January 15th, 2024

**Spinning Up React Native App**

Lab #1

**Prepared for**:

Southern Alberta Institute of Technology (SAIT)

Software Development Diploma Program

CPRG-303-A Mobile Application Development Course (Winter 2024)

**Instructor**: Nick Hamnett

**Prepared by:**

Sherif Mansour

Table of Contents

[Instructions 1](#_Toc156252731)

[Guidelines 1](#_Toc156252732)

[Notes 1](#_Toc156252733)

[Submission Guidelines 2](#_Toc156252734)

[GitHub 2](#_Toc156252735)

[D2L Submission 2](#_Toc156252736)

[Section A: Setting up the development environment for React Native 3](#_Toc156252737)

[Section B – Creating a new application 11](#_Toc156252738)

[Section C - Device Emulator Setup 14](#_Toc156252739)

[Section D – Run Project in Android Studio 17](#_Toc156252740)

[Section E – To run the app from the Command Line 21](#_Toc156252741)

[References 24](#_Toc156252742)

# Instructions

React Native is a significant framework in mobile app development. It allows developers to build mobile apps for both iOS and Android using a single TypeScript or JavaScript codebase. In this lab assignment, you will document setting up a React Native App which will be used for future lab assignments.

1. Review the Setting up the development environment to learn more about creating an app (<https://reactnative.dev/docs/environment-setup?guide=native>). Research any unfamiliar terms or concepts to ensure a clear understanding of the steps involved.
2. Document the step-by-step process of setting up a React Native environment. Use the Guidelines below on what is expected to be in the document.
3. Perform each step and as you go through each one, document the process of setting up a React Native environment.
4. If you encounter any difficulties, seek assistance from your instructor or classmates.
5. That's it! You now have a React Native app up and running.

# Guidelines

Come up with a Word or Markdown document that includes the following:

1. **System requirements**: Document specifications of the system being used for React Native development. This includes the CPU, RAM size, and Windows version.
2. **Installation instructions**: Include step-by-step instructions for installing the necessary tools and dependencies required for the framework, such as Node.js, and the React Native CLI.
3. **Configuration steps**: Detail any necessary configuration steps required to set up the framework, such as setting environment variables or configuring project settings.
4. **Project creation**: Outline the steps involved in creating a new project using the framework, including any necessary setup or configuration.
5. **Running the project**: Detail how to run the project in an Android Device Simulator.
6. **Troubleshooting**: Include information on how to troubleshoot common issues that may arise during setup or development, such as debugging and error messages. This may need to be updated in the future.
7. **Resources**: Provide links to additional resources and documentation that can be used for reference and further learning, such as official documentation, tutorials, and Stack Overflow answers.

# Notes

* Name your app something like "Incredible Todo List App".
* Use the React Native CLI (not Expo Go) to create the app.
* It’s recommended you use JavaScript instead of TypeScript.

