

Lập trình Mobile

Ths. Đỗ Phúc Thịnh

Giới thiệu khóa học



- Lập trình Mobile: Lập trình phần mềm cho các thiết bị di động
 - Sử dụng React Native để lập trình phần mềm cho điện thoại di động
- Thời lượng:
 - Số tín chỉ: 3 tín chỉ
 - Lý thuyết: 2 tín chỉ ~ 30 tiết
 - Thực hành: 1 tín chỉ ~ 30 tiết
 - Tự học: 6 tc ~ 120 tiết

- Khóa học này sẽ **hướng dẫn** về:
 - Cách **thiết kế** một ứng dụng trên **điện thoại di động**
 - Xử lý các **tác vụ cơ bản** trên ứng dụng
- Và **không hướng dẫn** về:
 - Lập trình game mobile

- Đi học đầy đủ và làm bài tập thực hành hàng tuần
- **Không** chơi game, gục đầu trong lớp
- Đeo thẻ tên (*đồng phục + cà vạt*) trong quá trình học
- Tôn chỉ:

Đã đi học thì phải thu được kiến thức

- Điểm thường kỳ: **50% (6 cột điểm)**
 - Bài tập thực hành và điểm chuyên cần
- Điểm giữa kỳ: **20%**
 - Báo cáo đề tài
- Điểm cuối kỳ: **30%**
 - Báo cáo đồ án cuối kỳ
- Điểm ẩn: **xx%**
 - Điểm thái độ học tập

- **Đề tài giữa kỳ** do thầy giao
- Tìm hiểu đề tài được giao
- Báo cáo theo nhóm đề tài được giao
- Thời gian báo cáo: **Tuần thứ sáu + bảy** của khóa học

- **Đồ án cuối kỳ** tự chọn
- Báo cáo theo nhóm đồ án đã chọn
- Thời gian báo cáo: **Tuần cuối cùng** của khóa học

- Mỗi nhóm **tối đa 3 người, tối thiểu 1 người**

React Native



facebook

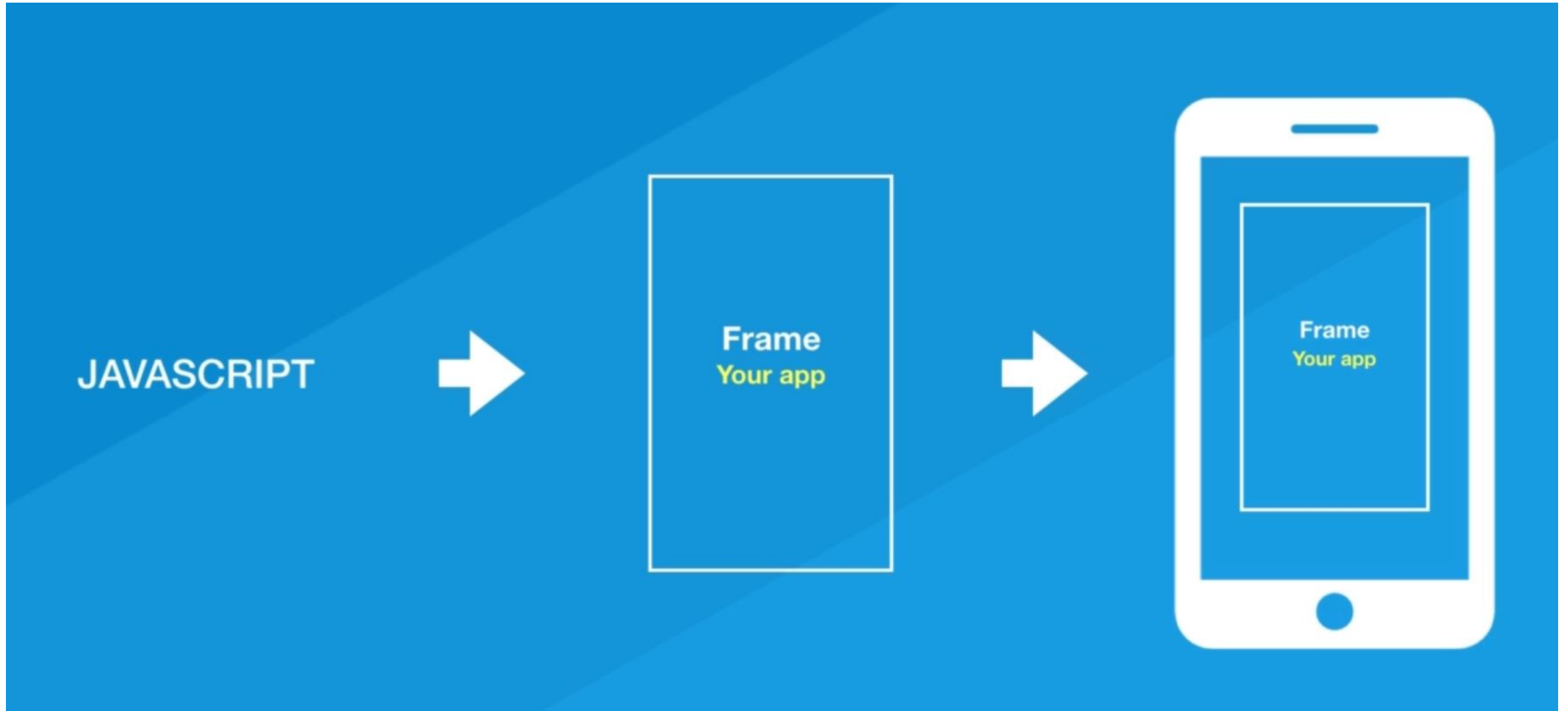
Walmart 

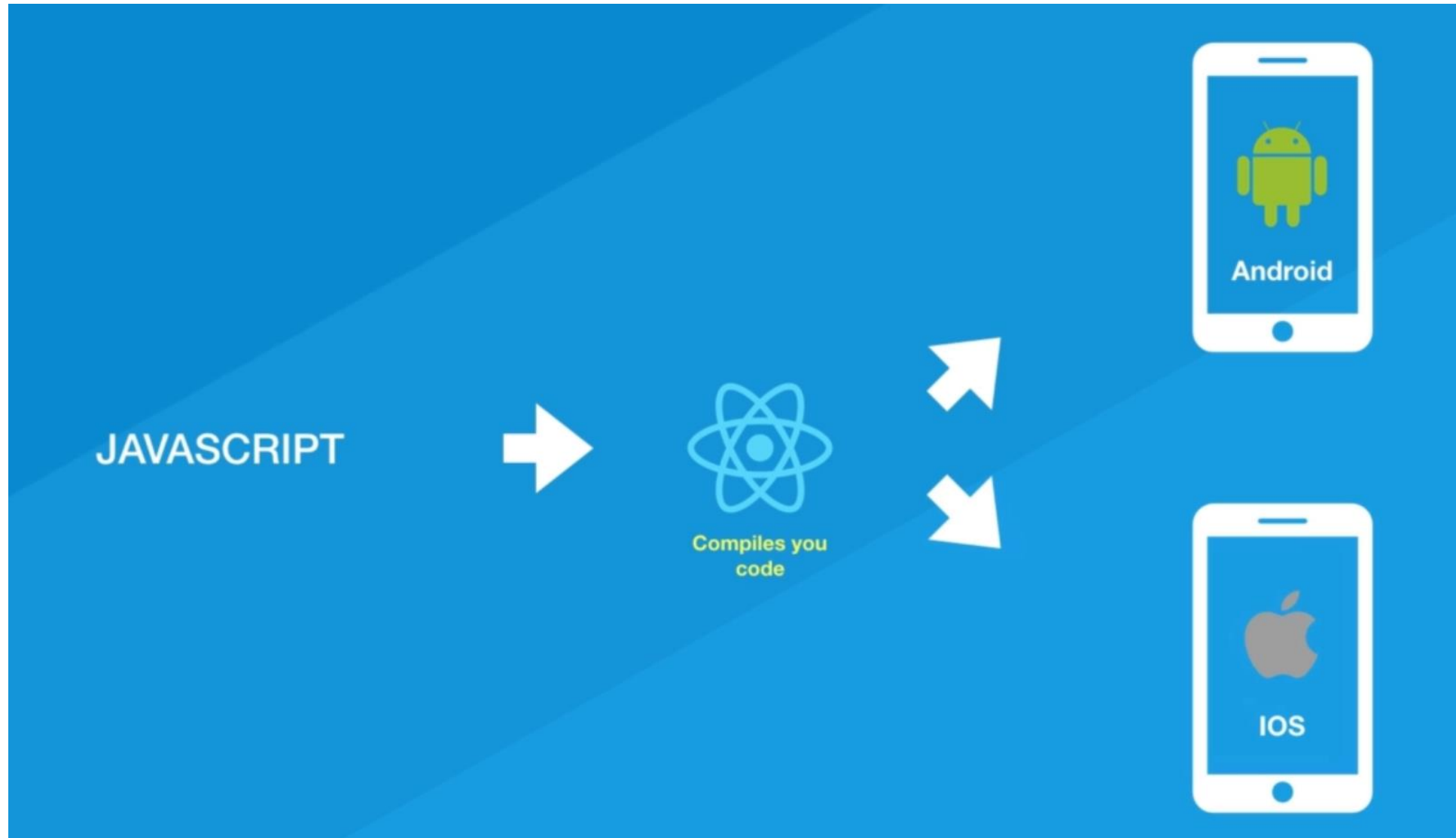
 airbnb

WIX.com









Tại sao có thể code 1 được 2?

