Apache Cordova Framework

Developer's Guide

Prepared by: Alejandro V. Matute Jr., Part-Time IT Instructor
Laguna State Polytechnic University – Los Baños Campus – College of Computer Studies

Content

1. Installation of required software.
2. Create your first Cordova app.
3. Converting Website to Mobile App.
4. Running Cordova Application.

1. Installation of Required software

Installing the Requirements

Java Development Kit (JDK)

Install Java Development Kit (JDK) 8 or you may download the lower version here.

When installing on Windows you also need to set `JAVA_HOME` Environment Variable according to your JDK installation path (see Setting Environment Variables)

Gradle

As of Cordova-Android 6.4.0, Gradle is now required to be installed to build Android.

When installing on Windows, you need to add Gradle to your path, (see Setting Environment Variables)

Android SDK

Install Android Studio. Follow the instructions at the linked Android Developer site to get started. Opening Android Studio for the first time will guide you through the process of installing the Android SDK.

Node.js®

Install Node.js®. Follow the instruction on how to install Node.js® by clicking this link.

Apache Cordova

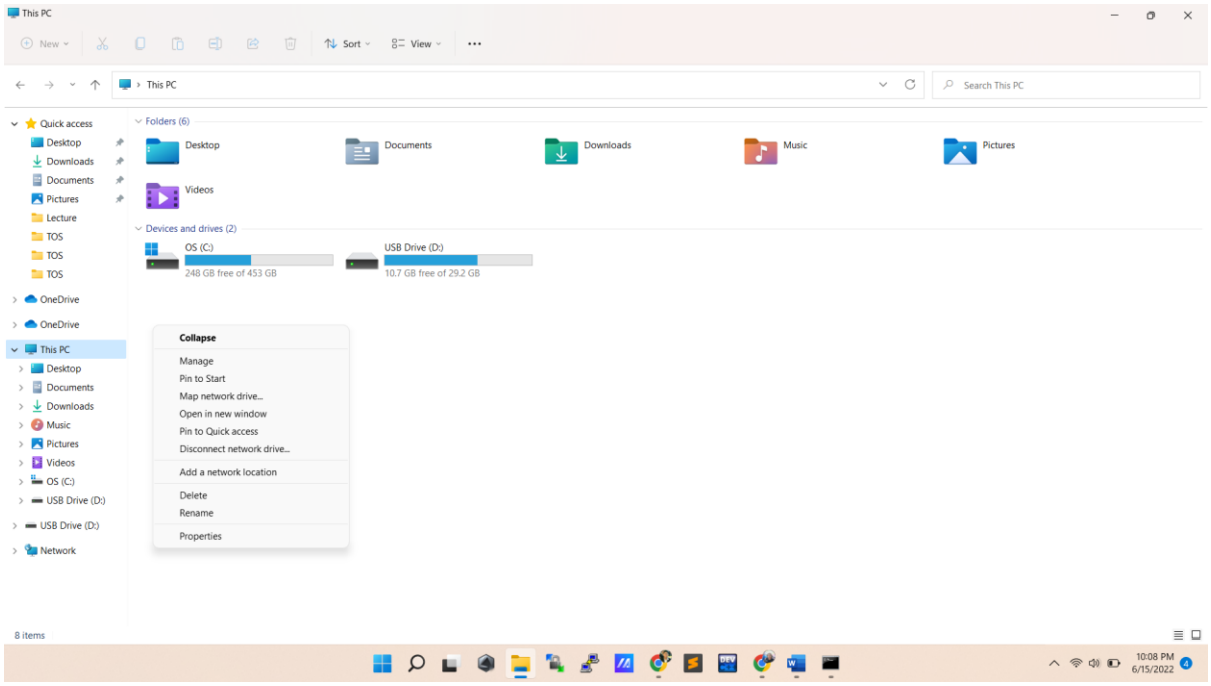
Install the Cordova CLI. Cordova is installed using the Node Package Manager (npm). Type the following in the command window to install:

```
npm install -g cordova
```

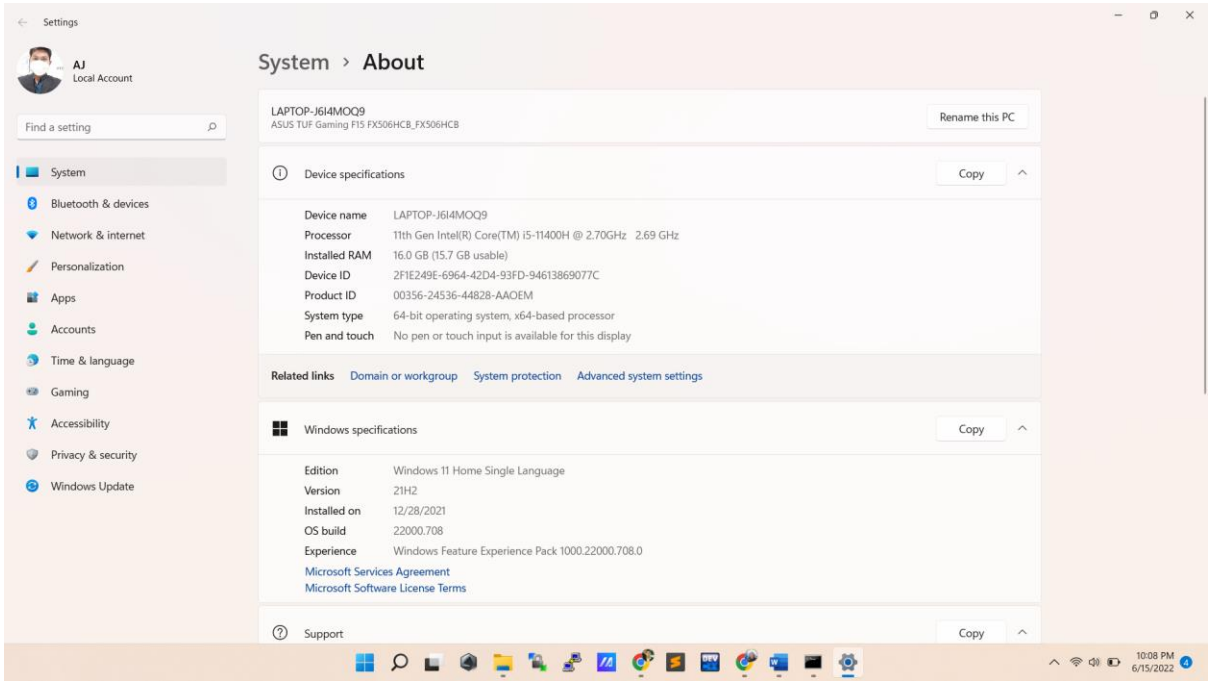
You can follow the instruction to setup apache cordova here.

Java Development Kit (JDK) – Setting Environment Variables

1. Right-click “This PC” and find “Properties”.

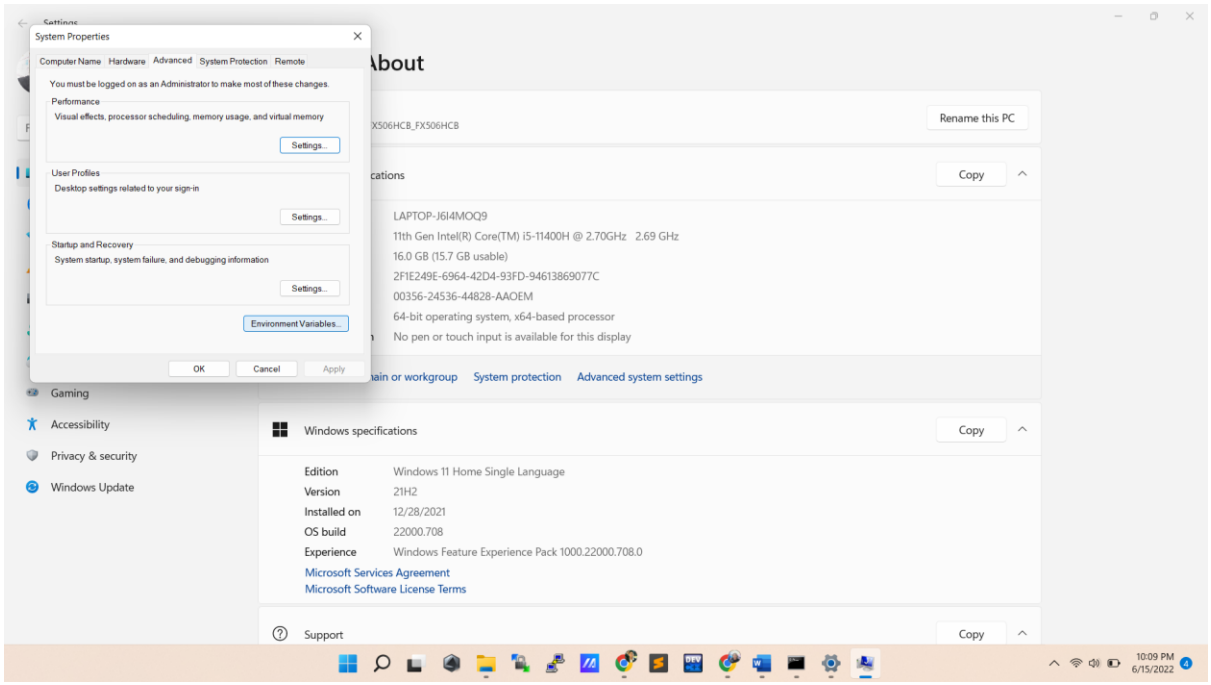


2. Find / Click “Advance system settings”.

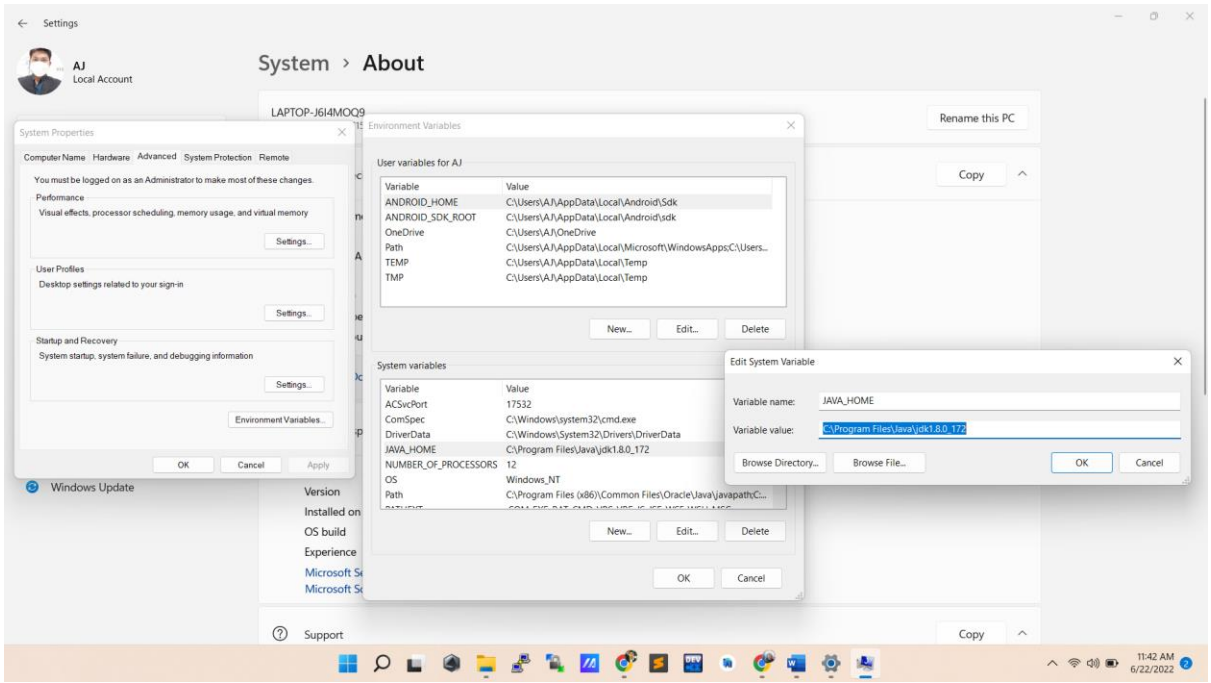


Prepared by: Alejandro V. Matute Jr., Part-Time IT Instructor
Laguna State Polytechnic University – Los Baños Campus – College of Computer Studies

3. Find / Click “Environment Variables”.

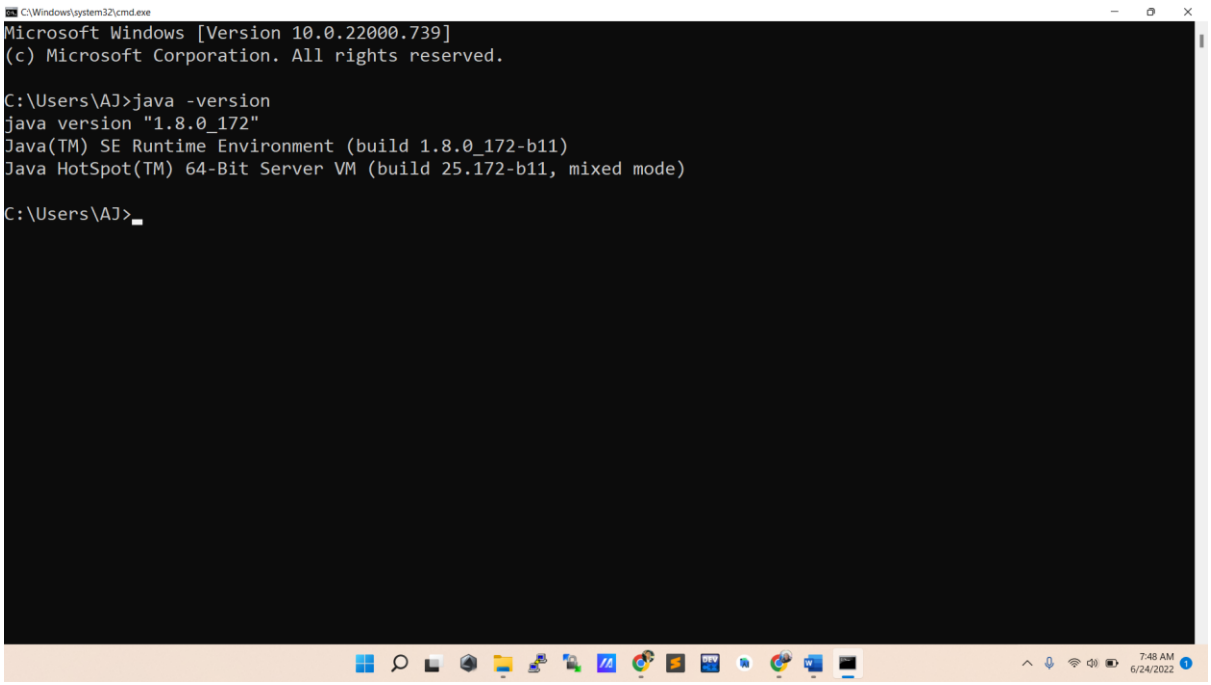


4. Set the JAVA_HOME environment variable to the location of your JDK installation. Under “System Variables”, click “New” and type “JAVA_HOME” as the environment **variable name** and the location of your JDK installation as a **value**.



5. Verify your installation

Open a console (or a Windows command prompt) and run java -version to run java and display the version, e.g.:



Gradle – Setting Environment Variables

Installing manually

Step 1. Download the latest Gradle distribution

The current Gradle release is version 7.4.2, released on 31 Mar 2022. The distribution zip file comes in two flavors:

- Binary-only
- Complete, with docs and sources

If in doubt, choose the binary-only version and browse docs and sources online.

