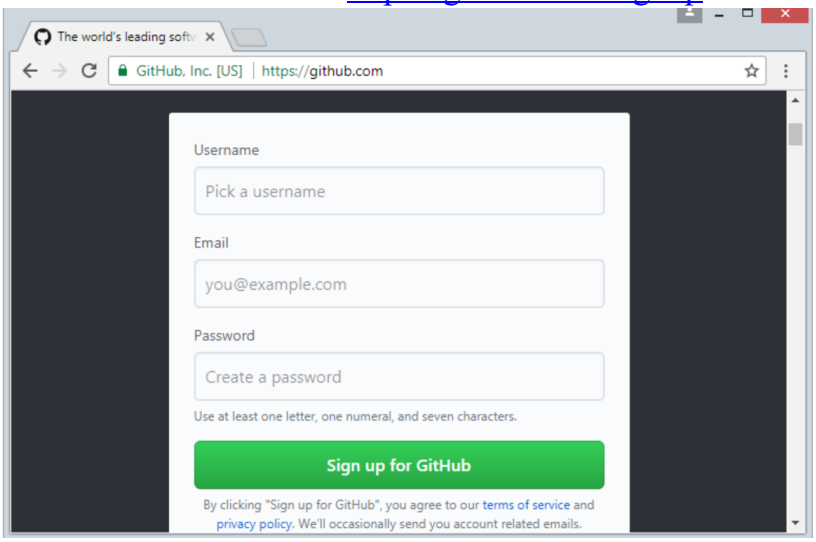
MOBILE PROGRAMMING

LAB 1

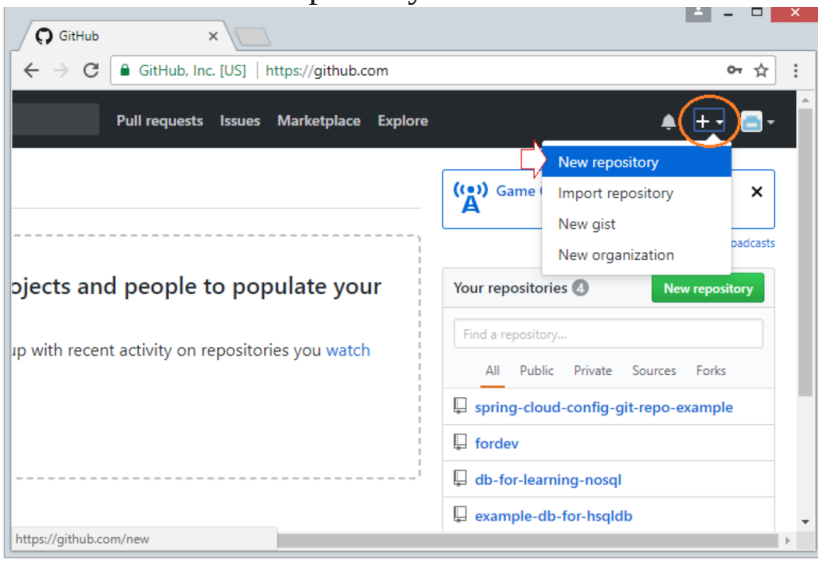
Contents

