

## 3SA04 – React-Native

เครื่องมือที่จำเป็น

1. Node.js ทำให้ JavaScript สามารถทำงานแบบ Runtime ได้
2. yarn ใช้ในการลงส่วนเสริมต่างๆ ของ react
3. Git เป็นเครื่องมือที่ช่วยอำนวยความสะดวกในการจัดการ Version ของโปรแกรม
4. Visual Studio Code ช่วยในการ Edit Code ต่างๆภายในโปรแกรม
5. create-react-native-app CLI เป็น Module ที่ใช้ในการสร้าง React-Native Application

ขั้นตอนการทำงานของโปรแกรม

เมื่อเรียก cmd โดยผ่าน administrator ทำการ cd เข้าไปยัง directory ที่ต้องการแล้วทำการติดตั้ง expo เพื่อให้สามารถเชื่อมต่อกับโทรศัพท์โดยคำสั่ง

```
>>npm install -g expo-cli
```

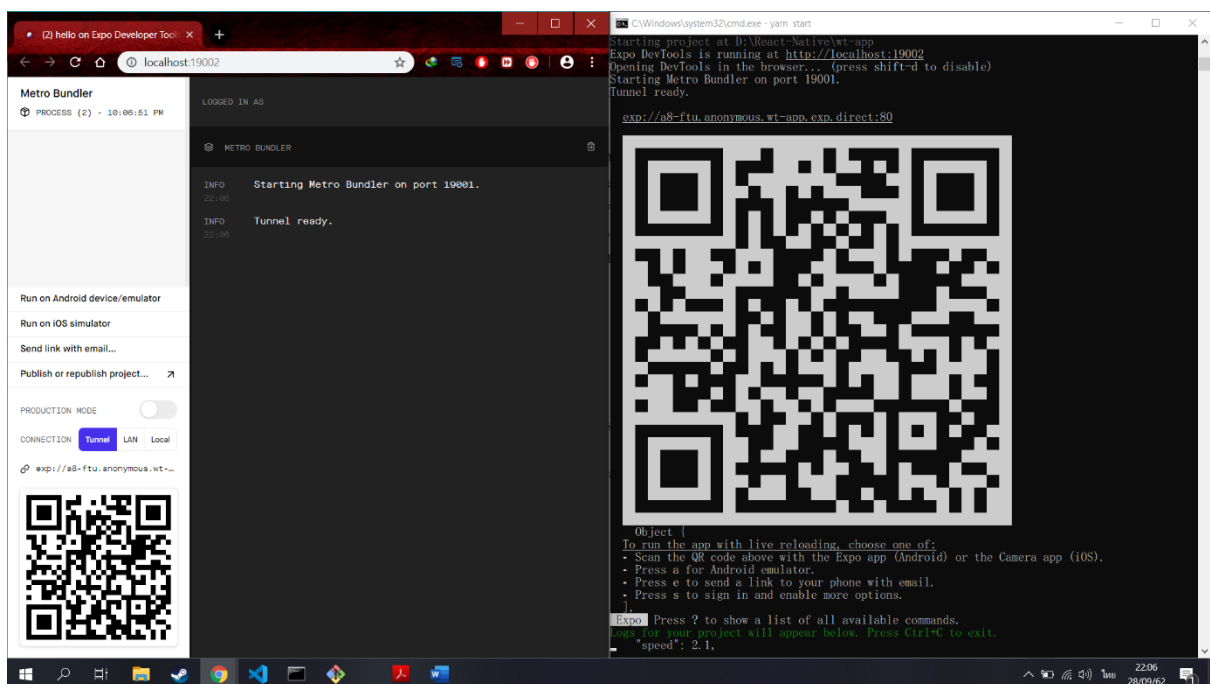
หลังจากนั้นทำการสร้าง app ที่มี directory ชื่อว่า wt-app

```
>>expo init wt-app
```