React for the Web	Native Component (Android)	Native Component (iOS)	React Native
<div>	android.view	UIView	<View>
<input>	EditText	UITextField	<TextInput>
...

Tại sao có thể code 1 được 2?

15

React + React Native App

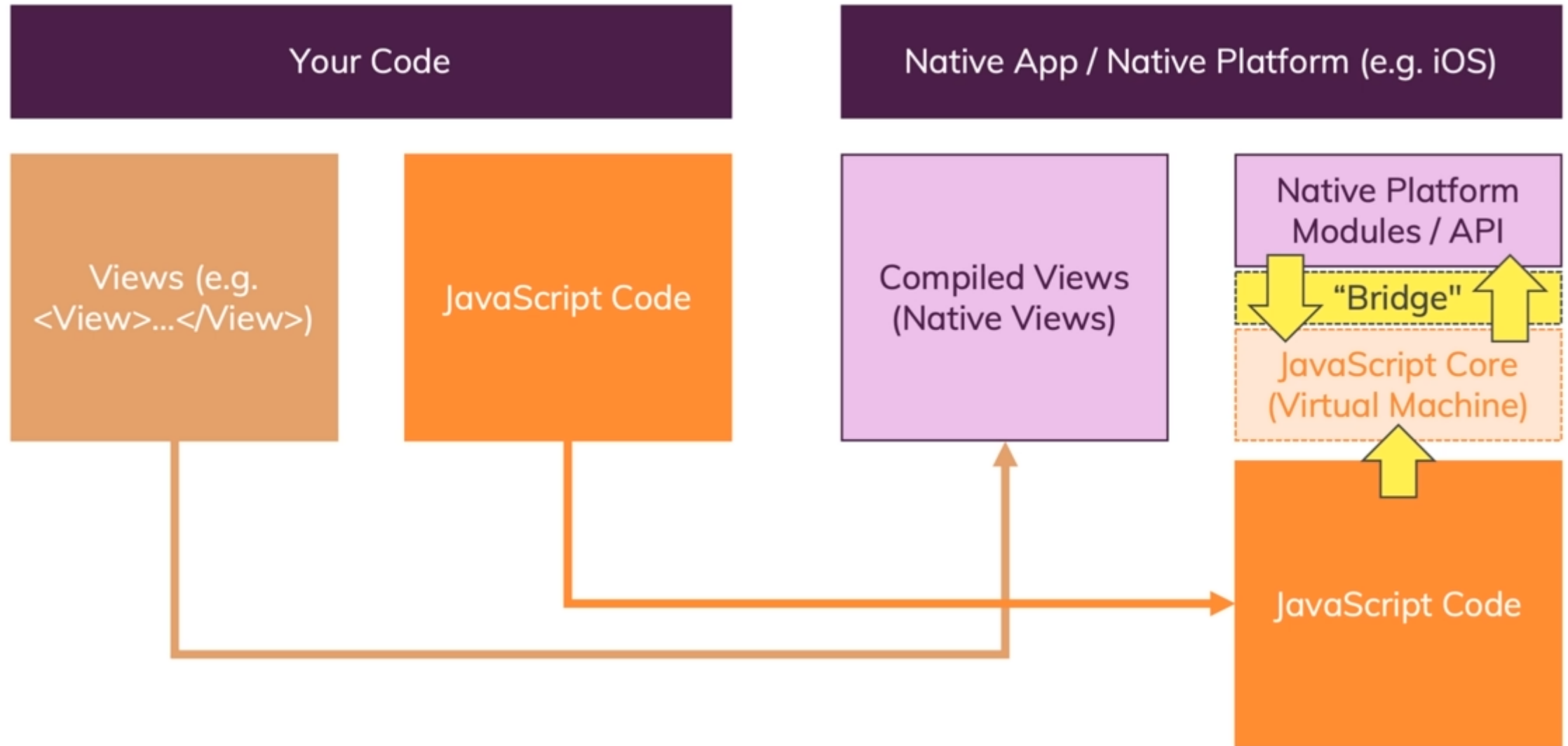
```
const App = props => {  
  return (  
    <View>  
      <Text>Hello there!</Text>  
    </View>  
  );  
}
```

Compiled to

Views are
compiled!

Real Native App

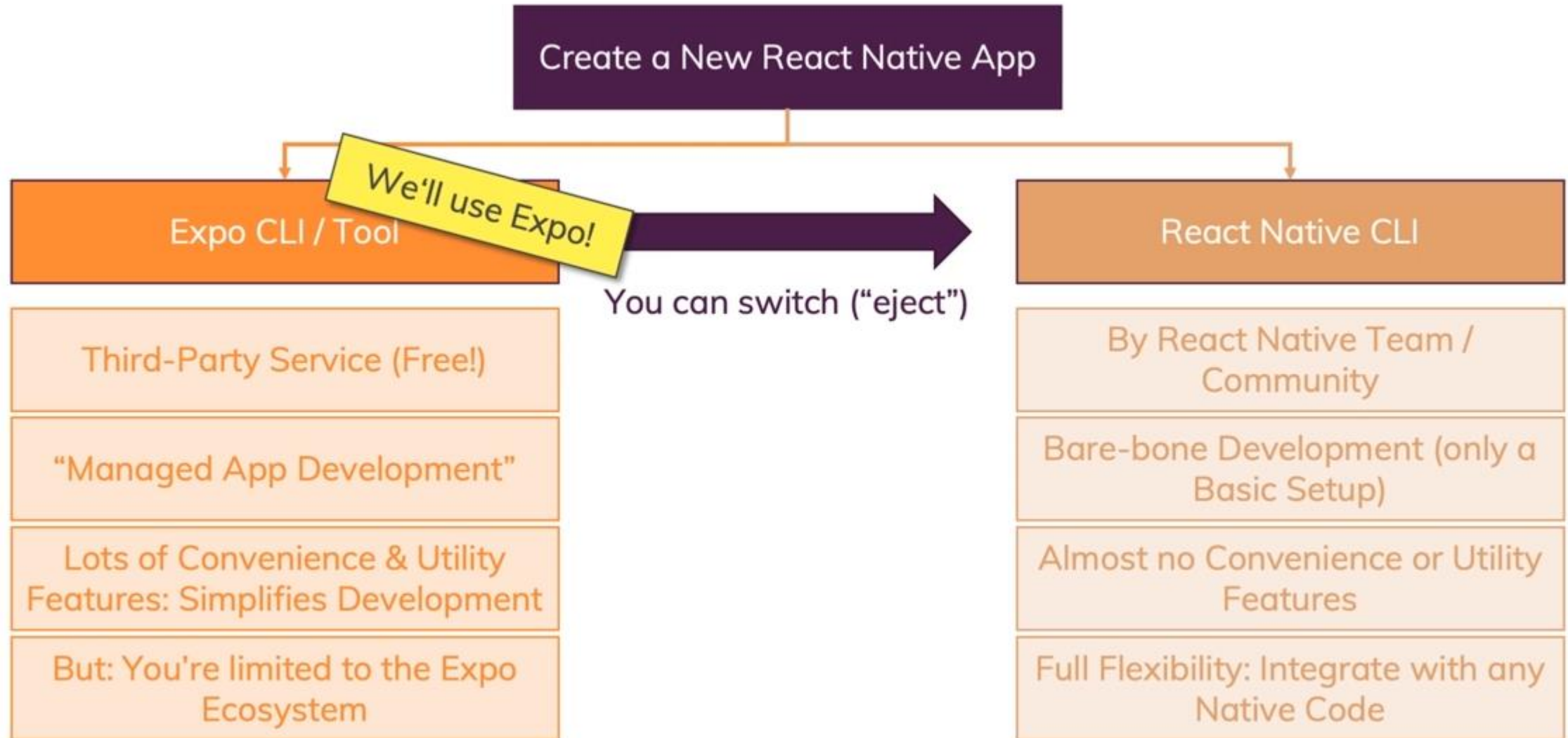
Tại sao có thể code 1 được 2?



React Native CLI



- **Có 2 loại CLI (Command line interfaces)**
 - **Expo CLI**
 - **React Native CLI**



Cách cài đặt



- Hướng dẫn: <https://reactnative.dev/docs/environment-setup>

Setting up the development environment

This page will help you install and build your first React Native app.

If you are new to mobile development, the easiest way to get started is with Expo CLI. Expo is a set of tools built around React Native and, while it has many [features](#), the most relevant feature for us right now is that it can get you writing a React Native app within minutes. You will only need a recent version of Node.js and a phone or emulator. If you'd like to try out React Native directly in your web browser before installing any tools, you can try out [Snack](#).

If you are already familiar with mobile development, you may want to use React Native CLI. It requires Xcode or Android Studio to get started. If you already have one of these tools installed, you should be able to get up and running within a few minutes. If they are not installed, you should expect to spend about an hour installing and configuring them.

