Lab Assignment

1. System requirements:

To develop with React Native, it's recommended to have a modern multi-core processor (like Intel i5/i7/i9 or AMD Ryzen), with at least 8GB of RAM, though 16GB or more is preferable for optimal performance. Your operating system should be Windows 10 or later (64-bit), or macOS Catalina (10.15) or newer for iOS development. A Solid-State Drive (SSD) with at least 100GB of free space is recommended for faster performance. Essential software includes Node.js, Java Development Kit (JDK) 8 or newer, Android Studio for Android development, and Xcode for macOS users targeting iOS. A stable and fast internet connection is necessary for downloading dependencies and updates.

1. Installation instruction:

Install Node.js:

* Go to the Node.js website.
* Download the installer for your operating system (Windows, macOS, Linux).
* Run the installer and follow the on-screen instructions to install Node.js and npm.

Install the React Native CLI:

* Open a terminal or command prompt.
* Install the React Native command line interface (CLI) by running npm install -g react-native-cli.

Install Android Studio (for Android development):

* Download Android Studio from the official website.
* Run the installer and choose a 'Custom' setup when prompted.
* Make sure to check the boxes next to 'Android SDK', 'Android SDK Platform', 'Performance (Intel ® HAXM)', and 'Android Virtual Device'.
* Once the installation is complete, launch Android Studio and go through the 'Android SDK Manager' to install any additional SDKs and tools as needed.

1. Configuration step:

Node.js and NPM:

* Ensure that Node.js and npm are added to your system's PATH. This is usually done automatically during the Node.js installation.
* To check, open Command Prompt and type node -v and npm -v. If you see version numbers, they are correctly set.

JAVA\_HOME:

* Install JDK (Java Development Kit), if not already installed.
* Set JAVA\_HOME in your environment variables.
* Locate your JDK installation directory (e.g., C:\Program Files\Java\jdk-11.0.1).
* Right-click on 'This PC' or 'My Computer' -> Properties -> Advanced system settings -> Environment Variables.
* Under System Variables, click New, and add JAVA\_HOME as the variable name and the path to your JDK installation as the variable value.

Configuring Android Studio

* Download and install Android Studio with Android SDK, Android SDK Platform, and Performance (Intel ® HAXM) (for Intel machines).
* Open SDK Manager in Android Studio and ensure you have the latest versions of SDK platforms and SDK tools.
* Set up an Android Virtual Device (AVD) for emulation.

1. Project Creation

Install React Native CLI:

* Open a terminal (or command prompt) and run npm install -g react-native-cli to install the React Native Command Line Interface globally.
* This allows you to use React Native commands from any directory in your system.

Create a New React Native Project:

* In the terminal, navigate to the directory where you want to create your project.
* Run npx react-native init ProjectName, replacing ProjectName with your desired project name (e.g., npx react-native init IncredibleTodoListApp).
* This command creates a directory named ProjectName and initializes a new React Native project inside it.
* The initialization process installs necessary dependencies and creates a basic project structure.

1. Running the project

- To run a React Native app in an Android simulator, first ensure Android Studio is installed and an Android Virtual Device (AVD) is set up. Open your project directory in the command prompt or terminal, start the Metro Bundler with npx react-native start, and then run npx react-native run-android in a new terminal window. This command compiles and launches your app on the emulator.

1. Troubleshooting

Installation and Setup:

* Ensure Node.js, NPM, and React Native CLI are correctly installed. Use node -v, npm -v, and react-native -v for verification.
* Check Android Studio and SDK setups, including environment variables like ANDROID\_HOME.

Emulator and Device Connectivity:

* For emulator issues, ensure AVD is correctly configured in Android Studio.
* If devices aren't recognized, check USB debugging settings and driver installations.

Build and Runtime Errors:

* Resolve dependency issues with npm install and clear cache if needed.
* Check build.gradle files for correct configuration.
* Use cd android && ./gradlew clean for Gradle-related problems.
* For runtime errors, carefully read the error messages. Use React Native Debugger for in-depth debugging.

1. Essential Resources for React Native Development

For React Native development, key resources include the Official React Native Documentation for foundational knowledge and updates. Beginners can benefit from the React Native Tutorial by Programming with Mosh on YouTube for a practical start. For community support and problem-solving, Stack Overflow's React Native section is invaluable. Those interested in deeper learning can explore the comprehensive Complete React Native + Hooks Course on Udemy. Additionally, the React Native Blog offers insightful articles and updates, while the React Native GitHub repository is great for keeping track of the latest developments and community contributions.