Step 1. Install Required Software :

The first step to install React Native is Download and Install below required software packages.

**NodeJS** : Download and install the latest NodeJS windows machine installer software package from **nodejs.org** .

Step 2. Install React Native :

After installing **NodeJS** we can access the nmp packages via Command Line Interface ( DOS ) in windows. So open Command prompt type the below command to install React Native .



|  |  |
| --- | --- |
| 1 | npm install -g react-native-cli |

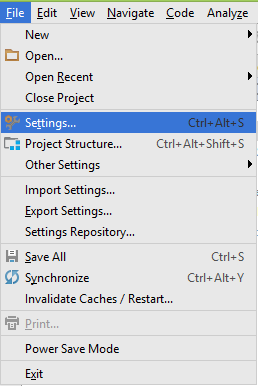
 Step 3. Install JDK( Java Development Kit ) :

**JDK :** Download and install JDK from Oracle’s official website **oracle.com** .

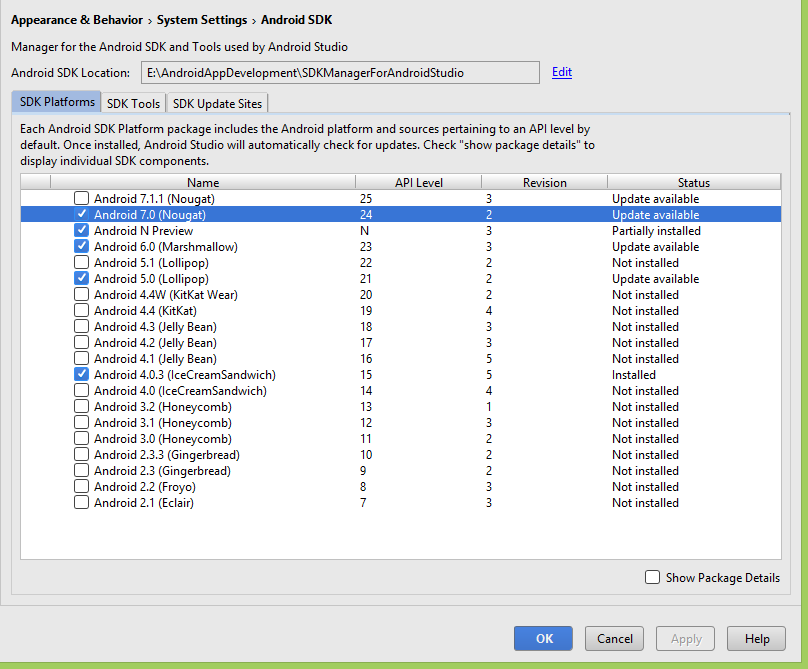
Step 4. Install Android Studio + SDK Manager :

Download the Android Studio latest version from Google Android Developers official Page **developer.android.com**.

After installing Android Studio Open it Goto **Files -> Settings .**

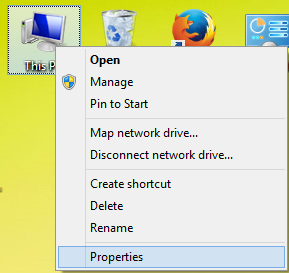
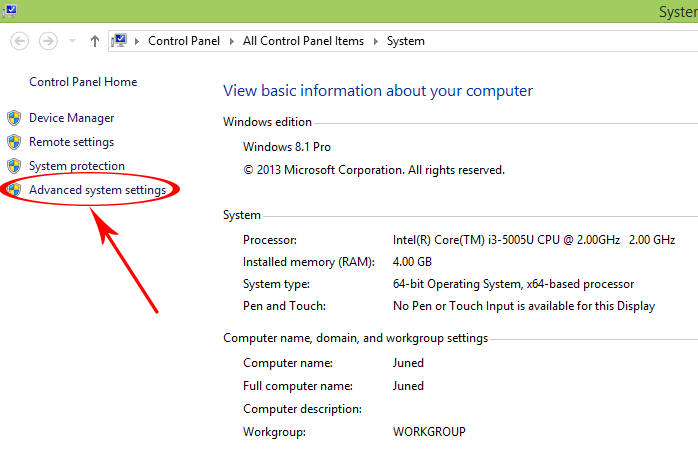
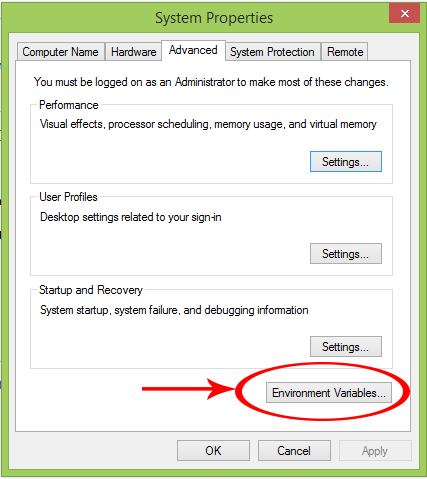
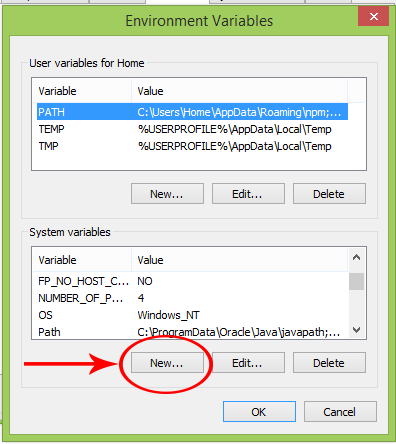
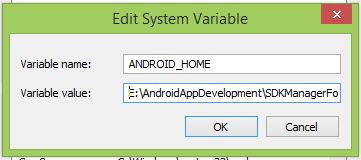