# Submission Guidelines

## GitHub

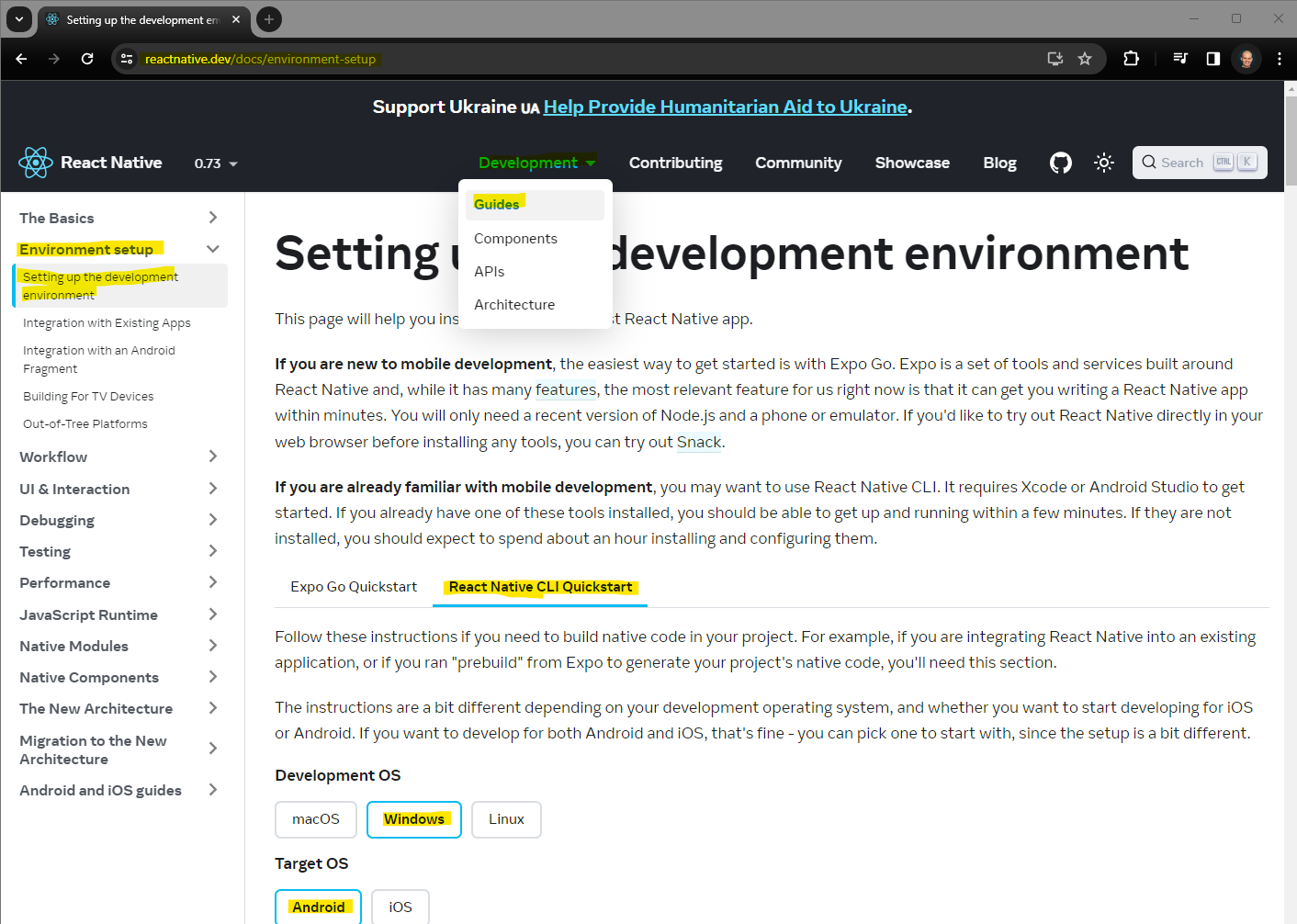
1. Create a new public GitHub repository to host your code.
2. Push all your app files (except the "node\_modules" folder) and setup instructions document to the GitHub repository.

## D2L Submission

1. Compress all the files and folders for your assignment (except the "node\_modules" folder) into a single ZIP file.
2. Upload the ZIP file to the appropriate assignment drop-box on D2L.
3. Include the link to your GitHub repository in the comments area.

# Section A: Setting up the development environment for React Native

1. Navigated to <https://reactnative.dev/docs/environment-setup>



1. Navigated to <https://nodejs.org/en> and downloaded and installed Node.js 20.11.0 LTS with built-in default settings.

A screenshot of a computer

Description automatically generated

1. Navigated to <https://developer.android.com/studio?gad_source=1&gclid=CjwKCAiAzJOtBhALEiwAtwj8to5iLIFewp3DQixn4ECo9FolxpkBtNAQFzH30rsUvFhk3KYF3qb3PBoCHg0QAvD_BwE&gclsrc=aw.ds> and downloaded and installed Android Studio Hedgehog.

A screenshot of a computer

Description automatically generated

1. Ensured that I have the Android SDK Installed

A screenshot of a computer

Description automatically generated

1. Accepted Terms and Conditions.
2. Upon Downloading Components, the Intel HAXM installation failed.A screenshot of a computer

   Description automatically generated
3. I was directed to <https://github.com/intel/haxm/wiki/Installation-Instructions-on-Windows#user-content-System_Requirements> for further instructions.