[Expo CLI Quickstart](#) [React Native CLI Quickstart](#)

Assuming that you have [Node 12 LTS](#) or greater installed, you can use npm to install the Expo CLI command line utility:

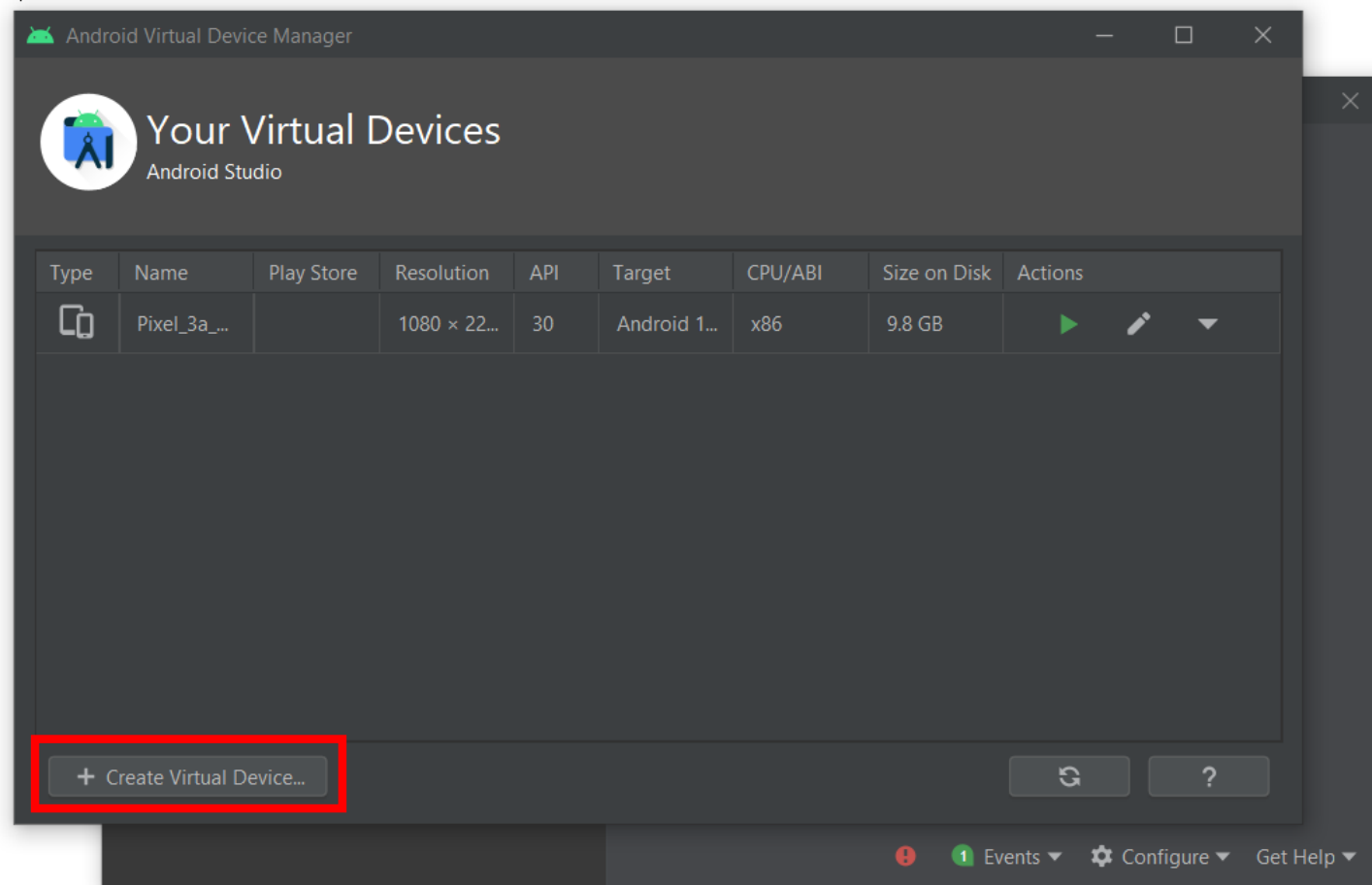
- Cài đặt **Visual Studio Code**: <https://code.visualstudio.com/>
- Có thể cài thêm các **Extension** (tùy chọn):
 - **Material Icon Theme**: Hỗ trợ đổi giao diện Icon
 - **React Native Snippet**: Hỗ trợ code nhanh
 - **Kite AutoComplete AI Code**: Hỗ trợ code siêu nhanh
 - ...

- Cài đặt Android Studio:

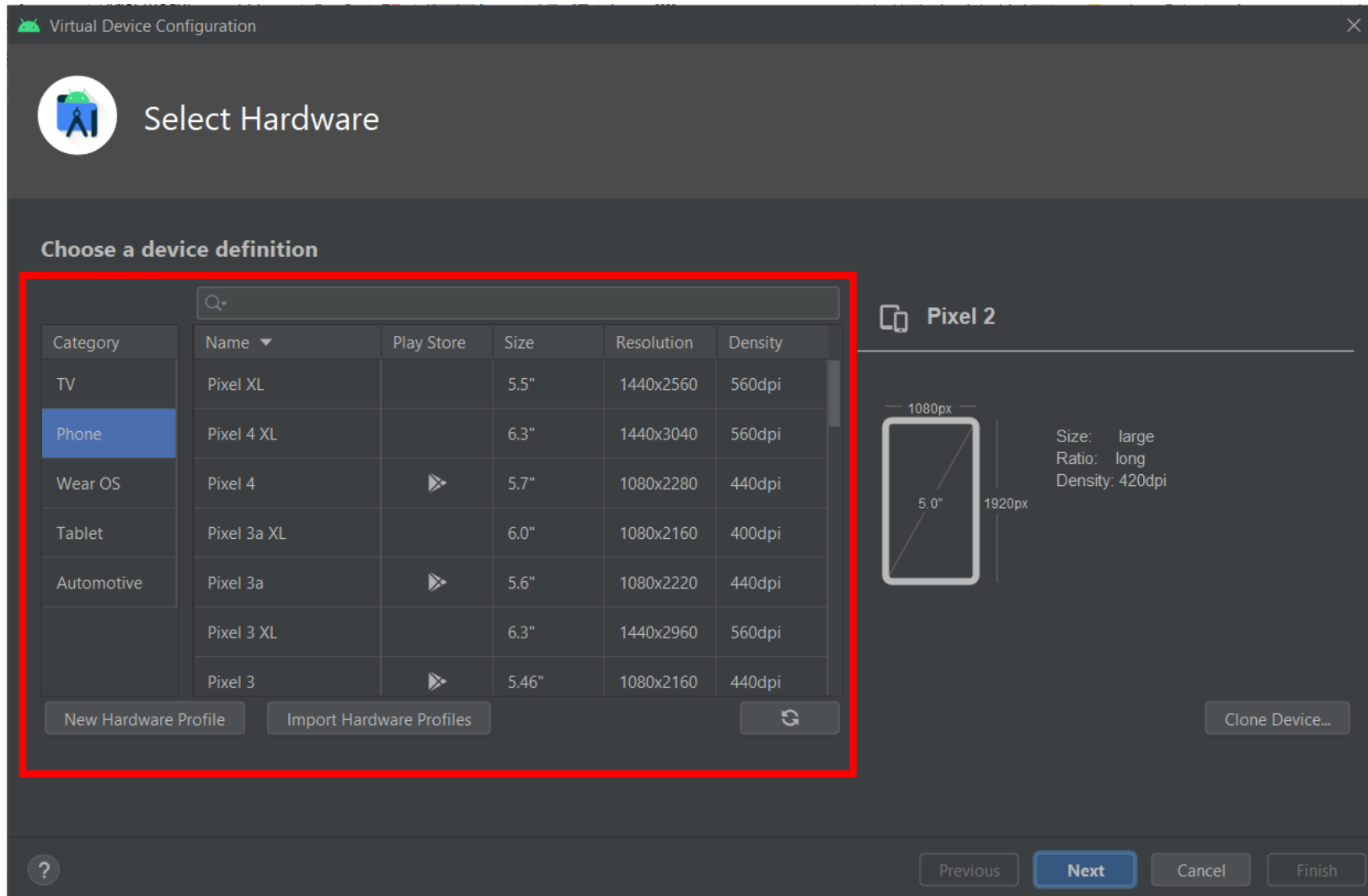
<https://developer.android.com/studio/index.html>

- Chọn các lựa chọn sau:
 - Android SDK
 - Android SDK Platform
 - **Android Virtual Device**

- Mở **Android Studio** → Chọn **Configure** → **AVD Manager** → **Create Virtual Device**