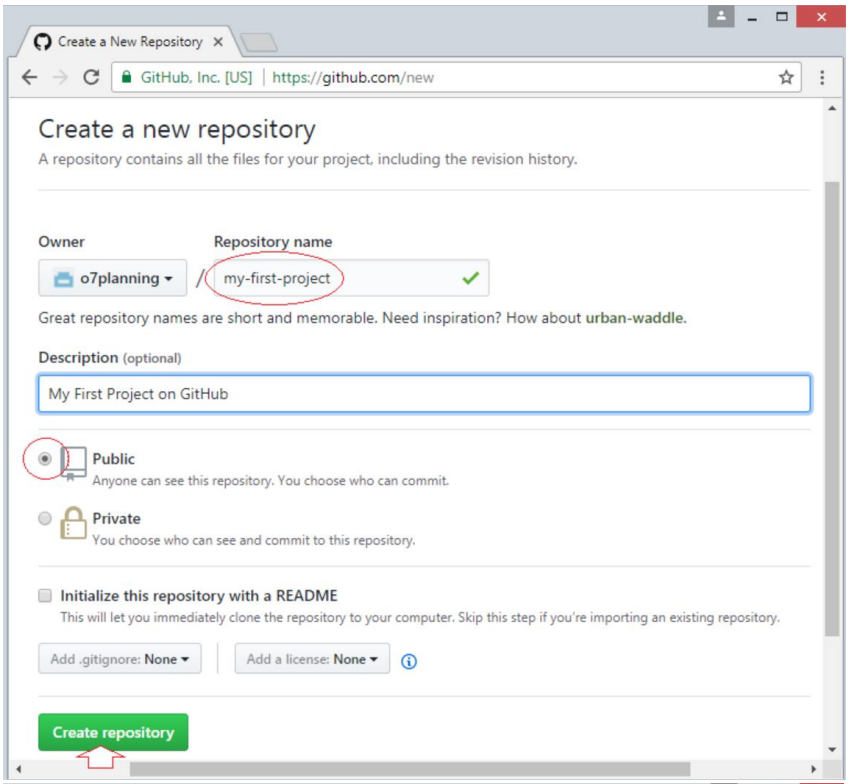
Question 1. Github

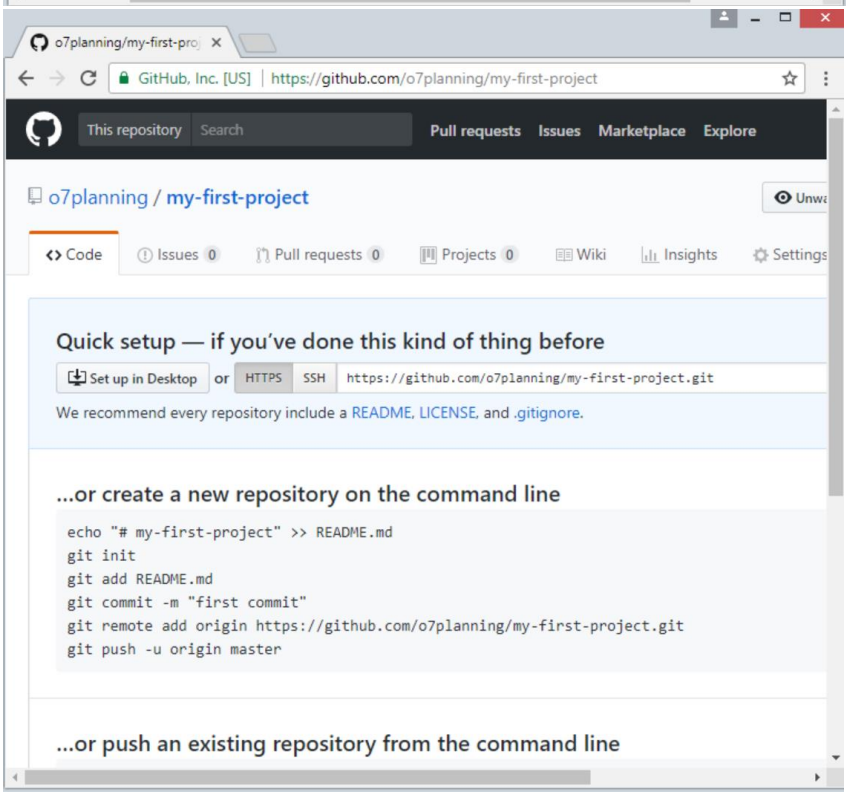
- Create a Github account: https://github.com/signup



