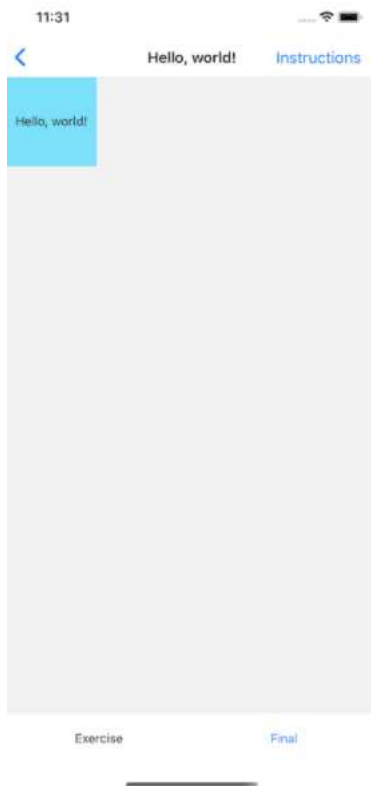


Lab 2. React – Native – Fundamentals

Phần 1. Các Projects Cơ Bản

Project 1. Hello world



Link tham khảo:

- [React Native View Docs](#)
- [React Native Text Docs](#)

Hướng dẫn:

Bước 1. Khởi tạo project

```
PS C:\Users\DESKTOP-HUU\Desktop\PTUDDDDNT\Projects> expo init Hello-world
```

```
PS C:\Users\DESKTOP-HUU\Desktop\PTUDDDDNT\Projects> cd Hello-world
```

Start project với Android emulator (Đã cài đặt)

```
PS C:\Users\DESKTOP-HUU\Desktop\PTUDDDDNT\Projects\Hello-world> expo start
```

Nhấn **a** để chạy với android.

Bước 2. Import các component cần thiết.

```
1 import React from 'react';
2 import {
3   View,
4   Text
5 } from 'react-native'
```

Bước 3. Thiết kế giao diện

Tạo function trả về view (JSX) của App

```
const App = ()=>
{
  return (
    <View>
      <Text > Hello World</Text>
    </View>
  );
}
```

- Chú ý tuân thủ cú pháp của JSX (XML) chỉ có 1 component (Element) tổng chứa tất cả các component con.
- Code ES6 trong JSX đặt trong cặp dấu {}

Export default component chính (App)

```
export default App;
```

Bước 4. Thiết kế style

```
<View style={
  {
    width:100,
    height:100,
    backgroundColor:'aqua',
    alignItems: 'center',
    justifyContent: 'center',
  }
}>
  <Text style={{color:'#fff'}}> Hello World</Text>
</View>
```

Bước 5. Hiệu chỉnh tạo và sử dụng style với hàm StyleSheet.create()

Import StyleSheet

```
2 import {  
3   View,  
4   Text,  
5   StyleSheet  
6 } from 'react-native'
```

Tạo MyStyle

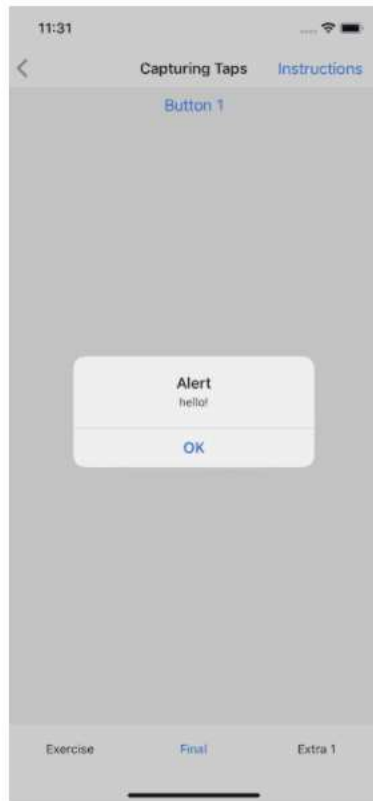
```
var MyStyle = StyleSheet.create(  
  {  
    ViewStyle: {  
      width:100,  
      height:100,  
      backgroundColor:'aqua',  
      alignItems: 'center',  
      justifyContent: 'center',  
    },  
    TextStyle:  
    {  
      color:'black'  
    }  
  }  
)
```