- Chọn máy ảo mong muốn



- **Android:**

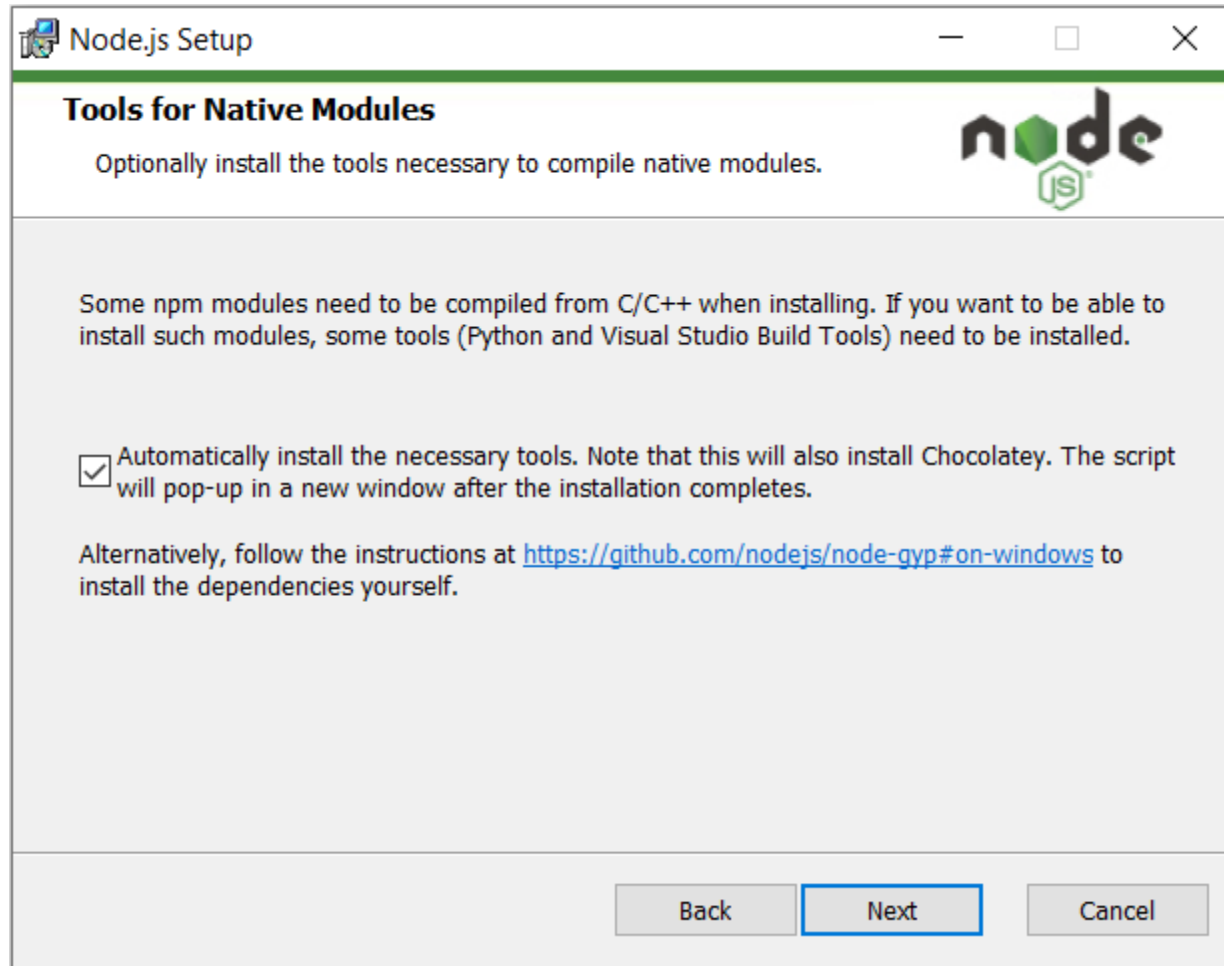
<https://play.google.com/store/apps/details?id=host.exp.exponent>

- **iOS:**

<https://apps.apple.com/app/apple-store/id982107779>



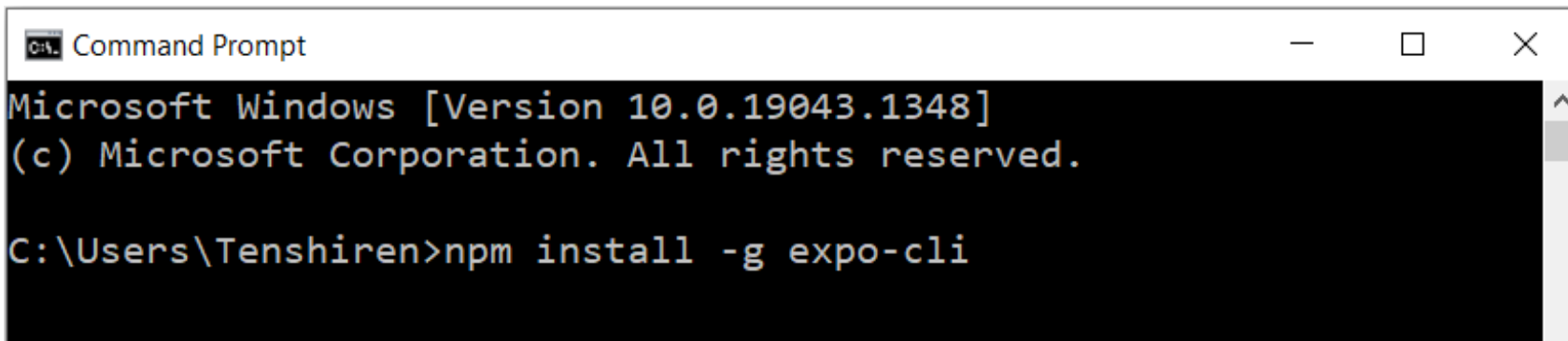
- Cài đặt **Node.js**: <https://nodejs.org/en/download/>



- Mở **Command Prompt** (Tìm từ khóa **cmd**)
- Gõ lệnh: vào Command Prompt, để cài đặt **Expo CLI** gõ

`npm install -g create-expo-app`

`Bản cũ: npm install -g expo-cli`



```
Command Prompt
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

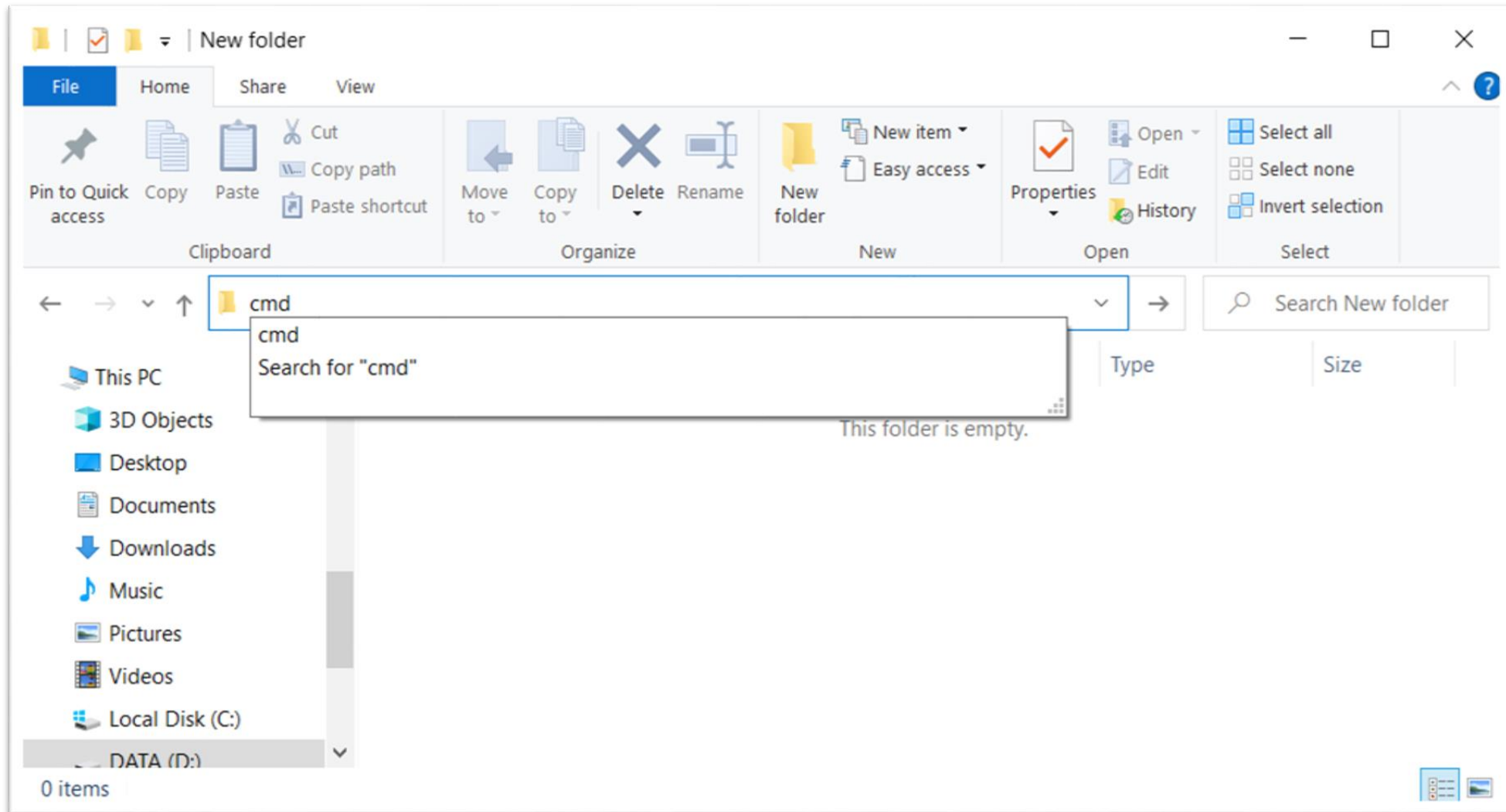
C:\Users\Tenshiren>npm install -g expo-cli
```