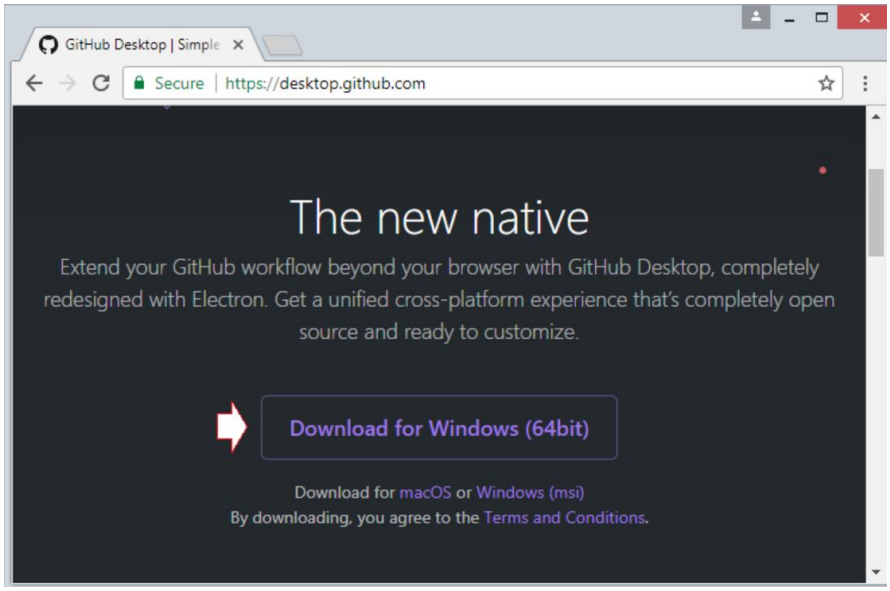
- Create GitHub Repository



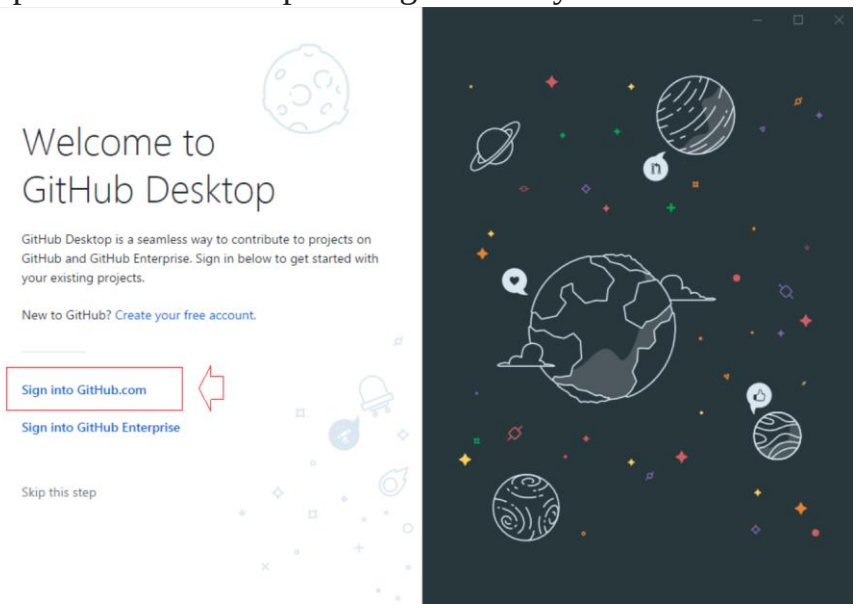




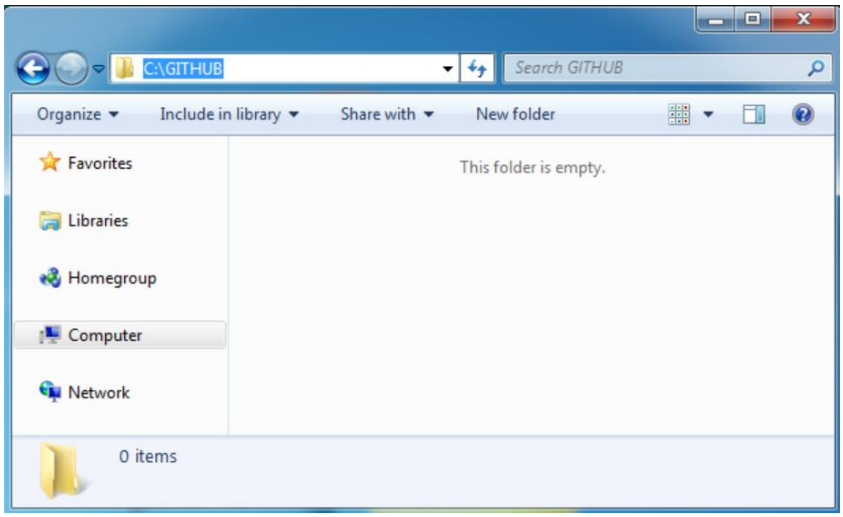
- Download Github Desktop and install: https://desktop.github.com/



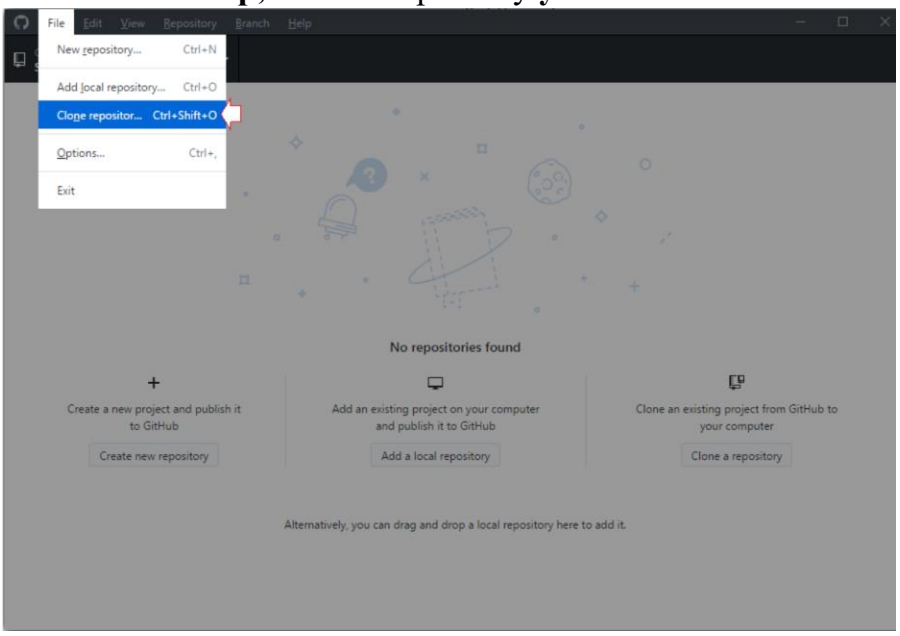
- Open Github Desktop and Sign in with your Github account