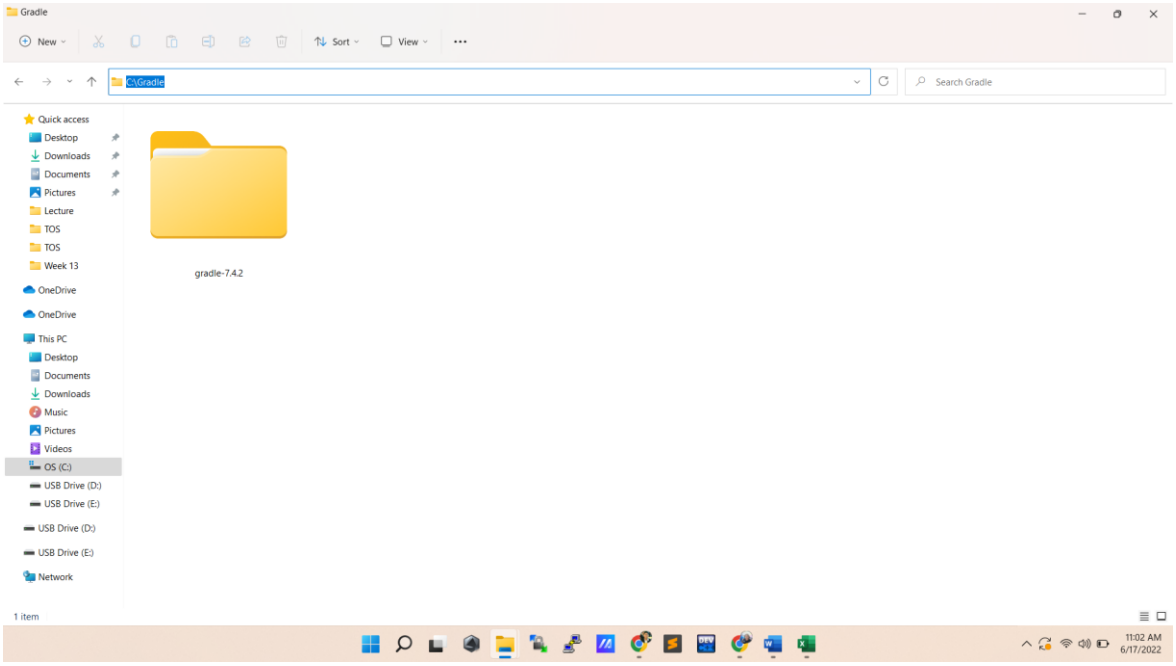
Need to work with an older version? See the releases page.

Step 2. Unpack the distribution

Microsoft Windows users

Create a new directory C:\Gradle with File Explorer.

Open a second File Explorer window and go to the directory where the Gradle distribution was downloaded. Double-click the ZIP archive to expose the content. Drag the content folder gradle-7.4.2 to your newly created C:\Gradle folder.



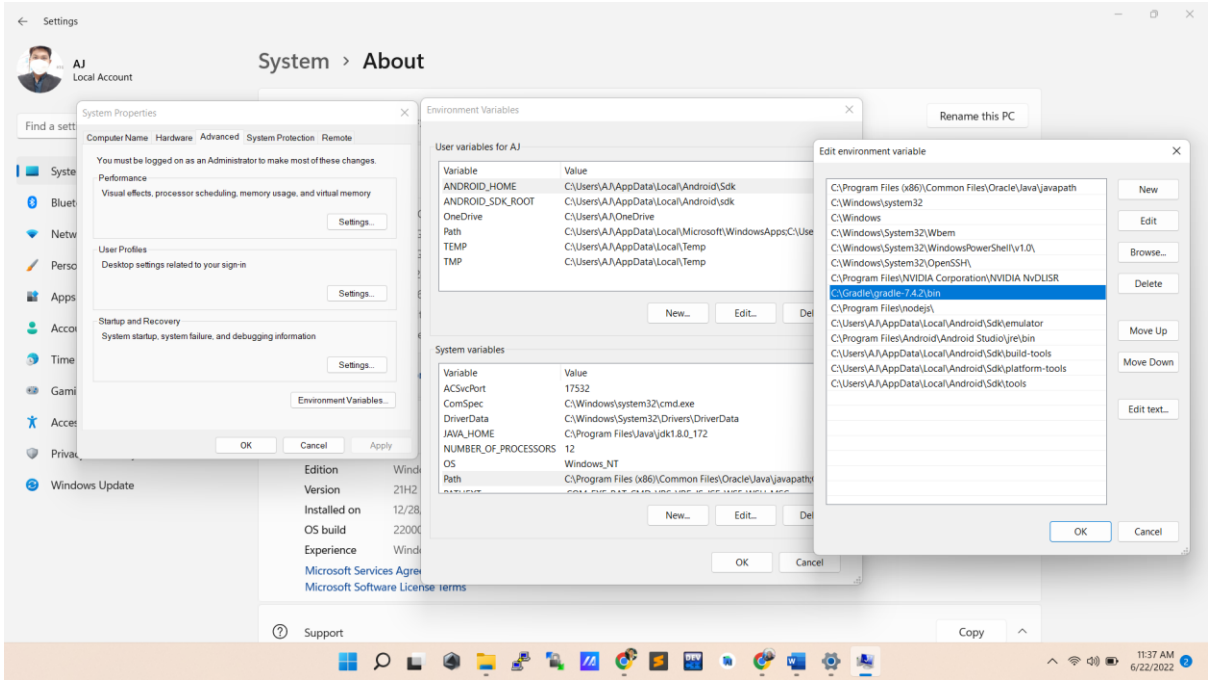
Alternatively you can unpack the Gradle distribution ZIP into C:\Gradle using an archiver tool of your choice.

Step 3. Configure your system environment

Microsoft Windows users

In File Explorer right-click on the This PC (or Computer) icon, then click Properties -> Advanced System Settings -> Environmental Variables.

Under System Variables select Path, then click Edit. Add an entry for C:\Gradle\gradle-7.4.2\bin. Click OK to save.



Step 4. Verify your installation

Open a console (or a Windows command prompt) and run `gradle -v` to run gradle and display the version, e.g.:

```
$ gradle -v

------------------------------------------------------------
Gradle 7.4.2
------------------------------------------------------------
```

For complete details about setting environment variables for gradle, click here.

Android (SDK)

Setting Environment Variables

Cordova's CLI tools require some environment variables to be set in order to function correctly. The CLI will attempt to set these variables for you, but in certain cases you may need to set them manually. The following variables should be updated:

1. Set the `ANDROID_SDK_ROOT` environment variable to the location of your Android SDK installation.
2. It is also recommended that you add the Android SDK's `cmdline-tools/latest/bin`, `emulator` and `platform-tools` directories to your `PATH`
3. For apksigner and zipalign, the Android SDK's `build-tools` must also be added to your `PATH`.

Windows

These steps may vary depending on your installed version of Windows. Close and reopen any command prompt windows after making changes to see them reflected.

1. Click on the **Start** menu in the lower-left corner of the desktop
2. In the search bar, search for **Environment Variables** and select **Edit the system Environment Variables** from the options that appear
3. In the window that appears, click the **Environment Variables** button

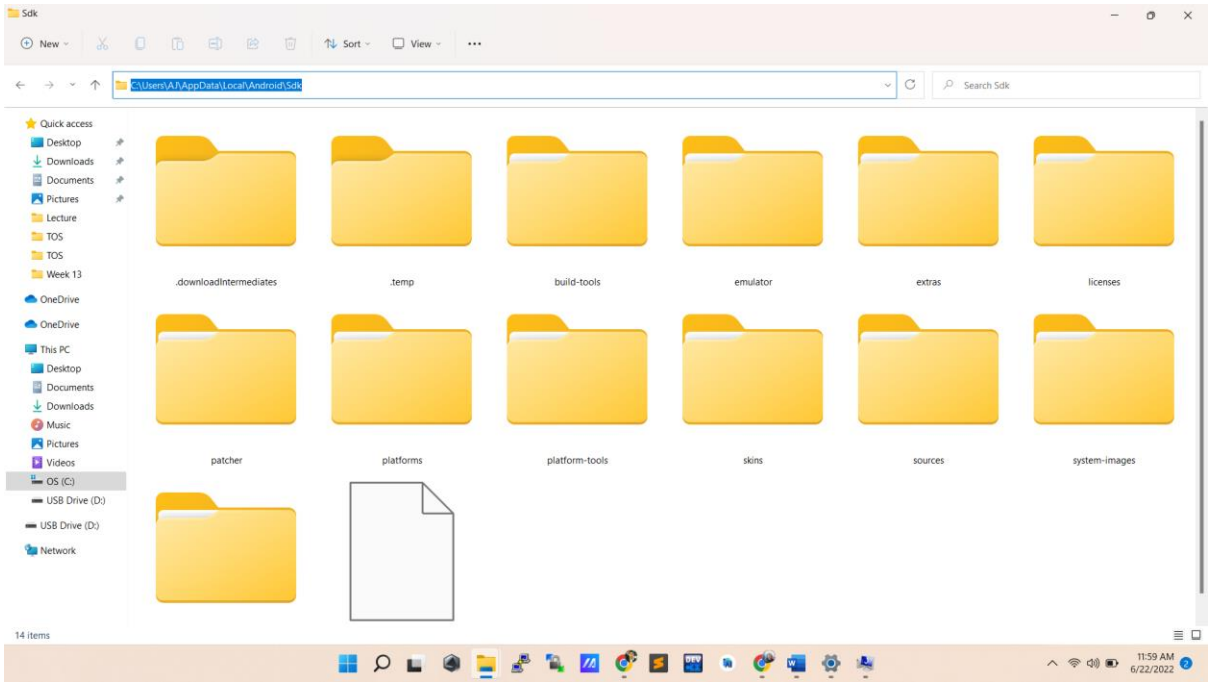
To create a new environment variable:

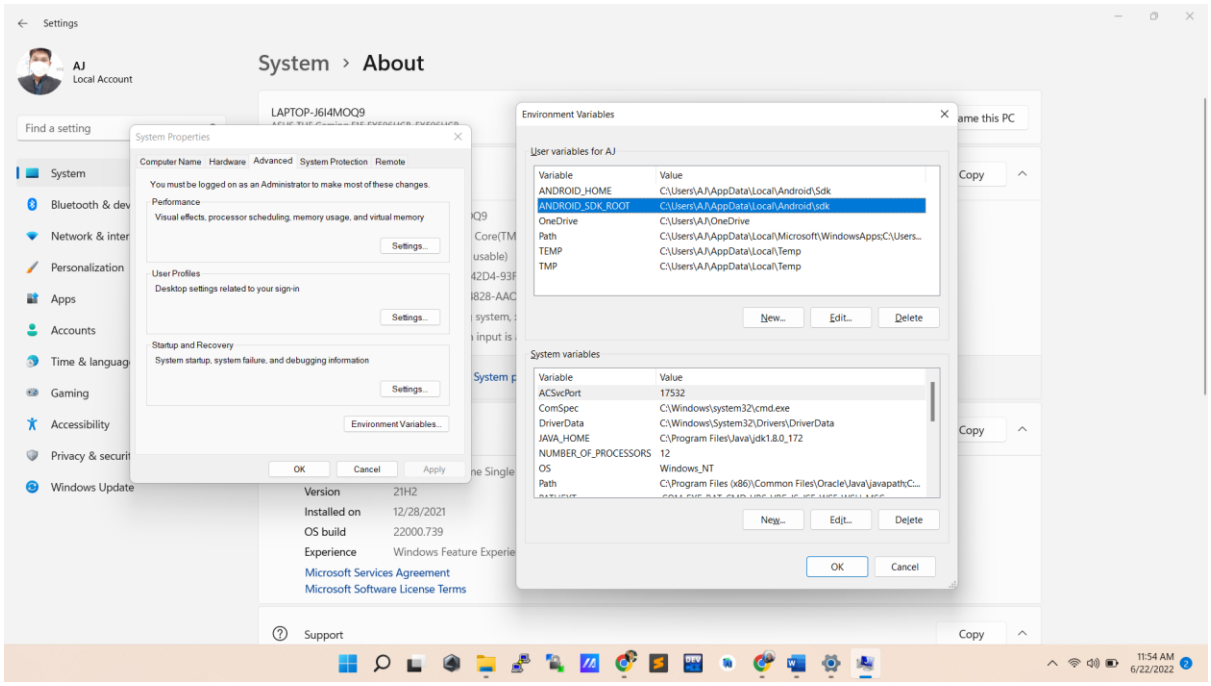
1. Click **New…** and enter the variable name and value

To set your PATH:

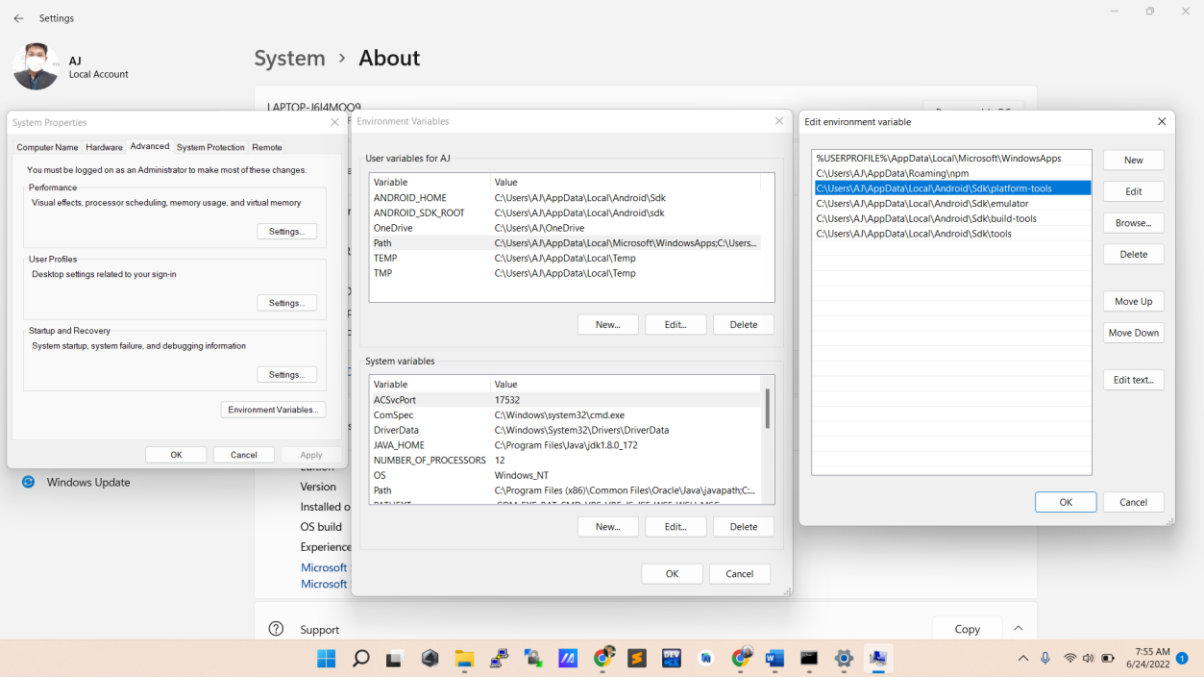
1. Select the **PATH** variable and press **Edit**.
2. Add entries for the relevant locations to the **PATH**. For example (substitute the paths with your local Android SDK installation's location):

```
C:\Users\[your user]\AppData\Local\Android\Sdk\platform-tools
C:\Users\[your user]\AppData\Local\Android\Sdk\cmdline-tools\latest\bin
C:\Users\[your user]\AppData\Local\Android\Sdk\tools\emulator
```





Prepared by: Alejandro V. Matute Jr., Part-Time IT Instructor
Laguna State Polytechnic University – Los Baños Campus – College of Computer Studies



For complete details about setting environment variables for Android, click here.