- **Visual Studio Code** để lập trình
- **Android Studio** để chạy **máy ảo** (hoặc chạy trực tiếp trên trình duyệt)
- **Expo Go** (cài đặt trên điện thoại) để chạy trên **máy thật**
- **Node.js**
- **Expo CLI** (`npm install -g create-expo-app`)

Tạo Project mới

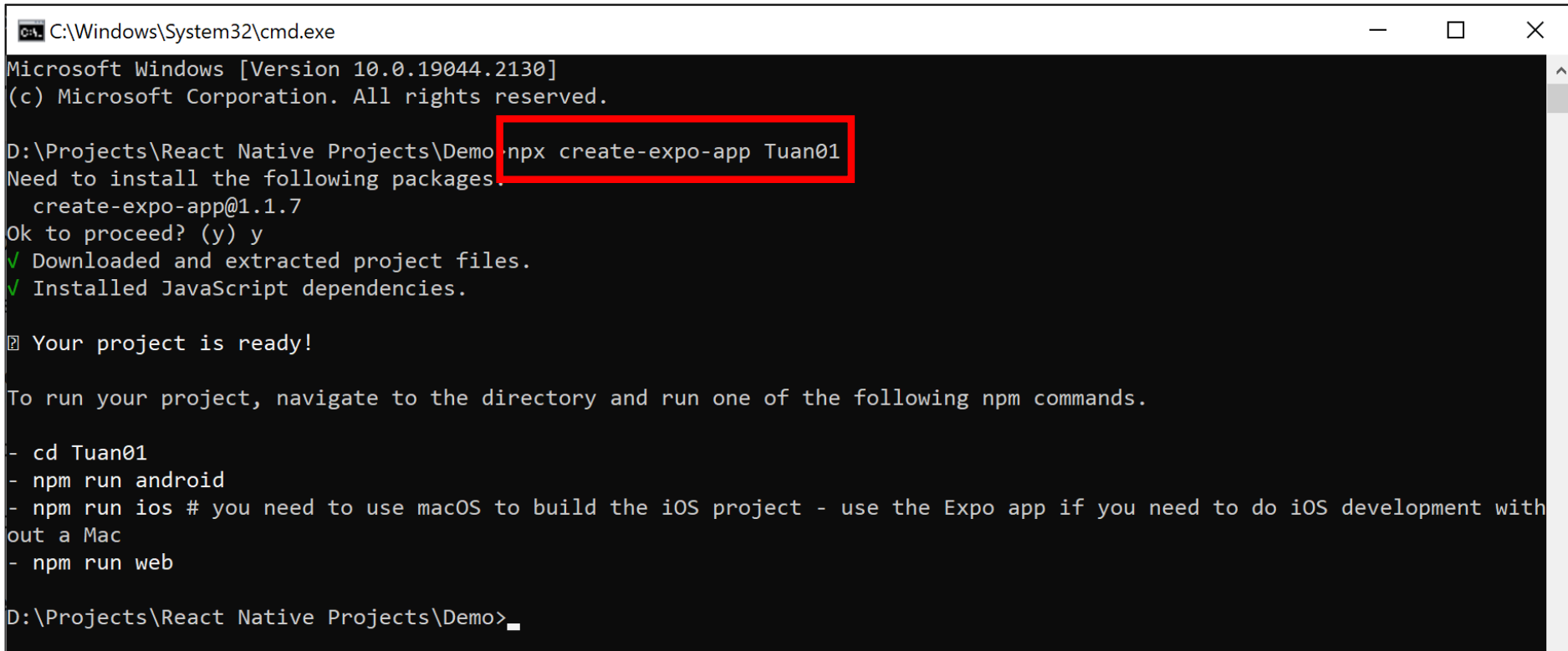


Bước 1: Vào thư mục bất kỳ và gõ cmd



Bước 2: Gõ lệnh: **npx create-expo-app** TenProject

Bản cũ: expo init Tên_Project (Chọn blank)

A screenshot of a Windows Command Prompt window. The title bar shows 'C:\Windows\System32\cmd.exe'. The window content shows the following text:

```
Microsoft Windows [Version 10.0.19044.2130]
(c) Microsoft Corporation. All rights reserved.

D:\Projects\React Native Projects\Demo>npx create-expo-app Tuan01
Need to install the following packages:
  create-expo-app@1.1.7
Ok to proceed? (y) y
✓ Downloaded and extracted project files.
✓ Installed JavaScript dependencies.

🎉 Your project is ready!

To run your project, navigate to the directory and run one of the following npm commands.

- cd Tuan01
- npm run android
- npm run ios # you need to use macOS to build the iOS project - use the Expo app if you need to do iOS development without a Mac
- npm run web

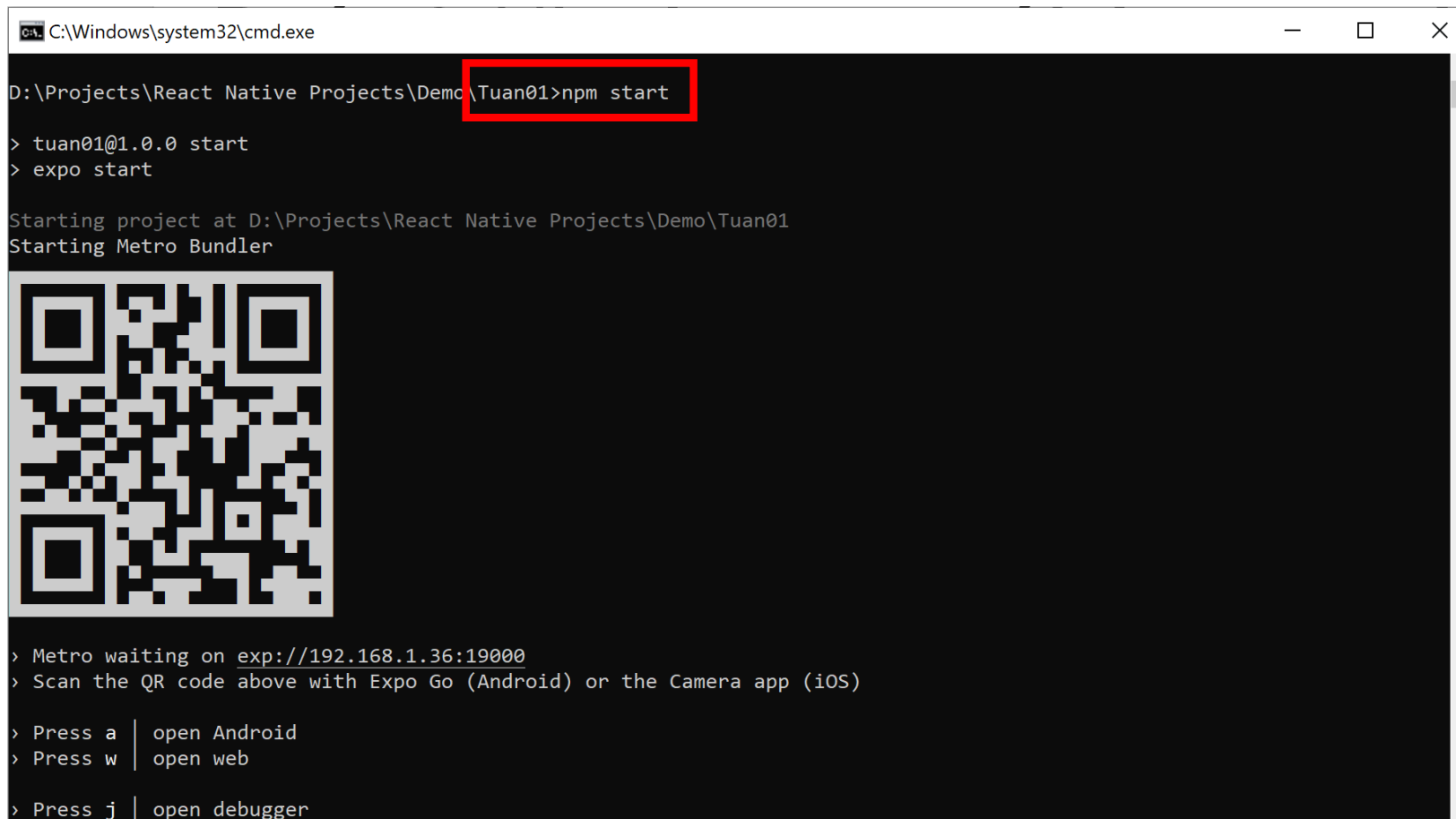
D:\Projects\React Native Projects\Demo>_
```

The command 'npx create-expo-app Tuan01' is highlighted with a red rectangular box.

