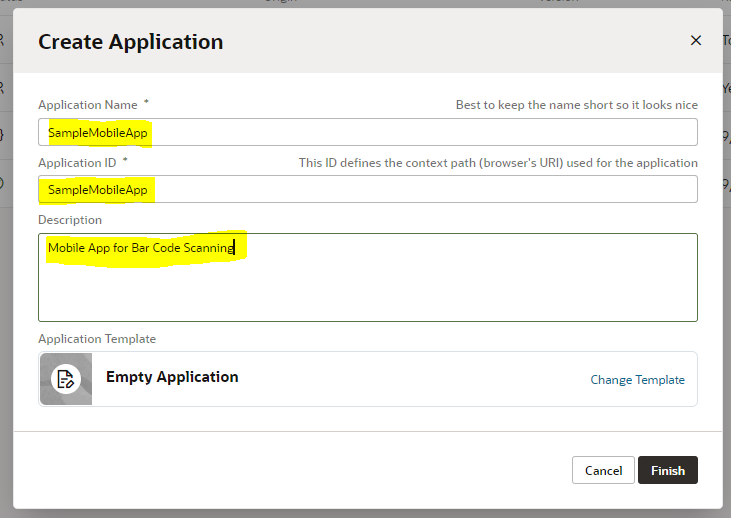
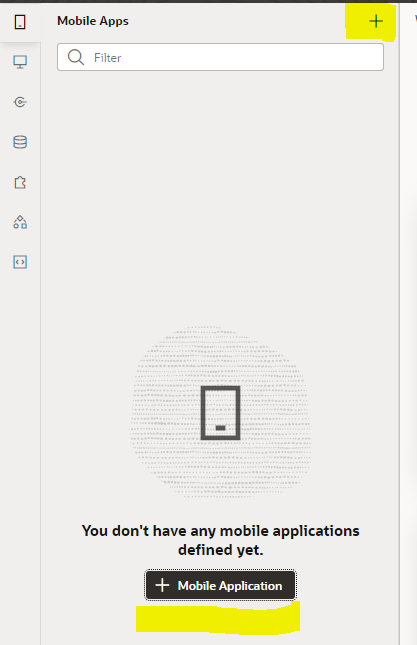
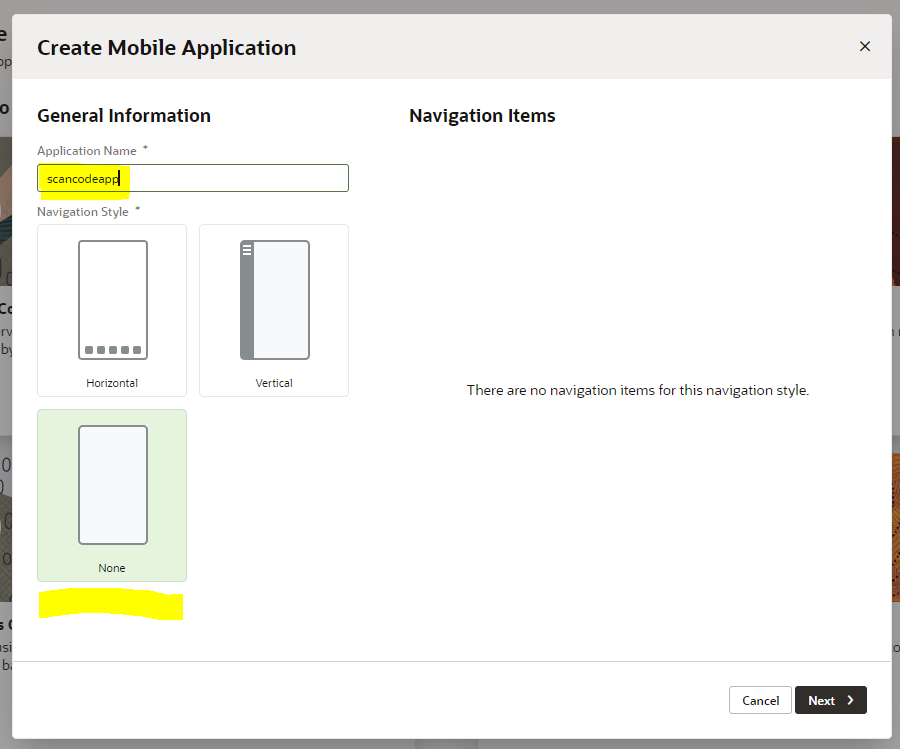
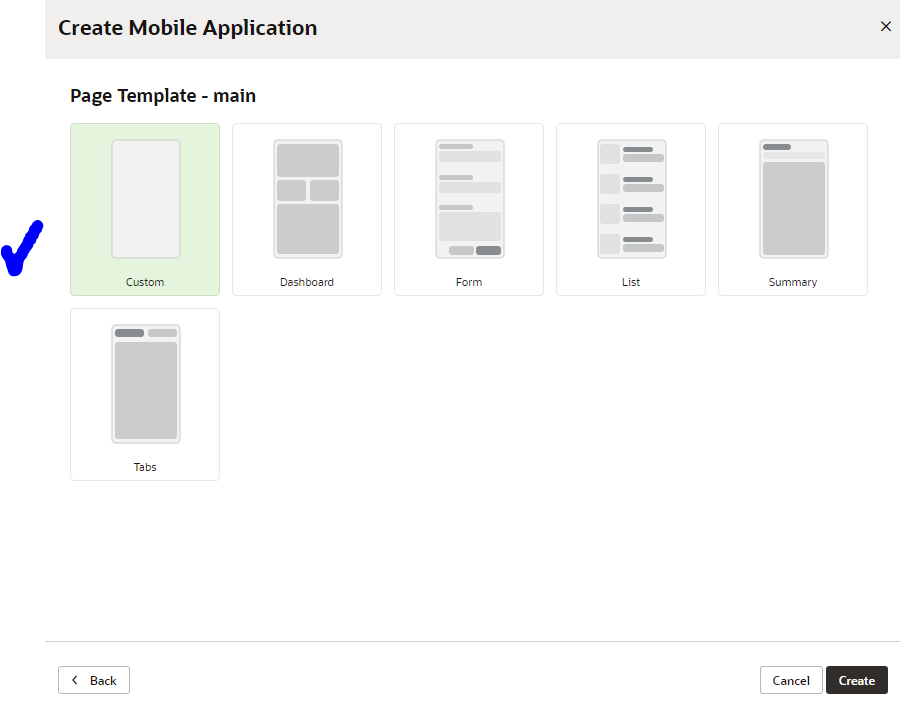
Creating the new VBCS application