Adding SDK Packages

After installing the Android SDK, you must also install the packages for whatever API level you wish to target. It is recommended that you install the highest SDK version that your version of cordova-android supports (see Requirements and Support).

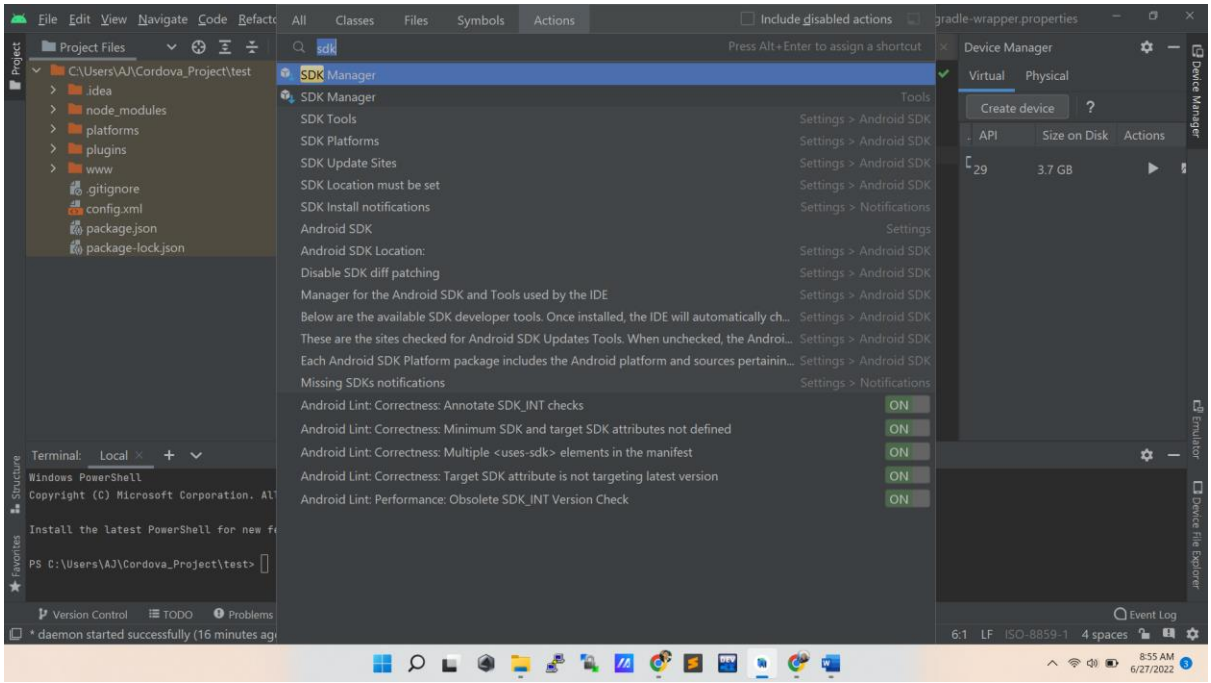
Open the Android SDK Manager (Tools > SDK Manager in Android Studio, or sdkmanager on the command line), and make sure the following are installed:

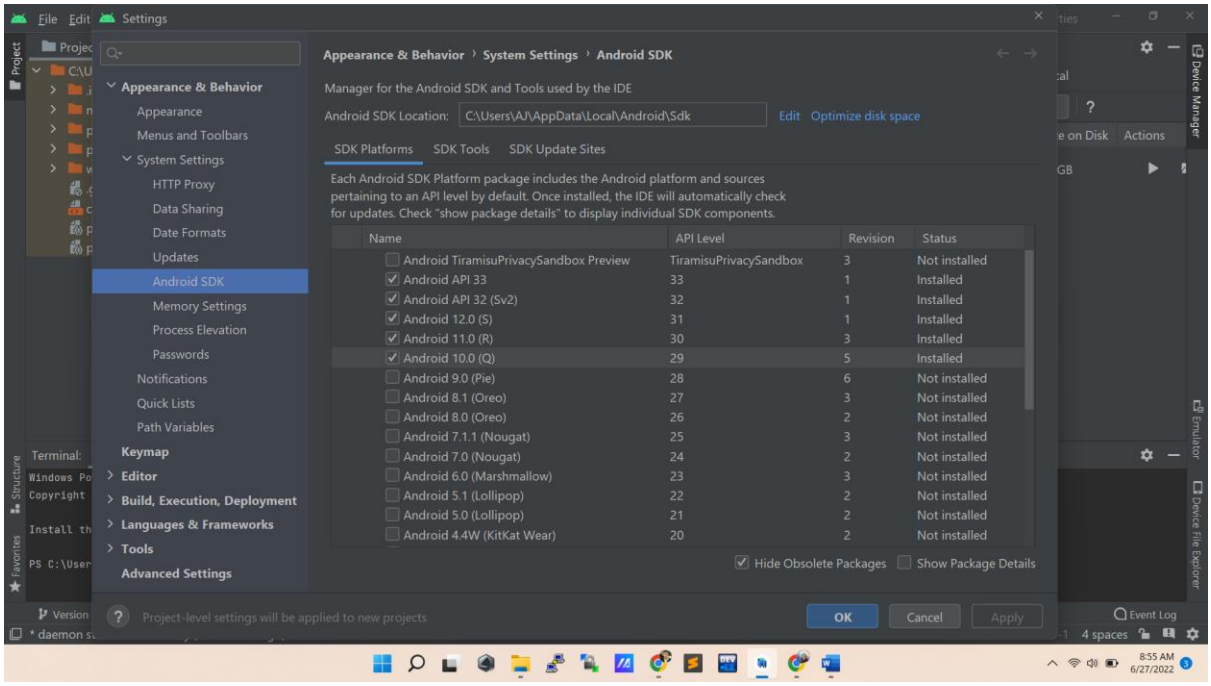
1. Android Platform SDK for your targeted version of Android
2. Android SDK build-tools version 29.0.2 or higher

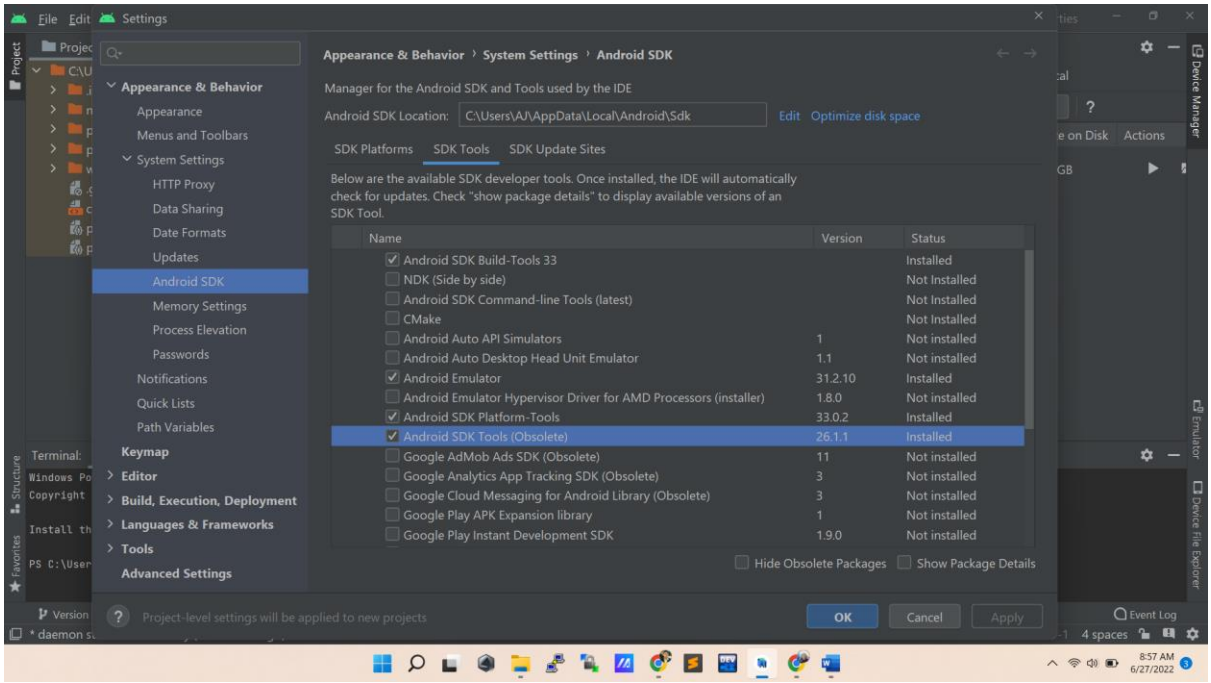
Android SDK Tools:

In Android Studio 3.6 or later, you need to manually add the old version of the Android SDK Tools. To do this:

1. Open the Android Studio **SDK Manager**
2. In the Android **SDK Tools** tab, uncheck Hide Obsolete Packages
3. Check Android SDK Tools (Obsolete)


File Edit View Navigate Code Refactc
All Classes Files Symbols Actions
Include disabled actions
sdk
Press Alt+Enter to assign a shortcut
SDK Manager
SDK Manager
SDK Tools
SDK Platforms
SDK Update Sites
SDK Location must be set
SDK Install notifications
Android SDK
Android SDK Location:
Disable SDK diff patching
Manager for the Android SDK and Tools used by the IDE
Below are the available SDK developer tools. Once installed, the IDE will automatically ch...
These are the sites checked for Android SDK Updates Tools. When unchecked, the Androi...
Each Android SDK Platform package includes the Android platform and sources pertainin...
Missing SDKs notifications
Android Lint: Correctness: Annotate SDK_INT checks
Android Lint: Correctness: Minimum SDK and target SDK attributes not defined
Android Lint: Correctness: Multiple <uses-sdk> elements in the manifest
Android Lint: Correctness: Target SDK attribute is not targeting latest version
Android Lint: Performance: Obsolete SDK_INT Version Check
Project Files
C:\Users\AJ\Cordova_Project\test
.idea
node_modules
platforms
plugins
www
.gitignore
config.xml
package.json
package-lock.json
Terminal: Local
Windows PowerShell
Copyright (C) Microsoft Corporation. Al
Install the latest PowerShell for new f
PS C:\Users\AJ\Cordova_Project\test>
Version Control
TODO
Problems
* daemon started successfully (16 minutes ag
gradle-wrapper.properties
Device Manager
Virtual
Physical
Create device
API
Size on Disk
Actions
29
3.7 GB
Event Log
6:1 LF ISO-8859-1 4 spaces
8:55 AM
6/27/2022

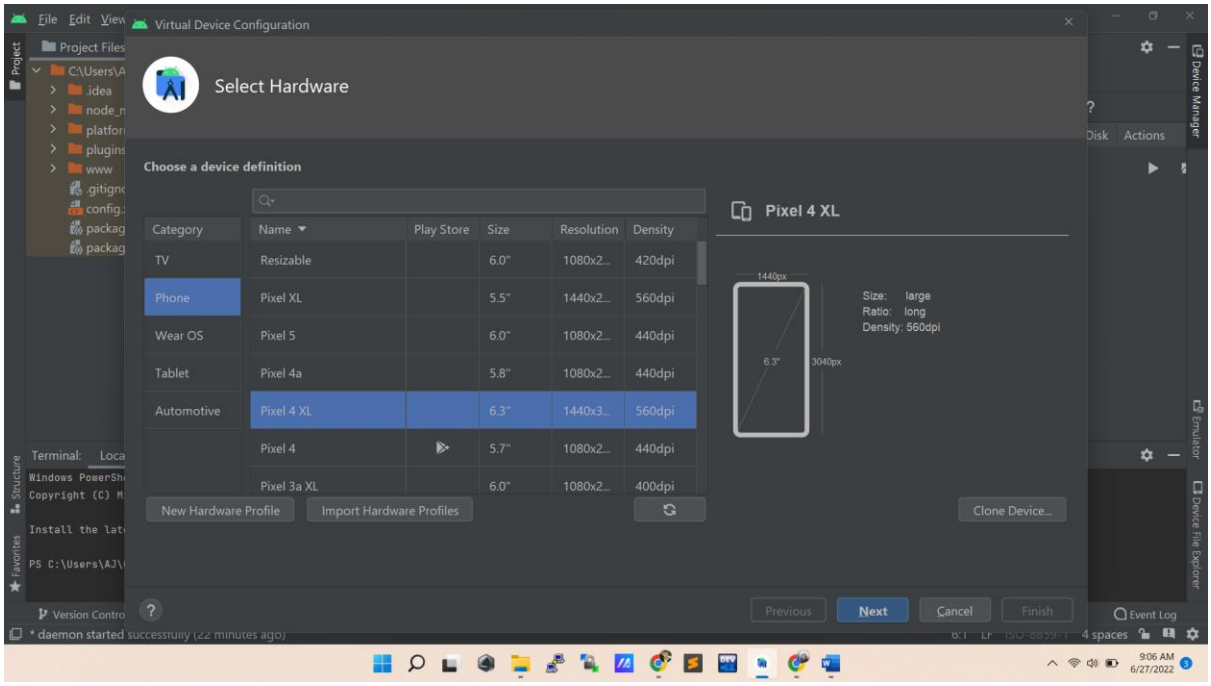

File Edit Settings
Appearance & Behavior
Appearance
Menus and Toolbars
System Settings
HTTP Proxy
Data Sharing
Date Formats
Updates
Android SDK
Memory Settings
Process Elevation
Passwords
Notifications
Quick Lists
Path Variables
Keymap
Editor
Build, Execution, Deployment
Languages & Frameworks
Tools
Advanced Settings
Appearance & Behavior › System Settings › Android SDK
Manager for the Android SDK and Tools used by the IDE
Android SDK Location: C:\Users\AJ\AppData\Local\Android\Sdk
Edit
Optimize disk space
SDK Platforms
SDK Tools
SDK Update Sites
Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, the IDE will automatically check for updates. Check "show package details" to display individual SDK components.
Name
API Level
Revision
Status
Android TiramisuPrivacySandbox Preview
TiramisuPrivacySandbox
3
Not installed
Android API 33
33
1
Installed
Android API 32 (Sv2)
32
1
Installed
Android 12.0 (S)
31
1
Installed
Android 11.0 (R)
30
3
Installed
Android 10.0 (Q)
29
5
Installed
Android 9.0 (Pie)
28
6
Not installed
Android 8.1 (Oreo)
27
3
Not installed
Android 8.0 (Oreo)
26
2
Not installed
Android 7.1.1 (Nougat)
25
3
Not installed
Android 7.0 (Nougat)
24
2
Not installed
Android 6.0 (Marshmallow)
23
3
Not installed
Android 5.1 (Lollipop)
22
2
Not installed
Android 5.0 (Lollipop)
21
2
Not installed
Android 4.4W (KitKat Wear)
20
2
Not installed
Hide Obsolete Packages
Show Package Details
Project-level settings will be applied to new projects
OK
Cancel
Apply
8:55 AM
6/27/2022



