

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN CUỐI KỲ
MÔN HỌC NHẬP ỨNG DỤNG DI ĐỘNG
ĐỒ ÁN: XÂY DỰNG APP DI ĐỘNG DỰ BÁO THỜI TIẾT

Giảng viên : Phan Nguyệt Minh

Sinh viên thực hiện: Nguyễn Tuấn Phương Nam – 15520519
Nguyễn Hoài Thu – 16521200

TP.HCM, Tháng 12/2019

LỜI CẢM ƠN

Lời đầu tiên chúng em xin chân thành gửi lời cảm ơn đến trường Đại học Công Nghệ Thông Tin đã tạo mọi điều kiện cho chúng em có môi trường học tập thật tốt, xin cảm ơn tất cả thầy cô đã hết lòng quan tâm, dạy dỗ và trang bị cho chúng em những kiến thức quý báu, cần thiết cho việc làm đồ án này cũng như công việc trong tương lai.

Chúng em xin chân thành cảm ơn cô Phan Nguyệt Minh đã tận tình hướng dẫn, chỉ bảo và giúp đỡ chúng em trong quá trình học tập. Trong thời gian học tập với cô, chúng em không ngừng tiếp thu thêm nhiều kiến thức bổ ích mà còn học tập được tinh thần làm việc, thái độ nghiêm cứu nghiêm túc, hiệu quả, đây là những điều rất cần thiết cho chúng em trong quá trình học tập và làm việc sau này.

Chúng em cũng chân thành cảm ơn những thầy cô trường Đại học Công Nghệ Thông đã trang bị cho chúng em những kiến thức quý báu trong thời gian vừa qua.

Mặc dù chúng em đã cố gắng hoàn thành đồ án này trong phạm vi khả năng cho phép nhưng vẫn không thể tránh khỏi những sai sót. Chúng em mong thầy cô và các bạn cảm thông và tận tình chỉ bảo.

TP.HCM , tháng 12 năm 2019

Nhóm sinh viên thực hiện

MỤC LỤC

PHẦN 1 : GIỚI THIỆU ĐỀ TÀI , MỤC TIÊU, CÁC CHỨC NĂNG CHÍNH

- | | |
|------------------------|---|
| 1. Giới thiệu đề tài | 4 |
| 2. Mục tiêu | 4 |
| 3. Các chức năng chính | 4 |

PHẦN 2 : PHÂN TÍCH CÁC YÊU CẦU CẦN THIẾT CỦA ĐỀ TÀI

- | | |
|---|---|
| 1. Lập các yêu cầu và chức năng cần thiết của app | 5 |
| 2. Phân tích chi tiết | 5 |

PHẦN 3 : THIẾT KẾ APP DI ĐỘNG

- | | |
|--|----|
| 1. Dữ liệu cung cấp cho app dự báo thời tiết | 6 |
| 2. Cấu trúc project, các components, các phương thức , kiểu dữ liệu, biến môi trường chính | 6 |
| a. Cấu trúc project | 6 |
| b. Các components | 8 |
| c. Các phương thức, kiểu dữ liệu, biến môi trường chính | 10 |
| 3. Giao diện | 11 |