- Tạo project thành công


```
👉 Your project is ready!  
  
To run your project, navigate to the directory and run one of the following npm commands.  
  
- cd ExpoDemo  
- npm start # you can open iOS, Android, or web from here, or run them directly with the commands below.  
- npm run android  
- npm run ios # requires an iOS device or macOS for access to an iOS simulator  
- npm run web  
  
D:\Projects\React-Native-Projects>
```

Bước 3: Vào thư mục mới được tạo → Gõ cmd vào ô địa chỉ → Gõ lệnh: **`npx expo start`** (Bản cũ: *`npm start`*)



```
C:\Windows\system32\cmd.exe
D:\Projects\React Native Projects\Demo\Tuan01>npx start
> tuan01@1.0.0 start
> expo start

Starting project at D:\Projects\React Native Projects\Demo\Tuan01
Starting Metro Bundler




> Metro waiting on exp://192.168.1.36:19000
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

> Press a | open Android
> Press w | open web
> Press j | open debugger
```

- Kết quả:

```
C:\Windows\system32\cmd.exe
Starting project at D:\Projects\React Native Projects\Demo\Tuan01
Starting Metro Bundler



> Metro waiting on exp://192.168.1.36:19000
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

> Press a | open Android
> Press w | open web

> Press j | open debugger
> Press r | reload app
> Press m | toggle menu

> Press ? | show all commands

Logs for your project will appear below. Press Ctrl+C to exit.
```

- Cài thư viện để chạy trên web (vào thư mục chứa project):

`npx expo install react-native-web@~0.18.9 react-dom@18.1.0 @expo/webpack-config@^0.17.2`

```
npx expo install react-native-web@~0.18.9 react-dom@18.1.0 @expo/webpack-config@^0.17.2
```

```
If you're not using web, please ensure you remove the "web" string from the platforms array in the project Expo config.
```

```
-
```

- Kết quả khi cài thành công:

```
C:\Windows\System32\cmd.exe

D:\Projects\React Native Projects\Demo\Tuan01>npx expo install react-native-web@~0.18.9 react-dom@18.1.0 @expo/webpack-c
onfig@^0.17.2
> Installing 3 SDK 47.0.0 compatible native modules using npm
> npm install
npm WARN deprecated stable@0.1.8: Modern JS already guarantees Array#sort() is a stable sort, so this library is depreca
ted. See the compatibility table on MDN: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Object
s/Array/sort#browser_compatibility
npm WARN deprecated chokidar@2.1.8: Chokidar 2 does not receive security updates since 2019. Upgrade to chokidar 3 with
15x fewer dependencies
npm WARN deprecated chokidar@2.1.8: Chokidar 2 does not receive security updates since 2019. Upgrade to chokidar 3 with
15x fewer dependencies
npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. new code should use the URLSearchParams
API instead.
npm WARN deprecated svgo@1.3.2: This SVGO version is no longer supported. Upgrade to v2.x.x.

added 687 packages, and audited 1835 packages in 44s

136 packages are looking for funding
  run `npm fund` for details

28 vulnerabilities (2 moderate, 22 high, 4 critical)

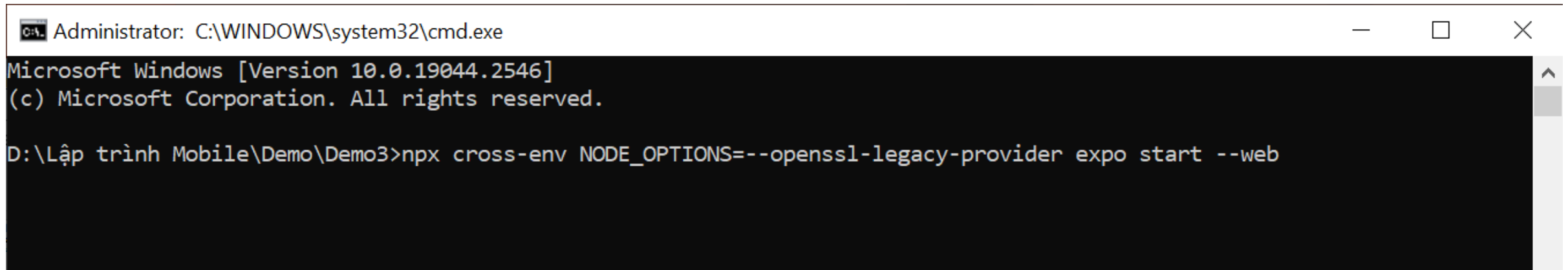
To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
```

- Để chạy trên trình duyệt, tại thư mục chứa project gõ:

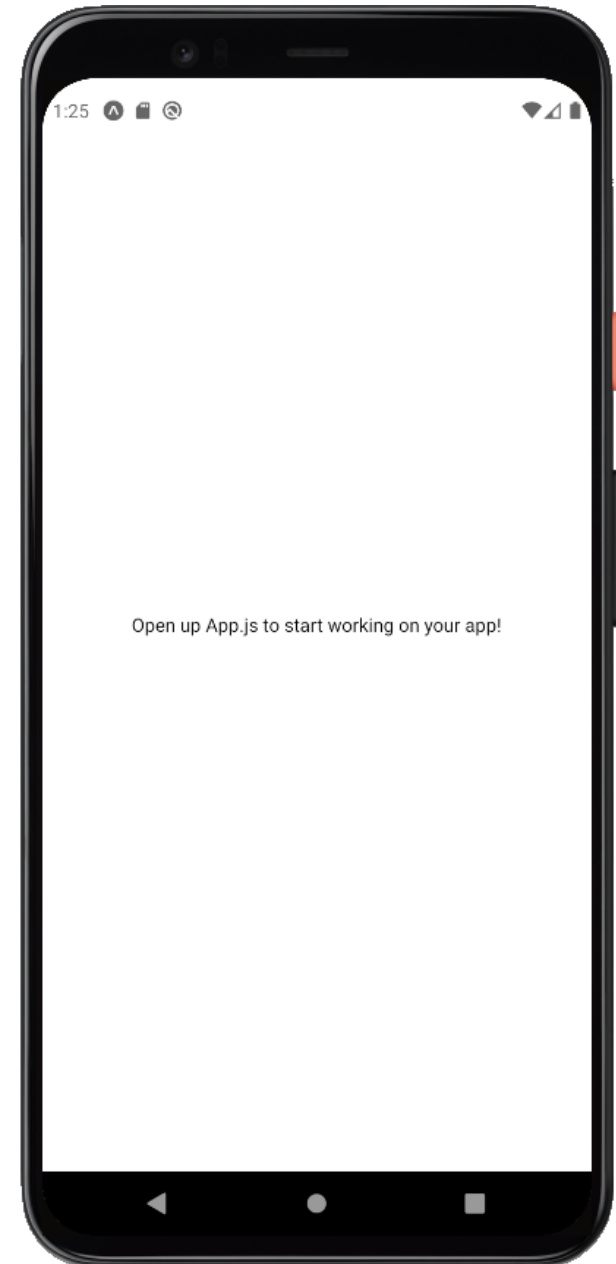
**`npx cross-env NODE_OPTIONS=--openssl-legacy-provider
expo start --web`**



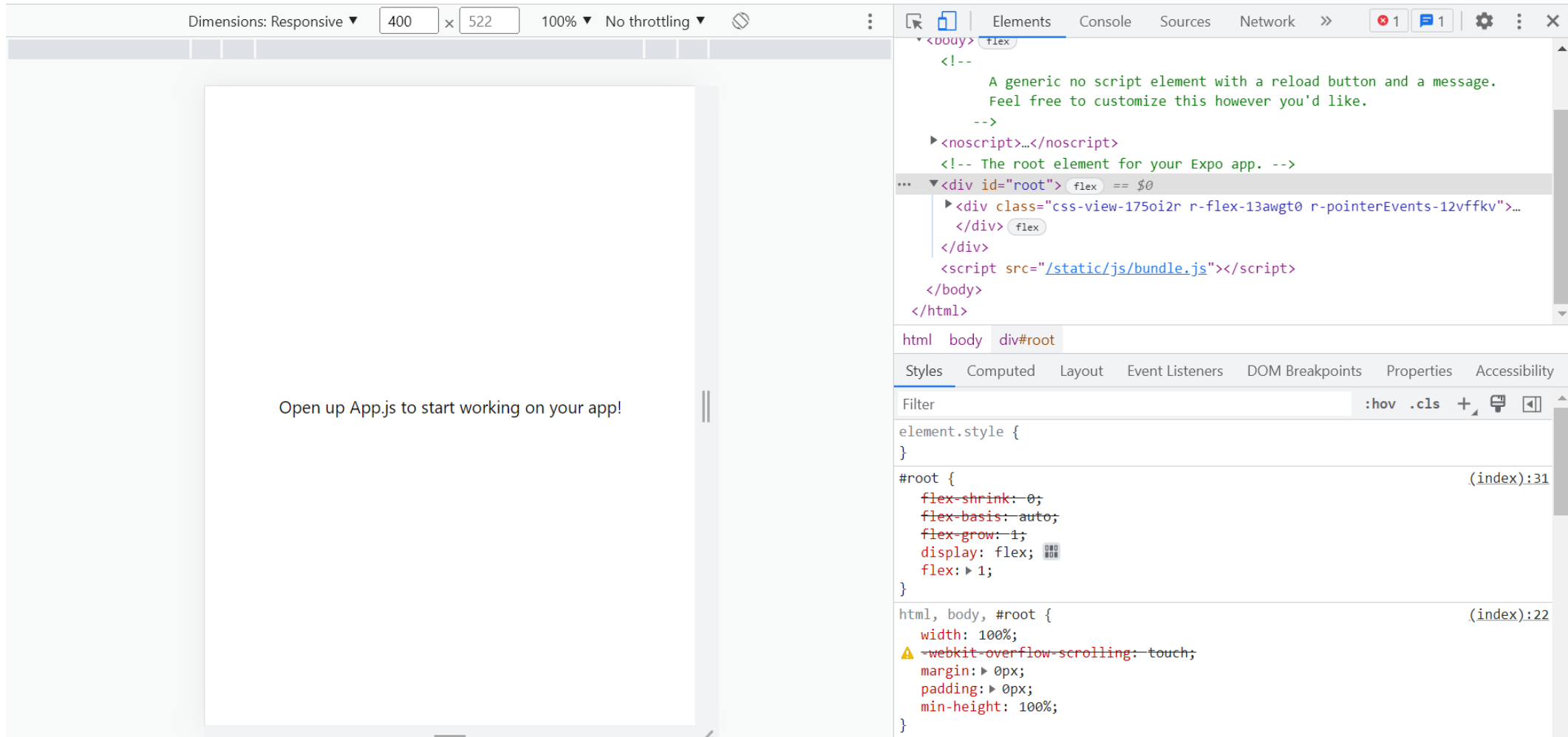
```
Administrator: C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19044.2546]
(c) Microsoft Corporation. All rights reserved.

D:\Lập trình Mobile\Demo\Demo3>npx cross-env NODE_OPTIONS=--openssl-legacy-provider expo start --web
```