Sử dụng style trong component

```
8 const App = ()=>  
9 {  
10   return (  
11     <View style={  
12       MyStyle.ViewStyle  
13     }>  
14       <Text style={MyStyle.TextStyle}> Hello World</Text>  
15     </View>  
16   );  
17 }
```

Project 2. Capturing Taps

Xây dựng project với button khi nhấn vào hiển thị thông báo với alert “hello”



Link tham khảo

- [React Native Button Docs](#)
- [React Native TouchableOpacity Docs](#)

Hướng dẫn

```
1  import React from "react";
2  import { View, Button } from "react-native";
3
4  export default () => {
5    return (
6      <View>
7        <Button title="Button 1" onPress={() => alert("hello!")} />
8      </View>
9    );
10 };
11
```

Hiệu chỉnh

```
import {
  View,
  Button,
  TouchableOpacity,
  Text } from "react-native";
```

```
8 | export default () => {
9 |   return (
10 |     <View style={{flex:1, justifyContent:'center'}}>
11 |       <Button title="Button 1" onPress={() => alert("hello!")} />
12 |       <TouchableOpacity
13 |         onPress={() => alert("hello 2!")}
14 |         style={{
15 |           backgroundColor: "blue",
16 |           padding: 10,
17 |           alignItems: "center",
18 |           marginTop: 10,
19 |         }}
20 |       >
21 |         <Text style={{color: "white", fontSize: 18 }}>Button 2</Text>
22 |       </TouchableOpacity>
23 |     </View>
24 |   );
25 | };
```

Hiệu chỉnh tạo và sử dụng style với hàm StyleSheet.create()

```

24  ✓ const MyStyle = StyleSheet.create(
25  ✓  {
26  ✓    container:{
27      flex:1,
28      justifyContent:"center"
29    },
30    button:{
31      backgroundColor: "blue",
32      marginTop:10,
33      alignItems: "center",
34      padding: 10
35    },
36    text:{
37      color: "white",
38      fontSize: 18
39    }
40  }
41  )

```

```

10  export default ()=>
11  {
12    return (
13      <View style={MyStyle.container}>
14        <Button title="Button 1" onPress={()=> alert("Hello 1")} />
15        <TouchableOpacity style={MyStyle.button} onPress={()=> alert("Hello 2")}>
16          <Text style={MyStyle.text}>
17            Button 2
18          </Text>
19        </TouchableOpacity>
20      </View>
21    )
22  }

```

Project 3. Custom Component

Tạo component Button với TouchableOpacity và Text với 2 props truyền vào là text, hàm xử lý sự kiện _onPress và buttonStyle truyền vào từ component cha.



Hướng dẫn code

```
import React from "react";
import {
  TouchableOpacity,
  Text,
  View }
from "react-native";
```

```

8   const Button = (props) => (
9     <TouchableOpacity
10      onPress={props.onPress}
11      style={{
12        backgroundColor: "#ff637c",
13        alignSelf: "center",
14        padding: 10,
15        margin: 10,
16        ...props.buttonStyle,
17      }}
18    >
19      <Text style={{ color: "fff" }}>{props.text}</Text>
20    </TouchableOpacity>
21  );
22
23  export default () => (
24    <View style={{ flex: 1, justifyContent: "center" }}>
25      <Button text="Say hello" onPress={() => alert("hello!")} />
26      <Button
27        text="Say goodbye"
28        onPress={() => alert("goodbye!")}
29        buttonStyle={{ backgroundColor: "#4dc2c2" }}
30      />
31    </View>
32  );

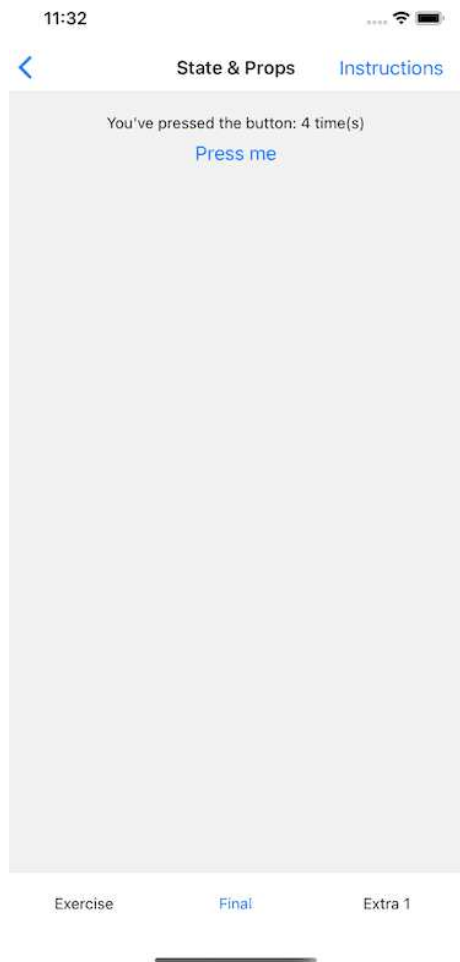
```