PHẦN 4 : CÀI ĐẶT VÀ KIỂM THỬ

- | | |
|-------------|----|
| 1. Cài đặt | 14 |
| 2. Kiểm thử | 15 |

PHẦN 5 :KẾT LUẬN VÀ HƯỚNG MỞ RỘNG

| | |
|--|----|
| | 16 |
|--|----|

PHẦN 1 : GIỚI THIỆU ĐỀ TÀI , MỤC TIÊU, CÁC CHỨC NĂNG CHÍNH

1. GIỚI THIỆU ĐỀ TÀI

Trong thời đại đổi mới hiện nay, công nghệ thông tin đang phát triển mạnh mẽ, những ứng dụng di động đã không còn xa lạ gì đối với mọi người , thậm chí nhu cầu sử dụng của tất cả mọi người ngày càng tăng. Có rất nhiều ứng dụng hữu ích hiện nay như: App học ngoại ngữ , app nghe nhạc , các app game , Đa dạng về đề tài . Từ đó , chúng em đã suy nghĩ và quyết định đưa ra đề tài thân thiện với mọi người ‘App Dự Báo Thời Tiết’ . Đây là ứng dụng miễn phí giúp hiển thị thông tin chi tiết về tình hình thời tiết trong ngày cũng như đưa ra dự báo thời tiết về các ngày tiếp theo trong tuần.

2. MỤC TIÊU

Xây dựng App Dự Báo Thời Tiết đáp ứng được nhu cầu của người dùng, không chỉ để theo dõi tình hình thời tiết trong những ngày sắp tới mà còn để trang trí cho màn hình smartphone của mình thêm đẹp mắt.

3. CÁC CHỨC NĂNG CHÍNH

App Dự Báo Thời Tiết là một ứng dụng chạy trên điện thoại sử dụng hệ điều hành Android gồm 2 chức năng chính :

- + Thông báo và hiển thị thời tiết hiện tại.
- + Dự báo thời tiết trong vòng 5 ngày tiếp theo , thông tin sẽ được cập nhật liên tục mỗi 3 tiếng.

PHẦN 2 : PHÂN TÍCH CÁC YÊU CẦU CẦN THIẾT CỦA ĐỀ TÀI

1. LẬP CÁC YÊU CẦU VÀ CHỨC NĂNG CẦN THIẾT CỦA APP

- Tìm kiếm địa điểm và những nơi cần xem thời tiết
- Bảng hiển thị thời tiết hiện tại
- Bảng hiển thị thời tiết thời gian sắp tới
- Mục chọn xem thời tiết theo thời gian (5 ngày tiếp theo, mỗi 3 tiếng).

2. PHÂN TÍCH CHI TIẾT

a. Tìm kiếm địa điểm và những nơi cần xem thời tiết

Cần phải đáp ứng nhu cầu tìm kiếm của người dùng , cung cấp nhiều địa điểm đa dạng để mọi người có thể nắm bắt nhiều thông tin hơn.

b. Bảng hiển thị thời tiết hiện tại

Cung cấp đầy đủ các thông tin hiển thị thời tiết hiện tại : Tên thành phố, nhiệt độ , nắng mưa , tốc độ gió,....

c. Bảng hiển thị thời tiết sắp tới

Cung cấp các thông tin hiển thị thời tiết sắp tới : Giờ, ngày, nhiệt độ , độ ẩm.

d. Mục chọn xem thời tiết theo thời gian

Cung cấp các lựa chọn để người dùng có thể chọn ngày giờ sắp tới từ đó thông tin thời tiết sẽ hiển thị trên bảng hiển thị thời tiết sắp tới.

PHẦN 3 : THIẾT KẾ APP DI ĐỘNG VỚI REACT NATIVE

1. DỮ LIỆU CUNG CẤP CHO APP DỰ BÁO THỜI TIẾT

Project dự báo thời tiết sẽ sử dụng việc Call API về để cung cấp data .

React Native cung cấp module Fetch API để sử dụng cho việc kết nối networking APIs. Fetch API là một API đơn giản cho việc gửi và nhận request bằng js. Với fetch thì việc thực hiện các yêu cầu và xử lý phản hồi dễ dàng hơn so với XMLHttpRequest cũ.

