

Chapter 1

INTRODUCTION TO MOBILE APPLICATION AND REACT
NATIVE

Mobile Application Development

- ❖ กระบวนการพัฒนาซอฟต์แวร์สำหรับอุปกรณ์เคลื่อนที่ เช่น โทรศัพท์เคลื่อนที่ แท็บเล็ต เป็นต้น
- ❖ ปกติแล้ว จะทำการพัฒนาแอปพลิเคชัน สำหรับระบบปฏิบัติการ 3 ระบบหลัก
 - ❖ Android
 - ❖ iOS
 - ❖ Windows

Ref.: <https://www.ibm.com/topics/mobile-application-development>

Types of Mobile Application

❖ Native Apps

- ❖ พัฒนาด้วยภาษาเฉพาะสำหรับแต่ละระบบปฏิบัติการ
- ❖ สามารถทำงานกับแพลตฟอร์ม และใช้ API และ Library ได้โดยตรง
- ❖ ข้อดี
 - ทำงานได้เร็ว มีประสิทธิภาพดี
 - เรียกใช้ API หรือฟีเจอร์ต่างๆ ได้ง่าย
- ❖ ข้อเสีย
 - กรณีที่ต้องพัฒนาแอปพลิเคชันสำหรับหลายแพลตฟอร์ม อาจทำให้ใช้เวลาพัฒนานาน

Ref.: <https://designli.co/blog/11-best-mobile-app-programming-languages-pros-and-cons/>

Ref.: <https://www.geeksforgeeks.org/introduction-of-mobile-applications/>

Types of Mobile Application

❖ Mobile Web Apps

- ❖ เป็นเว็บไซต์ที่สามารถปรับการแสดงผลให้เหมาะสมกับอุปกรณ์ได้
- ❖ Responsive Web Design, Progressive Web App
- ❖ ข้อดี
 - ใช้เวลาในการพัฒนาน้อย
 - ไม่จำเป็นต้องติดตั้งโปรแกรมเพิ่มเติม
- ❖ ข้อเสีย
 - ไม่สามารถเข้าถึงฟีเจอร์ต่างๆ ของระบบปฏิบัติการได้
 - มีข้อจำกัดในการพัฒนา UI, มี Community ที่ค่อนข้างจำกัด

Types of Mobile Application

❖ Cross-Platform/Hybrid Apps

- ❖ พัฒนาแอปพลิเคชันครั้งเดียว แต่สามารถ deploy ใช้บนแพลตฟอร์มที่แตกต่างกันได้
- ❖ จำเป็นต้องมีการใช้ library ที่ช่วยให้รันบนแพลตฟอร์มที่ต่างกันได้
- ❖ ข้อดี
 - พัฒนาแอปพลิเคชันสำหรับหลายแพลตฟอร์มได้อย่างรวดเร็ว
 - ค่าใช้จ่ายในการดูแลรักษา หรือแก้ไขปัญหาของแอปพลิเคชันต่ำกว่า
- ❖ ข้อเสีย
 - ประสิทธิภาพการทำงานช้ากว่า
 - มีข้อจำกัดในการใช้ฟีเจอร์เฉพาะของแต่ละระบบปฏิบัติการ

SKILLS BREAKDOWN

FULL STACK DEVELOPER



@code.know
www.codeknow.in

React

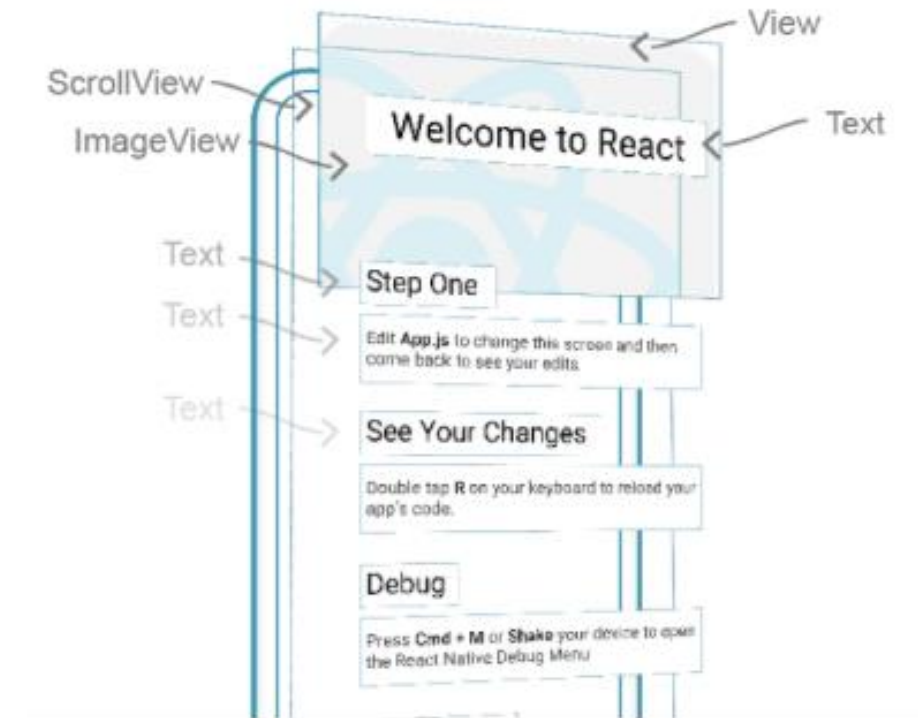
- ❖ **React** is a JavaScript library for building user interfaces.
- ❖ React Native runs on [React](#), a popular open-source library for building user interfaces with JavaScript.
- ❖ The core concepts behind React:
 - ❖ components
 - ❖ JSX
 - ❖ props
 - ❖ state