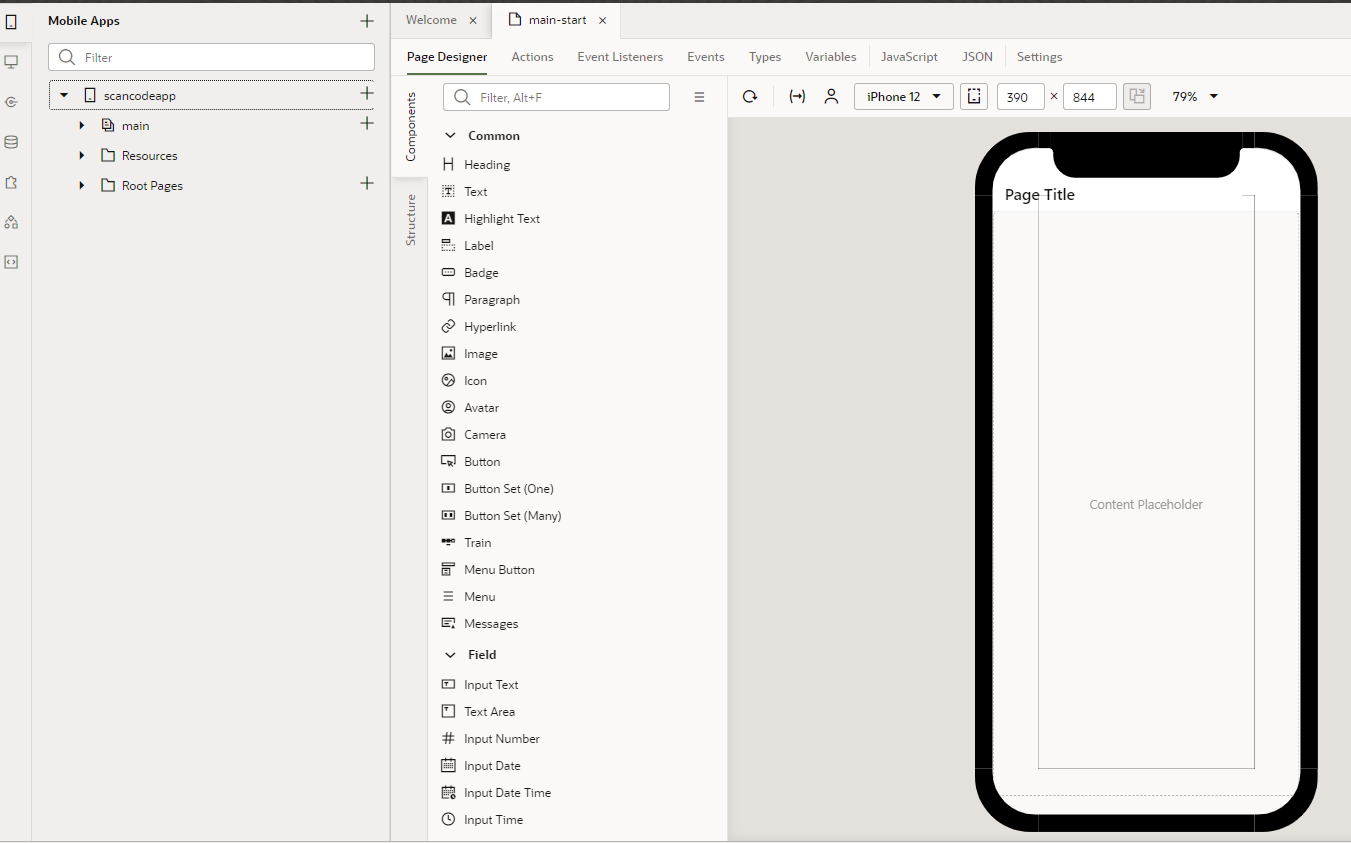


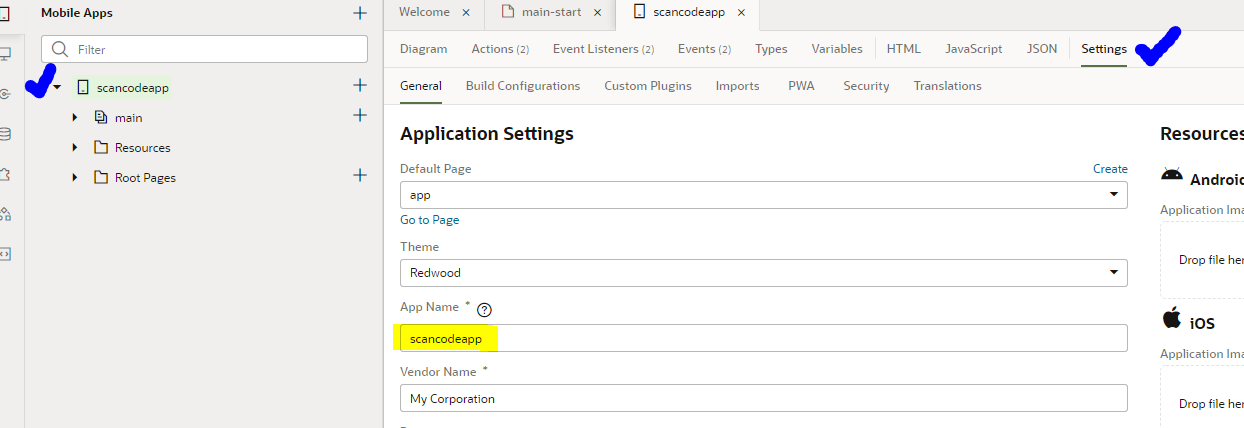
Creating a new Mobile Application

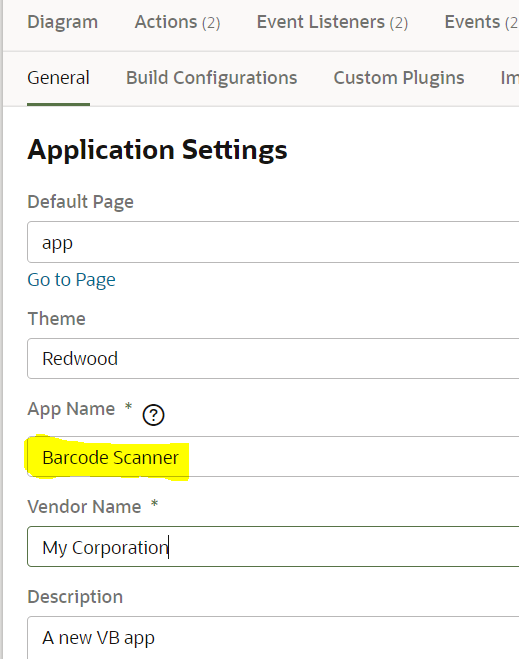