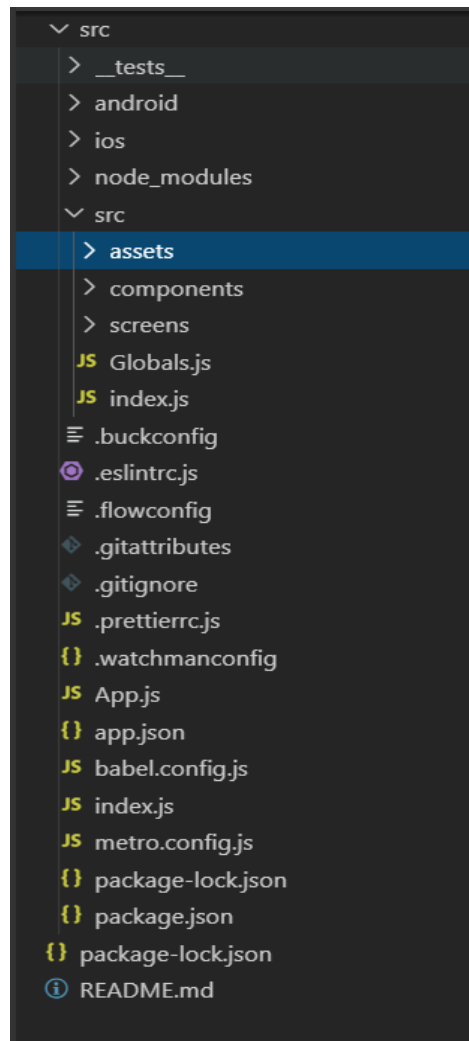
Phương thức fetch () nhận đầu vào là url để truy vấn rồi trả về response của request đó.

```
✓ getCurrentWeather(city) {  
  var url = this.CURRENTWEATHER_API + '&appid=' + this.API_KEY + '&q=' + city;  
  return fetch(url).then(response => response.json());  
},  
✓ getForecast(city) {  
  var url = this.FORECAST_API + '&appid=' + this.API_KEY + '&q=' + city;  
  return fetch(url).then(response => response.json());  
},  
✓ iconUrl(icon) {  
  const url = 'http://openweathermap.org/img/wn/' + icon + '.png';  
  return url;  
},  
};
```

Hình 3 . Cách sử dụng fetch()

2. CẤU TRÚC PROJECT, CÁC COMPONENTS, CÁC PHƯƠNG THỨC CHÍNH.

a. Cấu trúc project



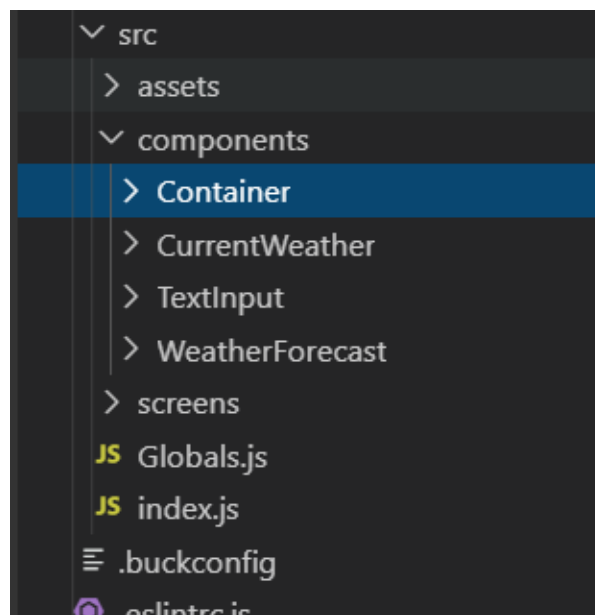
Hình 1 . Cấu trúc project

+ Ý nghĩa một số file cơ bản :

- **.babel.config.js** : định nghĩa cách JSX sẽ được chuyển đổi thành JS
- **.gitignore** : bỏ qua thay đổi tất cả các file được định nghĩa trong file gitignore.
- **.watchmanconfig** : xác định cách các file nên được xem để thay đổi.
- **App.js** là trung tâm của dự án, tương tự như root component trong React.js.
- **app.json** : cấu hình các phần trong ứng dụng.
- **package-lock.json**: nó được tạo ra để chỉ định và tới các kho lưu trữ thư viện mà chúng ta khai báo để khi build app.