React and React Native

- ❖ React คือ JavaScript Library ที่ถูกสร้างโดย Facebook โดยสร้างมาจากพื้นฐานแนวความคิดแบบ MVC (Model View Controller) เหมาะแก่การใช้พัฒนาเว็บไซต์ฝั่ง Front – end
- ❖ React Native คือ เครื่องมือสำหรับพัฒนา Mobile Application แบบ cross platform technology ที่สามารถแสดงผลได้ทั้ง iOS และ Android โดยใช้ JavaScript สำหรับการพัฒนา
- ❖ สรุปคือ React ใช้สำหรับเว็บไซต์ ส่วน React Native สำหรับพัฒนาแอปพลิเคชันบนอุปกรณ์เคลื่อนที่

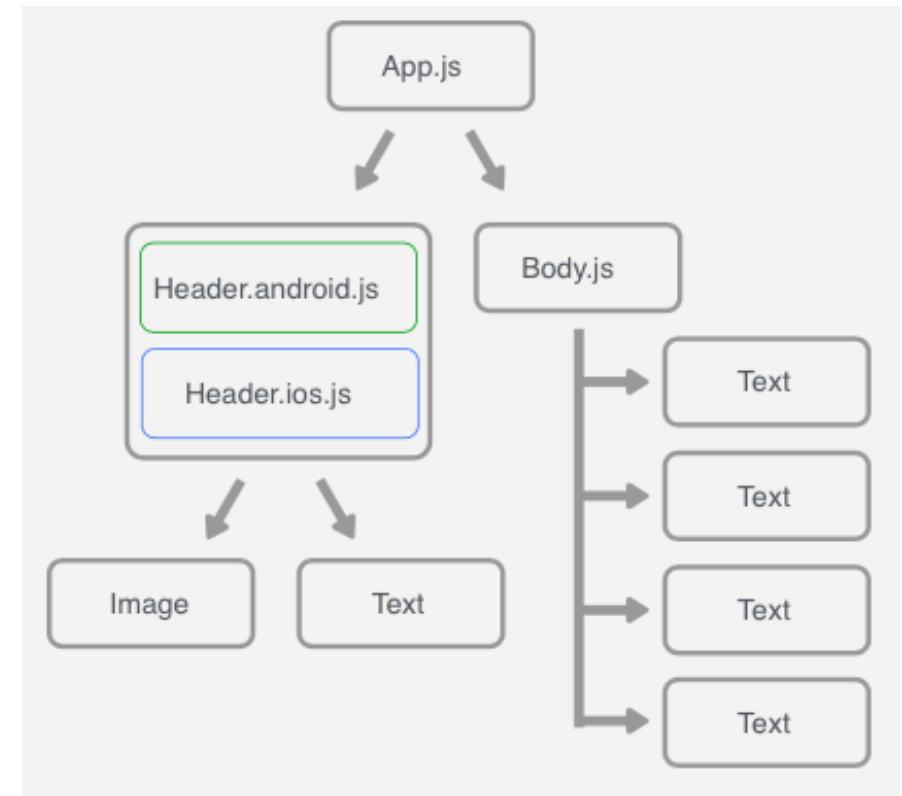
React Native

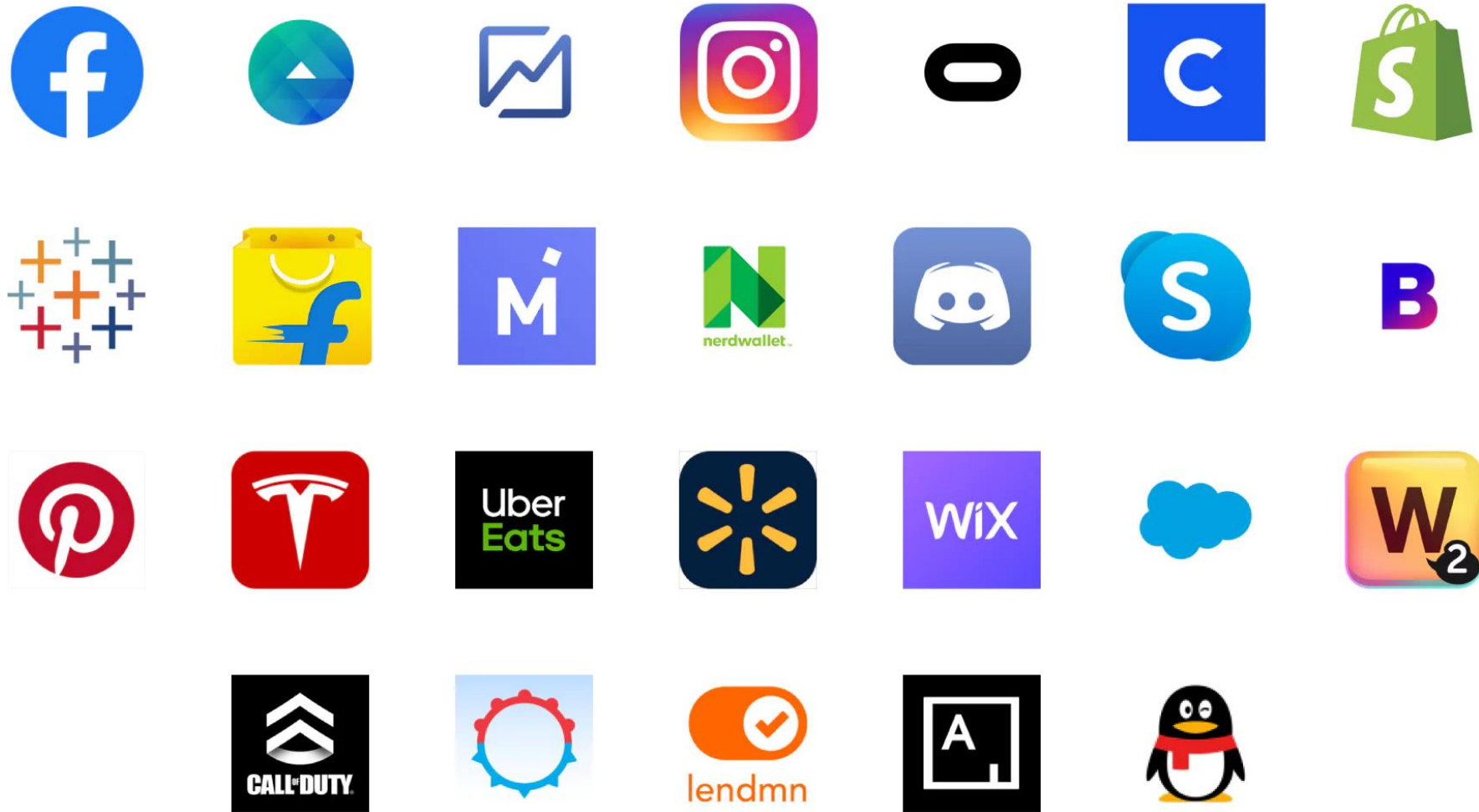
- ❖ เป็น JavaScript framework สำหรับเขียนและเรนเดอร์แอปพลิเคชันบนอุปกรณ์เคลื่อนที่สำหรับทั้ง iOS และ Android
- ❖ ทำให้นักพัฒนาเว็บสามารถสร้างโมบายแอปพลิเคชันที่มีหน้าตาเหมือน Native app ได้
- ❖ พัฒนาแอปพลิเคชันด้วย JavaScript และ JSX
- ❖ เตรียม Core component สำหรับใช้ในการสร้าง UI ที่สามารถ map เข้ากับ native UI สำหรับระบบปฏิบัติการนั้นๆ ได้