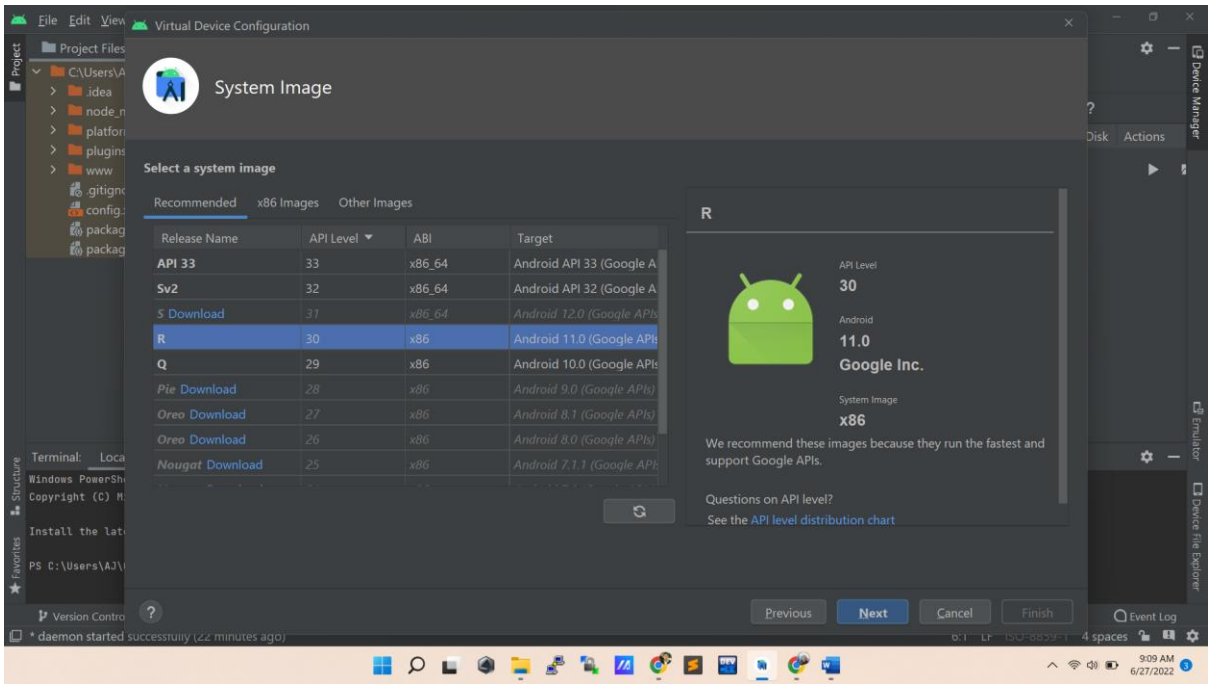
See Android's documentation on Installing SDK Packages for more details.

Virtual Device Configuration (Device Manager)

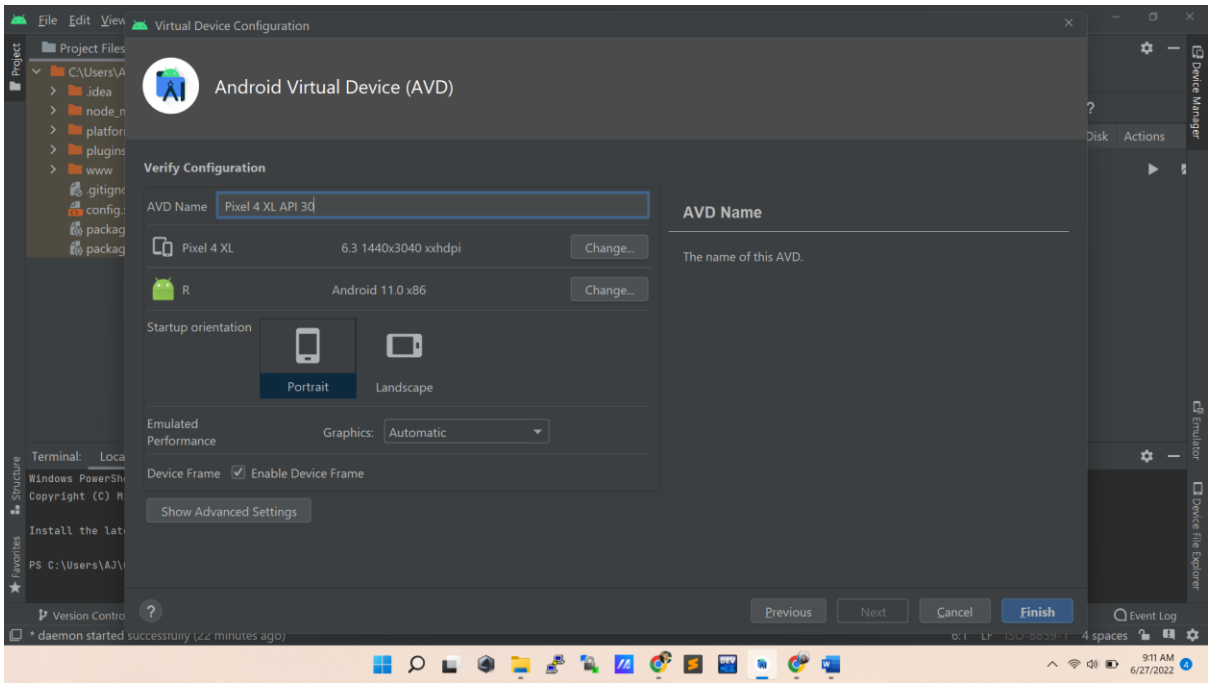
1. Choose a device definition
2. Select “Phone” as a hardware in the category including the size and resolution of your target device.



3. Select a system image. Download from recommended API Level. Example: Android 11.00.



4. Verify Configuration. Type in your chosen AVD Name and hit finish.

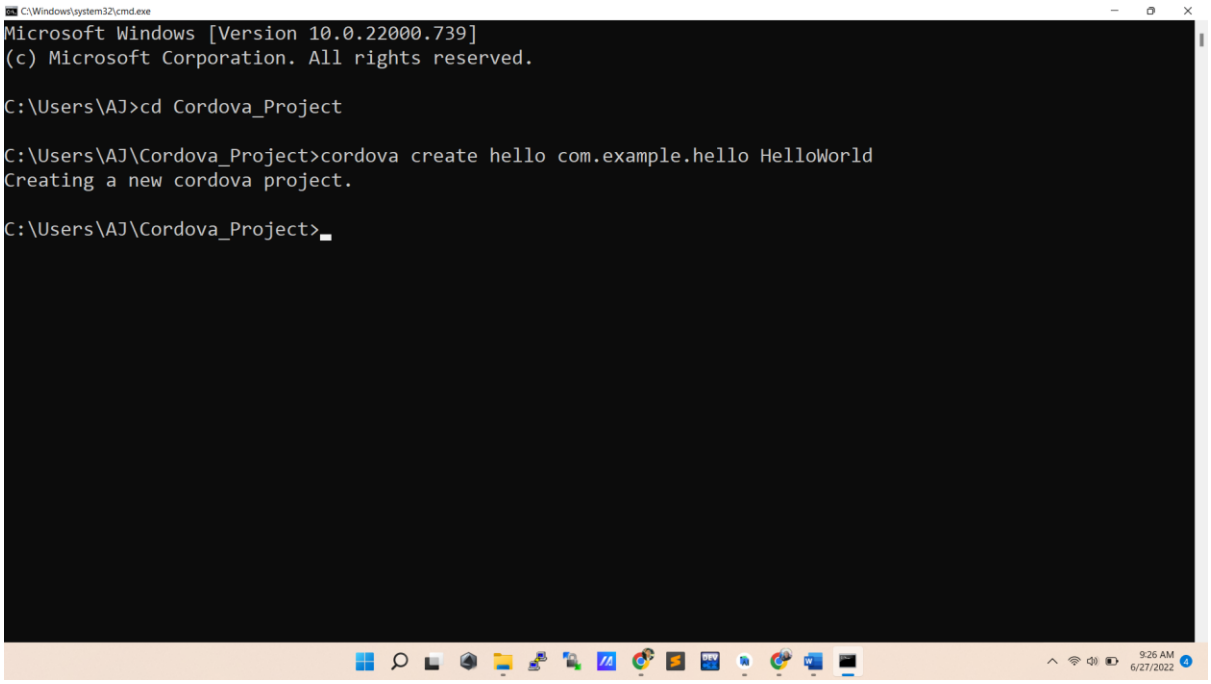


2. Create your first Cordova app

Create the App

Go to the directory where you maintain your source code, and create a cordova project:

```
$ cordova create hello com.example.hello HelloWorld
```



This creates the required directory structure for your cordova app. By default, the `cordova create` script generates a skeletal web-based application whose home page is the project's `www/index.html` file.

Add Platforms

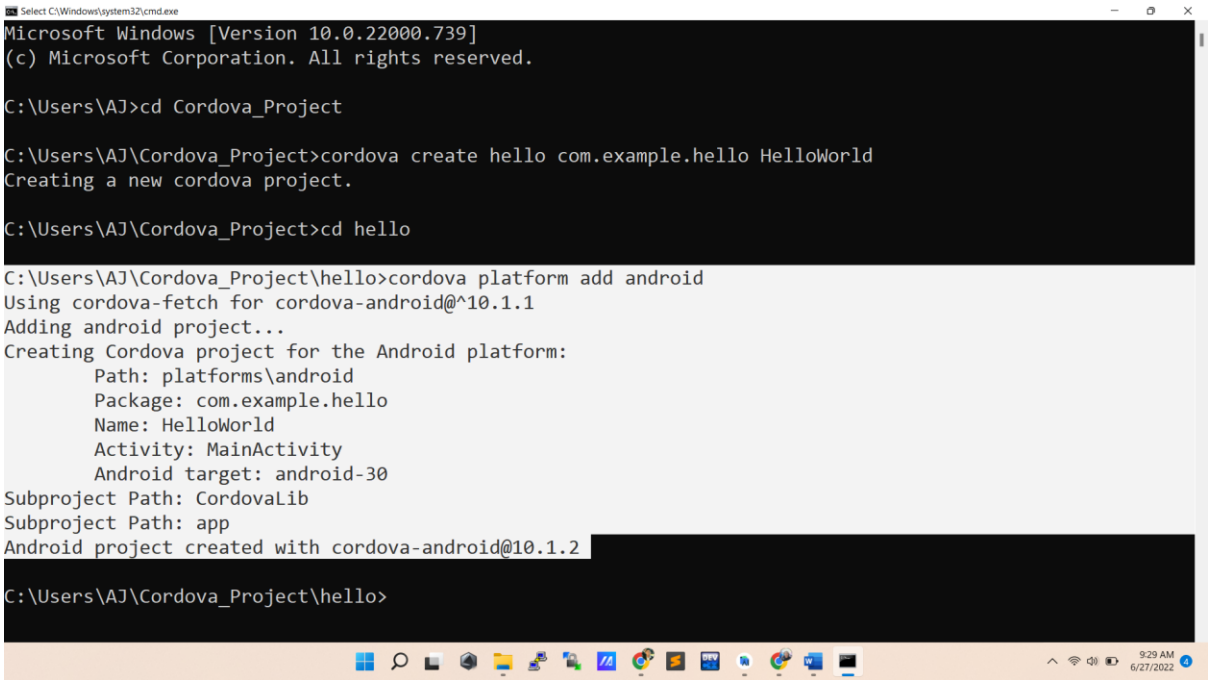
All subsequent commands need to be run within the project's directory, or any subdirectories:

```
$ cd hello
```

Add the platforms that you want to target your app. We will add the 'ios' and 'android' platform and ensure they get saved to `config.xml` and `package.json`:

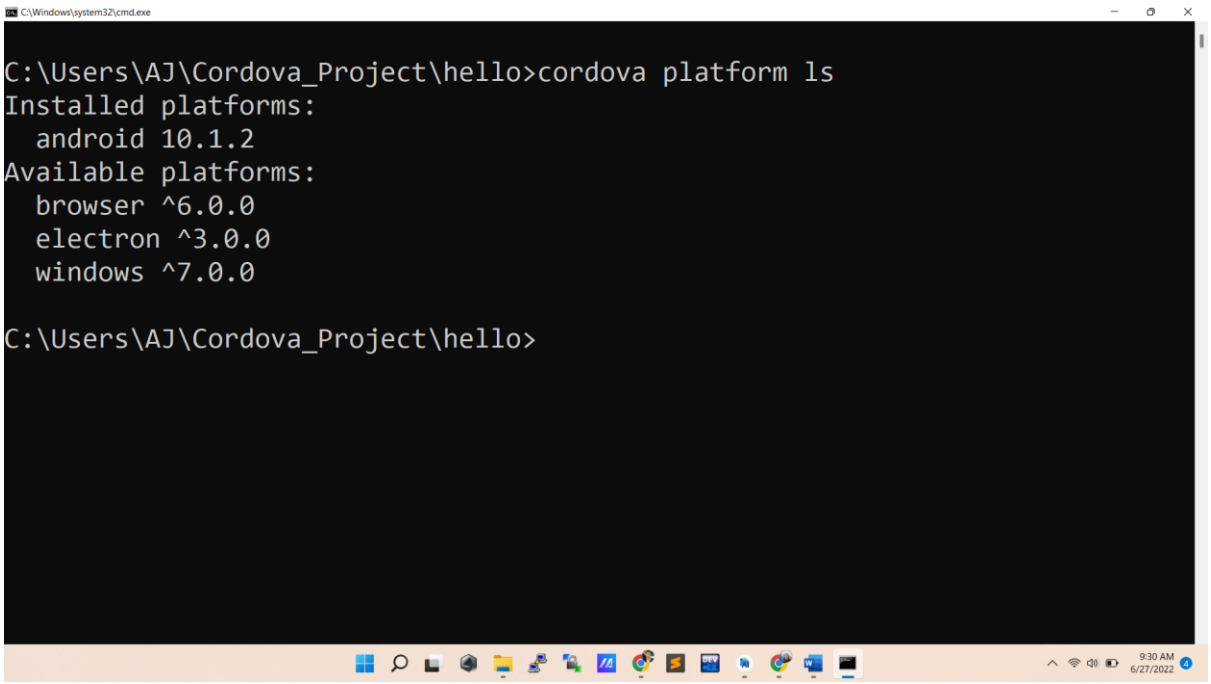
```
$ cordova platform add ios

$ cordova platform add android
```



To check your current set of platforms:

```
$ cordova platform ls
```



Running commands to add or remove platforms affects the contents of the project's *platforms* directory, where each specified platform appears as a subdirectory.

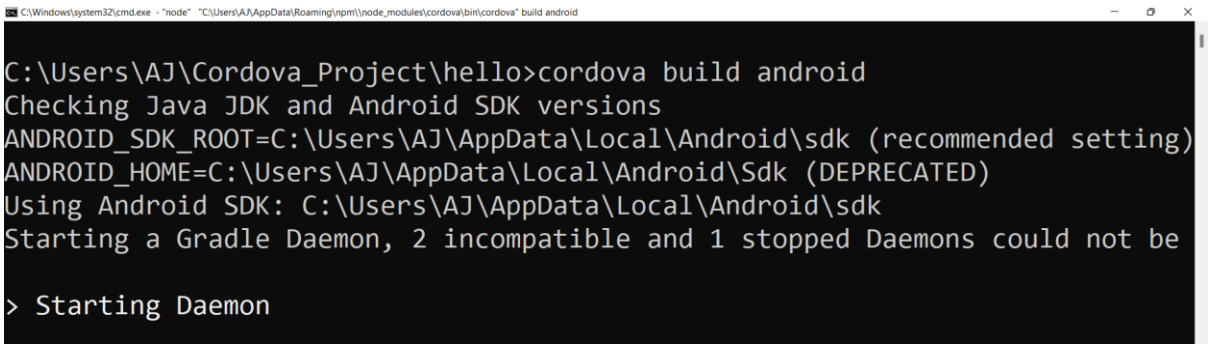
Note: When using the CLI to build your application, you should *not* edit any files in the /platforms/ directory. The files in this directory are routinely overwritten when preparing applications for building, or when plugins are re-installed.