- **package.json**: là nơi chứa tên ứng dụng phiên bản và các thư viện mà chúng ta khai báo khi chia sẻ source lên git hay đi đâu chúng ta sẽ không cần copy nhưng thư viện đó theo mà chỉ cần chia sẻ nhưng file cần thiết khi người dùng tải source và thực hiện lệnh npm install thì nó sẽ vào đây để tìm và download nhưng thư viện cần thiết giống như maven trong web.
- **Thư mục node_modules** : Thư mục này dùng để chứa tất cả các thư viện mà chúng ta sử dụng trong quá trình xây dựng app.

b. Các components



Hình 2 . Các components chính

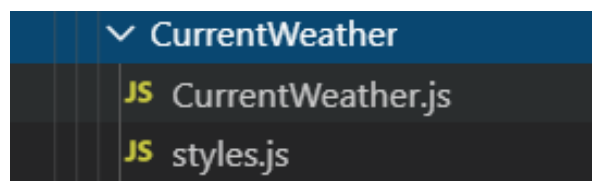
Concept cơ bản của React Native là việc chia nhỏ app di động thành các Component. Có thể nói tất những gì ta thấy trên app đều là Component.

Component chính là 1 trong những lí do khiến ứng dụng viết bằng React/React Native có khả năng tái sử dụng và tăng khả năng mở rộng. Đây là điểm mà tôi rất thích ở React, vì nó buộc bạn phải nghĩ đến việc thiết kế và lọc ra những thành phần chung trong một ứng dụng thay vì ném mọi thứ vào 1 cái container view.

React Native cung cấp cho chúng ta rất nhiều các component mặc định, ví dụ như Text. Khi chúng ta build app trên iOS, React Native sẽ đảm nhận việc chuyển component này thành UIView chứa đoạn text, còn khi build trên Android thì sẽ thành TextView native.

+ Ý nghĩa các components chính trong project ‘app dự báo thời tiết’:

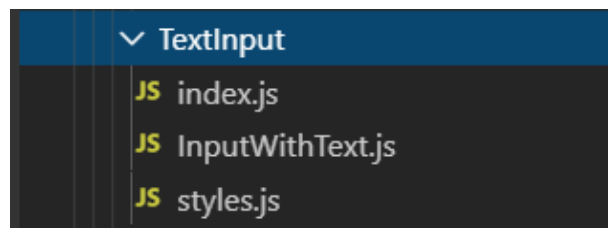
- **CurrentWeather :**



Đây là components code của bảng thời tiết hiện tại .

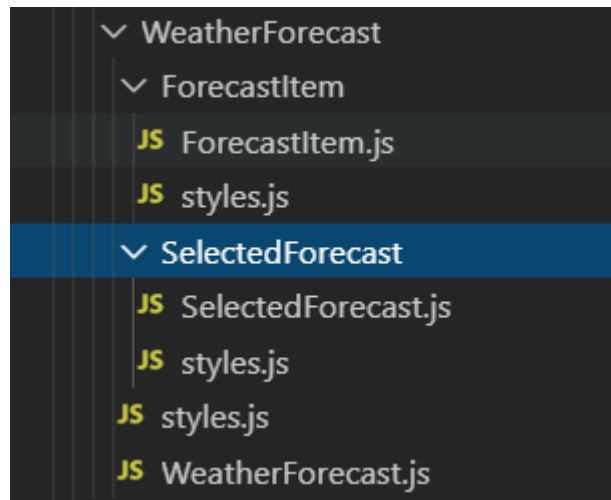
- **TextInput:**

Components code của thanh Search trong app.



