

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- Xây dựng ứng dụng hoàn chỉnh trong React Native

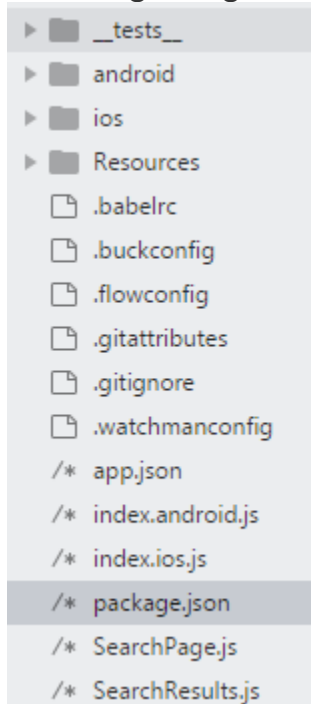
PHẦN I

Bài 1 (2 điểm):

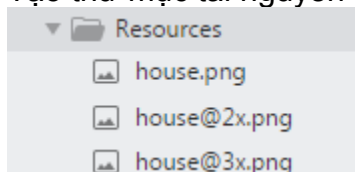
Yêu cầu: Tạo ứng dụng react native, sau đó thêm tài nguyên và thêm dữ liệu mẫu

Hướng dẫn

Tạo ứng dụng với cấu trúc như sau:



Tạo thư mục tài nguyên trong file đính kèm



Định nghĩa thuộc tính trong app.json

```

1 {
2   "name": "PropertyFinder",
3   "displayName": "PropertyFinder"
4 }
    
```

Định nghĩa dữ liệu trong package.json

```

1 {
2   "name": "PropertyFinder",
3   "version": "0.0.1",
4   "private": true,
5   "scripts": {
6     "start": "node node_modules/react-native/local-cli/cli.js start",
7     "test": "jest"
8   },
9   "dependencies": {
10    "react": "16.0.0-alpha.12",
11    "react-native": "0.46.0"
12  },
13  "devDependencies": {
14    "babel-jest": "20.0.3",
15    "babel-preset-react-native": "2.1.0",
16    "jest": "20.0.4",
17    "react-test-renderer": "16.0.0-alpha.12"
18  },
19  "jest": {
20    "preset": "react-native"
21  }
22 }
    
```

Sản phẩm phải nộp: Nộp toàn bộ dự án lên LMS

Bài 2 (2 điểm):

Yêu cầu: Sử dụng bài 1 và update file index.android.js nhằm mục đích sử dụng StyleSheet cho container, view, text

Hướng dẫn

Để định nghĩa các hàm cần gọi và styling
Ta code file index.android.js như sau:

```

1  import React, { Component } from 'react';
2  import {
3    AppRegistry,
4    StyleSheet,
5    Text,
6    View
7  } from 'react-native';
8
9  export default class PropertyFinder extends Component {
10   render() {
11     return (
12       <View style={styles.container}>
13         <Text style={styles.welcome}>
14           Welcome to React Native!
15         </Text>
16         <Text style={styles.instructions}>
17           To get started, edit index.android.js
18         </Text>
19         <Text style={styles.instructions}>
20           Double tap R on your keyboard to reload,{'\n'}
21           Shake or press menu button for dev menu
22         </Text>
23       </View>
24     );
25   }
26 }
27
28 const styles = StyleSheet.create({
29   container: {
30     flex: 1,
31     justifyContent: 'center',
32     alignItems: 'center',
33     backgroundColor: '#F5FCFF',
34   },
35   welcome: {
36     fontSize: 20,
37     textAlign: 'center',
38     margin: 10,
39   },
40   instructions: {
41     textAlign: 'center',
42     color: '#333333',
43     marginBottom: 5,
44   },
45 });
46
47 AppRegistry.registerComponent('PropertyFinder', () => PropertyFinder);
48

```

Sản phẩm phải nộp: Nộp toàn bộ dự án lên LMS

PHẦN II

Bài 3 (2 điểm):

Yêu cầu: Định nghĩa chức năng tìm kiếm trong React Native

Hướng dẫn

Tacode file SearchPage.js như sau:

```

1  |'use strict';
2
3  import React, { Component } from 'react';
4  import {
5    StyleSheet,
6    Text,
7    TextInput,
8    View,
9    Button,
10   ActivityIndicator,
11   Image,
12 } from 'react-native';
13 import SearchResults from './SearchResults';
14
15 function urlForQueryAndPage(key, value, pageNumber) {
16   const data = {
17     country: 'uk',
18     pretty: '1',
19     encoding: 'json',
20     listing_type: 'buy',
21     action: 'search_listings',
22     page: pageNumber,
23   };
24   data[key] = value;
25
26   const querystring = Object.keys(data)
27     .map(key => key + '=' + encodeURIComponent(data[key]))
28     .join('&');
29
30   return 'https://api.nestoria.co.uk/api?' + querystring;
31 };
32
33 export default class SearchPage extends Component {
34   constructor(props) {
35     super(props);
36     this.state = {
37       searchString: 'london',
38       isLoading: false,
39       message: '',
40     };
41   }
42