Mobile application is now created

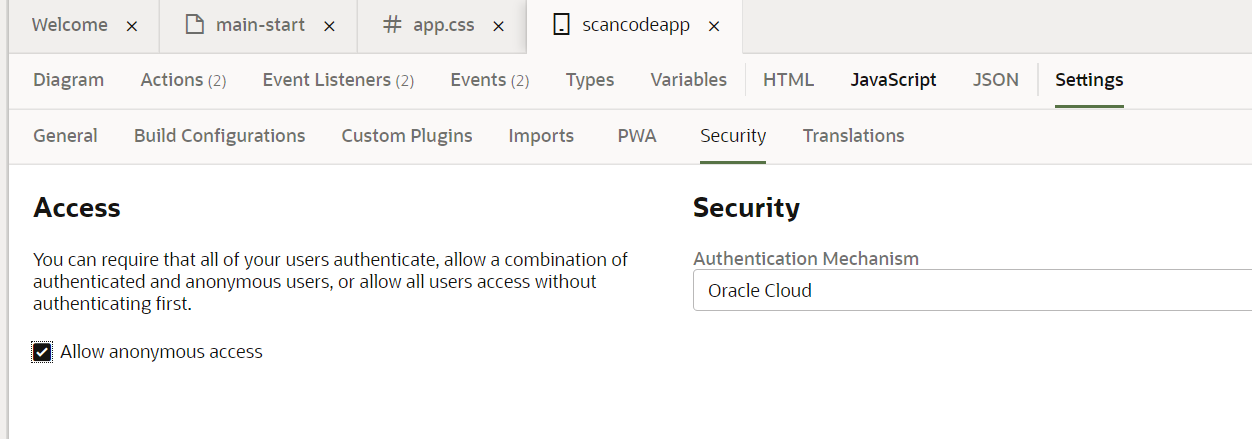


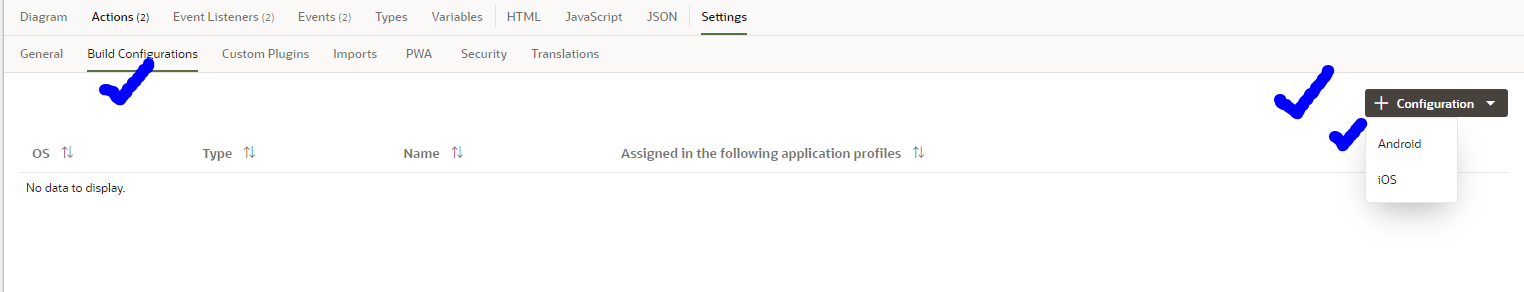