- **WeatherForecast:**

Components code của bảng hiển thị thời tiết sắp tới và mục chọn thời gian xem thời tiết.



c. Các phương thức, kiểu dữ liệu, biến môi trường chính

+ **fetch()**: như đã giới thiệu ở phần ‘DỮ LIỆU CUNG CẤP CHO APP DỰ BÁO THỜI TIẾT’.

+ **props**: là viết tắt của Properties. Khi sử dụng nó không bao giờ nên thay đổi giá trị của nó, hay nói cách khác, đây là một dữ liệu immutable.

Các component nhận props từ component cha. Chúng ta không được thay đổi giá trị của props trong các component này mà chỉ được phép đọc giá trị ra thôi. Trong React thì dữ liệu sẽ đi theo một chiều, có nghĩa là từ component cha => các component con.

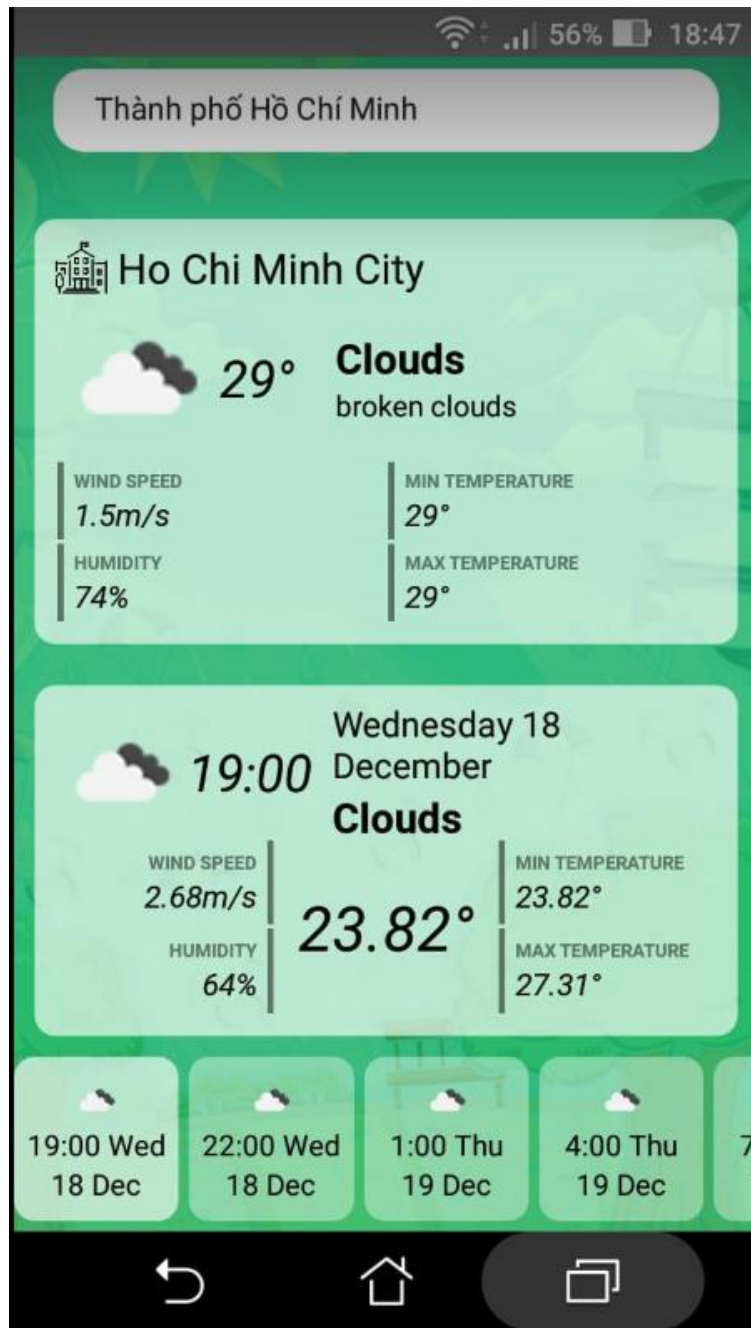
+ **State**: nó thì hoạt động khác với props. state là dữ liệu nội bộ của một Component, trong khi props là dữ liệu được truyền cho Component. Chính vì vậy chúng ta hoàn toàn có thể thay đổi state, và coi nó là một kiểu dữ liệu mutable.