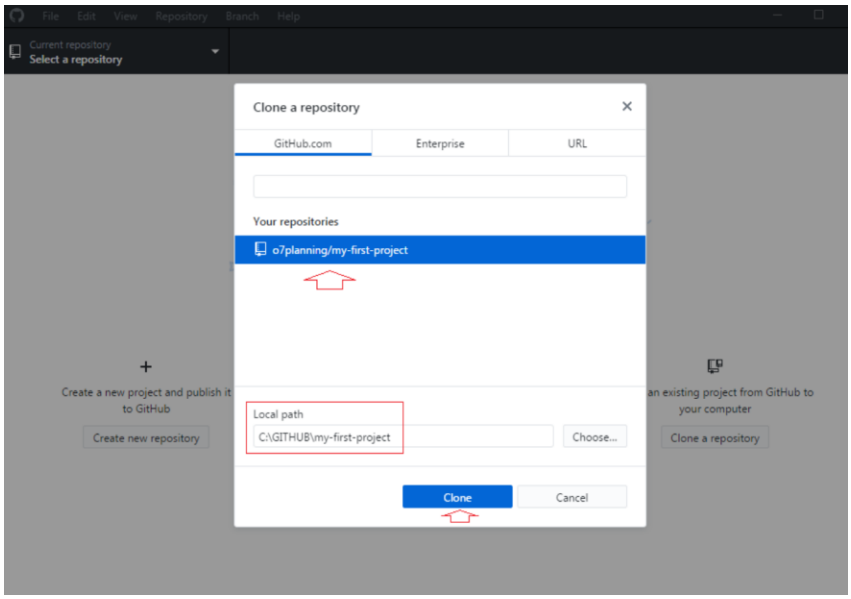


- Create an empty folder to store assignment/Project.