You can change the name of the application as

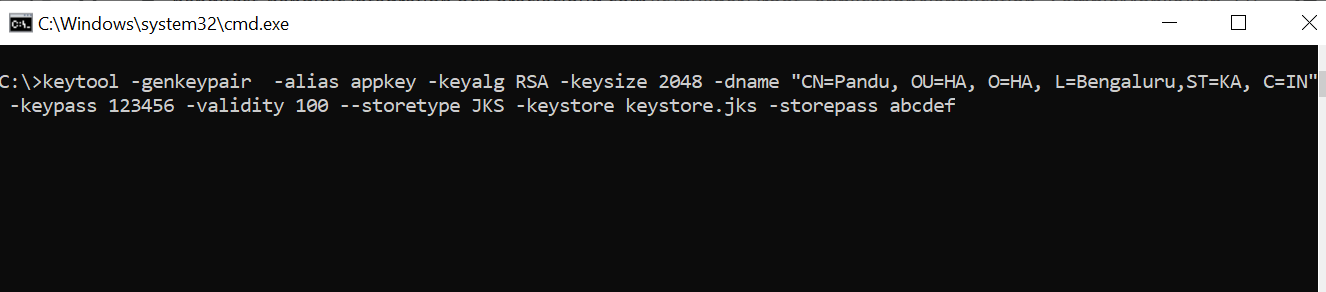


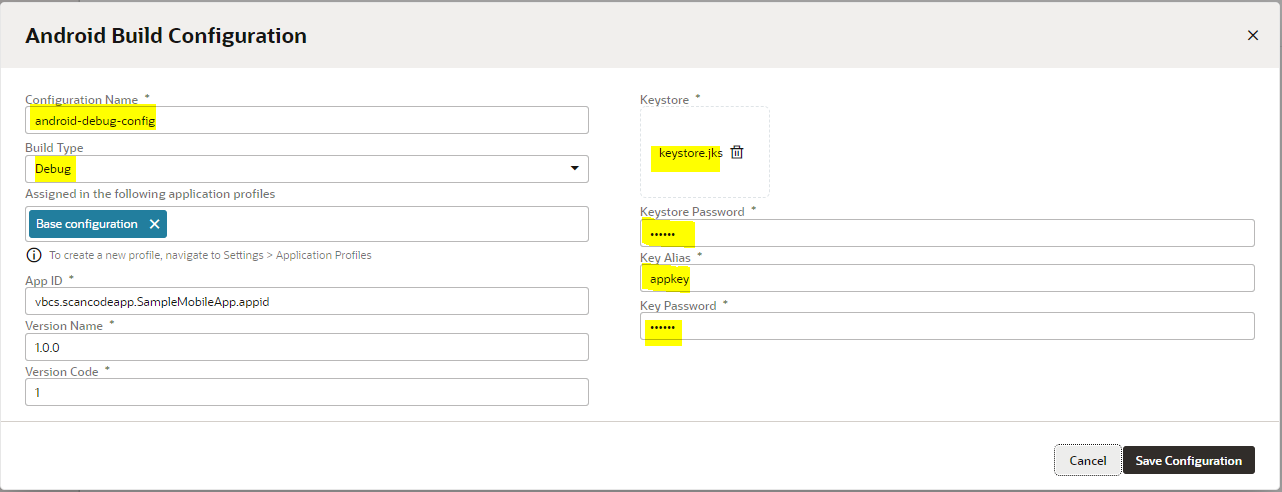
Allow anonymous access to the application