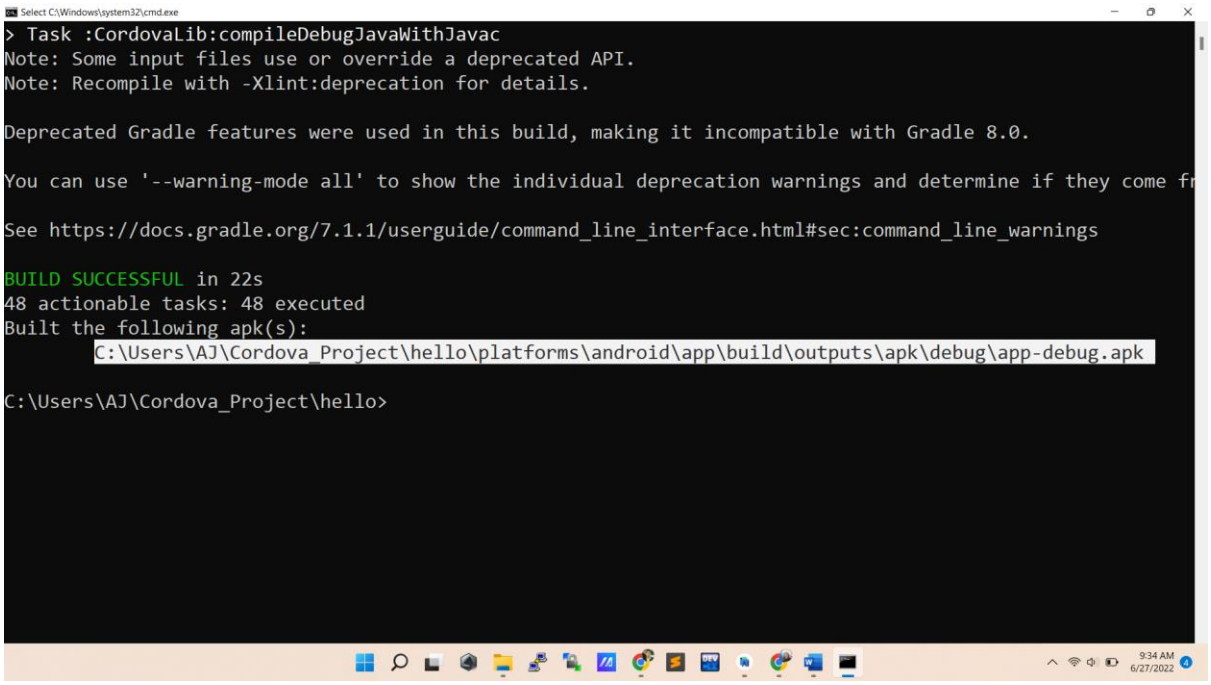
Build the App

By default, cordova create script generates a skeletal web-based application whose start page is the project's www/index.html file. Any initialization should be specified as part of the deviceready event handler defined in www/js/index.js.

Run the following command to build the project for *all* platforms:

```
$ cordova build
```





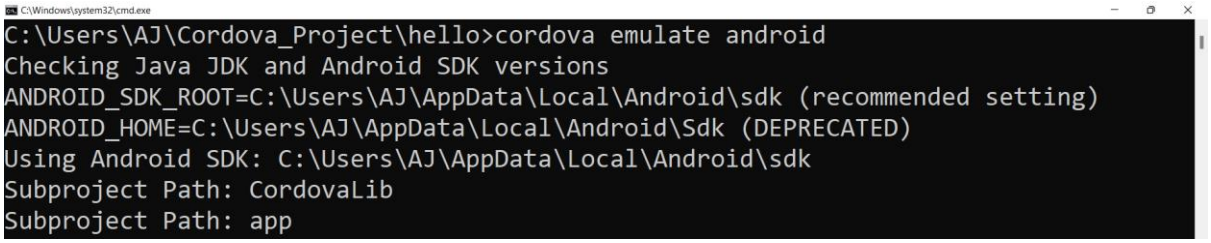
You can optionally limit the scope of each build to specific platforms - 'ios' in this case:

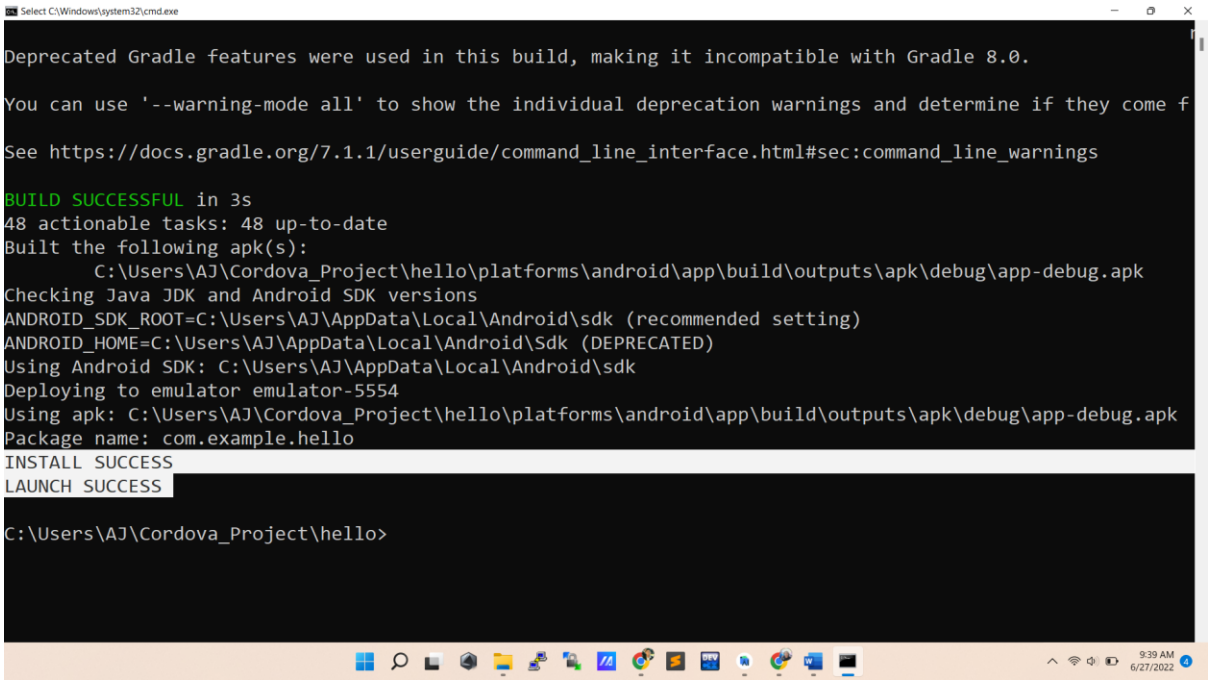
```
$ cordova build ios
```

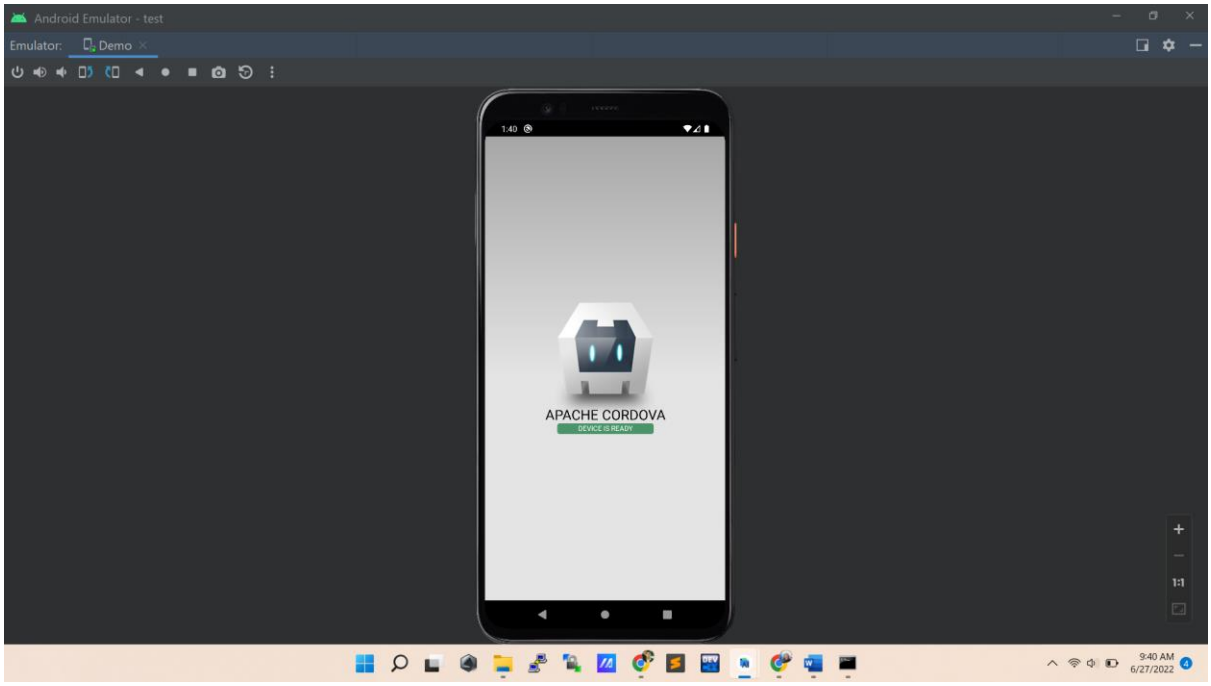
Test the App

SDKs for mobile platforms often come bundled with emulators that execute a device image, so that you can launch the app from the home screen and see how it interacts with many platform features. Run a command such as the following to rebuild the app and view it within a specific platform's emulator:

```
$ cordova emulate android
```





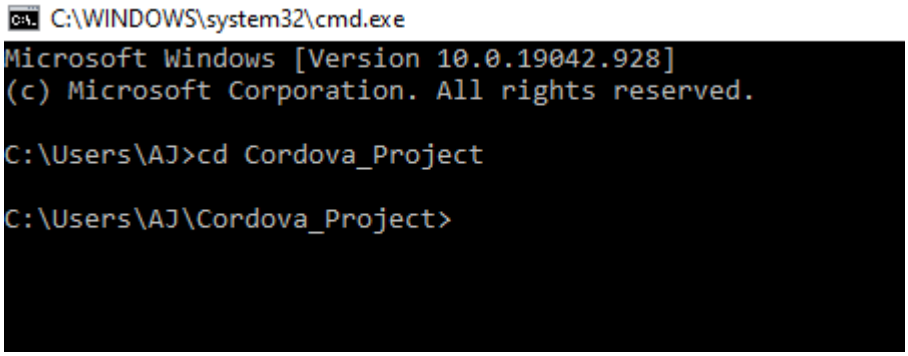


See Android's documentation on [Create your first Cordova app] for more details.

3. Converting Website to Mobile App

1. Creating a cordova project.

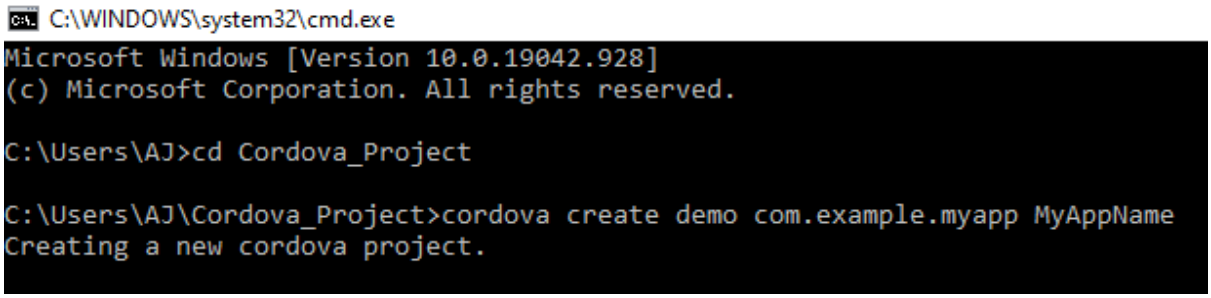
Open the command prompt (`command + r`) or any CLI tool and go the path where you want to create new cordova project.



Type the following command:

```
cordova create project_directory_name package_name app_title
```

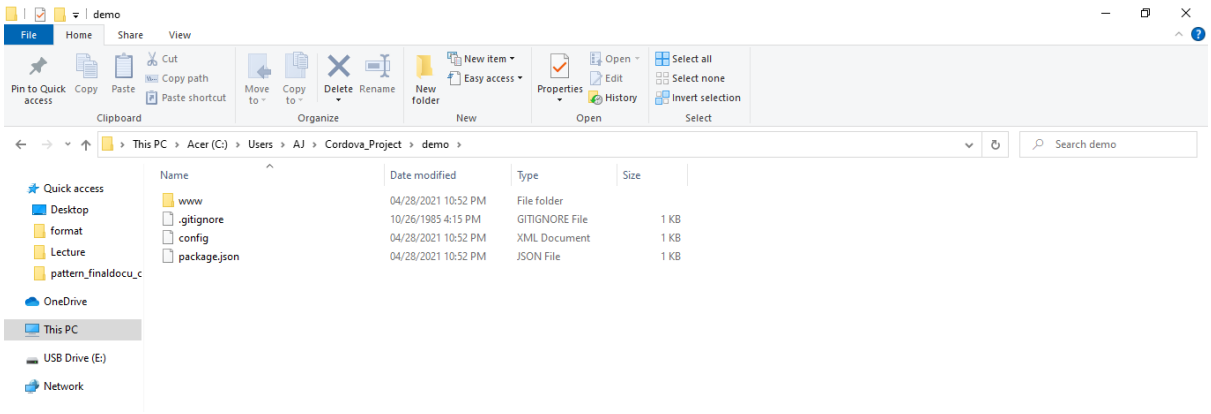
Example:




```
cordova create directory-name com.example.myapp MyAppName
```

The create command has three parameters: the first specifies the **folder** where the solution will be created, the second is a **reverse domain-style identifier** which must be unique, and the last one is the **application display name**.