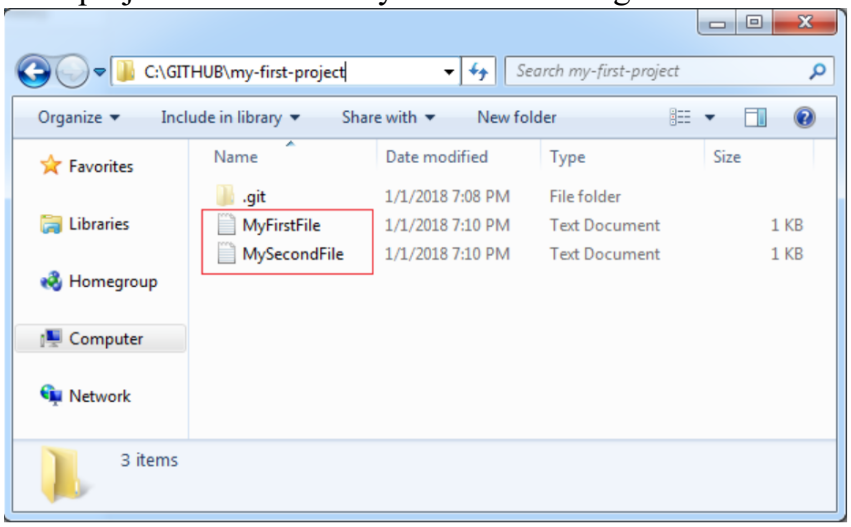


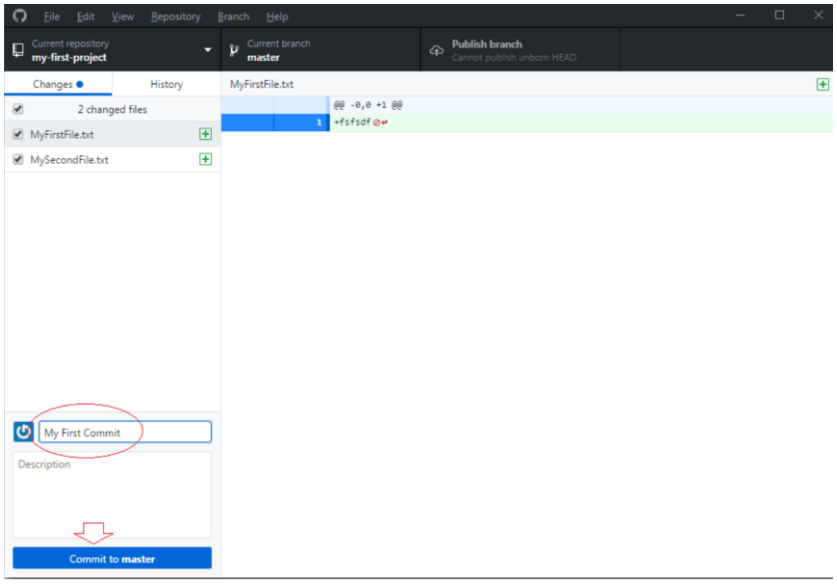
- On **GitHub Desktop, select a** repository **you created on** GitHub **to** clone





- Create a project to the directory and commit to github





Question 2. Development environment settings

1) Install an environment with Expo CLI

https://docs.expo.dev/get-started/installation/

or

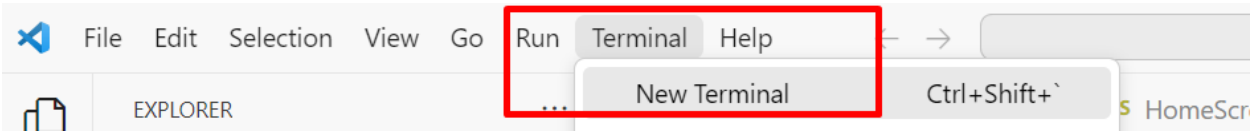
https://reactnative.dev/docs/environment-setup?guide=quickstart

2) Install an environment with React native CLI

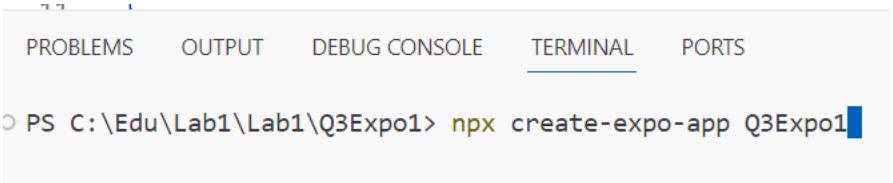
https://reactnative.dev/docs/environment-setup?guide=native

Question 3. Create Expo CLI apps

1. Open Visual code and New Terminal.



2. Create project:



3. Open project “Q3Expo1” on Visual code: “File/Open folder”
4. Open file ‘App.js’ and enter the following source code.

App.js

App.js > default

```
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 title="Press me" onPress={() => setPressCount(pressCount + 1)} />
11     </View>
12   );
13 };
14
```

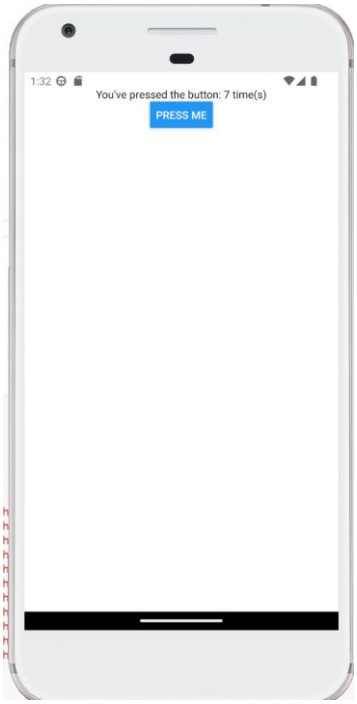
5. Run Expo CLI app.
   Open file “package.json’. View commands that run across different environments.