Tuy vậy, nhưng đừng bao giờ thay đổi trực tiếp biến `this.state`. Thay vào đó hãy dùng hàm `setState` để cập nhật giá trị. Sở dĩ chúng ta cần dùng hàm này là do nó sẽ kích hoạt việc render lại component và tất cả component con nằm trong nó, còn thay đổi `this.state` thì không. Còn một vấn đề nữa, đó là `setState` chạy bất đồng bộ, vậy nên nếu chúng ta tiến hành đọc ra giá trị state ngay sau khi `setState` thì chưa chắc giá trị sẽ được update lên mới nhất.

3. GIAO DIỆN



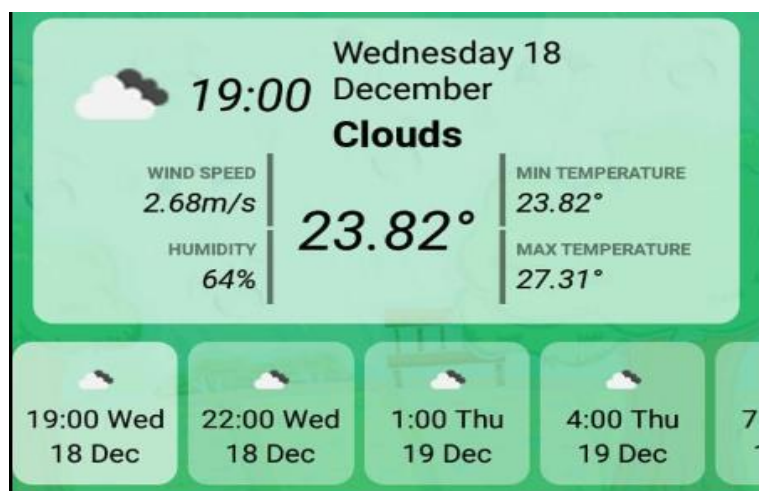
Hình 4 . Màn hình chờ



Hình 5 . Màn hình chính



Hình 6 . CurrentWeather



Hình 7. WeatherForecast

PHẦN 4 : CÀI ĐẶT VÀ KIỂM THỬ

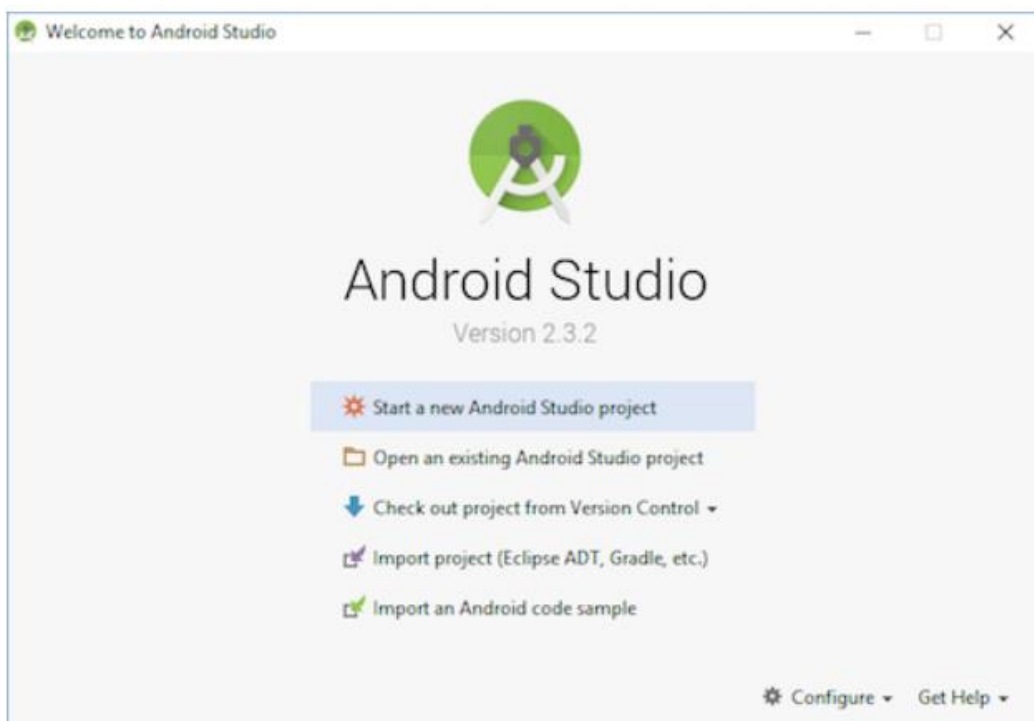
1. CÀI ĐẶT

a. Môi trường

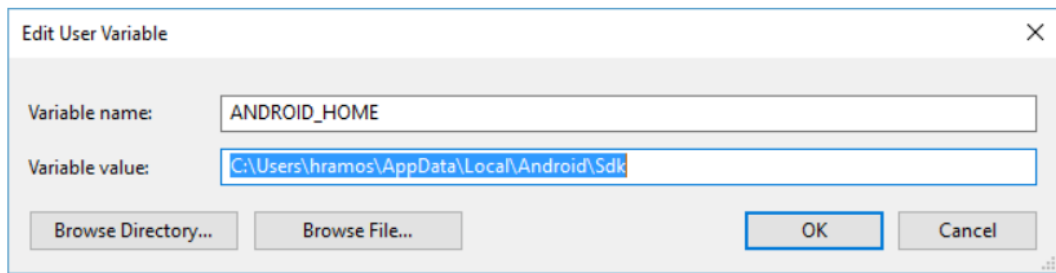
- + Chúng ta cần cài đặt React Native để chạy project dưới đây là sơ bộ:
 - Cài đặt node, python2, JDK thông qua Chocolatey: mở Administrator Command Prompt và nhập

```
choco install -y nodejs.install python2 jdk8
```

- Android development environment:
 - o Cài đặt Android Studio
