 0)

   {

    this.setState({

        SL: this.state.SL - 1

      });

   }

}

  render() {

    return (

        <View style={styles.product}>

        <View style={styles.containerImg}>

          <View>

            <Image style={styles.img} source={{uri: this.props.img}} />

          </View>

          <View >

            <Text style={styles.soLuongText}> Số lượng: {this.state.SL}</Text>

          </View>

          <View style={{width: 20, height: 20}}>

            <TouchableOpacity onPress={()=> this.tangSLuong()}><Text style={styles.tangSL}>+</Text></TouchableOpacity>

          </View>

          <View style={{width: 20, height: 20, marginLeft: 30, marginTop: -20}}>

            <TouchableOpacity onPress={()=> this.giamSLuong()}><Text style={styles.giamSL}>-</Text></TouchableOpacity>

          </View>

        </View>

        <View style={styles.containerContent}>

          <Text> Mã sản phẩm : {this.props.maSP} </Text>

          <Text> Tên sản phẩm : {this.props.tenSP} </Text>

        </View>

      </View>

    );

  }

}

const styles = StyleSheet.create({

  product: {

    justifyContent: 'center',

    //alignItems: 'center',

    width: 400,

    height: 110,

    marginBottom : 10,

    backgroundColor: 'rgb(234, 195, 196)'

  },

  productText: {

    marginBottom : 5,

  },

  img: {

    marginTop: -25,

    width: 70,

    height:70,

  },

  containerImg: {

    width: 100,

    height: 150,

    marginLeft: 10

  },

  containerContent: {

    marginTop:-150,

    marginLeft: 80

  },

  soLuong:{

    marginLeft: 20,

  },

  soLuongText: {

    fontSize: 12

  },

  giamSL: {

    backgroundColor: 'white',

    textAlign:'center',

    borderRadius: 5

  },

  tangSL: {

    backgroundColor: 'white',

    textAlign:'center',

    borderRadius: 5

    }

});

/\*Name: Le Quoc Dung \*/

import React from 'react';

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

import SanPham from './SanPham';

export default class FetchExample extends React.Component {

  constructor(props){

    super(props);

    this.state ={ isLoading: true}

  }

  componentDidMount(){

    return fetch('http://192.168.129.180:81/Project\_dd3/Select\_SanPham.php')

      .then((response) => response.json())

      .then((responseJson) => {

        this.setState({

          isLoading: false,

          dataSource: responseJson,

        }, function(){

        });

      })

      .catch((error) =>{

        console.error(error);

      });

  }

  render(){

    if(this.state.isLoading){

      return(

        <View style={{flex: 1, padding: 20}}>

          <ActivityIndicator/>

        </View>

      )

    }

    return(

      <View style={{flex: 1, paddingTop:20}}>

        <FlatList

          data={this.state.dataSource}

          renderItem={({item}) => <SanPham maSP={item.masp} tenSP={item.tensp} img={item.HinhAnh}/>}

        />

      </View>

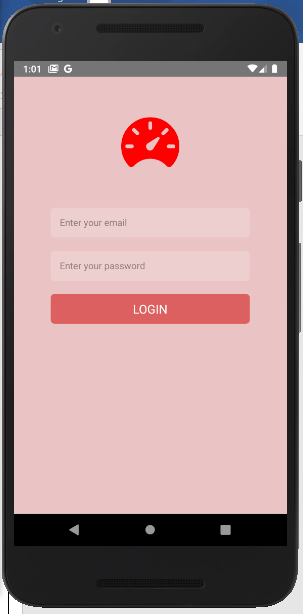
    );

  }

}

# CHƯƠNG 5: XÂY DỰNG ĐỒ ÁN

## 1 Màn hình đăng nhập (Nguyễn Ngọc Sơn)



Code:

import React, { Component } from 'react'

import {

    StyleSheet,

    Text,

    View,

    Image,

    TouchableWithoutFeedback,

    StatusBar,

    TextInput,

    Keyboard,

    TouchableOpacity,

  } from 'react-native';

import { red } from 'ansi-colors';

import IOnicons from 'react-native-vector-icons/Ionicons'

import FontAwesome from 'react-native-vector-icons/FontAwesome'