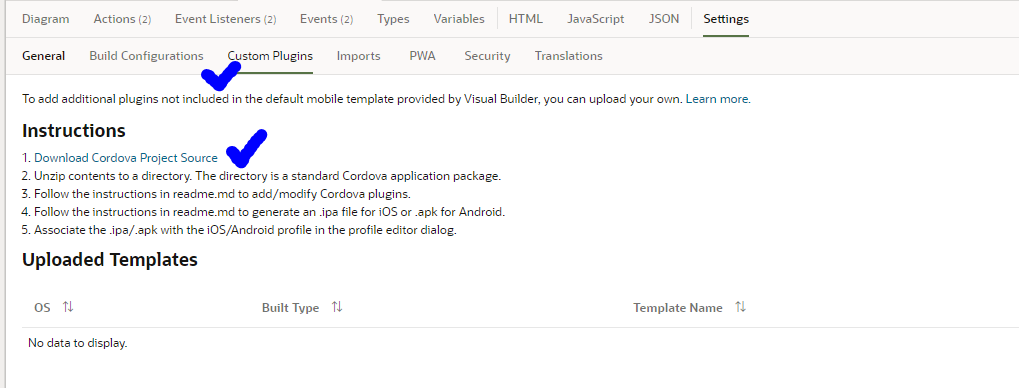


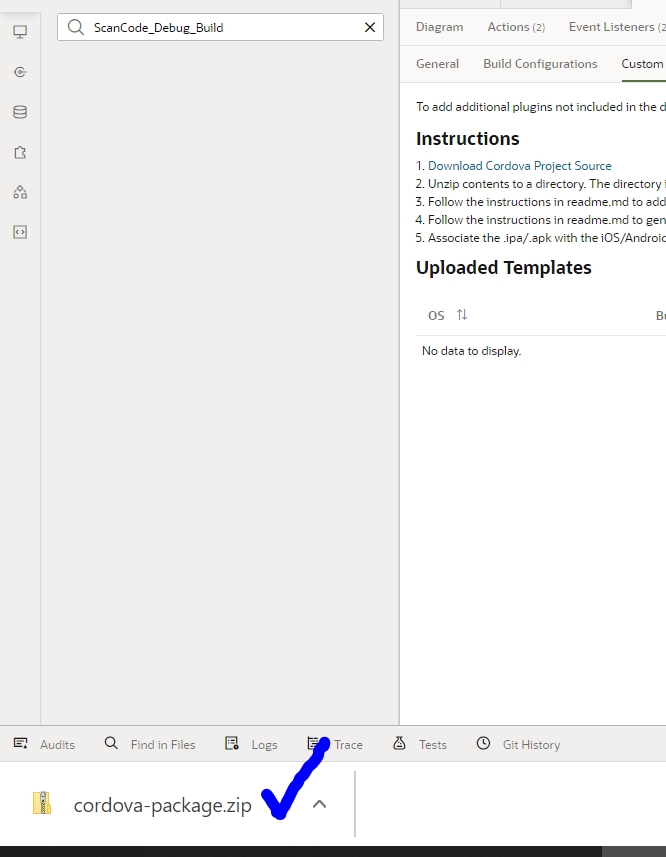


keytool -genkeypair -alias appkey -keyalg RSA -keysize 2048 -dname "CN=Pandu, OU=HA, O=HA, L=Bengaluru,ST=KA, C=IN" -keypass 123456 -validity 100 --storetype JKS -keystore keystore.jks -storepass abcdef

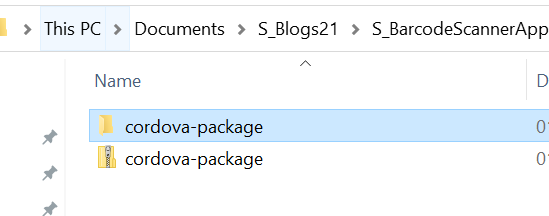








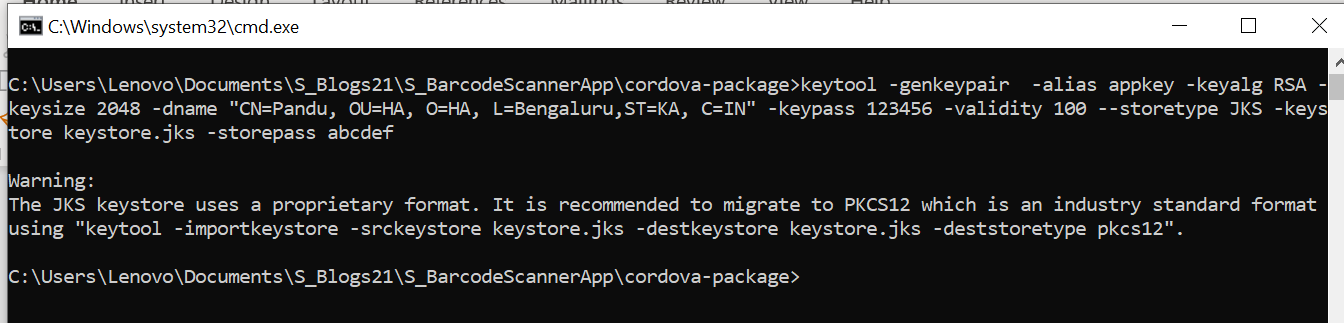
Copy the downloaded zip to a folder and extract it there