แล้วทำการ cd เข้าไปยัง wt-app หลังจากนั้นใช้คำสั่ง

```
>>yarn start
```

จะได้ดังรูปต่อไปนี้



- จากนั้นทำการทดลองตาม lab sheet ตั้งแต่ตอนที่ 3-6 เพื่อทำความเข้าใจคำสั่งและการทำงานต่างๆ ของ React-Native
- ซึ่งจะคล้ายกับการทำงานของ React การสร้างและใช้งาน Component, state, การส่งข้อมูลจาก Component หลักไปยัง Component ย่อยโดยใช้ props แต่ใน React-Native จะสร้าง UI แทนการใช้ HTML และ CCS

```

JS App.js JS Weather.js JS Forecast.js X JS ZipCodeScreen.js JS WeatherScreen.js
components > JS Forecast.js > [0] styles
1 import React, { Component } from "react";
2 import { StyleSheet, Text, View, TextInput, Animated } from "react-native";
3
4 export default class Forecast extends React.Component{
5   render(){
6     return(
7       <View style={styles.container}>
8         <Text style={styles.bea}>{this.props.location}</Text>
9         <Text style={styles.bea}>{this.props.main}</Text>
10        <Text style={styles.bea}>สถานะ : {this.props.description}</Text>
11        <Text style={styles.bea}>อุณหภูมิ : {this.props.temp} °C</Text>
12        <Text style={styles.bea}>ความชื้นสัมพัทธ์ : {this.props.humidity}%</Text>
13        <Text style={styles.bea}>อุณหภูมิสูงสุด : {this.props.temp_max} °C</Text>
14        <Text style={styles.bea}>อุณหภูมิต่ำสุด : {this.props.temp_min} °C</Text>
15      </View>
16    );
17  }
18 }

```

ตัวอย่างการสร้าง Component ต่างๆ ใน App ในไฟล์ Forecast.js

```

const styles = StyleSheet.create({
  container: {
    // paddingTop: 25,
    // flex: 1,
    // marginLeft: 24,
    // marginRight: 24,
    // marginBottom: 24
  },
  backdrop: {
    width: '100%',
    height: '100%',
    flexDirection: 'column',
    justifyContent: 'center',
    alignItems: 'center',
  },
  bea: {fontSize: 30 },
});

```

ตัวอย่างการสร้าง UI (: User Interface) ในไฟล์ Forecast.js

- (Checkpoint#1 - #4) แสดงสภาพอากาศที่หาได้ใหญ่ โดยมีการเพิ่มข้อมูลนอกจาก  
ในตอนนี้จะทำการเพิ่มในส่วนของคุณค่าใหม่เข้าไปคือ

- location: สถานที่
- main: สภาพอากาศโดยสรุป
- description: รายละเอียดสภาพอากาศ
- temp: อุณหภูมิ
- humidity: ความชื้นสัมพัทธ์
- temp\_max: อุณหภูมิสูงสุด
- temp\_min: อุณหภูมิต่ำสุด

```
fetchData = () => {
  fetch(`http://api.openweathermap.org/data/2.5/weather?q=${this.props.zipCode},th&units=metric&appid=fd68c0f2039c5a25f666a9ff374bc93e`)
    .then((response) => response.json())
    .then((json) => {
      console.log(json)
      this.setState({
        forecast: {
          location: json.name,
          main: json.weather[0].main,
          description: json.weather[0].description,
          temp: json.main.temp,
          humidity: json.main.humidity,
          temp_max: json.main.temp_max,
          temp_min: json.main.temp_min,
        }
      })
    })
    .catch((error) => {
      console.warn(error)
    })
}
```

ตัวอย่าง fetchData เพิ่มข้อมูลให้เพิ่มเติม

- (Checkpoint #5) ทำหน้าตาแสดงสภาพอากาศ (WeatherScreen) ซึ่งไปเรียก Component Weather จาก Weather.js
- และหน้าตา แสดงเมนูสถานที่ตามรหัสไปรษณีย์ (ZipCodeScreen)
  1. 77000 (อ.เมือง จ.ประจวบคีรีขันธ์)
  2. 76120 (อ.ชะอำ จ.เพชรบุรี)
  3. 96000 (อ.เมือง จ.นราธิวาส)

