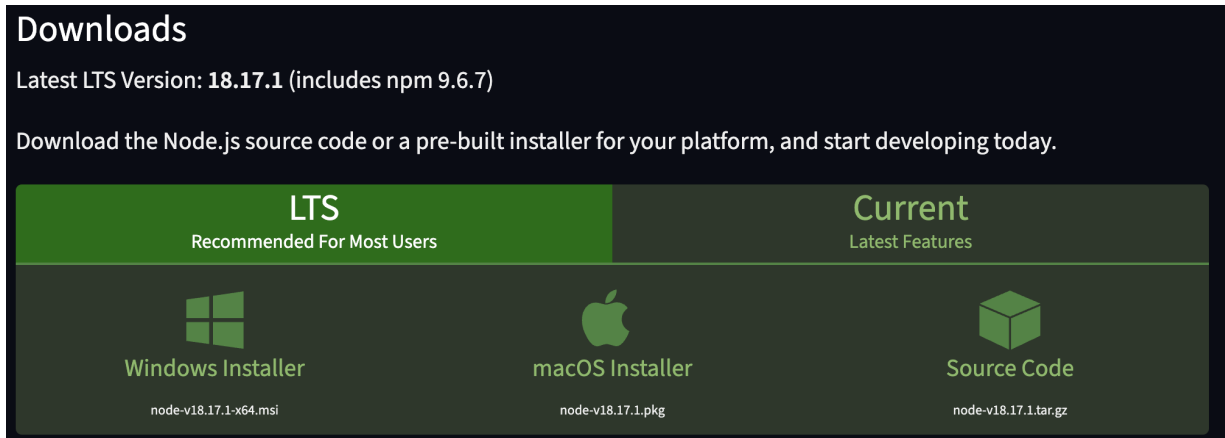


Activity No. 1 Getting Started with React Native	
Course Code:	Program: Computer Engineering
Course Title:	Date Performed:
Section:	Date Submitted:
Name:	Instructor:
<b>1. Objective(s)</b>	
This activity aims to enable students to build a basic application with local state management and Create React Native App.	
<b>2. Intended Learning Outcomes (ILOs)</b>	
After this module, the student should be able to: <ul style="list-style-type: none"> <li>• Configure a React Native development using the Expo Go environment;</li> <li>• Deploy a live-demo of a mobile application on both iOS and Android.</li> </ul>	
<b>3. Discussion</b>	
<p>There are two common ways to set up a React Native development environment: expo-cli and react-native CLI.</p> <p><b>expo-cli</b>  <i>This is a command-line utility for configuring and publishing React Native projects. This is the best way to get started as a beginner. This tool generates a QR code which you can scan to launch the app on your device. As you update your code, the changes will automatically be reflected on your device. In order to preview the app on your device, you'll be prompted to download the Expo app, which is a React Native app previewing client.</i></p> <p><i>The downside to this approach is that it only works for pure-JavaScript apps. If at any point you need to use native modules (Swift, Kotlin, etc), you'll need to eject. The expo-cli utility provides an eject command to export your app into the same format used by react-native CLI, which you can then add native modules to.</i></p> <p><i>The company behind expo-cli, Expo, has been heavily involved in the React Native codebase and community since its inception, so you can be confident that the tool is well-maintained and doesn't deviate too much from a "standard" React Native setup.</i></p> <p><i>We'll walk through setting up a project in the next section, Quick Start.</i></p> <p><b>React Native CLI</b>  <i>If you're integrating React Native into an existing native app, or if you know you'll need custom native modules in your app, you'll want to use the react-native CLI to create your app. You can read how to do this on the Facebook docs for environment setup. Click on the "React Native CLI Quickstart" tab.</i></p> <p><i>In terms of the JavaScript code, the main difference from expo-cli is that react-native apps require registering the root component of your app using <code>AppRegistry.registerComponent</code>, e.g:</i></p> <p><i>This allows you to optionally have multiple React Native root components, which you might embed into different native screens.</i></p> <p>In this class, we will use <b>expo-cli</b> to develop native applications.</p>	
<b>4. Materials and Equipment</b>	
To properly perform this activity, the student must have: <ul style="list-style-type: none"> <li>• Node.js LTS</li> </ul>	

- Expo
- VS Code (or any code editor).