React Native

- ❖ React Native “bridge” จะทำการเรียก API สำหรับเรนเดอร์คอมโพเนนต์เนทีฟ ด้วย Objective-C สำหรับระบบ iOS หรือด้วย Java สำหรับระบบ Android (ใช้ UI component สำหรับอุปกรณ์เคลื่อนที่จริงๆ ไม่ใช่เป็น web component)
- ❖ React Native ยังเตรียมอินเทอร์เฟซสำหรับ platform APIs อีกด้วย นั่นคือ แอปพลิเคชันสามารถเข้าถึงฟีเจอร์ของแพลตฟอร์ม เช่น การใช้กล้อง หรือตำแหน่งของอุปกรณ์ได้อีกด้วย



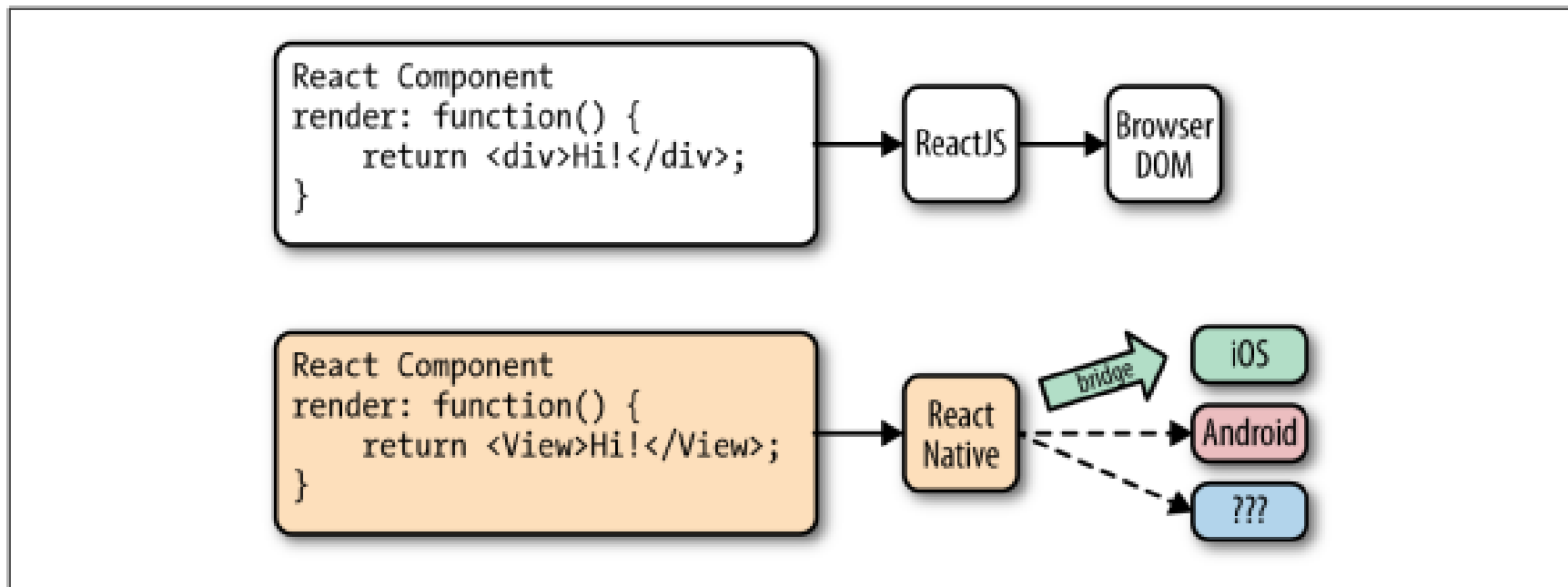


Apps using React Native

Ref.: <https://nix-united.com/blog/why-use-react-native-for-mobile-app-development/>

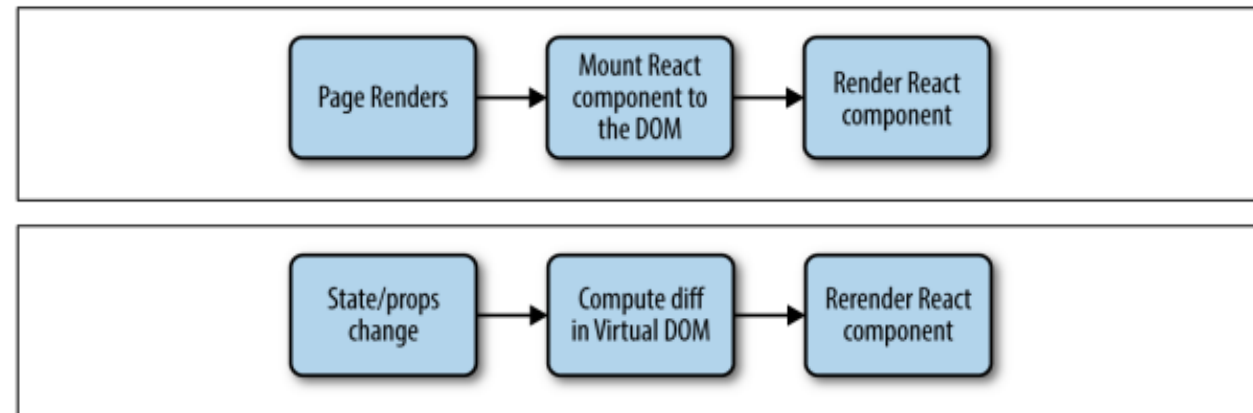
How Does React Native Work?

