React Native

* React native installation and setup

1:- Required software:-

A:-Node.js

B:-Vs Code in latest Version

C:-Android studio

D:-Xcode(App store)

E:- JDk (openJDk 11 version)

F:-Git(**Global Information Tracker)**

2:- Software Download and Installation Step:-

A:-NodeJs:- Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on a JavaScript Engine and executes JavaScript code outside a web browser, which was designed to build scalable network application.

1:-Download and Installion Step: -

Step 1:-open the any browser and Type in NodeJs Download click enter and otherwise ([url:-nodejs.org](file:///C:\Users\dpc48\Downloads\-nodejs.org)) url copy and paste in search bar .redirect Nodejs Offical site and click in Latest Version Download .

Step 2:-Nodejs.msi setup click in install it

Step 3:- Nodejs install it and open the Cmd (Command line) and Type it ( node –version)(npm –version) version number display that mean Nodejs Success install your Window system.

B: VsCode:- Visual Studio Code is **a streamlined code editor with support for development operations like debugging, task running, and version control**. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE.

1:-Download and Installion Step: -

Step 1:- open the Browser Search Bar click in Vs Code Download and otherwise ( url:-<https://code.visualstudio.com/download>) click in url and automatic Redirect VS Code Offical Site Select your System bit 32/64bit Select any bit automatic download it

Step 2:- Click in Vs Code.exe file and basic details required like this (Desktop icon required and not ,VS Code path) and Next button click and install it

C: Android Studio:- Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems.

1:-Download and Installion Step: -

Step 1:-

**(a) Install Android Studio:-** open the Browser Search Bar click in Android studio Download and otherwise ( url:- https://developer.android.com/studio) click in url and automatic Redirect Android studio Offical Site first Download Android Studio click ->Alert Message open and scroll Down Term&Condition button click ->Download button click in her Download it. While on Android Studio installation wizard, make sure the boxes next to all of the following items are checked:

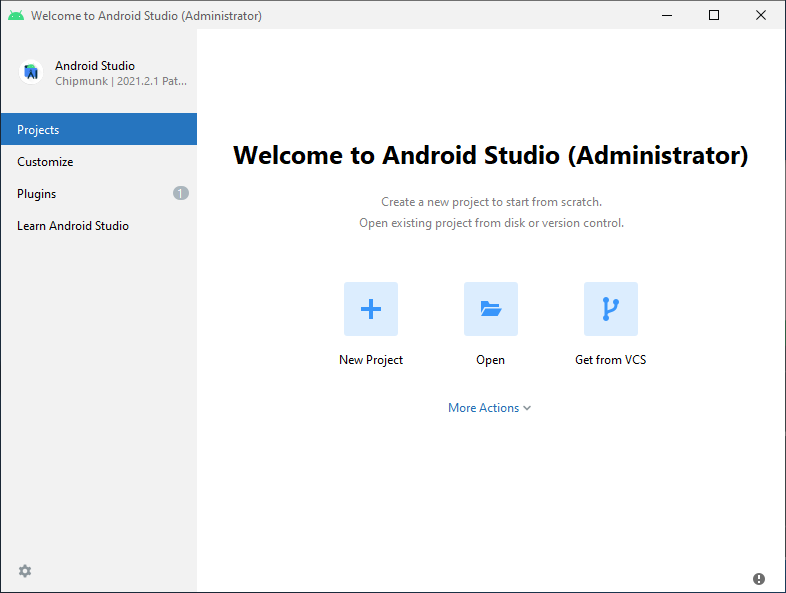
* Android SDK
* Android SDK Platform
* Android Virtual Device
* If you are not already using Hyper-V: Performance (Intel ® HAXM) (see here for Amd or Hyper-v)

Then, click "Next" to install all of these components.If the checkboxes are grayed out, you will have a chance to install these components later on.Once setup has finalized and you're presented with the Welcome screen, proceed to the next step.

Step 2:-

(b) Install the Android SDK:- Android Studio installs the latest Android SDK by default. Building a React Native app with native code, however, requires the Android 12 (S) SDK in particular. Additional Android SDKs can be installed through the SDK Manager in Android Studio.

To do that, open Android Studio, click on "More Actions" button and select "SDK Manager".