Open the cordova-package folder in command prompt and the keytool command as shown below.

***Please do not close the command prompt window. This is needed***

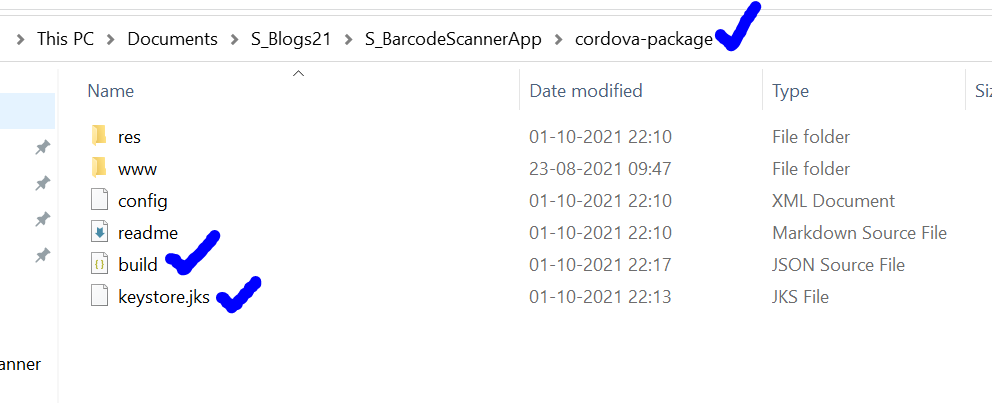
keytool -genkeypair -alias appkey -keyalg RSA -keysize 2048 -dname "CN=Pandu, OU=HA, O=HA, L=Bengaluru,ST=KA, C=IN" -keypass 123456 -validity 100 --storetype JKS -keystore keystore.jks -storepass abcdef