 - o Cài đặt Android SDK



- o Cấu hình ANDROID_HOME:



```
c:\Users\YOUR_USERNAME\AppData\Local\Android\Sdk
```

- Thêm platform-tools vào Path:

```
c:\Users\YOUR_USERNAME\AppData\Local\Android\Sdk\platform-tools
```

➔ Về chi tiết cài đặt chúng ta có thể tìm hiểu thêm trên trang chủ của React Native : <https://facebook.github.io/react-native/docs/getting-started>

b. Ngôn Ngữ

Trong project App Dự Báo Thời Tiết chúng ta sẽ sử dụng javascript

2. KIỂM THỬ

- + App hoạt động ổn định .
- + Các api dữ liệu được cập nhật và hiển thị đúng theo như mong muốn.
- + Giao diện phù hợp thân thiện với người dùng.

PHẦN 5 : KẾT LUẬN VÀ HƯỚNG MỞ RỘNG

1. KẾT LUẬN

- + Nhóm đã hoàn thành được đề tài cùng với mục tiêu đề ra.
- + Tạo ra được sản phẩm thân thiện với người dùng
- + Các chức năng phù hợp .

2. HƯỚNG MỞ RỘNG

- + Tìm hiểu thêm nhưng nhu cầu cần thiết của mọi người .Từ đó xây dựng thêm các chức năng khác cho App.
- + Thử áp dụng thêm các công nghệ hiện đại vào dự án.
- + Nâng cấp giao diện để người sử dụng ưa thích.