Once it's done, it will create a directory structure like this:



2. Add platform to project.

Once you create project add platform to project using the following command: (**Note:** Before you add a platform, make sure that you are in your project directory.)

```
cordova platform add <platform>
```

Example:

C:\WINDOWS\system32\cmd.exe

```
C:\Users\AJ\Cordova_Project\demo>cordova platform add android
Using cordova-fetch for cordova-android@^9.0.0
Adding android project...
Creating Cordova project for the Android platform:
        Path: platforms\android
        Package: com.example.myapp
        Name: MyAppName
        Activity: MainActivity
        Android target: android-29
Subproject Path: CordovaLib
Subproject Path: app
Android project created with cordova-android@9.1.0
Discovered plugin "cordova-plugin-whitelist". Adding it to the project
Installing "cordova-plugin-whitelist" for android
Adding cordova-plugin-whitelist to package.json
```

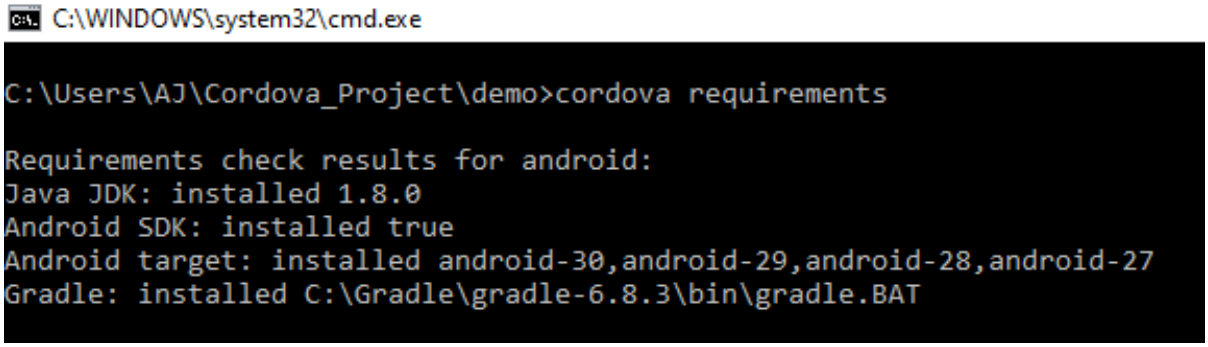
Example: (*Select the platform where you want to build your app.)*

```
cordova platform add ios  OR  cordova platform add android
```

Type following command to check if all requirements are configured and installed properly:

```
cordova requirements
```

Example:



3. Install necessary plug-ins.

Install any plugins you may need using the following command: (Example: Dialog Box, Check Network Connectivity, and Web View Functionality.)

Sample Plugin:

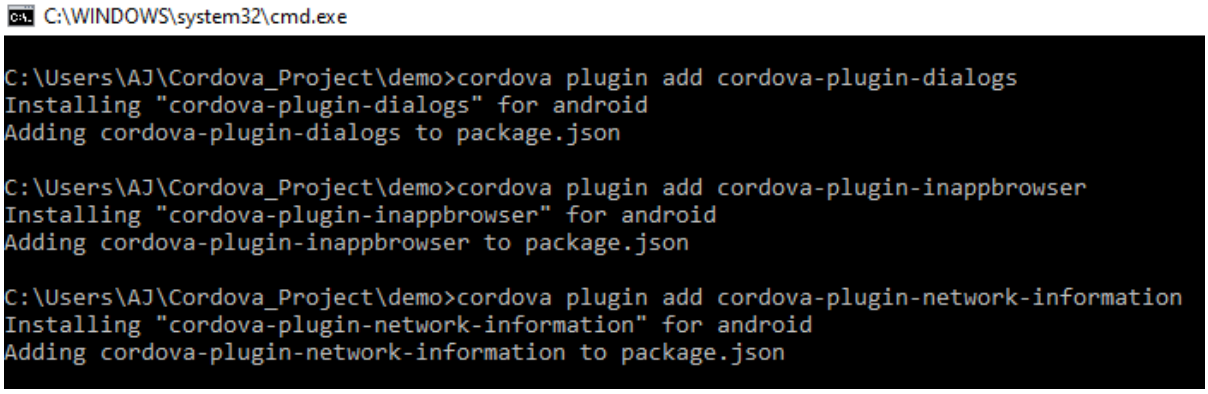
cordova-plugin-inappbrowser

cordova-plugin-dialogs

cordova-plugin-network-information

```
Cordova plugin add plug-in-name
```

Example:



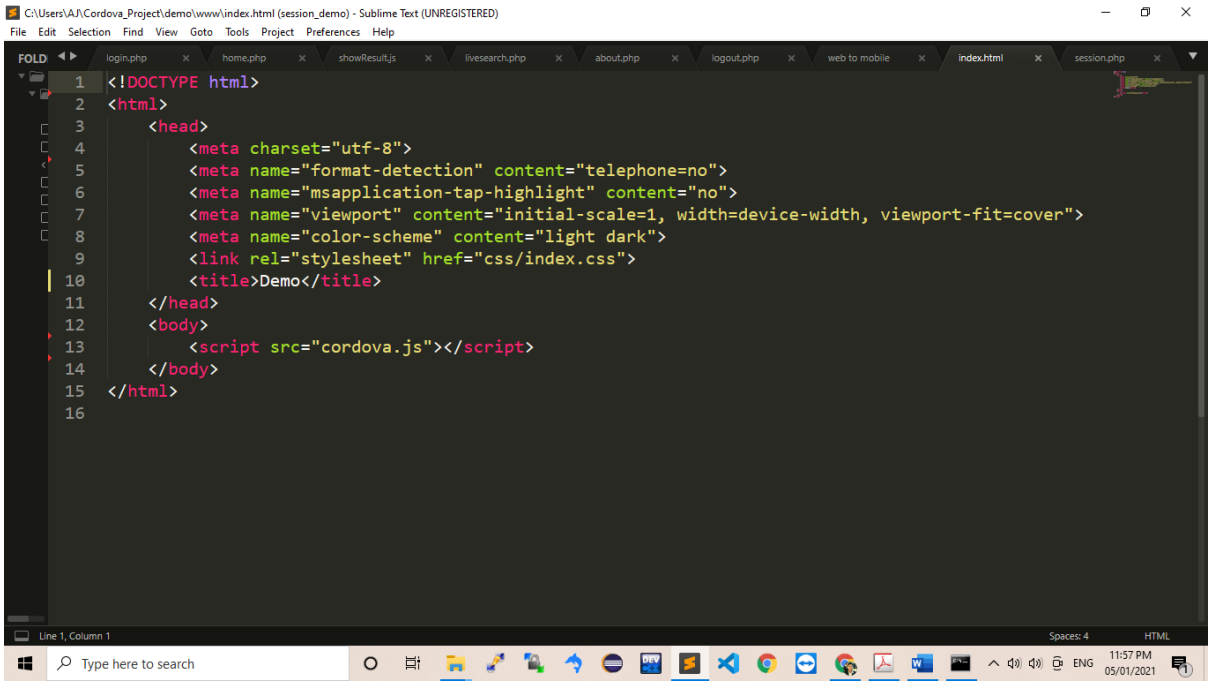
Click this link for complete cordova plugin information/commands.

www: (required) this directory contains the html, javascript and other assets that should be included in your application. This directory should contain a file called index.html that is the HTML root of your application.

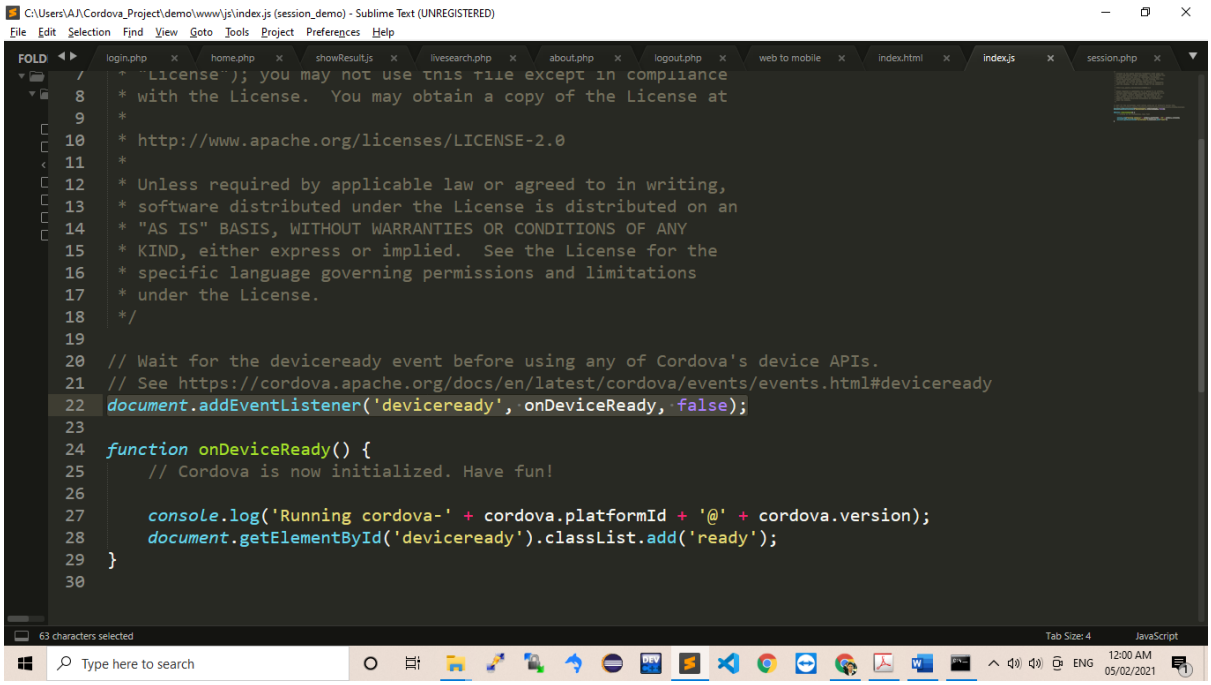
4. Build the App.

By default, cordova create script generates a skeletal web-based application whose start page is the project's **www/index.html file**. Any initialization should be specified as part of the deviceready event handler defined in **www/js/index.js**

1. From your project directory, go to */www* and open the *index.html file*. Remove unnecessary code meta tag/commentary/license information and retain *cordova.js* file.



2. Go to *www/js* directory and open the index.js file. Look and copy device ready event.



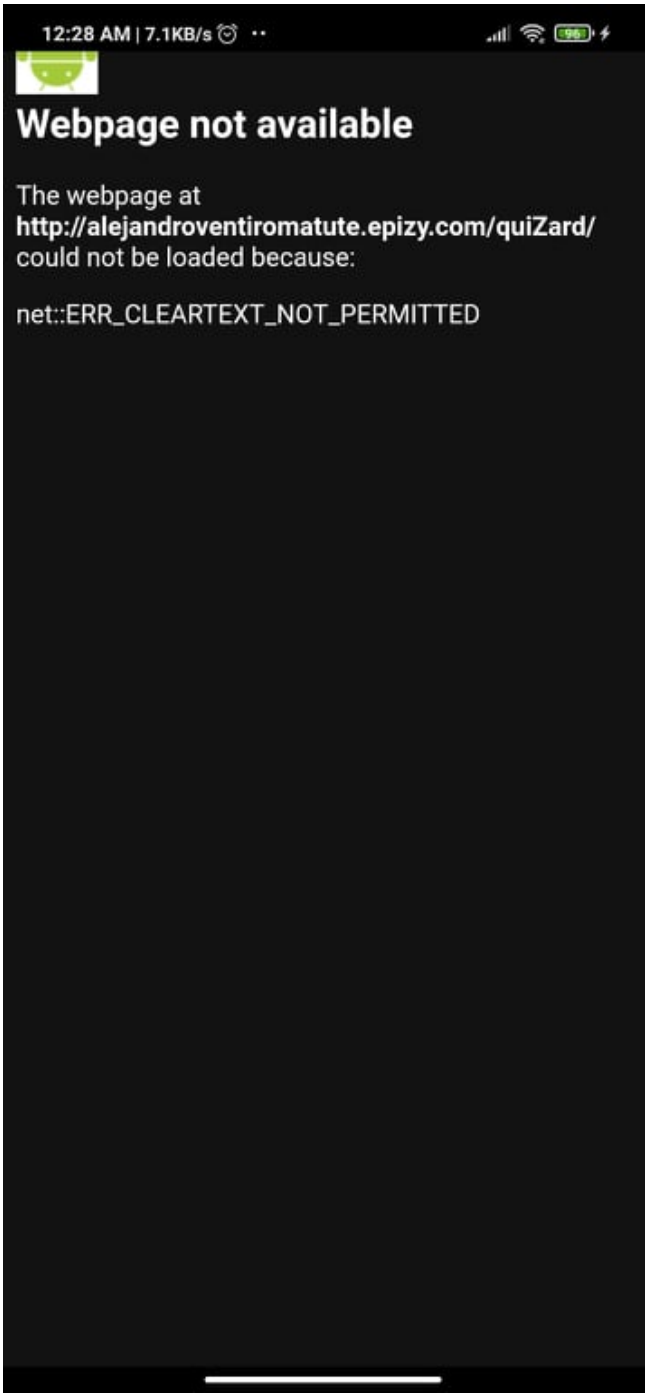
3. Go back to *www/index.html* file and create an External JavaScript tag below the existing <script> tag. Inside of the new created External JavaScript tag, and paste the device ready event. Do the following code below: (Set the *url* of your website as the first parameter of the *cordova.InAppBrowser.open()* function.)

Index.html

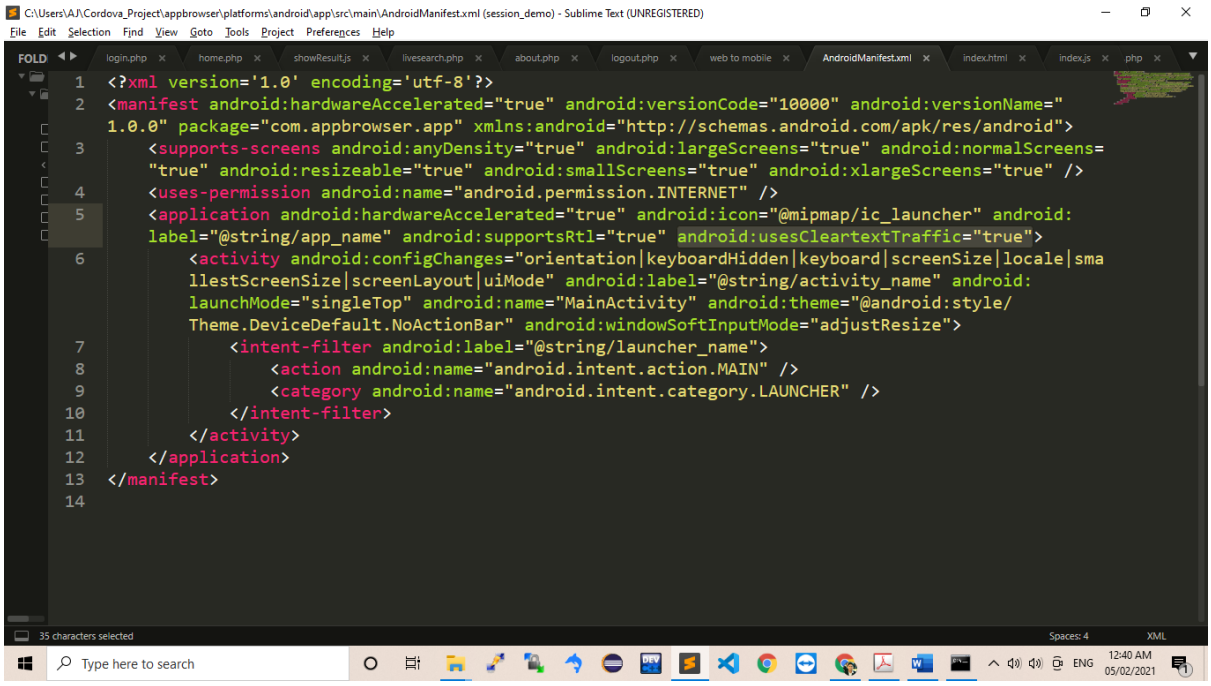
```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <meta name="format-detection" content="telephone=no">
        <meta name="msapplication-tap-highlight" content="no">
        <meta name="viewport" content="width=device-width, initial-scale=1.0,
minimum-scale=1.0, maximum-scale=1.0, user-scalable=no, viewport-fit=cover"/>
        <meta name="color-scheme" content="light dark">
        <link rel="stylesheet" href="css/index.css">
        <title>Demo</title>
    </head>
```

```
    <body>
        <script src="cordova.js"></script>
        <script type="text/javascript">
            document.addEventListener('deviceready', function(){
            cordova.InAppBrowser.open('http://alejandroventiromatute.epizy.com/e-learning/', '_blank', 'location=no, zoom=no, toolbar=no');
            });
        </script>
    </body>
</html>
```