A screenshot of a computer

Description automatically generated

1. My device and operating system specifications are as follows:

A screenshot of a computer

Description automatically generated

1. Since the GitHub instruction page didn’t contain Windows 11, I was unsure whether Windows 11 was the issue and would have to downgrade my OS to Windows 10 to be able to run the program, so I inquired with the instructor for further guidance and was advised to navigate to <https://stackoverflow.com/questions/54955446/intel-haxm-is-required-to-run-this-avd>.
2. The StackOverFlow questions forum page suggested that I check the “Hyper-V” feature under “Windows Features” in the Control Panel.

A screenshot of a computer

Description automatically generated

1. I was able to check the “Hyper-V Management Tools”, however, was unable to check the “Hyper-V Platform”
2. Upon further research, I came across a [YouTube video](https://www.youtube.com/watch?v=TUlK-zAdhQk&t=4s) that explained how to get the “Hyper-V Platform” selected. It involved restarting the computer, navigating to the Virtualization Settings in the BIOS and changing the setting to “Yes”.
3. Upon changing the BIOS settings, I was able to check the “Hyper-V Platform” feature in Windows Features

A screenshot of a computer

Description automatically generated

1. My SDK Manager now looks like this:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Add the appropriate environment variables for Android Studio. Under “Edit Environment Variables for your account” in the Control Panel and added:

* the “ANDROID\_HOME” variable with the corresponding value location, and
* the Platform “Path” for Android Studio variable with the corresponding value location.

**Note**: The AppData folder was hidden and thus had to “Show Hidden Items” in the View options under File Explorer.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Created a folder in my C: Drive

A screenshot of a computer

Description automatically generated

1. Open the created folder from Step 16 in the command prompt by typing cmd in the File Explorer address bar

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

# Section B – Creating a new application

1. To create a new application, I entered the command “npx react-native@latest init” followed by “MySecondApp” and then entering ‘y’

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. If npm is out of date, run the command “npm i -g npm”

A screen shot of a computer

Description automatically generated

1. Open the newly created application in VSC.

A screenshot of a computer

Description automatically generated

1. Change TypeScript (includes data types) to JavaScript

* Change file extension of “App.tsx” to “App.jsx”
* Delete the errors in file App.jsx that are unique to TypeScript by removing:
  + Line 9 – import type {PropsWithChildren} from 'react';
  + Line 27 to 29 – only

type SectionProps = PropsWithChildren<{

title: string;

}>;

* + Line 27 – only “: SectionProps” & “: React.JSX.Element”
  + Line 53 – only “: React.JSX.Element”
  + Save

Before:

A screenshot of a computer

Description automatically generated

After:

A screenshot of a computer

Description automatically generated

# Section C - Device Emulator Setup

1. To create a virtual machine image, open “Virtual Device Manager” in Device Manager and click “Create Device” and select “Pixel 7” under Phone.

A screenshot of a computer

Description automatically generated

1. Select the “R” System Image, download the associated files and click Next

A screenshot of a computer

Description automatically generated

1. Configure the AVD Name and Startup Orientation

A screenshot of a computer

Description automatically generated

1. To start the device, click the play button

A screenshot of a computer

Description automatically generated

A screen shot of a cell phone

Description automatically generated

# Section D – Run Project in Android Studio

1. Open the project in Android Studio. File>Open>… Locate the “android” file in your application folder

A screenshot of a computer

Description automatically generated

1. It is recommended to ignore Android Studio prompt to update the project.
2. In Android Studio, click Build>Make Project and wait until prompt Build Successful is shown.

A screenshot of a computer

Description automatically generated

1. To run the application, click Run ‘app’ so it will run it in the Android Emulator

A screenshot of a computer

Description automatically generated

1. Upon successful running the device, the content of the device screen show:  
   A screenshot of a cell phone

   Description automatically generated
2. To fix the “Unable to load script” prompt, go to the command line inside of the directory where the application was created and run the command “npm run start” and then enter “r” to reload the app.

A screen shot of a cell phone

Description automatically generated

1. Changes can be made to the “App.jsx” file and saved to verify that the app is live and showcases the changes made.

# Section E – To run the app from the Command Line

1. Open the application folder in the command prompt by typing cmd in the File Explorer address bar

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Run the command “npm run start” & enter “a”

A screenshot of a computer

Description automatically generated

1. It will take some time for the application to build in the command prompt, the Android emulator will start on its own.

A screenshot of a cell phone

Description automatically generated

# References