- ❖ การทำงานของ React Native จะคล้ายกับ React แต่แทนที่จะเรนเดอร์ DOM ในเว็บเบราว์เซอร์ แต่ React Native จะทำการเรียก Objective-C APIs เพื่อเรนเดอร์คอมโพเนนต์ของ iOS, หรือเรียก Java APIs เพื่อเรนเดอร์คอมโพเนนต์ของ Android



Rendering Lifecycle - React

- ❖ เมื่อ React รันบนเบราว์เซอร์ กระบวนการเรนเดอร์ (render lifecycle) จะเริ่มจากการ mount คอมโพเนนต์ของ React เข้ากับ DOM ก่อน หลังจากนั้น เมื่อเกิดการเปลี่ยนแปลง React จะทำการเรนเดอร์ คอมโพเนนต์ใหม่เท่าที่จำเป็นเท่านั้น
- ❖ ขั้นตอนการเรนเดอร์ ผู้พัฒนาจะรีเทิร์น HTML Markup ผ่านเมธอด render()



Rendering Lifecycle – React Native

- ❖ สำหรับ React Native, กระบวนการเรนเดอร์ (render lifecycle) เป็นเช่นเดียวกับกระบวนการใน React แต่ขั้นตอนการเรนเดอร์จะมีความแตกต่างเล็กน้อย เนื่องจาก React Native มีการทำงานขึ้นอยู่กับ bridge
- ❖ โดยที่ bridge จะทำการแปลการเรียก JavaScript แล้วทำการเรียก API และ คอมโพเนนต์ UI ที่สอดคล้องตามแพลตฟอร์มที่แอปพลิเคชันรันอยู่
- ❖ เนื่องจาก React Native ไม่ได้รันบนเธรดหลักของ UI ทำให้ React Native สามารถทำงานในรูปแบบ asynchronous ได้

Components in React Native

- ❖ When writing in React for the web, you render normal HTML elements (<div>, <p>, , <a>, etc.).
- ❖ With React Native, all of these elements are replaced by platform-specific React components

React	React Native
<code><div></code>	<code><View></code>
<code></code>	<code><Text></code>
<code>, </code>	<code><FlatList></code> , child items
<code></code>	<code><Image></code>