4. From your project directory, go to platforms/android/app/src/main and look for **AndroidManifest**.**xml** file. Open the file and add the following code after the **android:supportsRtl="true"** like the example below, which will resolve this error '*net::ERR_NAME_NOT_RESOLVED*':



```
android:usesCleartextTraffic="true"
```



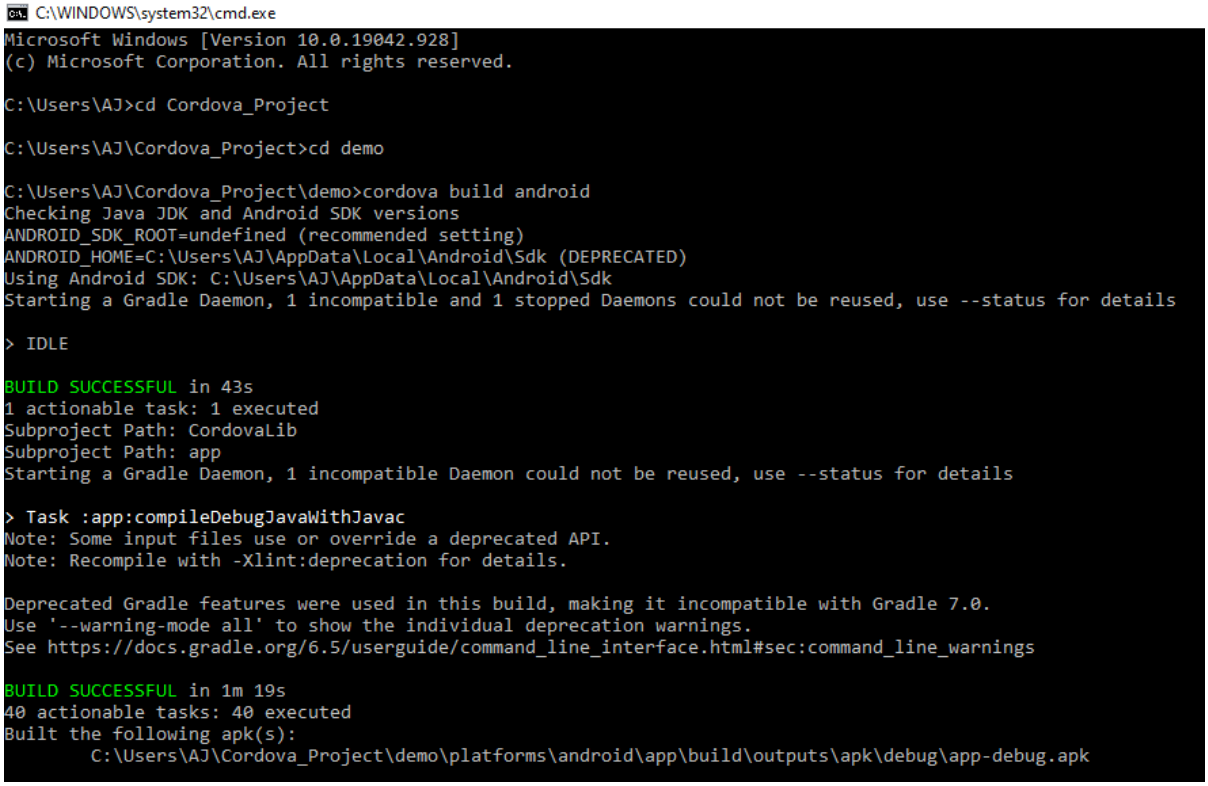
Build Cordova on android/ios platform using the following command:

```
cordova build android
```

OR

```
cordova build ios
```

Example: (You should something like this in your CLI)



Go to the directory to see and test your apk file.

Example:
C:\Users\AJ\Cordova_Project\demo\platforms\android\app\build\outputs\apk\debug\app-debug.apk

5. Add App Icon

Provide any icon you want for your app. Resize your app icon to all needed resolutions. You may use any online app resizer such as resizeappicon.com.

From your project directory, create */res* folder to store the values for the resources (images, etc.).

Example:

C:\Users\AJ\Cordova_Project\demo\res\images

Copy and Paste the code below into the config.xml file. (The config.xml file is located in the project directory.)

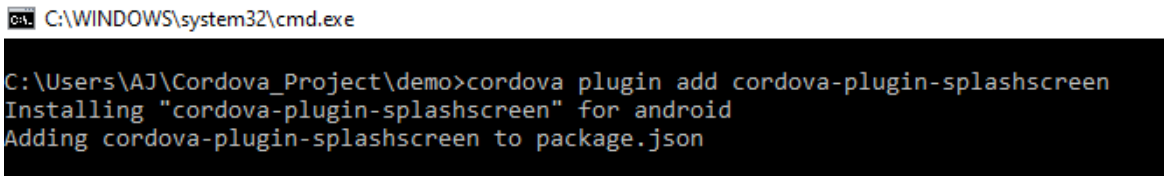
```
<platform name="android">
    <allow-intent href="market:*" />
    <icon density="ldpi" src="res/images/ldpi.png" />
    <icon density="mdpi" src="res/images/mdpi.png" />
    <icon density="hdpi" src="res/images/hdpi.png" />
    <icon density="xhdpi" src="res/images/xhdpi.png" />
    <icon density="xxhdpi" src="res/images/xxhdpi.png" />
    <icon density="xxxhdpi" src="res/images/xxxhdpi.png" />
</platform>
```

6. Add splashscreen

To add splashscreen, install cordova splashscreen plugin.

```
cordova plugin add cordova-plugin-splashscreen
```

Example:



Platform Splash Screen Image Configuration

Directory structure:

```
projectRoot
    node_modules

    platforms

        android
                app
                        src
                                main
                                        res
    plugins
    res
    www
```

Splash Screens for the Android Platform

Place 9-patch image files in the Android project's directories.

The size for each should be:

- xlarge (xhdpi): at least 960 × 720
- large (hdpi): at least 640 × 480
- medium (mdpi): at least 470 × 320
- small (ldpi): at least 426 × 320

Example: C:\Users\AJ\Cordova_Project\demo\platforms\android\app\src\main\res

Default Cordova Screen

Location: YourProjectDirectory\platforms\android\app\src\main\res\drawable-port-hdpi

Copy and Paste the code below into the config.xml file. (The config.xml file is located in the project directory.)

```
<platform name="android">
    <preference name="SplashScreen" value="elearning" />
    <preference name="SplashScreenDelay" value="10000" />
    <preference name="ShowSplashScreenSpinner" value="true" />
    <preference name="SplashScreenSpinnerColor" value="blue" />
</platform>
```

For complete Icons and Splash Screens reference, click here.

4. Running Cordova Application

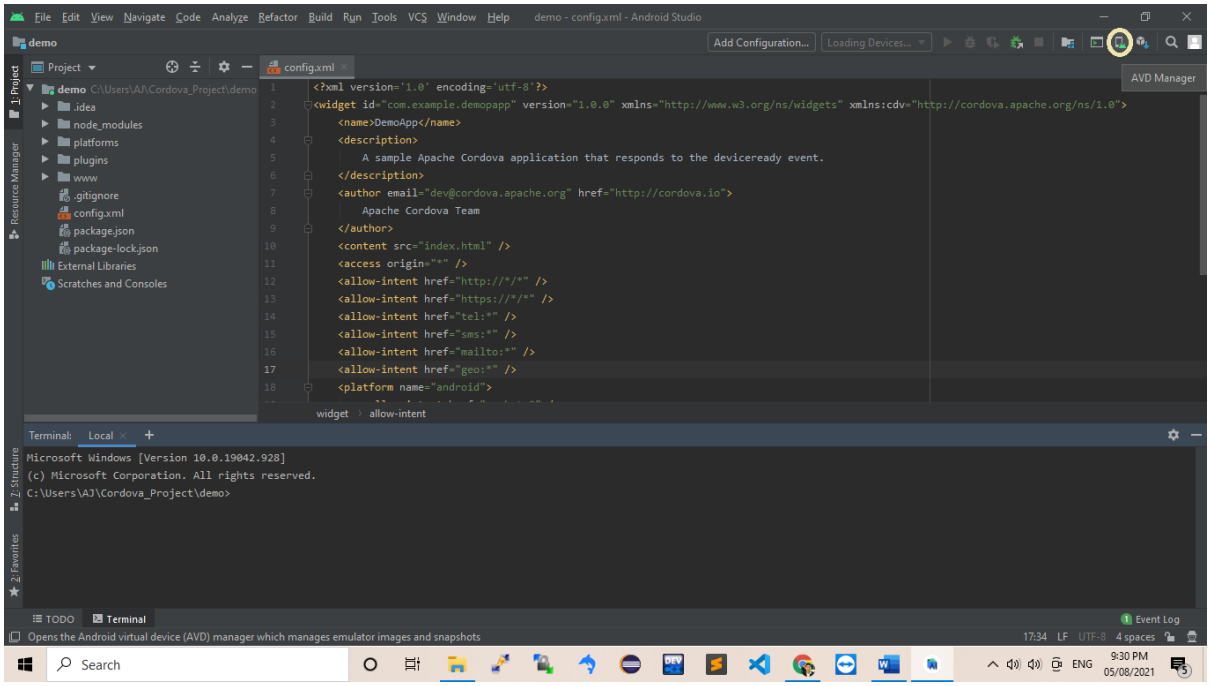
Option 1: Using Android Emulator

Option 2: Running on Actual Device

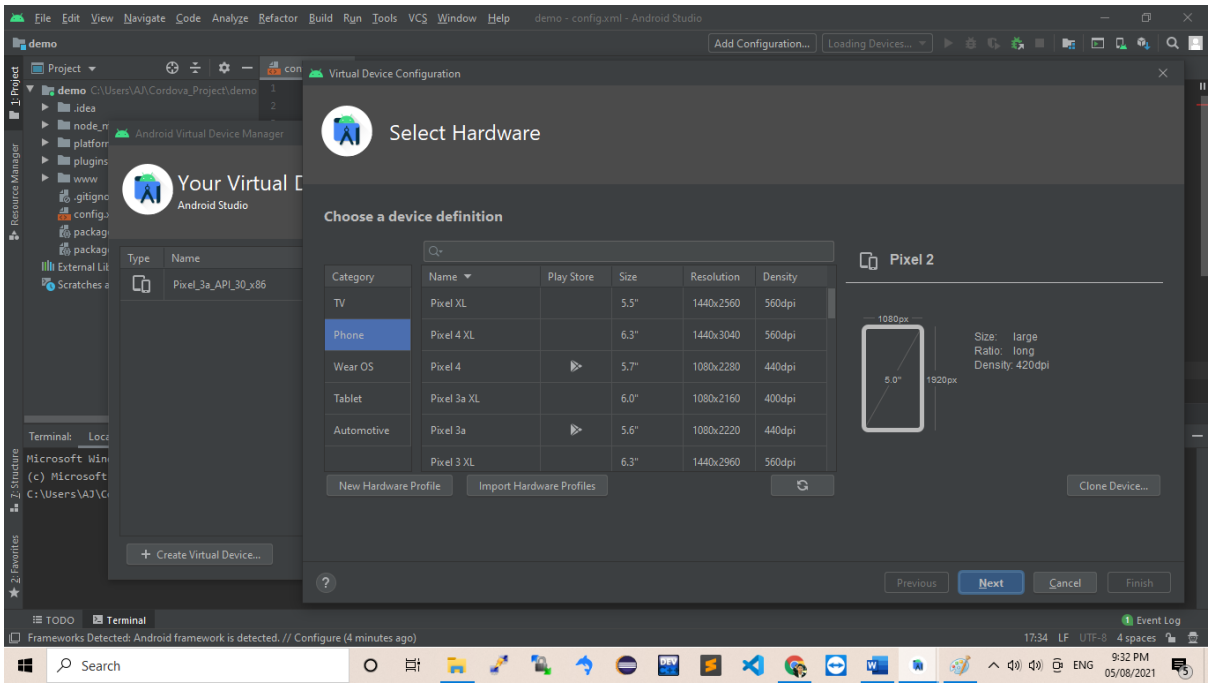
Option 1: Using Android Emulator

Step 1: Run Android Studio

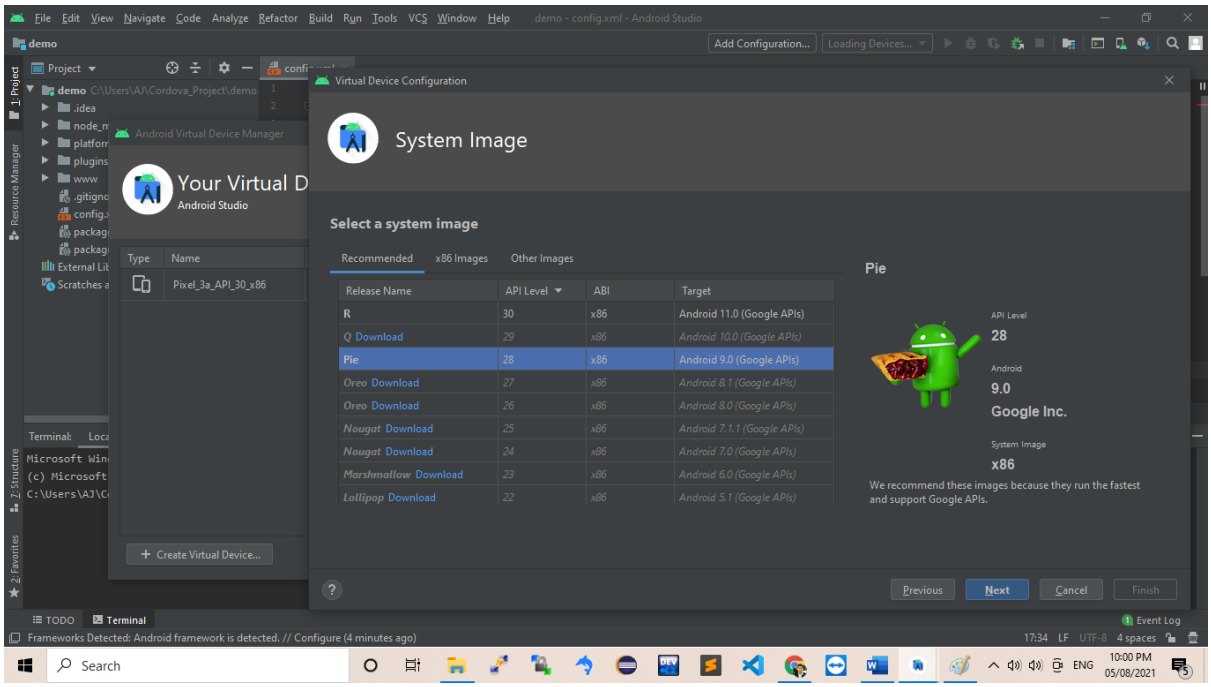
Step 2: Locate Android Virtual Device (AVD) Manager



Step 3: Create Virtual Device based on your own preference. Choose a device definition.