Now create a “build.json” file in the same location, where the above keystore.jks is present



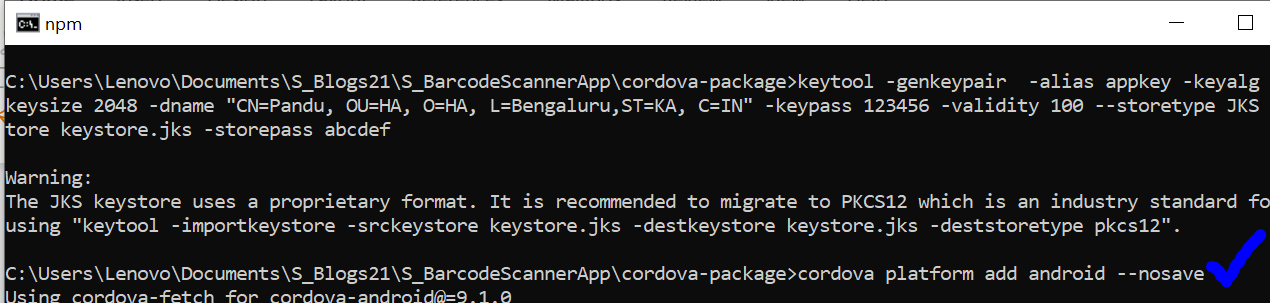
So now the folder structure will look as below

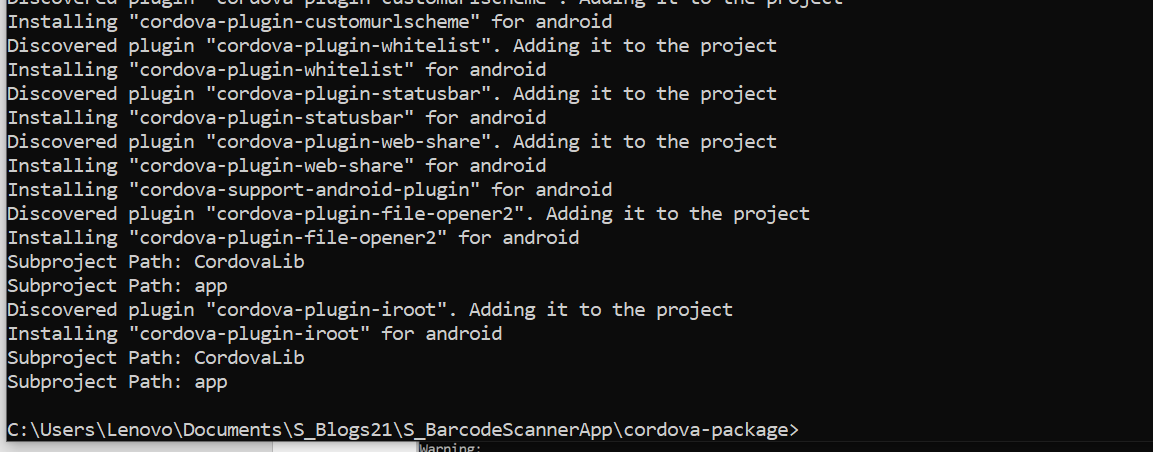


Next, we proceed to add android platform, using the below command.

To do this, we go back to the command prompt which was kept open in the above steps.

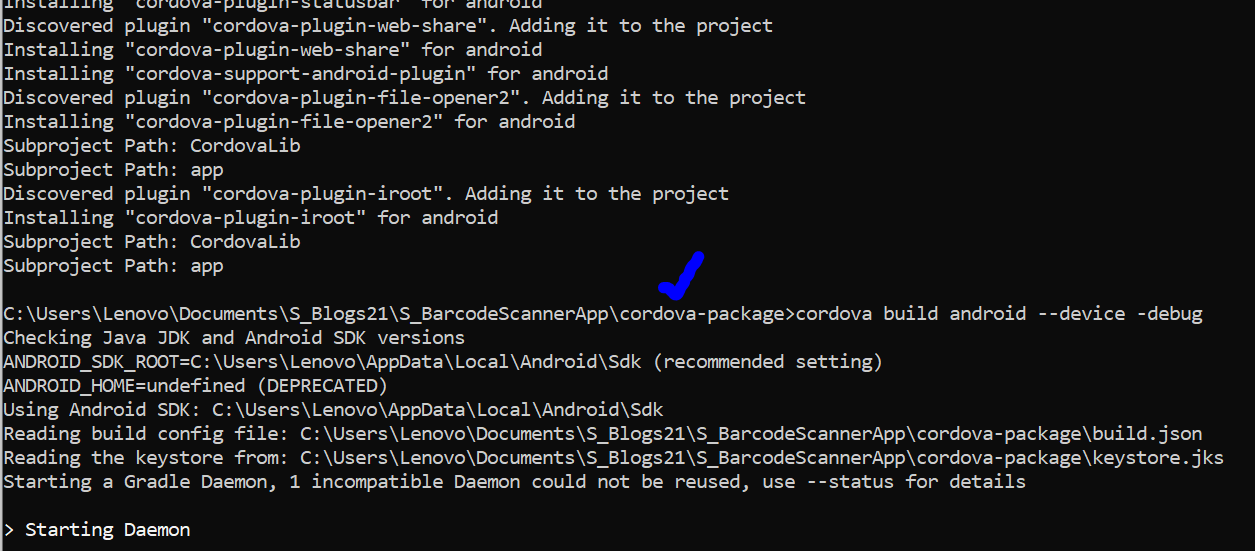
***cordova platform add android –nosave***