App.js package.json

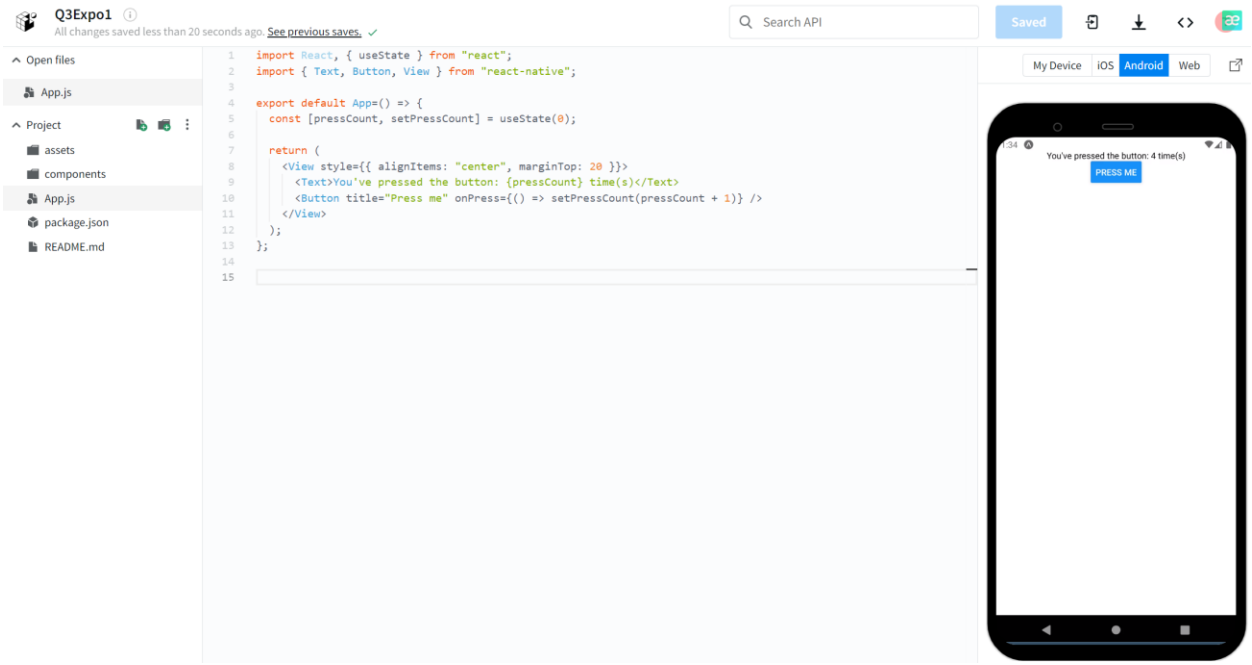
package.json > {} dependencies

```
1  {
2    "name": "q3expo1",
3    "version": "1.0.0",
4    "main": "node_modules/expo/AppEntry.js",
     Debug
5    "scripts": {
6      "start": "expo start",
7      "android": "expo start --android",
8      "ios": "expo start --ios",
9      "web": "expo start --web"
10   },
11   "dependencies": {
12     "expo": "~49.0.13",
13     "expo-status-bar": "~1.6.0",
14     "react": "18.2.0",
15     "react-native": "0.72.5"
16   },
17   "devDependencies": {
18     "@babel/core": "^7.20.0"
19   },
20   "private": true
21 }
22
```

```
> expo start --android
```

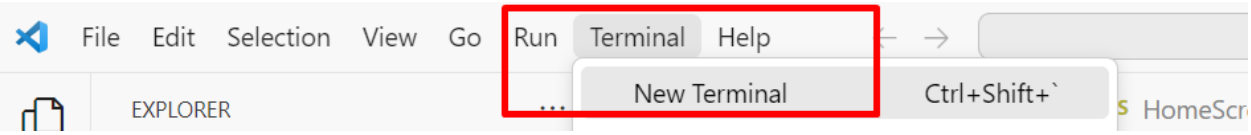


6. Another way to run the Expo app. Go to: https://snack.expo.dev/
   You can view results with Web/Android/iOS interface



Question 4. Create React native CLI apps

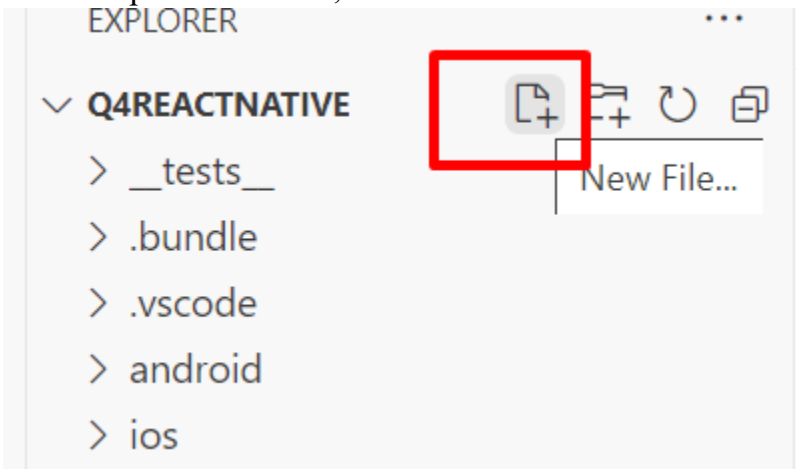
1. Open Visual code and New Terminal.