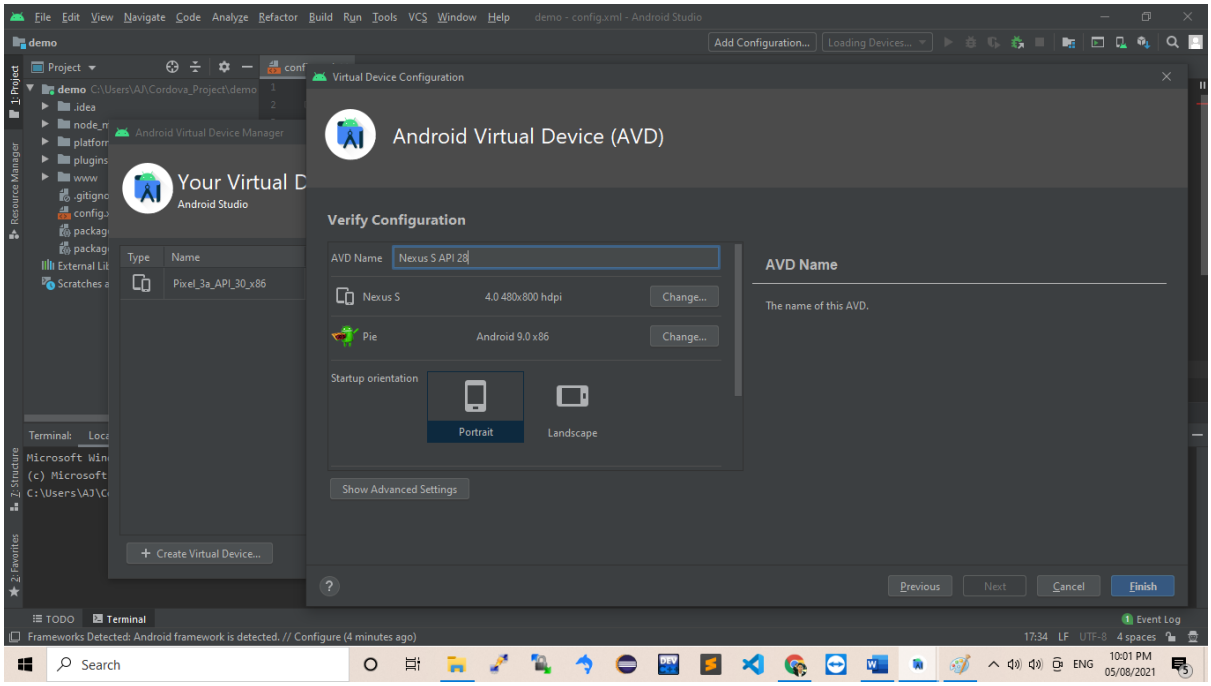


Step 4: Select a system image.

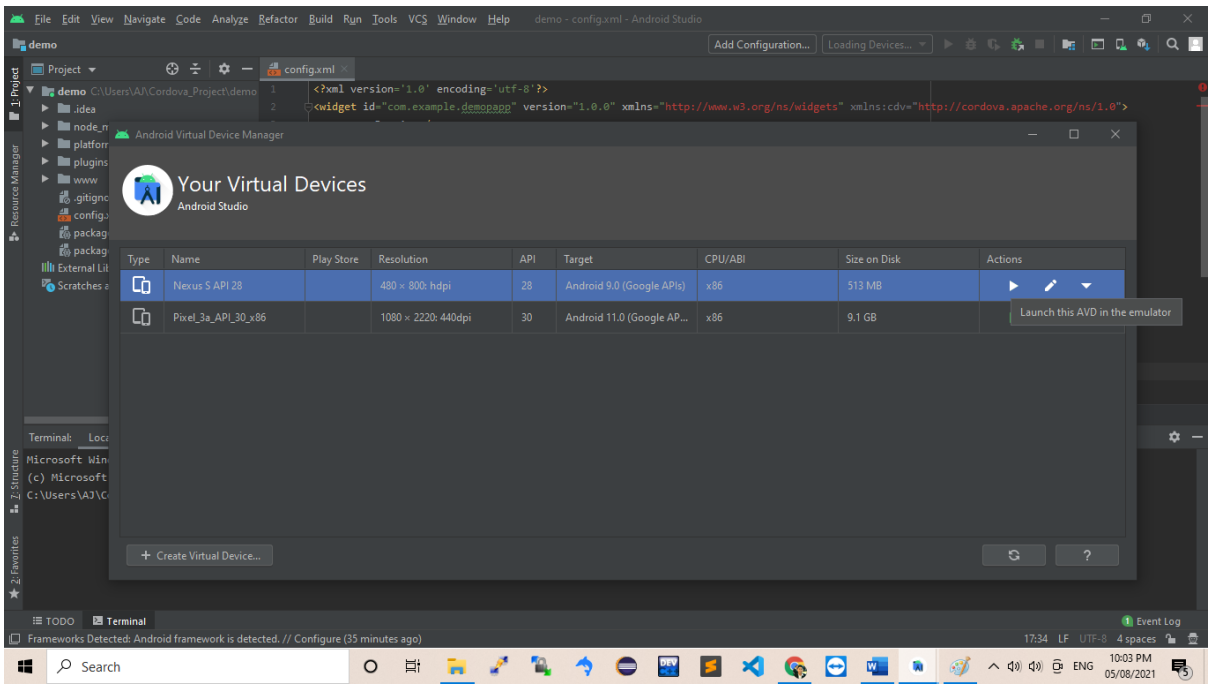


Prepared by: Alejandro V. Matute Jr., Part-Time IT Instructor
Laguna State Polytechnic University – Los Baños Campus – College of Computer Studies

Step 5: Verify Configuration.



Step 6: Select and Lunch the AVD in the Emulator



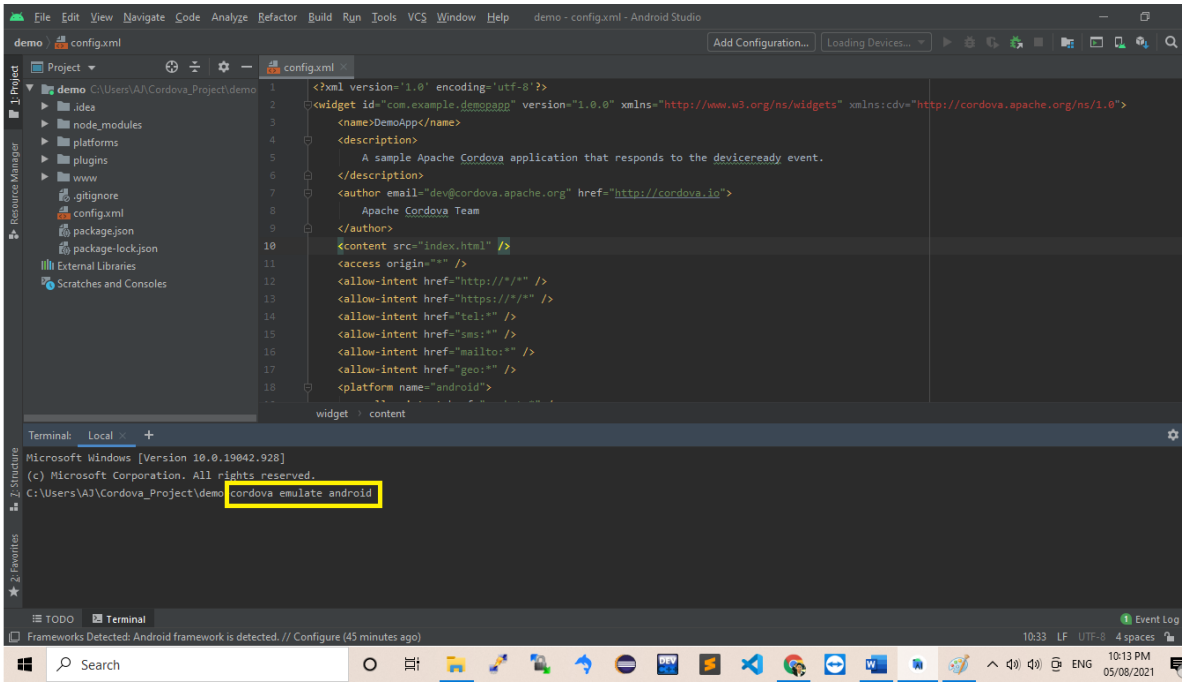
Prepared by: Alejandro V. Matute Jr., Part-Time IT Instructor
Laguna State Polytechnic University – Los Baños Campus – College of Computer Studies

Step 7: Deploy to Emulator

Use the cordova CLI utility to deploy the application to the emulator from the command line:

```
cordova emulate android
```

Example:



Option 2: Running on Actual Device

Step 1: Connect your mobile device to your machine (pc or laptop).

Step 2: In your Mobile Device, go to ‘Settings’, Find and Tap **‘About Phone’**.

Step 3: Look for ‘Build Number’ or ‘MIUI version’ and Tap it 7 times until it says, “**You are now a Developer!**”.

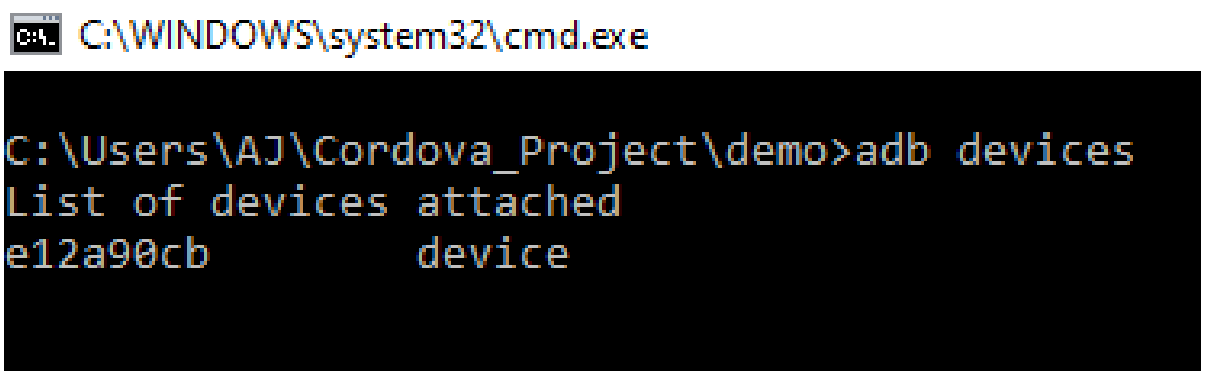
Step 4: Go back to settings and now you will see the **‘Developer Options’** or go to **Additional Settings** and look for **‘Developer Options’**. Enable **‘Developer Options’**.

Step 5: In developer options, you should be able to see ‘USB debugging’. **Enable** or **Allow** ‘USB debugging’.

Step 6: Go back to your machine (pc or laptop), specifically to command prompt or Terminal. In your cmd or terminal, type the following command to check if your mobile device is properly connected: (check devices)

```
adb devices
```

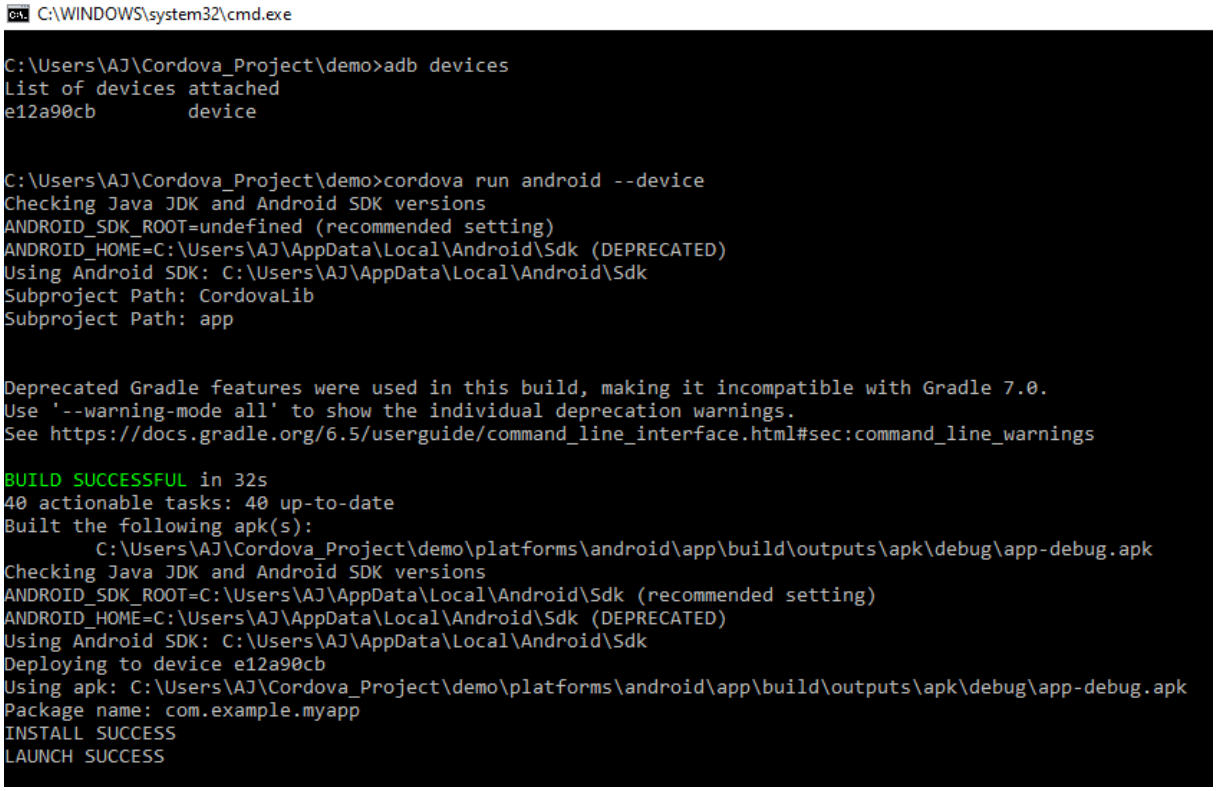
Example:



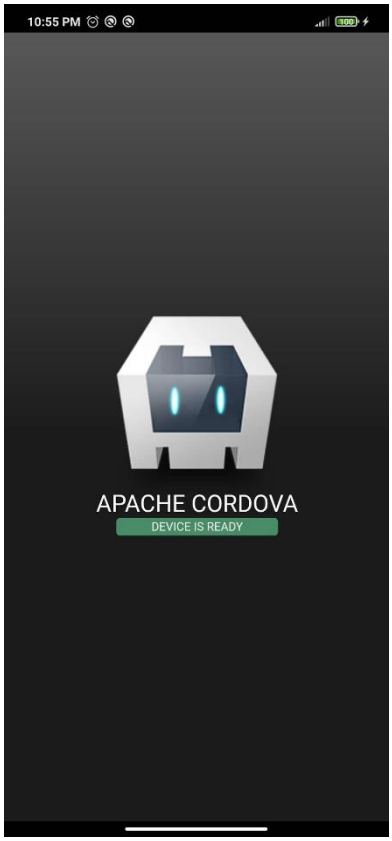
Step 7: In your project directory (cmd or terminal), type the following command to run your output. (In this case, APK file):

```
cordova run android --device
```

Example:



You should be able to see the output in your mobile device: (Device Ready)



***Note:** If there is a problem/error connecting, make sure you check on any of the following:

1. Turn Off/On "system optimization" or "device optimization" and Restart your device.
2. Turn On "USB Debugging".
3. Turn On "Install via USB" (Need to connect to network).
4. Set USB Configuration to Charging.
5. Turn On "install via USB (Need to connect to network).