The SDK Manager can also be found within the Android Studio "Preferences" dialog, under **Appearance & Behavior** → **System Settings** → **Android SDK**.

Select the "SDK Platforms" tab from within the SDK Manager, then check the box next to "Show Package Details" in the bottom right corner. Look for and expand the Android 12 (S) entry, then make sure the following items are checked:

* Android SDK Platform 31
* Intel x86 Atom\_64 System Image or Google APIs Intel x86 Atom System Image

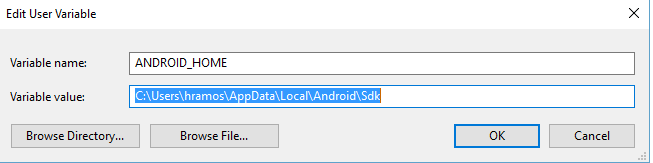
Next, select the "SDK Tools" tab and check the box next to "Show Package Details" here as well. Look for and expand the Android SDK Build-Tools entry, then make sure that 31.0.0 is selected.

Finally, click "Apply" to download and install the Android SDK and related build tools.

Step3:-

#### (c) Configure the ANDROID\_HOME environment variable:- The React Native tools require some environment variables to be set up in order to build apps with native code.

1. Open the **Windows Control Panel.**
2. Click on **User Accounts,** then click **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 to your Android SDK:



The SDK is installed, by default, at the following location:

%LOCALAPPDATA%\Android\Sdk

You can find the actual location of the SDK in the Android Studio "Settings" dialog, under **Appearance & Behavior** → **System Settings** → **Android SDK**.

Open a new Command Prompt window to ensure the new environment variable is loaded before proceeding to the next step.

1. Open powershell
2. Copy and paste **Get-ChildItem -Path Env:\** into powershell
3. Verify ANDROID\_HOME has been added

Step 4 :-

(d) Add platform-tools to Path:-

1. Open the **Windows Control Panel.**
2. Click on **User Accounts,** then click **User Accounts** again
3. Click on **Change my environment variables**
4. Select the **Path** variable.
5. Click **Edit.**
6. Click **New** and add the path to platform-tools to the list.

The default location for this folder is:

C:\Users\dpc48\Desktop\platformtools.PNG

#### Note 1:- Please check all Environment variable set are not

#### Open the Control panel ->UserAccounts clicks->change my environment variable click ->open panel->Android side Environment variable set in here