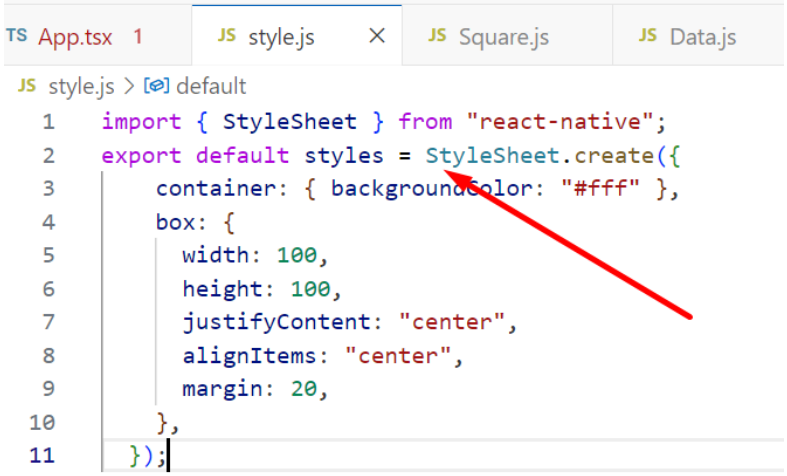
2. Create project with React native CLI:

```
PS C:\Edu\Lab1\Lab1> npx react-native init Q4ReactNative
```

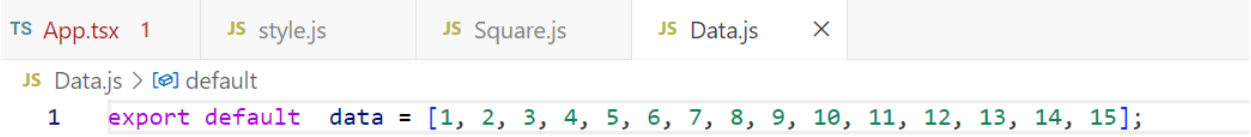
3. Open project “Q4reactNative” on Visual code: “File/Open folder”
4. In the Explorer window, create a new file with the name "style.js"



5. Add the following code to the "style.js" file



6. Create a new file named "Data.js" and enter the following code.



7. Create a new file named "Square.js" and enter the following code.

```
TS App.tsx 1   JS style.js   JS Square.js ×   JS Data.js
JS Square.js > ...
1  import React from "react";
2  import { View,Text, Alert,Button } from "react-native";
3  import styles from "./style";
4  function ClickOnTheSquare(value)
5  {
6    Alert.alert(value);
7  }
8  export default Square = ({ text}) => (
9    <View style={[styles.box, { backgroundColor: "#7ce0f9" }]}>
10     <Text>{text}</Text>
11     <Button title = 'CLick' onPress = {() => ClickOnTheSquare(text)}></Button>
12   </View>
13 );
```

Import from style.js

8. Open the "App.tsx" file and enter the following code.

```
TS App.tsx 1 ×   JS style.js   JS Square.js   JS Data.js
TS App.tsx > default
1  import React from "react";
2  import { View, Text, StyleSheet, ScrollView, Button, Alert } from "react-native";
3  import data from './Data'
4  import Square from './Square'
5  import styles from "./style";
6
7  export default App=()=> {
8    return (
9      <ScrollView style={styles.container}>
10       {data.map((item, index) => (
11         <Square key={item} text={`Square ${index + 1}`} />
12       ))}
13     </ScrollView>
14   );
15 };
16
```

