Lab Assignment 01 – Spinning Up React Native App

We will be setting up using the React Native CLI Quickstart guide, found here: <https://reactnative.dev/docs/environment-setup?guide=native&platform=android>

# System Requirements

* Windows 10 or later (64-bit), Windows 11 recommended
* Node.js and NPM (node package manager)
* JDK (Java Development Kit) 8 or newer
* Android Studio with Android SDK and AVD Manager
* React Native CLI
* Android Emulator or a physical Android device
* Intel i5 CPU or better
* 8GB RAM or better
* 10GB Storage space

# Installation & Configuration

## Install Node, JDK using Chocolatey

1. Run PowerShell as administrator.
2. Check Execution Policy (Get-ExecutionPolicy) is set to either Bypass or AllSigned.
3. Run the following command:

Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))

1. Once installed, run the following command:

choco install -y nodejs-lts microsoft-openjdk17

## Setting Up Android Development Environment

1. Download and install Android Studio installation wizard.
2. Ensure the installation of Android SDK, Android SDK Platform, Android Virtual Device, and Hyper-V (if using an AMD processor).
3. Accept the recommended settings for the Android environment.
4. Once installation is finished, run the Android Studio program.
5. On the home page, select More Actions > SDK Manager.
6. The React Native app requires the Android 13 (Tiramisu) SDK, so select Android 13, click on “Show Package Details” to expand the features, and make sure Android SDK Platform 33 is checked, as well as check Intel x86\_64 Atom System Image.
7. Move to the SDK Tools tab on the same page, click on “Show Package Details” to expand the features, then enable the 33.0.0 version.
8. Click “Apply” to install.

## Configure ANDROID\_HOME environment variable

1. Open the Windows Control Panel.
2. Open User Accounts, then User Accounts again.
3. Click on Change my environment variables.
4. Click on New… to create a new ANDROID\_HOME user variable that points to the path of your Android SDK, installed here by default:

%LOCALAPPDATA%\Android\Sdk

1. Insure it has been properly installed by opening PowerShell and ensuring the directory exists with the following command: **Get-ChildItem -Path Env:\**
2. Add platform-tools to Path by using Change my environment variables, select the Path variable, click Edit, Click New, and then path to the following default location:

%LOCALAPPDATA%\Android\Sdk\platform-tools

# Project Creation

1. Run the following command in the Windows console (as administrator):

npx react-native@latest init StarterProject

1. Open the StarterProject using Android Studio and select the android folder. Once open, select Device Manager on the right of the API to see the list of available Android Virtual Devices (AVD).
2. Create a new AVD by selecting “Create Virtual Device”, selecting any phone, and the API 33 “Tiramisu” as the system image.
3. Click on the play button to run the virtual device.

## Running the Project

1. Open a new command terminal, go to the directory of your StarterProject, then run the following command:

npm start

1. Once the server has finished setting up, execute run on Android (“a”), causing it to open in a new terminal and run the Android emulator.
2. The emulator can also be run directly from Android Studio.
3. Be sure to be patient, especially the first time, as it can take a long time for the system to finish setting up and finally run the React Native app.