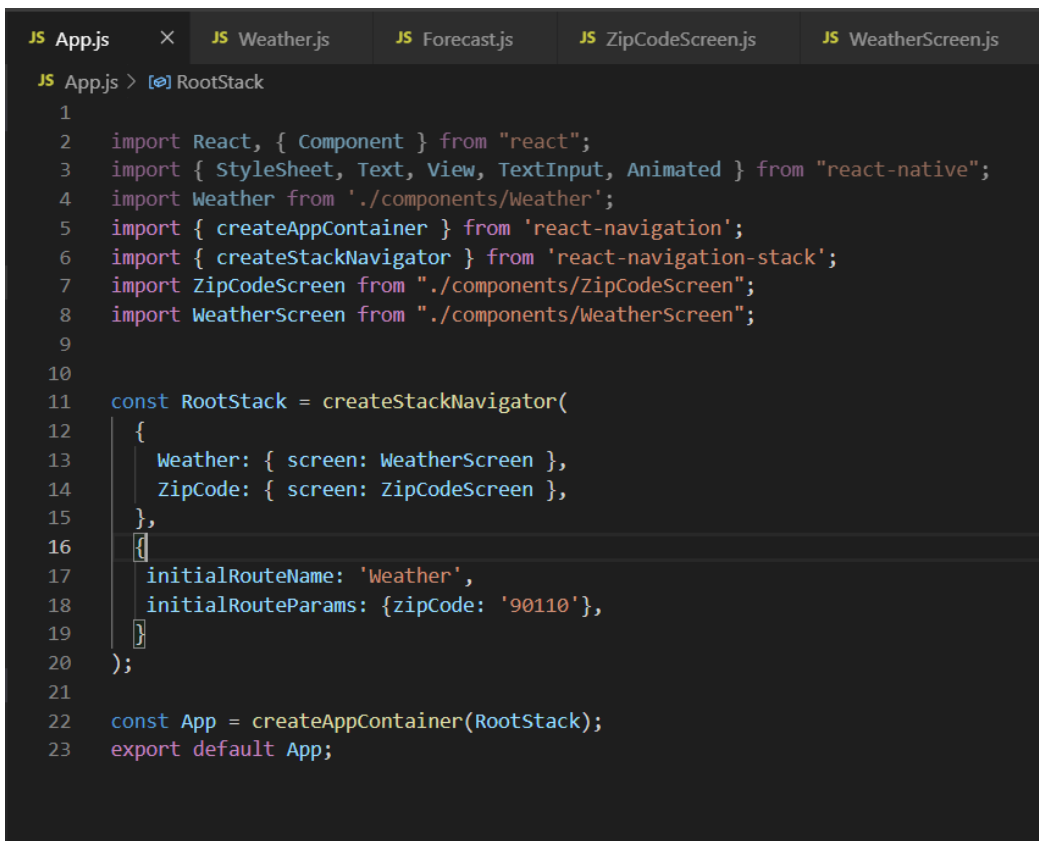
<a href="#">Weather</a> Choose a zip code	
Hat Yai, Songkhla	90110
Cha-am, Phetchaburi	76120
Prachuap Khiri Khan	77000
Narathiwat	96000
Trang	92000
Chiang Mai	50000
Khonkaen	40000
Chonburi	20000

- ในส่วนของ App.js สำหรับทำระบบสลับหน้าต่าง วิธีการจะแตกต่างไปจาก Labsheet เพราะ ไลบรารี react-native มีการเปลี่ยนแปลง โดยขั้นตอนในการทำ Routing มีดังนี้

1. ติดตั้งไลบรารีสามตัว ดังนี้

```
>> yarn add react-navigation
>> yarn add react-navigation-stack
>> expo install react-native-gesture-handler
```