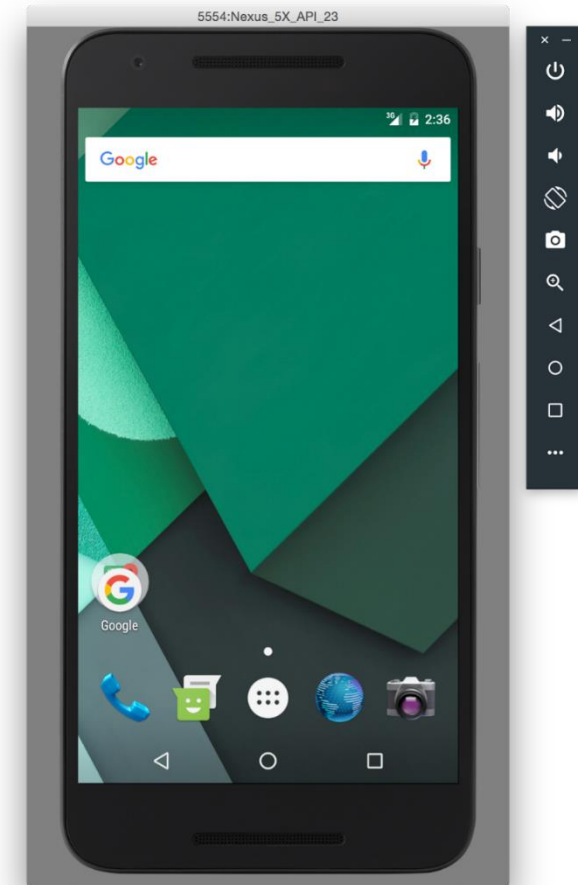
การติดตั้งและใช้งาน React Native

- ❖ ติดตั้ง Node.js เพื่อให้สามารถเรียกใช้ npx, npm ใน Node.js ได้
- ❖ ติดตั้ง Visual Studio Code
- ❖ ติดตั้ง Extension ต่างๆ ใน Visual Studio Code
- ❖ สร้างแอปพลิเคชันอย่างง่าย
 - ❖ `npx create-expo-app <project-name>`
- ❖ เข้าไปยังโฟลเดอร์ของโปรเจกต์ที่สร้าง แล้วทำการทดสอบการทำงานของแอปพลิเคชัน ผ่าน expo
 - ❖ `npx expo start`
 - ❖ เลือกหมวดที่ต้องการรันแอปพลิเคชัน เช่น กด a เพื่อรันผ่าน Android Emulator
- ❖ ทดลองรันแอปพลิเคชันผ่าน Emulator (Android Studio), web และ Expo Go (Client)



Android Emulator

❖ **Android Emulator** is used to run, debug and test the android application. If you don't have the real device, it can be the best way to run, debug and test the application.



The disadvantage of using emulator

- ❖ No support for placing or receiving actual phone calls. You can simulate phone calls (placed and received) through the emulator console, however.
- ❖ No support for USB connections
- ❖ No support for camera/video capture (input).
- ❖ No support for device-attached headphones
- ❖ No support for determining connected state
- ❖ No support for determining battery charge level and AC charging state
- ❖ No support for determining SD card insert/eject
- ❖ No support for Bluetooth

Virtual device

- ❖ An Android Virtual Device (AVD) is a configuration that defines the characteristics of an Android phone, tablet, Wear OS, Android TV, or Automotive OS device that you want to simulate in the [Android Emulator](#). The AVD Manager is an interface you can launch from Android Studio that helps you create and manage AVDs.

To open the AVD Manager, do one of the following:

- Select **Tools > AVD Manager**.

- Click **AVD Manager**



- in the toolbar.

- ❖ An AVD contains a hardware profile, system image, storage area, skin, and other properties.