export default class Loginfb extends Component {

    render() {

        return (

          <TouchableWithoutFeedback onPress={Keyboard.dismiss}>

            <View style={styles.container}>

            <StatusBar barStyle="light-content" />

              <View style={styles.up}>

                <IOnicons

                  name="ios-speedometer"

                  size={100}

                  color={'red'}></IOnicons>

              </View>

              <View style={styles.down}>

                <View style={styles.textInputContainer}>

                  <TextInput

                    style={styles.textInput}

                    textContentType="emailAddress"

                    keyboardType="email-address"

                    placeholder="Enter your email"></TextInput>

                </View>

                <View style={styles.textInputContainer}>

                  <TextInput

                    style={styles.textInput}

                    placeholder="Enter your password"

                    secureTextEntry={true}></TextInput>

                </View>

                <TouchableOpacity style={styles.loginButton}>

                  <Text style={styles.loginButtonTitle}>LOGIN</Text>

                </TouchableOpacity>

              </View>

            </View>

          </TouchableWithoutFeedback>

        );

    }

}

const styles = StyleSheet.create({

    container:{

        flex: 1,

        flexDirection:'column',

        justifyContent:'center',

        alignItems:'stretch',

        backgroundColor:'rgb(234, 195, 196)',

    },

    up:{

        flex:3,

        flexDirection:'column',

        justifyContent:'center',

        alignItems:'center',

    },

    down:{

        flex:7,

        flexDirection:'column',

        justifyContent:'flex-start',

        alignItems:'center'

    },

    title:{

        color:'white',

        textAlign:'center',

        color:'rgb(255,119,34)',

        width:400,

        fontSize:23

    },

    textInputContainer:{

        paddingHorizontal:10,

        borderRadius:6,

        marginBottom:20,

        backgroundColor:'rgba(255,255,255,0.2)'

    },

    textInput:{

        width:280,

        height:45,

    },

    loginButton:{

        width:300,

        height:45,

        borderRadius:6,

        justifyContent:'center',

        alignItems:'center',

        backgroundColor:'rgb(221, 97,97)'

    },

    loginButtonTitle:{

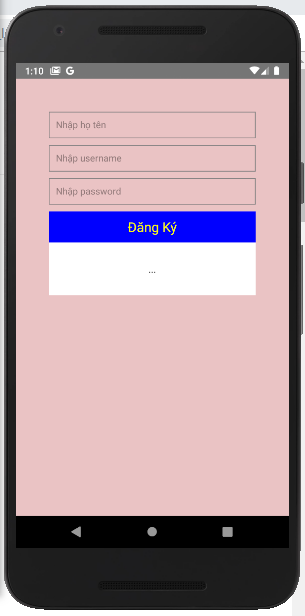
        fontSize:18,

        color:'white'

    }

})

## 2 Màn hình đăng ký (Nguễn Ngọc Sơn)



Code:

import React, {Component} from 'react';

import {

View, Text,StyleSheet,TextInput, TouchableOpacity

} from 'react-native';

import {red} from 'ansi-colors';

export default class DangKy extends Component {

    constructor(props){

        super(props);

        this.state={

            HOTEN:"",

            USERNAME:"",

            PASSWORD:"",

            resulf:"...",

        }

    }

    clickPlus(){

        fetch('http://192.168.1.3:81/Project\_dd3/dangky.php',{

            method:"POST",

            headers: {

                Accept: 'application/json',

                'Content-Type': 'application/json',

              },

            body: JSON.stringify({

                "HOTEN":this.state.HOTEN,

                "USERNAME":this.state.USERNAME,

                "PASSWORD":this.state.PASSWORD,

            })

        })

        .then((response)=>response.json())

        .then((reponseJson)=>{

            this.setState({resulf:reponseJson.id})

        })

        .catch((error)=>{console.log(error)});