Project 4. State & Props

Props: cho phép component truyền khởi tạo giá trị cho các properties ban đầu của component và không thay đổi trong quá trình thức thi.

State: hoạt động tương tự như props, cũng khởi tạo giá trị ban đầu cho các component nhưng có thể thay đổi theo tác động của người dùng.

- [Using the State Hook](#)
- [Introducing Hooks](#)

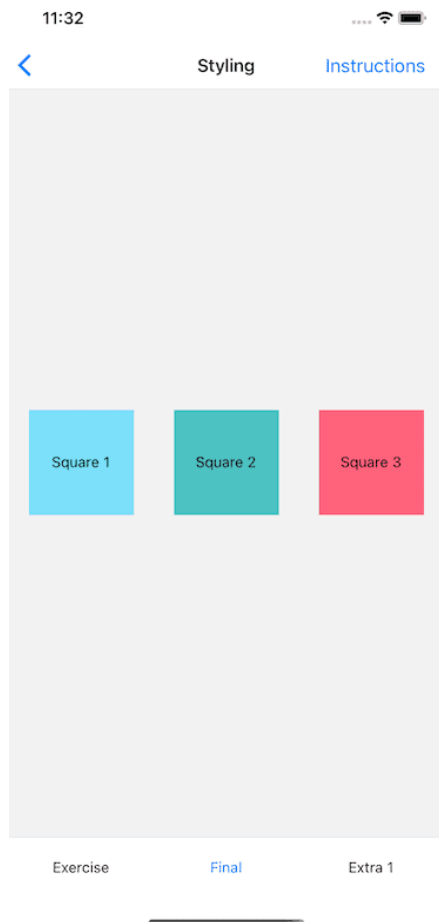


Hướng dẫn code

```
1  import React, { useState } from "react";
2  import { Text, Button, View } from "react-native";
3
4  export default () => {
5    const [pressCount, setPressCount] = useState(0);
6
7    return (
8      <View style={{ alignItems: "center", marginTop: 20 }}>
9        <Text>You've pressed the button: {pressCount} time(s)</Text>
10       <Button
11         title={`Pressed ${pressCount} time(s)`}
12         onPress={() => setPressCount(pressCount + 1)}
13       />
14     </View>
15   );
16 }
```

Project 5. Styling

- [StyleSheet API Documentation](#)
- [Layout with Flexbox](#)