#### Example:-

#### C:\Users\dpc48\Videos\first.PNG

#### Path variables click ->open pannel ->

#### android ndk-> java bin -> gradle bin ->Android Platform Tools ->Android sdk -> set in here all environment variable

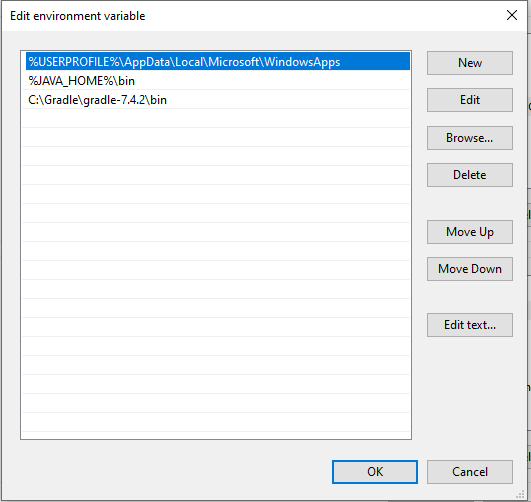
#### Example:-

#### C:\Users\dpc48\Videos\second.PNG

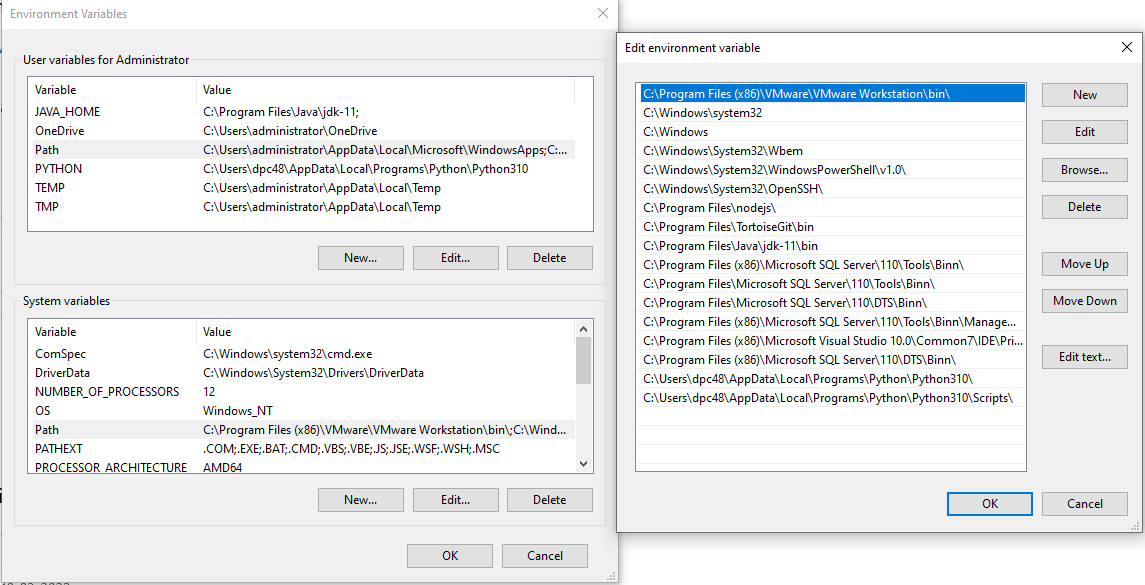
#### Note2:-This pc left button click ->ScrollDown->Properties click ->System Properties panel open ->Environment Variable click ->jdk path set are not check it

#### 

Note 3:-User variable for Administrator in-> Path variable click ->check in gradle and jdk path set are not



Note 4:-System Variables ->path variable click->jdk path set are not



### (E) Jdk :- **The JDK is the development platform for building Java applications. Learn about different JDK versions, then install a JDK in your development environment and use it to compile a Java program.** React Native also requires [Java SE Development Kit (JDK)](https://openjdk.java.net/projects/jdk/11/), which can be installed using Chocolatey as well.

1:-Download and Installion Step: -

Step 1:- Open an Administrator Command Prompt (right click Command Prompt and select "Run as Administrator"), then run the following command:(choco install -y openjdk11) and Download and Install it Jdk .regain open it cmd Type it (jdk --version). It mean version number Display so Jdk is install it



Sf

Note:- If you're using the latest version of Java Development Kit, you'll need to change the Gradle version of your project so it can recognize the JDK. You can do that by going to {project root folder}\android\gradle\wrapper\gradle-wrapper.properties and changing the distributionUrl value to upgrade the Gradle version. You can check out [here the lastest releases of Gradle](https://gradle.org/releases/).

(F) Git :-**Git** is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

1:-Download and Install Step:-

Step 1 (Url:- <https://git-scm.com/download/win>) open the url and choose the 32/64 bit as per Your system bit and Download and install it.

Step2:-open cmd and Type it (git –version) version number display that mean git successfully install it

## 3:- Creating a new application:-

Step 1:- If you previously installed a global react-native-cli package, please remove it as it may cause unexpected issues:

Example :- npm uninstall -g react-native-cli @react-native-community/cli

Step 2:- React Native has a built-in command line interface, which you can use to generate a new project. You can access it without installing anything globally using npx, which ships with Node.js. Let's create a new React Native project called "AwesomeProject":

Ex:- npx react-native init AwesomeProject

Note:- This is not necessary if you are integrating React Native into an existing application, if you "ejected" from Expo, or if you're adding Android support to an existing React Native project (see [Integration with Existing Apps](https://reactnative.dev/docs/integration-with-existing-apps)). You can also use a third-party CLI to init your React Native app, such as [Ignite CLI](https://github.com/infinitered/ignite).

Step3:-

### [Optional] Using a specific version or template:- If you want to start a new project with a specific React Native version, you can use the --version argument:- npx react-native init AwesomeProject --version X.XX.X

Step 4:- You can also start a project with a custom React Native template, like TypeScript, with --template argument:-

Ex:- npx react-native init AwesomeTSProject --template react-native-template-typescript

Step5 :-

## Preparing the Android device

You will need an Android device to run your React Native Android app. This can be either a physical Android device, or more commonly, you can use an Android Virtual Device which allows you to emulate an Android device on your computer.