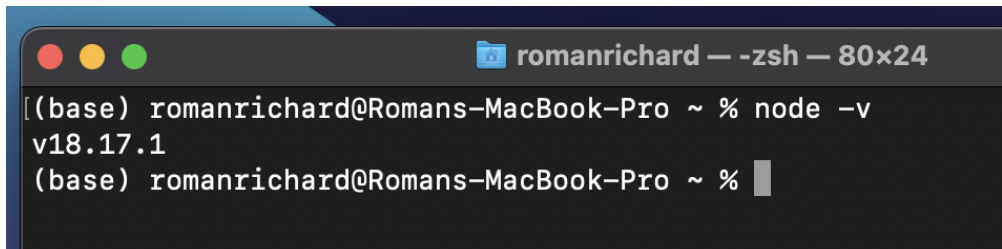
## 5. Procedure

1. Download node.js LTS.



It is recommended that you install the LTS version. This is intended for most users due to stability. To confirm installation, you can open your terminal and check the version of your node.js.

It should show version 18.17.1 like the figure below.



2. Create a new React Native project called "awesomeProject" using expo.  
Run the command `npx create-expo-app awesomeProject`

```
romanrichard — -zsh — 95x24
(base) romanrichard@Romans-MacBook-Pro ~ % sudo npx create-expo-app awesomeProject
Password:
✓ Downloaded and extracted project files.
> npm install
npm WARN deprecated @npmcli/move-file@1.1.2: This functionality has been moved to @npmcli/fs
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use
Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/
blog/math-random for details.
npm WARN deprecated uglify-es@3.3.9: support for ECMAScript is superseded by `uglify-js` as of
v3.13.0

added 1222 packages, and audited 1223 packages in 2m

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

5 moderate severity vulnerabilities

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

Run `npm audit` for details.
```

Let expo install dependencies such as shown in the figure. If you're able to achieve something similar to the screenshot below, then you can proceed to the next step.

```
✓ Your project is ready!

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

- cd awesomeProject
- npm run android
- npm run ios
- npm run web
(base) romanrichard@Romans-MacBook-Pro ~ %
```

3. Run expo using `npx expo start`.

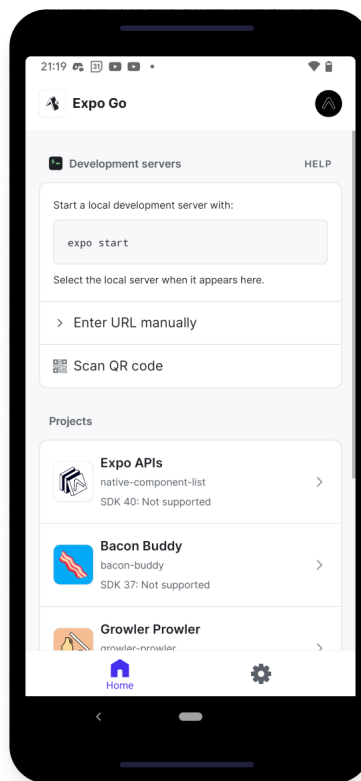
```
awesomeProject — node • sudo — 95x24
((base) romanrichard@Romans-MacBook-Pro awesomeProject % sudo npx expo start
[Password:
Starting project at /Users/romanrichard/awesomeProject
Starting Metro Bundler

<img alt="QR code for Expo Go" data-bbox="135 140 358 320"/>

> Metro waiting on exp://172.30.10.89:8081
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

> Using Expo Go
> Press s | switch to development build
```

Make sure to download Expo Go on your phone to scan the code shown to connect. Alternatively, you can also input the shown socket.



The following sections will have no screenshots provided as a guide, you are expected to include all your answers here under section 6: output.

4. Open VSCode (or any compatible editor) and explore the folder.
5. Provide a quick summary of the contents of the awesomeProject application.

## 6. Output

Provide an output of your work here.

## 7. Supplementary Activity

Modify **app.js** to reflect all names of the members of the team. This change must then reflect on your live demo. Provide the following:

- What did you change in app.js?
- Provide a picture of the live demo on Android and iOS device.

## 8. Conclusion

## 9. Assessment Rubric