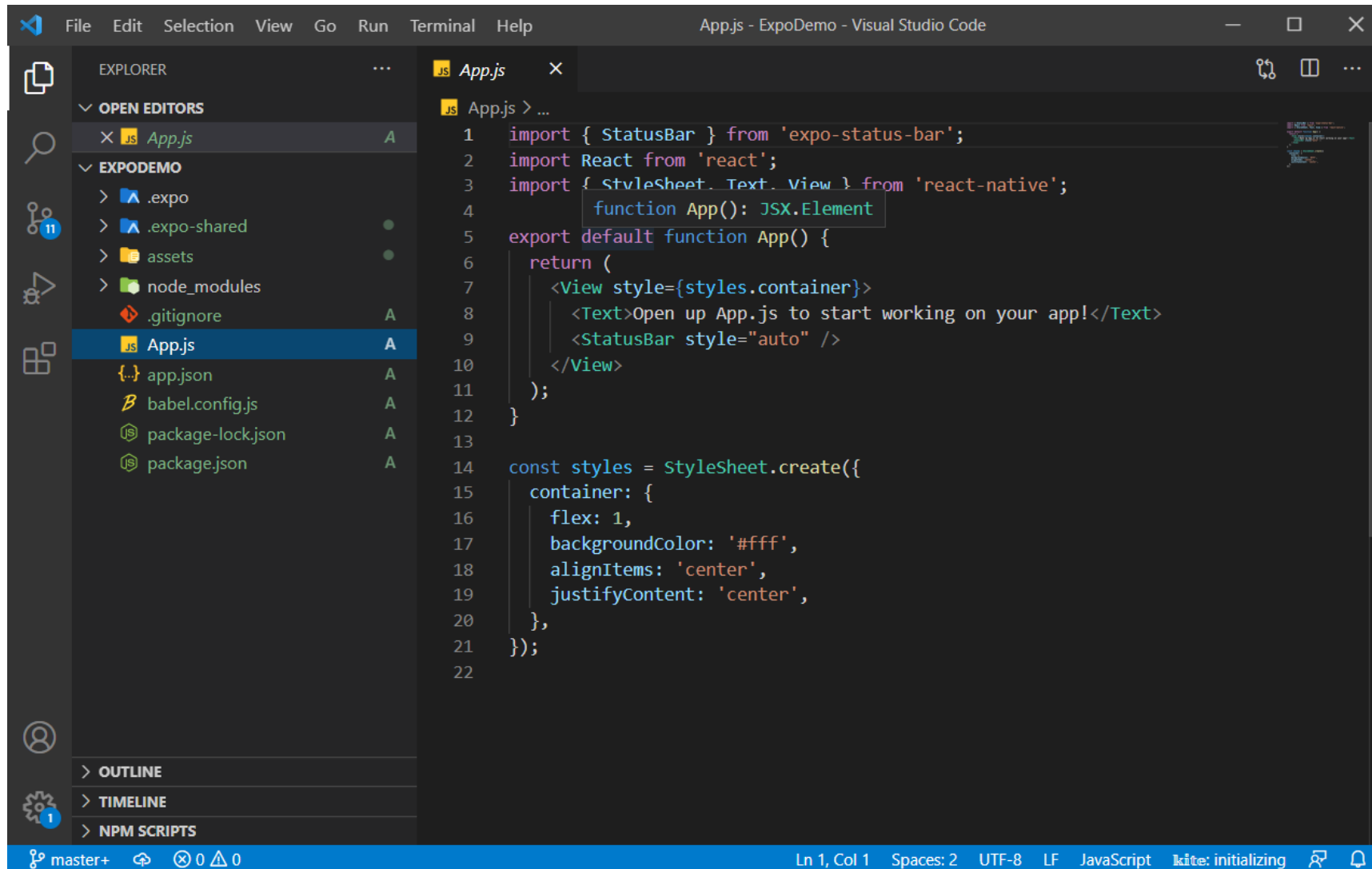
- Kết quả khi chạy bằng máy ảo Android



- Kết quả khi chạy bằng Web



- Thư mục Project khi mở bằng Visual Studio Code:



The screenshot shows the Visual Studio Code interface with the 'App.js' file open. The Explorer sidebar on the left shows the project structure for 'ExpoDemo', including folders like '.expo', '.expo-shared', 'assets', and 'node_modules', and files like '.gitignore', 'App.js', 'app.json', 'babel.config.js', 'package-lock.json', and 'package.json'. The 'App.js' file is selected and its content is displayed in the main editor. The code defines a function 'App()' that returns a React component structure using 'View', 'Text', and 'StatusBar' from 'react-native'. It also defines a 'styles' object for the 'View' component.

```
1 import { StatusBar } from 'expo-status-bar';
2 import React from 'react';
3 import { StyleSheet, Text, View } from 'react-native';
4 function App(): JSX.Element
5 export default function App() {
6   return (
7     <View style={styles.container}>
8       <Text>Open up App.js to start working on your app!</Text>
9       <StatusBar style="auto" />
10    </View>
11  );
12 }
13
14 const styles = StyleSheet.create({
15   container: {
16     flex: 1,
17     backgroundColor: '#fff',
18     alignItems: 'center',
19     justifyContent: 'center',
20   },
21 });
22
```

Cấu trúc Project



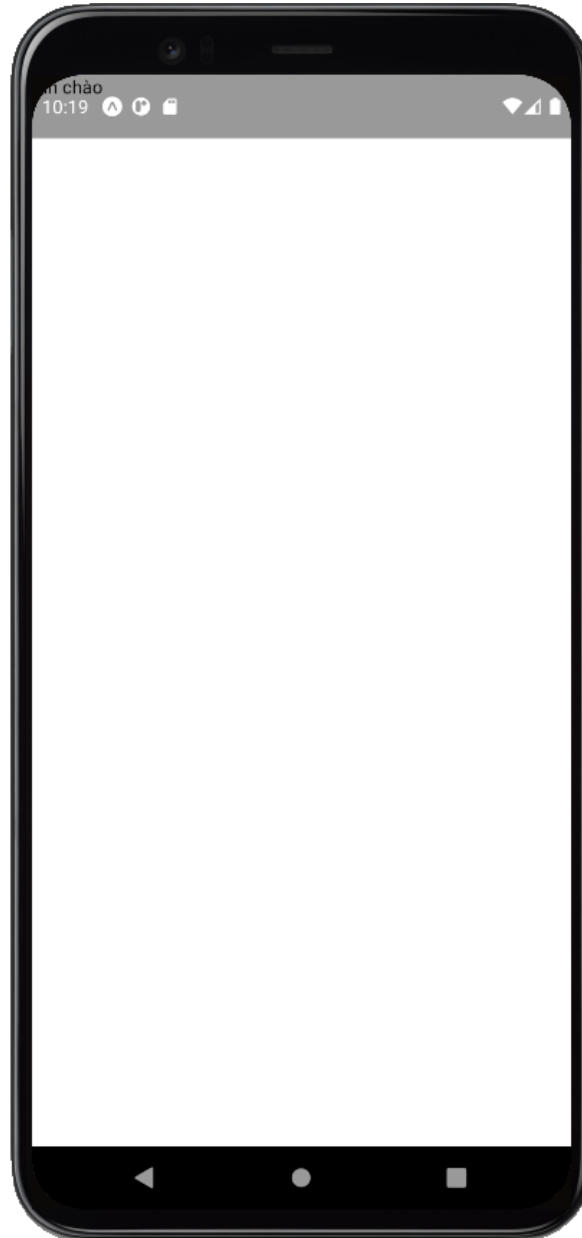
```
1  import { StatusBar } from 'expo-status-bar';
2  import React from 'react';
3  import { StyleSheet, Text, View } from 'react-native';
4
5  export default function App() {
6    return (
7      <View style={styles.container}>
8        <Text>Open up App.js to start working on your app!</Text>
9        <StatusBar style="auto" />
10     </View>
11   );
12 }
13
14 const styles = StyleSheet.create({
15   container: {
16     flex: 1,
17     backgroundColor: 'white',
18     alignItems: 'center',
19     justifyContent: 'center',
20   },
21 });
```

1. Phần import các thư viện cần dùng
2. Class, hàm chính, nội dung trong return(); sẽ là giao diện của App
 - <View>**: Tương ứng với **<div>** trong web
 - <Text>**: Tương ứng với **** trong web
3. Phần định dạng
4. Export