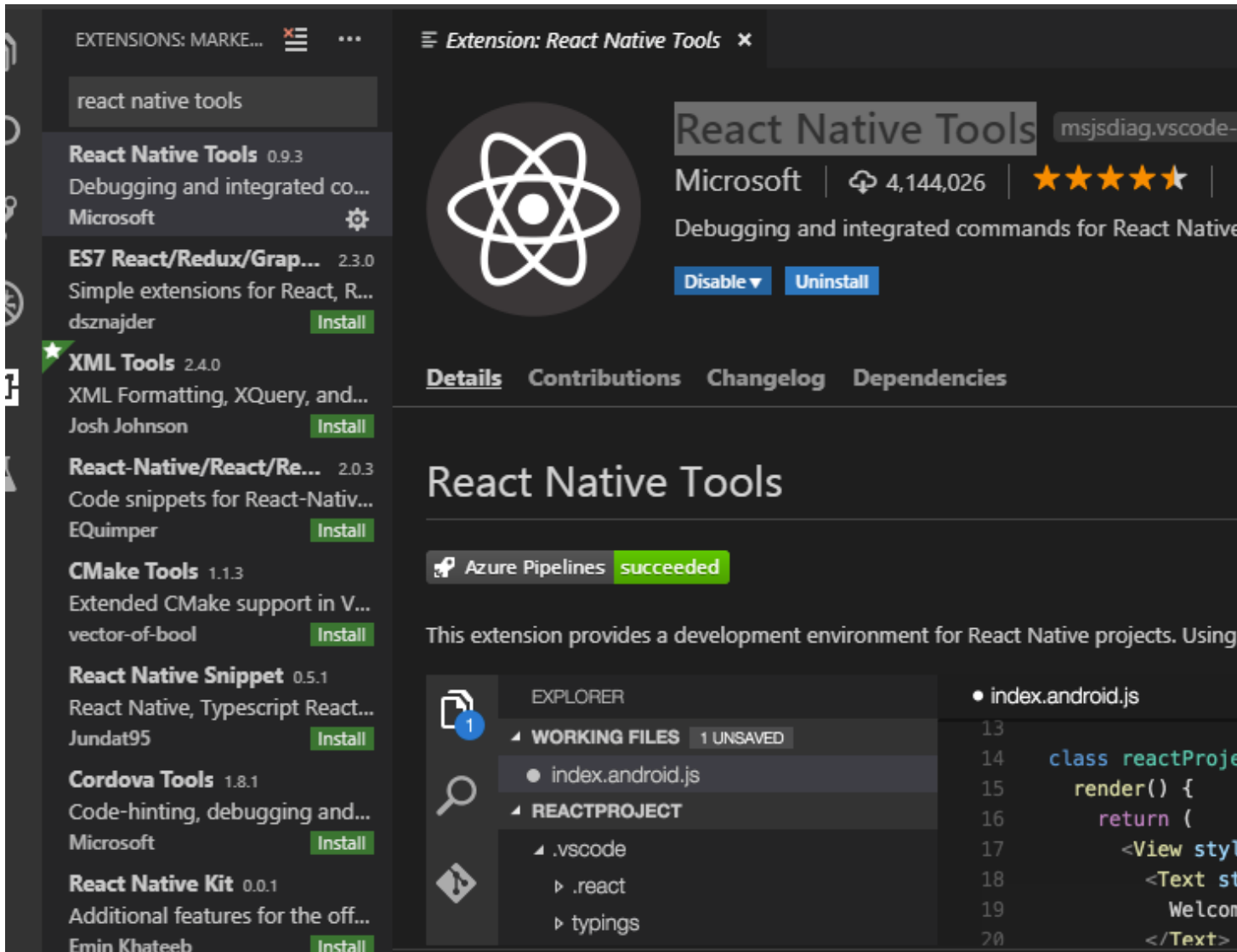


Your Virtual Devices

Android Studio

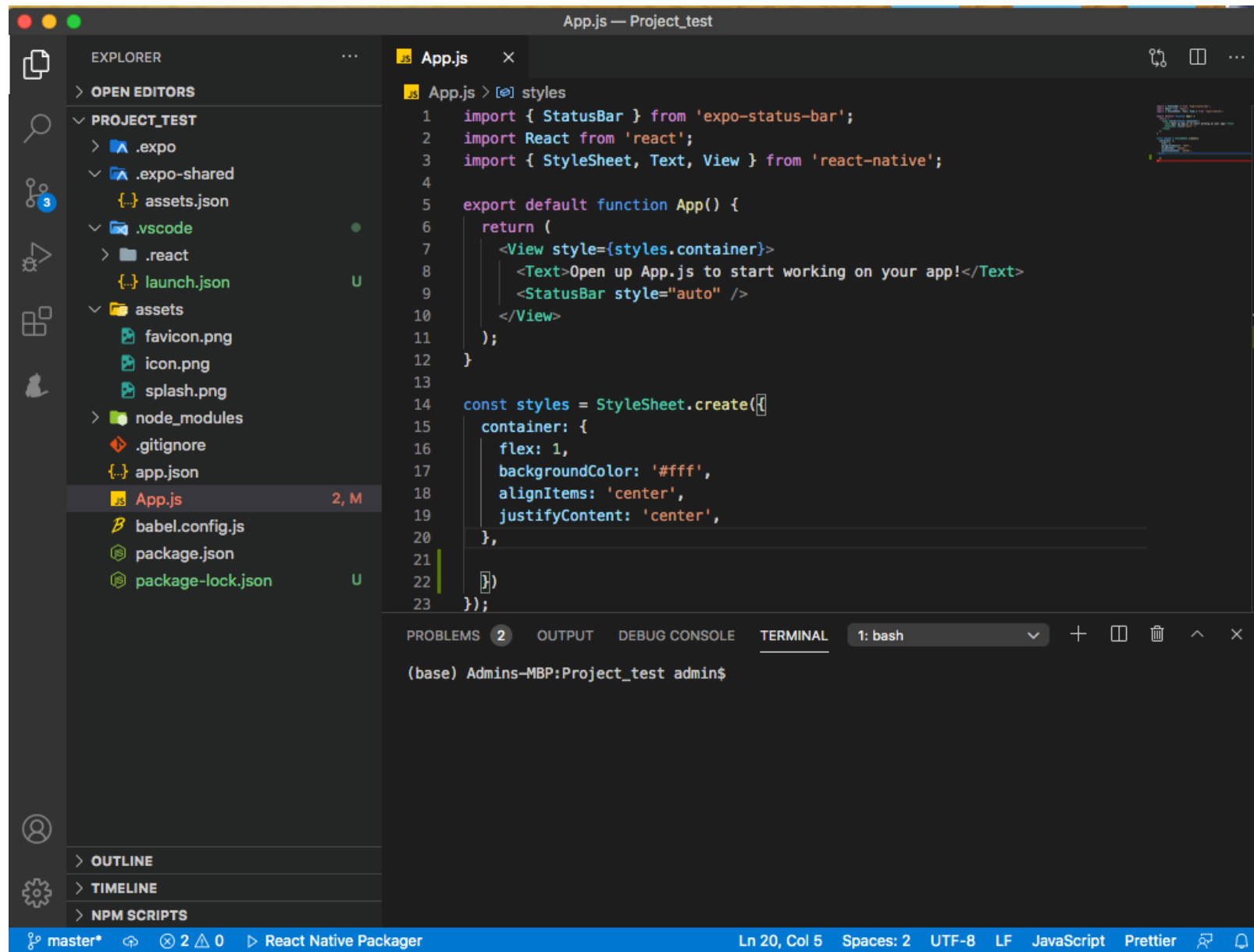
Type	Name ▲	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	10.1 WXGA (Tablet) ...		800 × 1280: mdpi	28	Android 9.0 (Google APIs)	x86	513 MB	  ▼
	Android TV (1080p) ...		1920 × 1080: xhdpi	28	Android 9.0 (Android TV)	x86	513 MB	  ▼
	Android Wear Round ...		320 × 320: hdpi	28	Android 9.0 (Wear OS)	x86	513 MB	  ▼
	Automotive (1024p la...		1024 × 768: mdpi	28	Android 9.0 (Automotive)	x86	513 MB	  ▼
	Pixel 3 API 28		1080 × 2160: 440dpi	28	Android 9.0 (Google Play)	x86	8.1 GB	  ▼