Import from Data.js, Square.js, style.js

7. Run React native CLI app.
   Open file “package.json’. View commands that run across different environments.

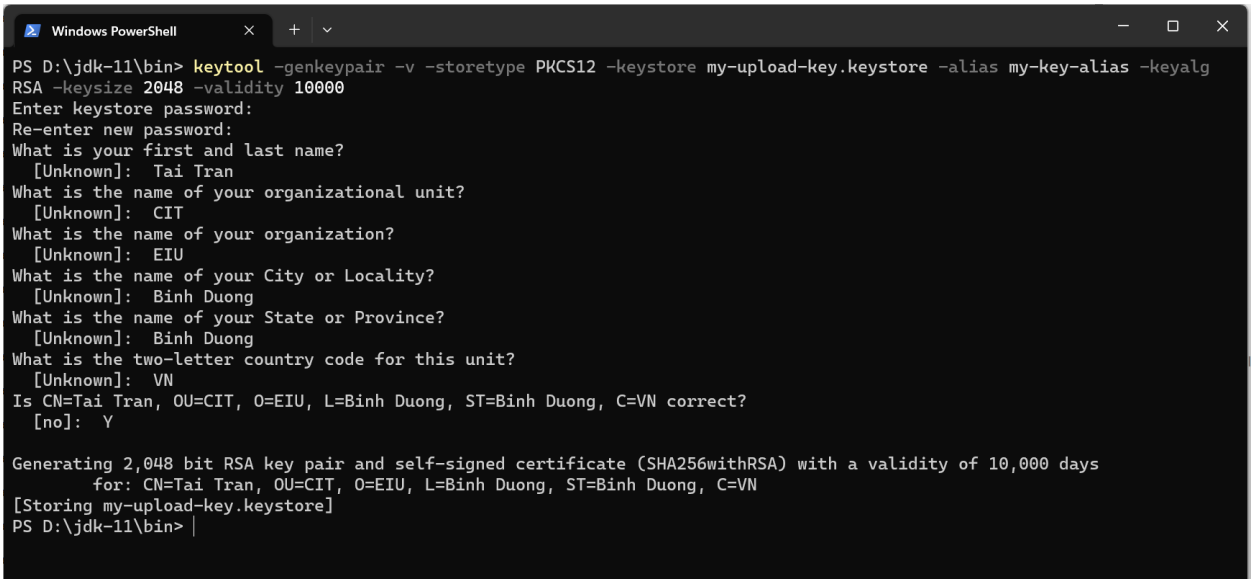
```
Debug
"scripts": {
  "android": "react-native run-android",
  "ios": "react-native run-ios",
  "lint": "eslint .",
  "start": "react-native start",
  "test": "jest"
},
PS C:\Edu\Lab1\Lab1\Q4ReactNative> react-native run-android
```

Question 5. Publishing

Use the Project **Q4reactNative** in Question 4. Create the "**Q5**" folder in the "**.\Lab1"** folder. Save 2 * files. AAB and *. APK to folder.

1. On Windows keytool must be run from C:\Program Files\Java\jdkx.x.x_x\bin, as administrator. Open powershell.

Run *keytool -genkeypair -v -storetype PKCS12 -keystore my-upload-key.keystore -alias my-key-alias -keyalg RSA -keysize 2048 -validity 10000*



2. Setting up Gradle variables

Place the **my-upload-key.keystore** file under the **android/app** directory in your project folder

Edit the file **~/.gradle/gradle.properties** or **android/gradle.properties**, and add the following (replace ***** with the correct keystore password, alias and key password)

```
MYAPP_UPLOAD_STORE_FILE=my-upload-key.keystore
MYAPP_UPLOAD_KEY_ALIAS=my-key-alias
MYAPP_UPLOAD_STORE_PASSWORD=*****
MYAPP_UPLOAD_KEY_PASSWORD=*****
```

3. Edit the file **android/app/build.gradle** in your project folder, and add the signing config

```
...
android {
    ...
    defaultConfig { ... }
    signingConfigs {
        release {
            if (project.hasProperty('MYAPP_UPLOAD_STORE_FILE')) {
                storeFile file(MYAPP_UPLOAD_STORE_FILE)
                storePassword MYAPP_UPLOAD_STORE_PASSWORD
                keyAlias MYAPP_UPLOAD_KEY_ALIAS
                keyPassword MYAPP_UPLOAD_KEY_PASSWORD
            }
        }
```

```
        }
        buildTypes {
            release {
                ...
                signingConfig signingConfigs.release
            }
        }
    }
    ...
```

4. Generating the release AAB (Support file deployed to google play)

npx react-native build-android --mode=release

Optional: divided by CPU type

```
android {
    splits {
        abi {
            reset()
            enable true
            universalApk false
            include "armeabi-v7a", "arm64-v8a", "x86", "x86_64"
        }
    }
}
```

Output file: ./android\app\build\outputs\bundle\release

5. Build React native App to **apk** file

./gradlew assembleRelease

Output file: ./android\app\build\outputs\apk\release

Question 6. Build apps with props and state.

1. Build an employee information entry screen with: full name, age, occupation specialized in training and an update button (display success message) (Component, Props).

2. Write a program to sum the first digit and the last digit of a number. (Component, State)
3. Write a program to find the minimum between three numbers. (Component & state)
4. Write a program that displays the Hailstone sequence: With some positive number (n > 0): (Component & State)
    a. If n is an even number, divide by 2.
    b. If n is an odd number, multiply it by 3 and add 1.
    c. Repeat two steps above until n equals 1.

Note: Use Github commit. Lab1 folder and send Github link to Moodle (Note: link Git to public)