**Goto Appearance and Behavior -> System Settings -> Android SDK** and install latest android platform.



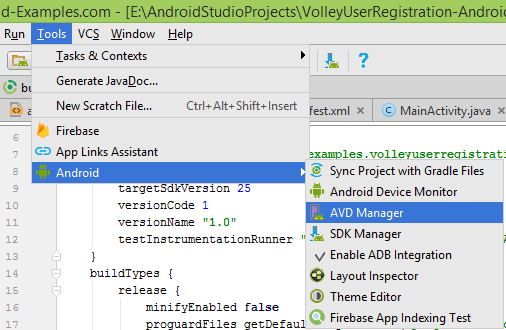
Step 5. Add Environment Variable ANDROID\_HOME in Windows :

React Native required ANDROID\_HOME variable to compile and run apps. So define the ANDROID\_HOME variable.

1. Right Click on My Computer.
2. Goto **Properties**.
3. Click on **Advanced System Settings**.
4. Click on **Environment Variables**.
5. Under System Variables click on New .
6. Set variable name as **ANDROID\_HOME** and variable value as your **SDK Manager’s Path**.
7. Here you go now your ANDROID\_HOME variable has been successfully set.

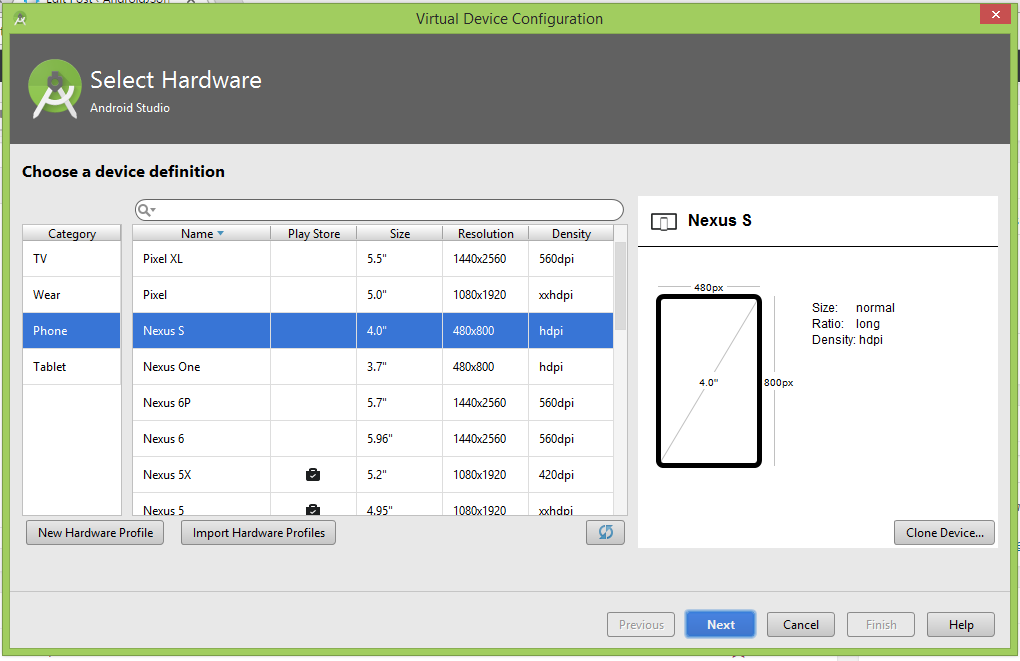
6. Start Android Virtual Device (AVD) :

1. Android virtual device is used to see the test result of our coding inside a virtual android machine just like a real android mobile phone. To setup AVD Open Android Studio -> Tools -> Android -> AVD Manager.