?
+ Create Virtual Device...

VSCODE - Extension

- ❖ React Native Tools
- ❖ React-Native /React/Redux....
- ❖ Prettier-Format
- ❖ Material icon theme
- ❖ Android Simulation Launcher
- ❖ etc.



App.js — Project_test

EXPLORER

OPEN EDITORS

PROJECT_TEST

- .expo
- .expo-shared
- assets.json
- .vscode
- .react
- launch.json
- assets
 - favicon.png
 - icon.png
 - splash.png
- node_modules
- .gitignore
- app.json
- App.js 2, M
- babel.config.js
- package.json
- package-lock.json

App.js > [?] styles

```

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: 'fff',
18     alignItems: 'center',
19     justifyContent: 'center',
20   },
21 });
22
23 
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL 1: bash

(base) Admins-MBP:Project_test admin\$

master* 2 0 React Native Packager Ln 20, Col 5 Spaces: 2 UTF-8 LF JavaScript Prettier

Lab 1

- ❖ ให้ศ.ทดลองติดตั้งโปรแกรมที่เกี่ยวข้องในการสร้างแอปพลิเคชันด้วย React Native ให้เรียบร้อย
- ❖ ให้ศ.ทดลองสร้างแอปพลิเคชันอย่างง่าย และทดลองรันแอปพลิเคชันด้วยรูปแบบต่างๆ
 - ❖ รันผ่าน Android Emulator
 - ❖ รันผ่าน Web (อาจต้องมีการ install package เกี่ยวกับ web เพิ่มเติม ก็ให้ install ตามคำชี้แจง)
 - ❖ รันผ่าน Expo Go บนโทรศัพท์มือถือ (กรณีที่เครื่องคอมพิวเตอร์สามารถเข้าใช้ wifi ได้ และกำหนดให้เครื่องคอมพิวเตอร์ที่พัฒนาโปรแกรม และโทรศัพท์ที่ใช้ทดลองรันโปรแกรม ใช้ wifi เดียวกัน)
- ❖ ให้ศ.ทดลองปรับเปลี่ยนโค้ดโปรแกรมใน App.js อย่างง่าย แล้วดูผลลัพธ์ที่ได้จากการรันโปรแกรม

การส่งงาน Lab

- ❖ ให้นศ.ส่งงานในห้อง Lab เท่านั้น
- ❖ กำหนดการส่ง Lab ในแต่ละสัปดาห์
 - ❖ นศ.ส่งงานในห้อง Lab ได้ภายในวันที่เรียน สัปดาห์นั้น (ได้คะแนนเต็ม)
 - ❖ นศ.สามารถส่งงานในสัปดาห์ถัดไปได้ 1 สัปดาห์ โดยไม่หักคะแนน
 - ❖ นศ.สามารถส่งงานล่าช้า ในสัปดาห์ที่ 2 ได้ (นับจากวันที่เรียนเนื้อหาวันนั้น) โดยจะคิดเป็นส่งงานช้า และหักคะแนน 50% ของชิ้นงานนั้น
- ❖ นศ.ต้องมีความซื่อสัตย์ และทำงานด้วยตัวเอง หากตรวจสอบพบว่า นศ.คัดลอกงานเพื่อนมาส่ง จะไม่ได้คะแนนงานชิ้นนั้น