Hướng dẫn code

```
import React from "react";
import { View, Text, StyleSheet } from "react-native";

const styles = StyleSheet.create({
  container: {
    flex: 1,
    alignItems: "center",
    flexDirection: "row",
    justifyContent: "space-around",
  },
  box: {
    width: 100,
    height: 100,
```

```

      justifyContent: "center",
      alignItems: "center",
    },
  });

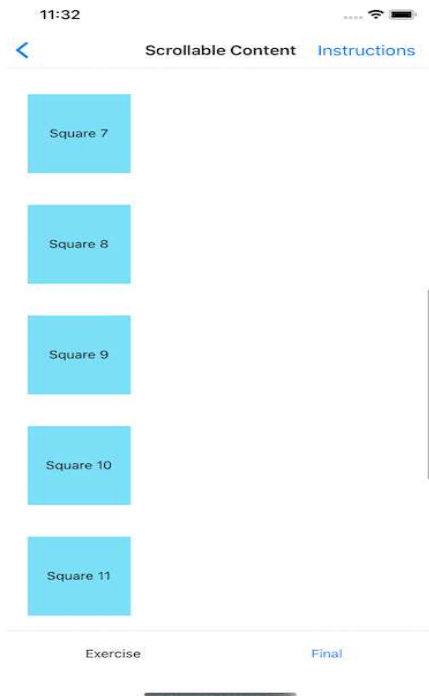
const Square = ({ text, bgColor = "#7ce0f9" }) => (
  <View style={[styles.box, { backgroundColor: bgColor }]}>
    <Text>{text}</Text>
  </View>
);

export default () => {
  return (
    <View style={styles.container}>
      <Square text="Square 1" />
      <Square text="Square 2" bgColor="#4dc2c2" />
      <Square text="Square 3" bgColor="#ff637c" />
    </View>
  );
};

```

Project 6. Scrollable Content

- [ScrollView Docs](#)
- [Explained: Each child in a list should have a unique "key" prop.](#)



```
import React from "react";
import { View, Text, StyleSheet, ScrollView } from "react-native";

const styles = StyleSheet.create({
  container: { backgroundColor: "#fff" },
  box: {
    width: 100,
    height: 100,
    justifyContent: "center",
    alignItems: "center",
    margin: 20,
  },
});

const Square = ({ text, bgColor = "#7ce0f9" }) => (
  <View style={[styles.box, { backgroundColor: bgColor }]}>
    <Text>{text}</Text>
  </View>
);

const data = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15];

export default () => {
  return (
    <ScrollView style={styles.container}>
      {data.map((item, index) => (
```

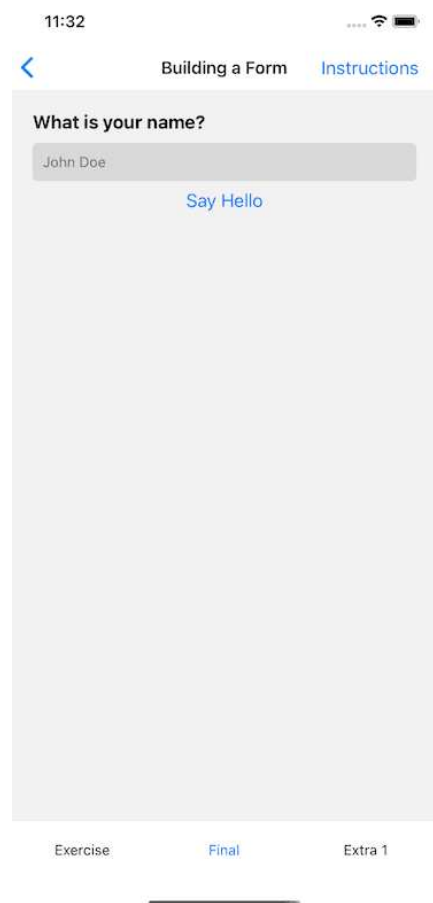
```

        <Square key={item} text={`Square ${index + 1}`} />
      )}
    </ScrollView>
  );
};

```

Project 7. Building a Form

- TextInput Docs



Hướng dẫn code

```

import React, { useState } from "react";
import { TextInput, Text, View, StyleSheet, Button } from "react-native";

const styles = StyleSheet.create({
  container: {
    padding: 20,