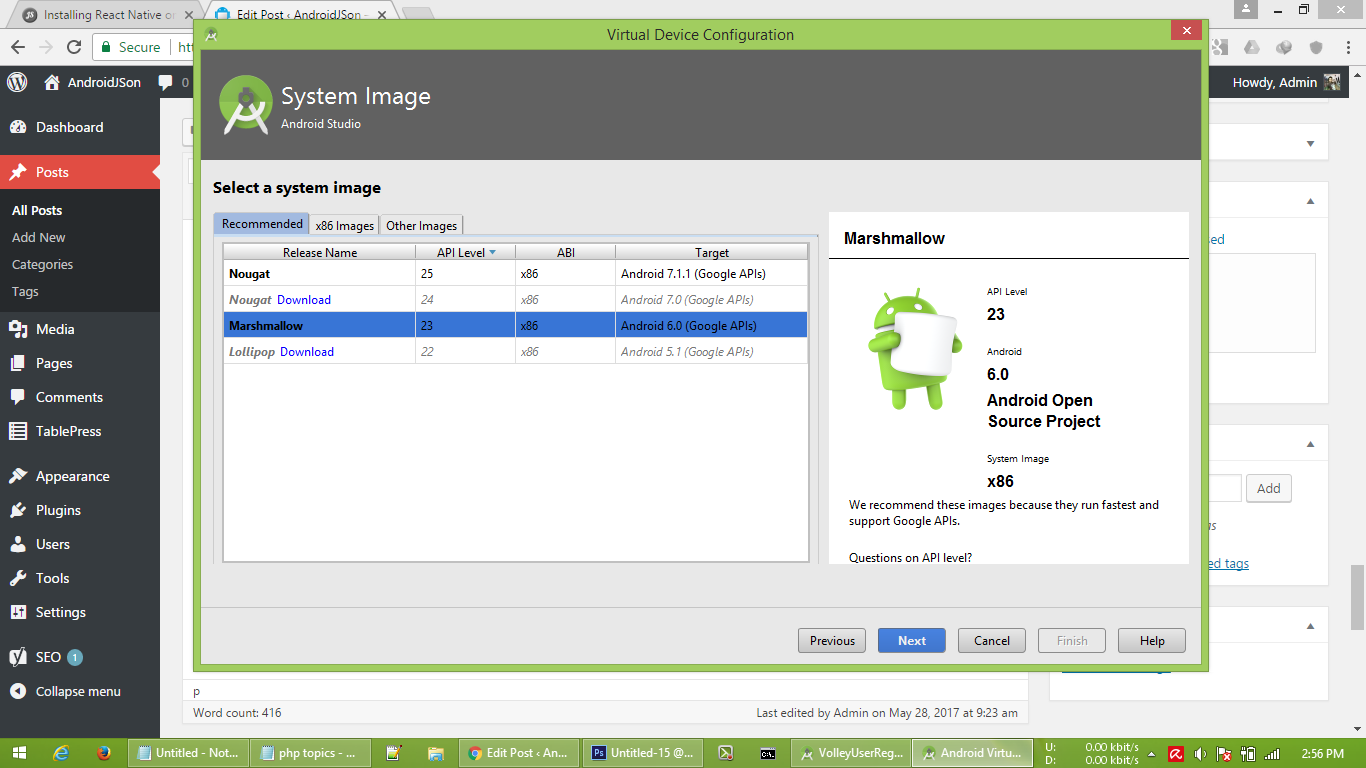


2. Click on **Create New Virtual Device.**

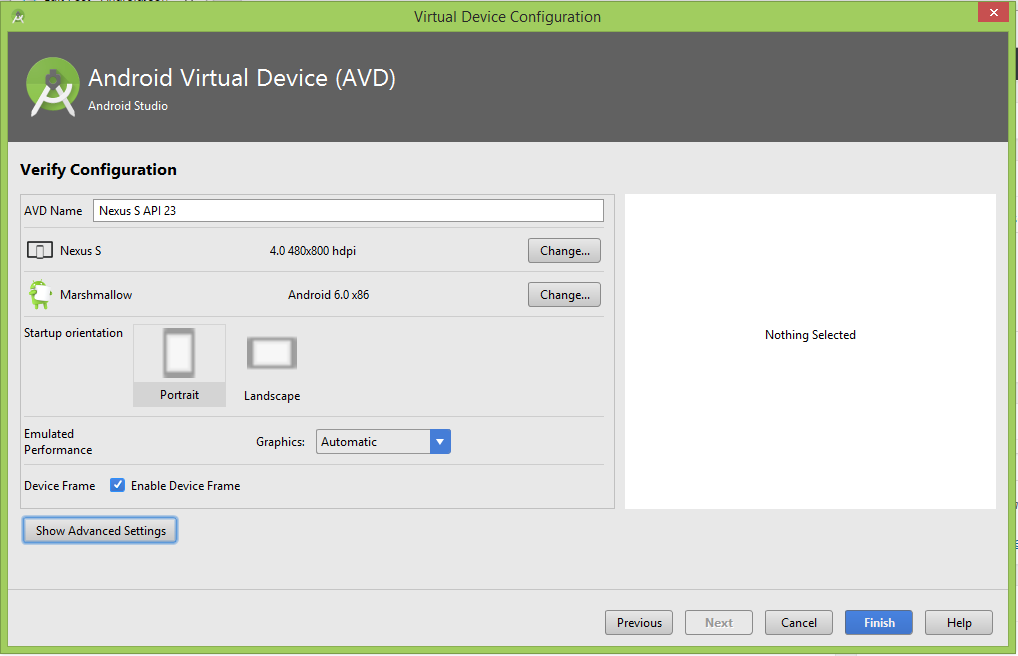
3. Select your device.



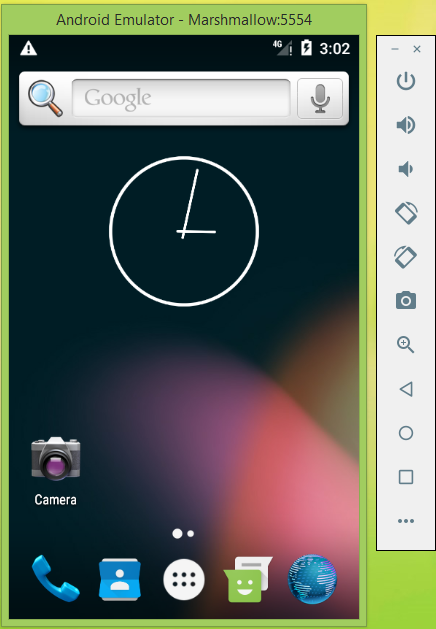
4. Select Android OS version System Image.



5. Name the AVD and hit the **Finish** button.



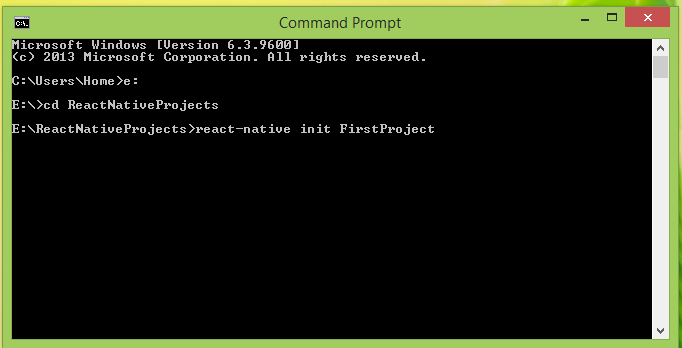
6. After done creating AVD just hit the Run button to start the AVD.



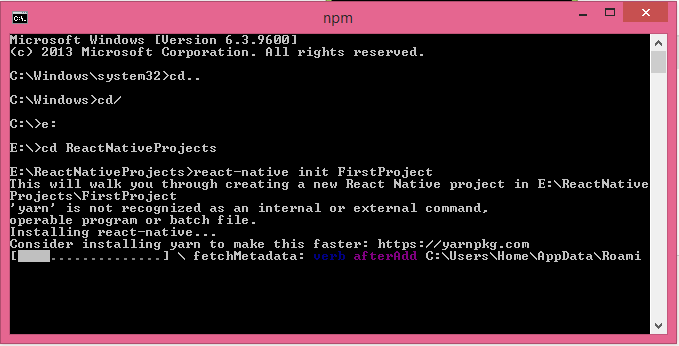
Step 7. Create your first React Native project :

1. Once you have finish all the installing than Create a folder in your drive in which you will store your all React Native projects. Than start command prompt and goto that folder inside command prompt. You can use **cd..**command to go back individually from folder to folder.