- Ví dụ:

```
export default function App() {  
  return (  
    <View>  
      <Text>Xin chào</Text>  
    </View>  
  );  
}
```

- Kết quả:



- Cho đoạn mã nguồn như sau:

```
export default function App() {  
  return (  
    <View>  
      <Text>Xin chào</Text>  
    </View>  
  );  
}
```

- Để thay đổi định dạng cho các thẻ, dùng thuộc tính **style**
- Cú pháp: `style={{thuocTinh: giatri}}`
- Ví dụ: `style={{marginTop: 50}}`

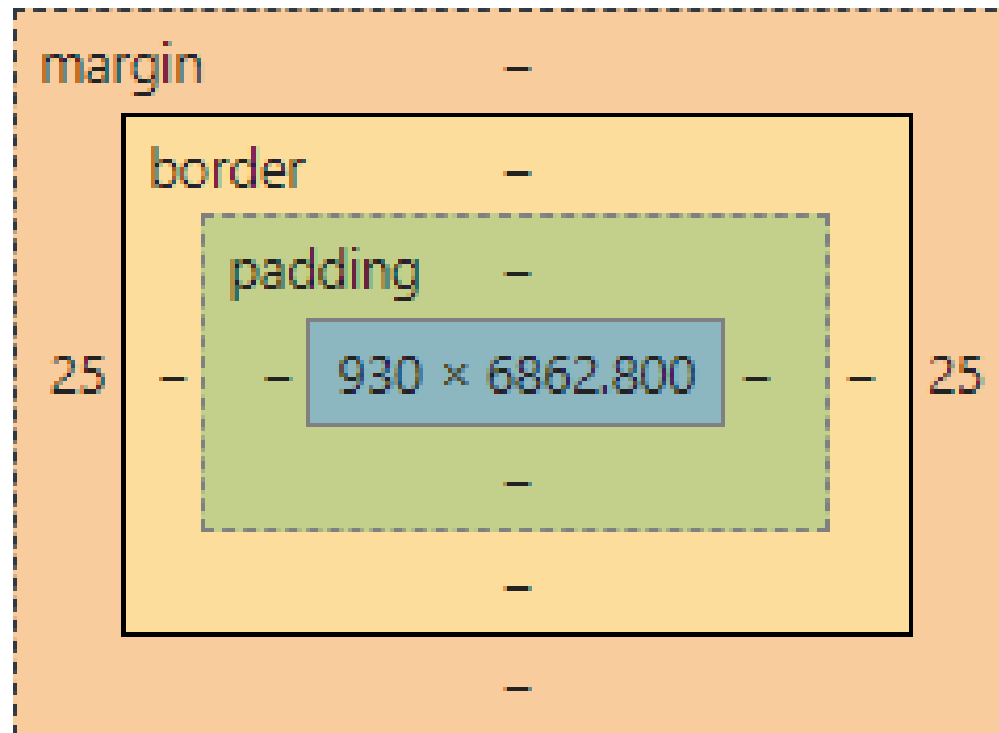
- Các **thuộc tính trong style tương tự** như các **thuộc tính** khi thiết kế trang Web, tuy nhiên cách viết sẽ dùng **cú pháp lạc đà**
- Ví dụ:

background-**color** → background**C**olor

font-**size** → font**S**ize

text-**align** → text**A**lign

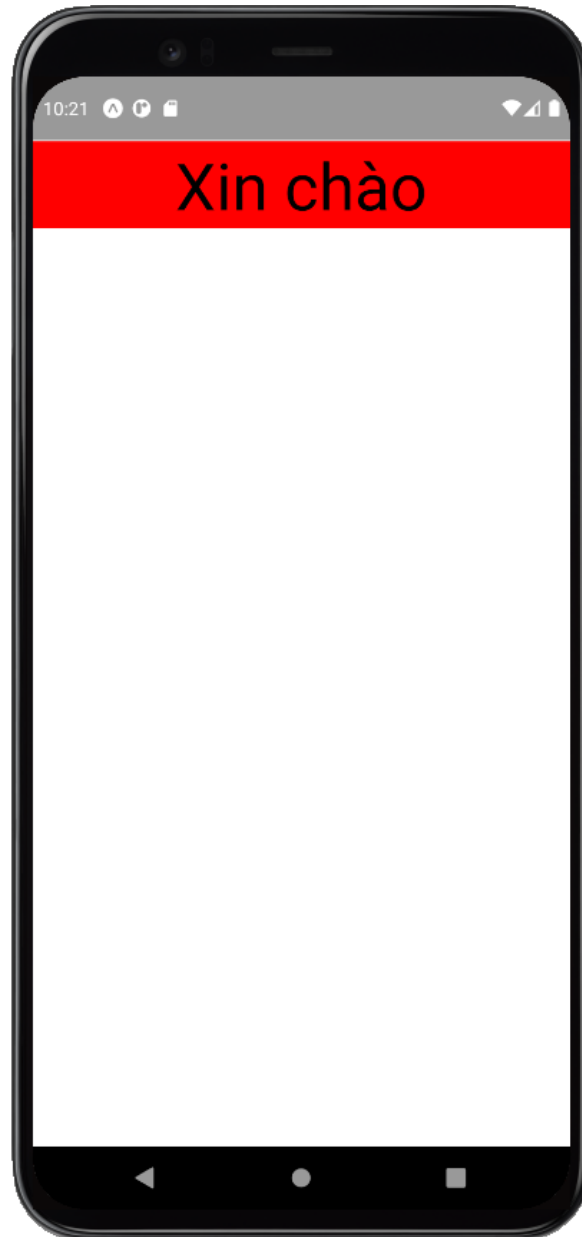
- Nhắc lại một số thuộc tính để định dạng



- Ví dụ:

```
export default function App() {  
  return (  
    <View style={{backgroundColor: 'red', marginTop: 50}}>  
      <Text style={{fontSize: 50, textAlign: 'center'}}>Xin chào</Text>  
    </View>  
  );  
}
```

- Kết quả:



- Lưu ý:
 - View sẽ không hiện lên màn hình nếu không có kích thước
 - Chỉ return về 1 View duy nhất

- Cheat Sheet: <https://github.com/vhpoet/react-native-styling-cheat-sheet>



Bài tập thực hành

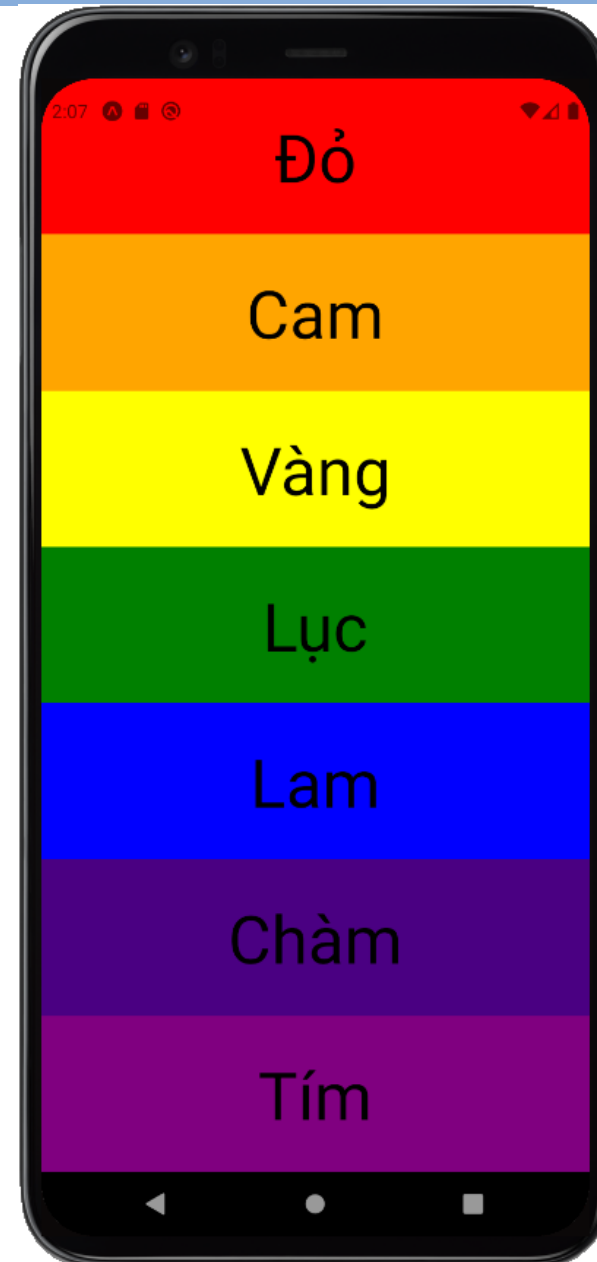


1. Cài đặt theo các bước đã được hướng dẫn
2. Tạo project mới và chụp lại kết quả khi chạy chương trình

3. Tạo giao diện như sau
(**xxx** là thông tin tương ứng
các thành viên trong nhóm)



4. Tạo giao diện như sau



- **Chụp hình, mô tả** quá trình làm và kết quả đạt được vào trong file word
- Lưu và đặt tên file word theo dạng: **MSSV.doc** hoặc **MSSV.docx**
- Nộp file word lên hệ thống **Canvas** theo tuần