```

```

43 | _onSearchTextChanged = (event) => {
44 |   this.setState({ searchString: event.nativeEvent.text });
45 | };
46 |
47 | _executeQuery = (query) => {
48 |   console.log(query);
49 |   this.setState({ isLoading: true });
50 |   fetch(query)
51 |     .then(response => response.json())
52 |     .then(json => this._handleResponse(json.response))
53 |     .catch(error =>
54 |       this.setState({
55 |         isLoading: false,
56 |         message: 'Something bad happened ' + error
57 |       }));
58 | };
59 |
60 | _handleResponse = (response) => {
61 |   this.setState({ isLoading: false, message: '' });
62 |   if (response.application_response_code.substr(0, 1) === '1') {
63 |     this.props.navigator.push({
64 |       title: 'Results',
65 |       component: SearchResults,
66 |       passProps: { listings: response.listings }
67 |     });
68 |   } else {
69 |     this.setState({ message: 'Location not recognized; please try again.' });
70 |   }
71 | };
72 |
73 | _onSearchPressed = () => {
74 |   const query = urlForQueryAndPage('place_name', this.state.searchString, 1);
75 |   this._executeQuery(query);
76 | }
77 |

```

```

78   render() {
79     const spinner = this.state.isLoading ?
80       <ActivityIndicator size='large' /> : null;
81
82     return (
83       <View style={styles.container}>
84         <Text style={styles.description}>
85           Search for houses to buy!
86         </Text>
87         <Text style={styles.description}>
88           Search by place-name or postcode.
89         </Text>
90         <View style={styles.flowRight}>
91           <TextInput
92             style={styles.searchInput}
93             value={this.state.searchString}
94             onChange={this._onSearchTextChanged}
95             placeholder='Search via name or postcode' />
96           <Button
97             onPress={this._onSearchPressed}
98             color='#48BBEC'
99             title='Go'
100          />
101        </View>
102        <Image source={require('./Resources/house.png')} style={styles.image} />
103        {spinner}
104        <Text style={styles.description}>{this.state.message}</Text>
105      </View>

```

```

105     </View>
106   );
107 }
108 }
109
110 ▼ const styles = StyleSheet.create({
111 ▼   description: {
112     marginBottom: 20,
113     fontSize: 18,
114     textAlign: 'center',
115     color: '#656565'
116   },
117 ▼   container: {
118     padding: 30,
119     marginTop: 65,
120     alignItems: 'center'
121   },
122 ▼   flowRight: {
123     flexDirection: 'row',
124     alignItems: 'center',
125     alignSelf: 'stretch',
126   },
127
126 },
127   searchInput: {
128     height: 36,
129     padding: 4,
130     marginRight: 5,
131     flexGrow: 1,
132     fontSize: 18,
133     borderWidth: 1,
134     borderColor: '#48BBEC',
135     borderRadius: 8,
136     color: '#48BBEC',
137   },
138   image: {
139     width: 217,
140     height: 138,
141   },
142 });
  
```

Sản phẩm phải nộp: Nộp toàn bộ dự án lên LMS

Bài 4 (2 điểm):

Yêu cầu: Hiển thị kết quả tìm kiếm trong ứng dụng react native

Hướng dẫn

Ta định nghĩa chức năng hiển thị trong **SearchResults.js** như sau:

```

1  |'use strict';
2
3  import React, { Component } from 'react'
4  import {
5    StyleSheet,
6    Image,
7    View,
8    TouchableHighlight,
9    FlatList,
10   Text,
11  } from 'react-native';
12
13  class ListItem extends React.PureComponent {
14    _onPress = () => {
15      this.props.onPressItem(this.props.index);
16    }
17
18  render() {
19    const item = this.props.item;
20    const price = item.price_formatted.split(' ')[0];
21    return (
22      <TouchableHighlight
23        onPress={this._onPress}
24        underlayColor='#dddddd'>
25        <View>
26          <View style={styles.rowContainer}>
27            <Image style={styles.thumb} source={{ uri: item.img_url }} />
28            <View style={styles.textContainer}>
29              <Text style={styles.price}>{price}</Text>
30              <Text style={styles.title}
31                numberOfLines={1}>{item.title}</Text>
32            </View>
33          </View>
34          <View style={styles.separator}/>
35        </View>
36      </TouchableHighlight>
37    );
38  }
39  }
40
    
```

```

41 export default class SearchResults extends Component {
42   _keyExtractor = (item, index) => index;
43
44   _renderItem = ({item, index}) => (
45     <ListItem
46       item={item}
47       index={index}
48       onPressItem={this._onPressItem}
49     />
50   );
51
52   _onPressItem = (index) => {
53     console.log("Pressed row: "+index);
54   };
55
56   render() {
57     return (
58       <FlatList
59         data={this.props.listings}
60         keyExtractor={this._keyExtractor}
61         renderItem={this._renderItem}
62       />
63     );
64   }
65 }

```

```

66
67 const styles = StyleSheet.create({
68   thumb: {
69     width: 80,
70     height: 80,
71     marginRight: 10
72   },
73   textContainer: {
74     flex: 1
75   },
76   separator: {
77     height: 1,
78     backgroundColor: '#dddddd'
79   },
80   price: {
81     fontSize: 25,
82     fontWeight: 'bold',
83     color: '#48BBEC'
84   },
85   title: {
86     fontSize: 20,
87     color: '#656565'
88   },
89   rowContainer: {
90     flexDirection: 'row',
91     padding: 10
92   },
93 });

```

Sản phẩm phải nộp: Nộp toàn bộ dự án lên LMS

Bài 5 (2 điểm):

Giảng viên cho thêm

Chú ý:

- ✓ *Phần I và Phần II chỉ áp dụng cho dạy tích hợp. Sinh viên làm phần 1 và phần 2 theo 2 bài khác nhau tương ứng với 2 phần lý thuyết đã dạy trong bài học.*
- ✓ *Nếu giảng dạy theo phương pháp truyền thống thì sinh viên phải thực hiện tất cả các bài trong một buổi thực hành.*