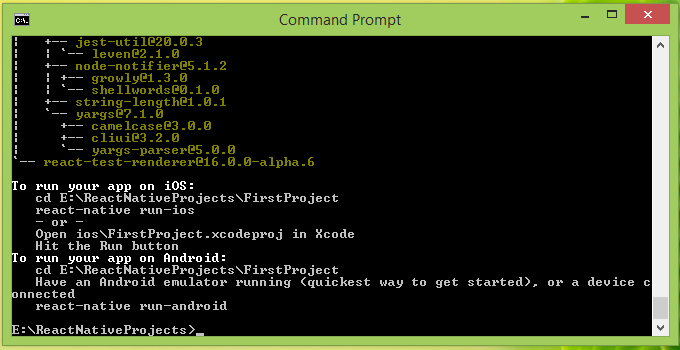
2. Now type **react-native init FirstProject** and press enter.



3. Now it will start downloading the React Native app support libraries from internet.



4. The final output will like below screenshot.



5. Now build the created app and run it into Android Emulator  using below command.



|  |  |
| --- | --- |
| 1  2  3  4  5 | E:\>cd reactnativeprojects    E:\ReactNativeProjects>cd firstproject    E:\ReactNativeProjects\FirstProject>react-native run-android |

6. If your project build successfully then you can see a message on your command prompt screen **BUILD SUCCESSFUL** .



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68 | E:\ReactNativeProjects\FirstProject>react-native run-android  Scanning 555 folders for symlinks in E:\ReactNativeProjects\FirstProject\node\_mo  dules (134ms)  Starting JS server...  Building and installing the app on the device (cd android && gradlew.bat install  Debug)...  Download https://jcenter.bintray.com/com/facebook/react/react-native/0.42.3-atla  ssian-1/react-native-0.42.3-atlassian-1.pom  Incremental java compilation is an incubating feature.  :app:preBuild UP-TO-DATE  :app:preDebugBuild UP-TO-DATE  :app:checkDebugManifest  :app:preReleaseBuild UP-TO-DATE  :app:prepareComAndroidSupportAppcompatV72301Library  :app:prepareComAndroidSupportSupportV42301Library  :app:prepareComFacebookFbuiTextlayoutbuilderTextlayoutbuilder100Library  :app:prepareComFacebookFrescoDrawee101Library  :app:prepareComFacebookFrescoFbcore101Library  :app:prepareComFacebookFrescoFresco101Library  :app:prepareComFacebookFrescoImagepipeline101Library  :app:prepareComFacebookFrescoImagepipelineBase101Library  :app:prepareComFacebookFrescoImagepipelineOkhttp3101Library  :app:prepareComFacebookReactReactNative0442Library  :app:prepareComFacebookSoloaderSoloader010Library  :app:prepareOrgWebkitAndroidJscR174650Library  :app:prepareDebugDependencies  :app:compileDebugAidl  :app:compileDebugRenderscript  :app:generateDebugBuildConfig  :app:mergeDebugShaders  :app:compileDebugShaders  :app:generateDebugAssets  :app:mergeDebugAssets  :app:generateDebugResValues  :app:generateDebugResources  :app:mergeDebugResources  :app:bundleDebugJsAndAssets SKIPPED  :app:processDebugManifest  :app:processDebugResources  :app:generateDebugSources  :app:incrementalDebugJavaCompilationSafeguard  :app:compileDebugJavaWithJavac  :app:compileDebugJavaWithJavac - is not incremental (e.g. outputs have changed,  no previous execution, etc.).  :app:compileDebugNdk UP-TO-DATE  :app:compileDebugSources  :app:transformClassesWithDexForDebug  Running dex in-process requires build tools 23.0.2.  For faster builds update this project to use the latest build tools.  :app:mergeDebugJniLibFolders  :app:transformNative\_libsWithMergeJniLibsForDebug  :app:processDebugJavaRes UP-TO-DATE  :app:transformResourcesWithMergeJavaResForDebug  :app:validateSigningDebug  :app:packageDebug  :app:assembleDebug  :app:installDebug  Installing APK 'app-debug.apk' on 'Marshmallow(AVD) - 6.0' for app:debug  Installed on 1 device.    <strong>BUILD SUCCESSFUL</strong>    Total time: 4 mins 53.489 secs  'adb' is not recognized as an internal or external command,  operable program or batch file.  Starting the app (E:\AndroidAppDevelopment\SDKManagerForAndroidStudio/platform-t  ools/adb shell am start -n com.firstproject/.MainActivity...  Starting: Intent { cmp=com.firstproject/.MainActivity } |

Here you go now you can see Your app is build successfully and run into Android emulator. Your emulator screenshot should look like this :