Either way, you will need to prepare the device to run Android apps for development.

### Using a physical device

If you have a physical Android device, you can use it for development in place of an AVD by plugging it in to your computer using a USB cable and following the instructions [here](https://reactnative.dev/docs/running-on-device).

### Using a virtual device

If you use Android Studio to open ./AwesomeProject/android, you can see the list of available Android Virtual Devices (AVDs) by opening the "AVD Manager" from within Android Studio. Look for an icon that looks like this:

If you have recently installed Android Studio, you will likely need to [create a new AVD](https://developer.android.com/studio/run/managing-avds.html). Select "Create Virtual Device...", then pick any Phone from the list and click "Next", then select the **S** API Level 31 image.

If you don't have HAXM installed, click on "Install HAXM" or follow [these instructions](https://github.com/intel/haxm/wiki/Installation-Instructions-on-Windows) to set it up, then go back to the AVD Manager.

Click "Next" then "Finish" to create your AVD. At this point you should be able to click on the green triangle button next to your AVD to launch it, then proceed to the next step.

## 4:-Running your React Native application

First, you will need to start Metro, the JavaScript bundler that ships with React Native. Metro "takes in an entry file and various options, and returns a single JavaScript file that includes all your code and its dependencies."—[Metro Docs](https://facebook.github.io/metro/docs/concepts)

To start Metro, run npx react-native start inside your React Native project folder:- npx react-native start

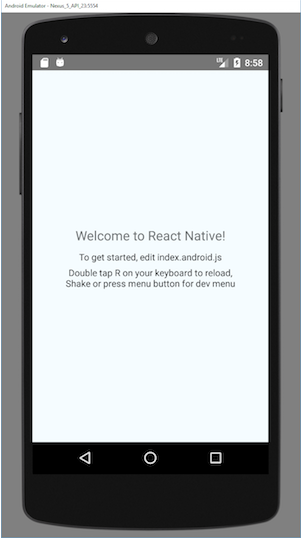
Note:- If you use the Yarn package manager, you can use yarn instead of npx when running React Native commands inside an existing project.

If you're familiar with web development, Metro is a lot like webpack—for React Native apps. Unlike Kotlin or Java, JavaScript isn't compiled—and neither is React Native. Bundling isn't the same as compiling, but it can help improve startup performance and translate some platform-specific JavaScript into more widely supported JavaScript.

### Step 2:

### Start your application:- Let Metro Bundler run in its own terminal. Open a new terminal inside your React Native project folder. Run the following:

Ex:- npx react-native run-android



5:- Ionic Install in windows

{A} :- Ionic cli :- Ionic apps are created and developed primarily through the Ionic [command-line](https://ionicframework.com/docs/reference/glossary#cli) utility. The Ionic CLI is the preferred method of installation, as it offers a wide range of dev tools and help options along the way. It is also the main tool through which to run the app and connect it to other services, such as Appflow.

Note :- Before proceeding, make sure your computer has [Node.js](https://ionicframework.com/docs/reference/glossary#node) installed. See [these instructions](https://ionicframework.com/docs/intro/environment) to set up an environment for Ionic.

Install the Ionic CLI with npm:

npm install -g @ionic/cli

If there was a previous installation of the Ionic CLI, it will need to be uninstalled due to a change in package name.

$ npm uninstall -g ionic  
$ npm install -g @ionic/cli

### Note :- With Node and NPM setup, let’s install the Ionic and Cordova CLI.

$ npm install -g ionic cordova

Note: The -g means this is a global install, so for Windows you will need to open an Admin command prompt. For Mac/Linux, you might need to run the command with sudo.

Once that’s done, create your first Ionic app:

$ ionic start helloWorld blank --**type=**ionic-angular

To run your app, cd into the directory that was created and then run the ionic serve command to test your app right in the browser!

$ cd helloWorld

$ ionic serve

Note:- more details in follow this link (url:- https://ionicframework.com/docs/intro/cdn)