```

```

    },
    label: {
      fontWeight: "bold",
      fontSize: 18,
    },
    input: {
      marginTop: 10,
      backgroundColor: "rgba(0, 0, 0, 0.1)",
      padding: 10,
      borderRadius: 5,
    },
  },
});

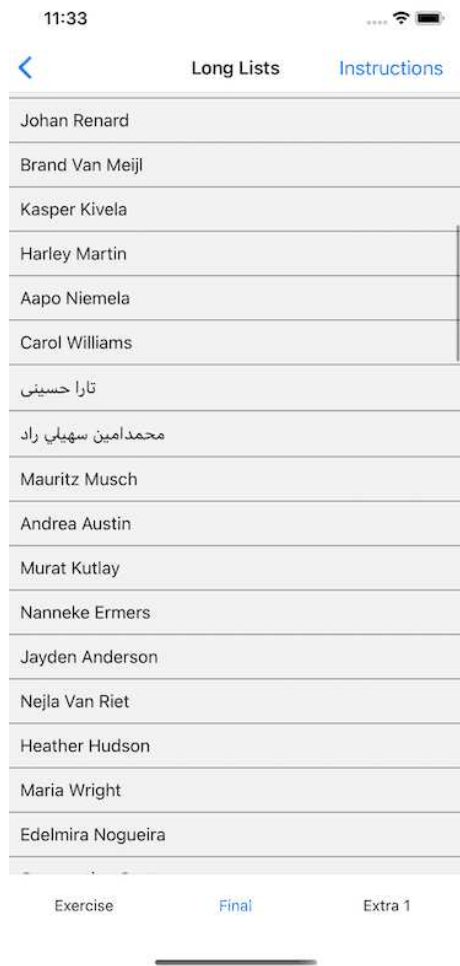
export default () => {
  const [name, setName] = useState("");

  return (
    <View style={styles.container}>
      <Text style={styles.label}>What is your name?</Text>
      <TextInput
        style={styles.input}
        placeholder="John Doe"
        placeholderTextColor="rgba(0, 0, 0, 0.5)"
        onChangeText={(text) => setName(text)}
        value={name}
      />
      <Button
        title="Say Hello"
        onPress={() => {
          alert(`Hello, ${name}!`);
          setName("");
        }}
      />
    </View>
  );
};

```

Project 8. Long Lists

- [FlatList](#)
- [SectionList](#)



```
import React from "react";
import {
  SectionList,
  Text,
  View,
  StyleSheet,
  SafeAreaView,
} from "react-native";

const styles = StyleSheet.create({
  row: {
    paddingHorizontal: 10,
    paddingVertical: 10,
  },
  name: {
    fontSize: 16,
  },
});
```

```

separator: {
  backgroundColor: "rgba(0, 0, 0, 0.5)",
  height: 1,
},
sectionHeader: {
  paddingHorizontal: 10,
  paddingVertical: 10,
  backgroundColor: "rgb(170, 170, 170)",
},
});

const groupPeopleByLastName = (_data) => {
  const data = [..._data];
  const groupedData = data.reduce((accumulator, item) => {
    const group = item.name.last[0].toUpperCase();

    if (accumulator[group]) {
      accumulator[group].data.push(item);
    } else {
      accumulator[group] = {
        title: group,
        data: [item],
      };
    }

    return accumulator;
  }, {});

  const sections = Object.keys(groupedData).map((key) => {
    return groupedData[key];
  });

  return sections.sort((a, b) => {
    if (a.title > b.title) {
      return 1;
    }
    return -1;
  });
};

export default () => {
  return (
    <SafeAreaView>
      <SectionList
        sections={groupPeopleByLastName(PEOPLE)}

```



```

    keyExtractor={({item}) => `${item.name.first}-${item.name.last}`}
    renderSectionHeader={({ section }) => {
      return (
        <View style={styles.sectionHeader}>
          <Text>{section.title}</Text>
        </View>
      );
    }}
    renderItem={({ item }) => {
      return (
        <View style={styles.row}>
          <Text style={styles.name}>
            {item.name.first} {item.name.last}
          </Text>
        </View>
      );
    }}
    ItemSeparatorComponent={() => <View style={styles.separator} />}
  />
</SafeAreaView>
);
};

const PEOPLE = [
  {
    name: {
      title: "Ms",
      first: "Maeva",
      last: "Scott",
    },
  },
  {
    name: {
      title: "Ms",
      first: "Maëlle",
      last: "Henry",
    },
  },
  {
    name: {
      title: "Mr",
      first: "Mohamoud",
      last: "Faaij",
    },
  },
],

```

];