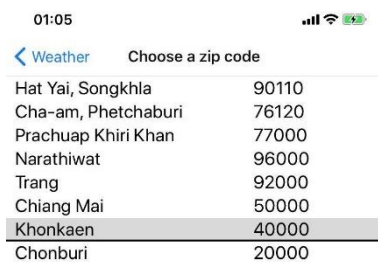
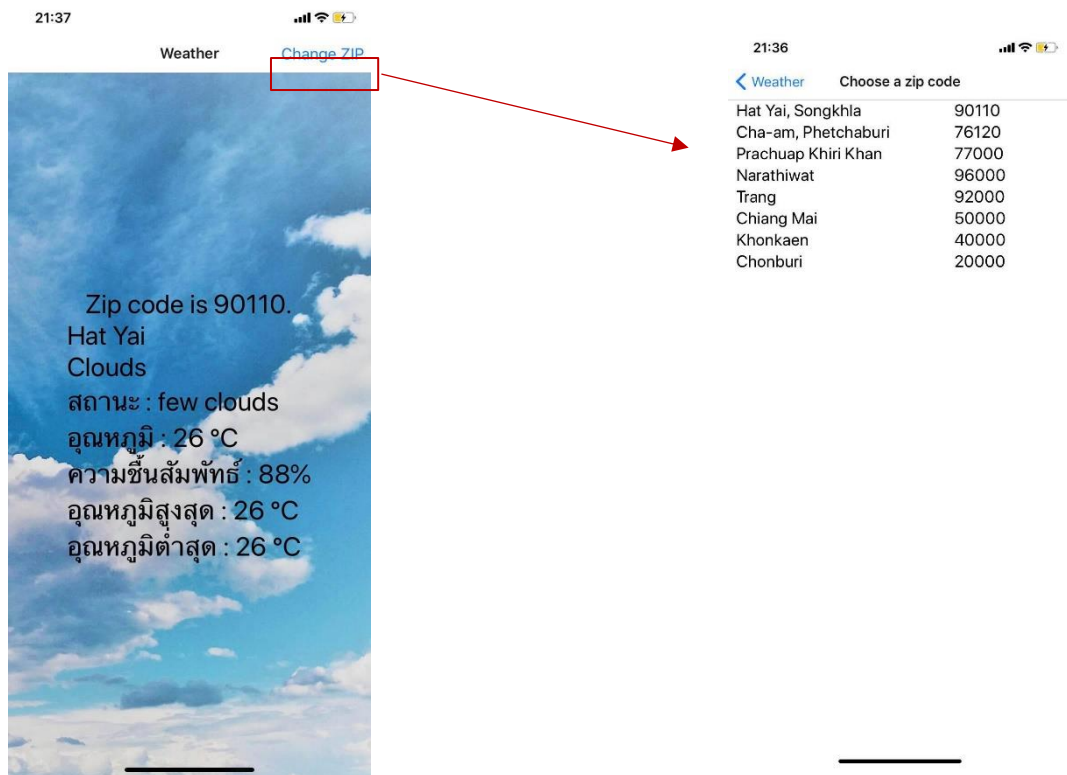
2. เปลี่ยน App.js เป็นดังนี้



```
JS App.js  X  JS Weather.js  JS Forecast.js  JS ZipCodeScreen.js  JS WeatherScreen.js
JS App.js > [0] RootStack
1
2  import React, { Component } from "react";
3  import { StyleSheet, Text, View, TextInput, Animated } from "react-native";
4  import Weather from './components/Weather';
5  import { createStackNavigator } from 'react-navigation';
6  import { createStackNavigator } from 'react-navigation-stack';
7  import ZipCodeScreen from './components/ZipCodeScreen';
8  import WeatherScreen from './components/WeatherScreen';
9
10
11  const RootStack = createStackNavigator(
12    {
13      Weather: { screen: WeatherScreen },
14      ZipCode: { screen: ZipCodeScreen },
15    },
16    {
17      initialRouteName: 'Weather',
18      initialRouteParams: {zipCode: '90110'},
19    }
20  );
21
22  const App = createStackNavigator(RootStack);
23  export default App;
```

ตัวอย่างหน้า App.js ที่เพิ่มเติมใหม่ใหม่

- สร้างปุ่ม Change ZIP เพื่อเปลี่ยนหน้าต่างจาก Weather ( WeatherScreen ) เป็น Choose a zip code
- หน้าต่างแสดงสภาพอากาศ ( WeatherScreen )



สามารถดูโค้ดเพิ่มเติม และ ตัวเต็มได้ที่ :

<https://github.com/Pichaya98/3SA04-6010110695.git>