    }

  render() {

    return (

        <View style={styles.wrapper}>

            <TextInput

            style={{height:40, borderColor:'gray', borderWidth:1}}

            onChangeText={(HOTEN) => this.setState({HOTEN})}

            placeholder="  Nhập họ tên"

            value={this.state.HOTEN}

            />

            <TextInput

            style={{height:40, borderColor:'gray', borderWidth:1,marginTop:10}}

            onChangeText={(USERNAME) => this.setState({USERNAME})}

            placeholder="  Nhập username"

            value={this.state.USERNAME}

            />

            <TextInput

            style={{height:40, borderColor:'gray', borderWidth:1,marginTop:10,marginBottom:10}}

            onChangeText={(PASSWORD) => this.setState({PASSWORD})}

            placeholder="  Nhập password"

            secureTextEntry={true}

            value={this.state.PASSWORD}

            />

            <TouchableOpacity style={styles.plus} onPress={()=>{this.clickPlus()}}>

                <Text style={styles.plusText,{fontSize:20, color:'yellow'}}> Đăng Ký </Text>

            </TouchableOpacity>

            <View style={styles.resulf}>

                <Text>{this.state.resulf}</Text>

            </View>

        </View>

    )

  }

}

var styles = StyleSheet.create({

    wrapper:{flex:1, backgroundColor:'rgb(234, 195, 196)', padding:50},

    plus:{padding:10, backgroundColor:"blue",justifyContent:"center",alignItems:"center"},

    plusText:{color:"white"},

    resulf:{padding:30,backgroundColor:"white",justifyContent:"center",alignItems:"center",}

});

## 3 Màn hình danh sách sản phẩm (Lê Quốc Dũng)



Code:

Sanpham.js

/\*\*

 \* Sample React Native App

 \* https://github.com/facebook/react-native

 \* @flow

 \*/

/\*Name: Le Quoc Dung \*/

import React, { Component } from 'react';

import {

  TouchableOpacity,

  StyleSheet,

  Text,

  View,

  Image,

} from 'react-native';

var icon;

export default class SanPham extends Component{

}

  render() {

    return (

        <View style={styles.product}>

        <View style={styles.containerImg}>

          <View>

            <Image style={styles.img} source={{uri: this.props.img}} />

          </View>

          <View >

            <Text style={styles.soLuongText}> Số lượng: {this.state.SL}</Text>

          </View>

        </View>

        <View style={styles.containerContent}>

          <Text> Mã sản phẩm : {this.props.maSP} </Text>

          <Text> Tên sản phẩm : {this.props.tenSP} </Text>

        </View>

      </View>

    );

  }

}

const styles = StyleSheet.create({

  product: {

    justifyContent: 'center',

    //alignItems: 'center',

    width: 400,

    height: 110,

    marginBottom : 10,

    backgroundColor: 'rgb(234, 195, 196)'

  },

  productText: {

    marginBottom : 5,

  },

  img: {

    marginTop: -25,

    width: 70,

    height:70,

  },

  containerImg: {

    width: 100,

    height: 150,

    marginLeft: 10

  },

  containerContent: {

    marginTop:-150,

    marginLeft: 80

  },

  soLuong:{

    marginLeft: 20,

  },

  soLuongText: {

    fontSize: 12

  },

  giamSL: {

    backgroundColor: 'white',

    textAlign:'center',

    borderRadius: 5

  },

  tangSL: {

    backgroundColor: 'white',

    textAlign:'center',

    borderRadius: 5

    }

});

FetchExample.js

import React from 'react';

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

import SanPham from './SanPham';

export default class FetchExample extends React.Component {

  constructor(props){

    super(props);

    this.state ={ isLoading: true}

  }

  componentDidMount(){

    return fetch('http://192.168.1.3:81/SanPham/Select\_SanPham.php')

      .then((response) => response.json())

      .then((responseJson) => {

        this.setState({

          isLoading: false,

          dataSource: responseJson,

        }, function(){

        });

      })

      .catch((error) =>{

        console.error(error);

      });

  }

  render(){

    if(this.state.isLoading){

      return(

        <View style={{flex: 1, padding: 20}}>

          <ActivityIndicator/>

        </View>

      )

    }

    return(

      <View style={{flex: 1, paddingTop:20}}>

        <FlatList

          data={this.state.dataSource}

          renderItem={({item}) => <SanPham maSP={item.masp} tenSP={item.tensp} img={item.HinhAnh}/>}

        />

      </View>

    );

  }

}

## 4 Màn hình thêm sản phẩm (Lê Quốc Dũng)



Code:

/\*Name: Le Quoc Dung \*/

import React, { Component } from 'react'

import {

    StyleSheet,

    Text,

    View,

    Image,

    TouchableWithoutFeedback,

    StatusBar,

    TextInput,

    Keyboard,

    TouchableOpacity,

  } from 'react-native';

import { red } from 'ansi-colors';

import IOnicons from 'react-native-vector-icons/Ionicons'

import FontAwesome from 'react-native-vector-icons/FontAwesome'

export default class ThemSamPham extends Component {

    constructor(props){

        super(props);

        this.state={

            MASP:"",

            TENSP:"",

            SOLUONG:"",

            HINHANH:"",

        }

    }

    clickPlus(){

        fetch('http://192.168.1.3:81/Project\_dd3/ThemSanPham.php',{

            method:"POST",

            headers: {

                Accept: 'application/json',

                'Content-Type': 'application/json',

              },

            body: JSON.stringify({

                "MASP":this.state.MASP,

                "TENSP":this.state.TENSP,

                "SOLUONG":this.state.SOLUONG,

                "HINHANH":this.state.HINHANH,

            })

        })

        .then((response)=>response.json())

        .then((reponseJson)=>{

            this.setState({resulf:reponseJson.id})

        })

        .catch((error)=>{console.log(error)});

    }

    render() {

        return (

          <TouchableWithoutFeedback onPress={Keyboard.dismiss}>

            <View style={styles.container}>

            <StatusBar barStyle="light-content" />

                <View style={styles.up}>

                <IOnicons

                  name="ios-speedometer"

                  size={100}

                  color={'red'}></IOnicons>

              </View>

              <View style={styles.down}>

                <View style={styles.textInputContainer}>

                  <TextInput

                    style={styles.textInput}

                    placeholder="Mã sản phẩm"

                    onChangeText={(MASP) => this.setState({MASP})}

                    value={this.state.MASP}></TextInput>

                </View>

                <View style={styles.textInputContainer}>

                  <TextInput

                    style={styles.textInput}

                    placeholder="Tên sản phẩm"

                    onChangeText={(TENSP) => this.setState({TENSP})}

                    value={this.state.TENSP}></TextInput>

                </View>

                <View style={styles.textInputContainer}>

                  <TextInput

                    style={styles.textInput}

                    placeholder="Số Lượng"

                    onChangeText={(SOLUONG) => this.setState({SOLUONG})}

                    value={this.state.SOLUONG}></TextInput>

                </View>

                <View style={styles.textInputContainer}>

                  <TextInput

                    style={styles.textInput}

                    placeholder="Đường dẫn Hình ảnh"

                    onChangeText={(HINHANH) => this.setState({HINHANH})}

                    value={this.state.HINHANH}></TextInput>

                </View>

                <TouchableOpacity style={styles.loginButton} onPress={()=>{this.clickPlus()}}>

                  <Text style={styles.loginButtonTitle}>Thêm Sản Phẩm</Text>

                </TouchableOpacity>

              </View>

            </View>

          </TouchableWithoutFeedback>

        );

    }

}

const styles = StyleSheet.create({

    container:{

        flex: 1,

        flexDirection:'column',

        justifyContent:'center',

        alignItems:'stretch',

        backgroundColor:'rgb(234, 195, 196)',

    },

    up:{

        flex:3,

        flexDirection:'column',

        justifyContent:'center',

        alignItems:'center',

    },

    down:{

        flex:7,

        flexDirection:'column',

        justifyContent:'flex-start',

        alignItems:'center'

    },

    title:{

        color:'white',

        textAlign:'center',

        color:'rgb(255,119,34)',

        width:400,

        fontSize:23

    },

    textInputContainer:{

        paddingHorizontal:10,

        borderRadius:6,

        marginBottom:20,

        backgroundColor:'rgba(255,255,255,0.2)'

    },

    textInput:{

        width:280,

        height:45,

    },

    loginButton:{

        width:300,

        height:45,

        borderRadius:6,

        justifyContent:'center',

        alignItems:'center',

        backgroundColor:'rgb(221, 97,97)'

    },

    loginButtonTitle:{

        fontSize:18,

        color:'white'

    }

})

# CHƯƠNG 5: kết luận và kiến nghị

Tài liệu tham khảo:

<https://cameoplus.com/tai-sao-nen-su-dung-let-const-thay-cho-var-trong-javascript/>

<https://facebook.github.io/react-native/docs/components-and-apis.html>

<https://facebook.github.io/react-native/docs/getting-started>

Và một số trang web khác.