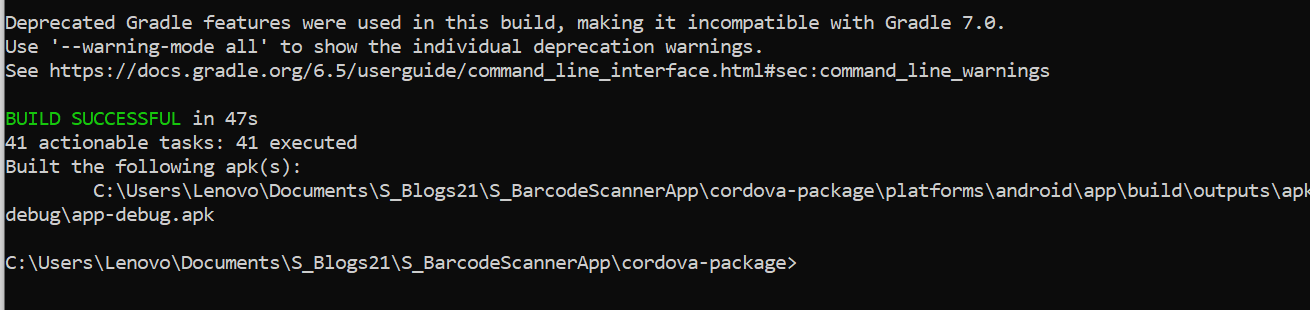




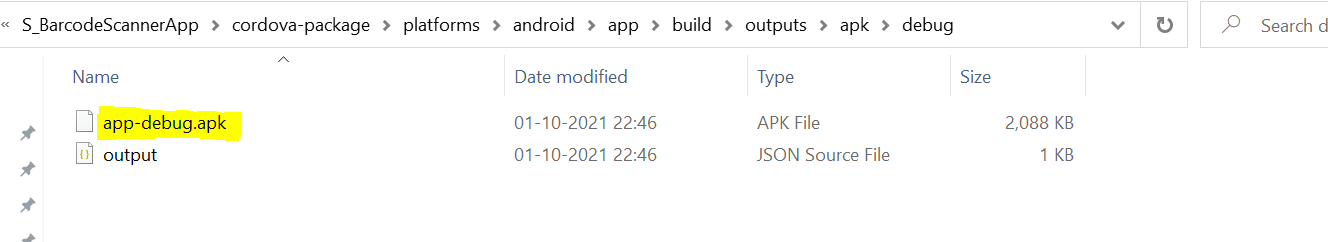
Next we proceed to create the APK file, in debug mode, as we have configured our app in debug mode in VBCS, using the below command. Again, this command will still needs to be run in command prompt, at the same one location where we have cordova-package, keystore, build.json and android platform added

***cordova build android --device –debug***





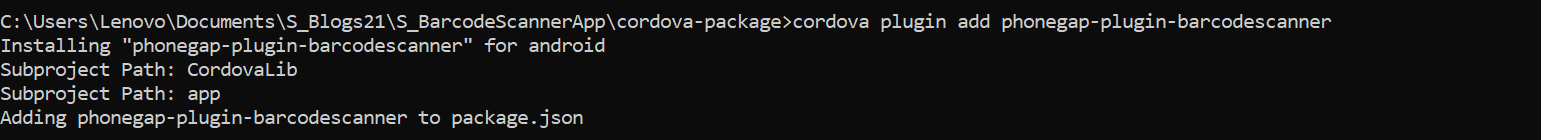
Let’s check the APK folder, as shown below



Let’s continue by adding the barcode scanner plugin. We add it using the below command.

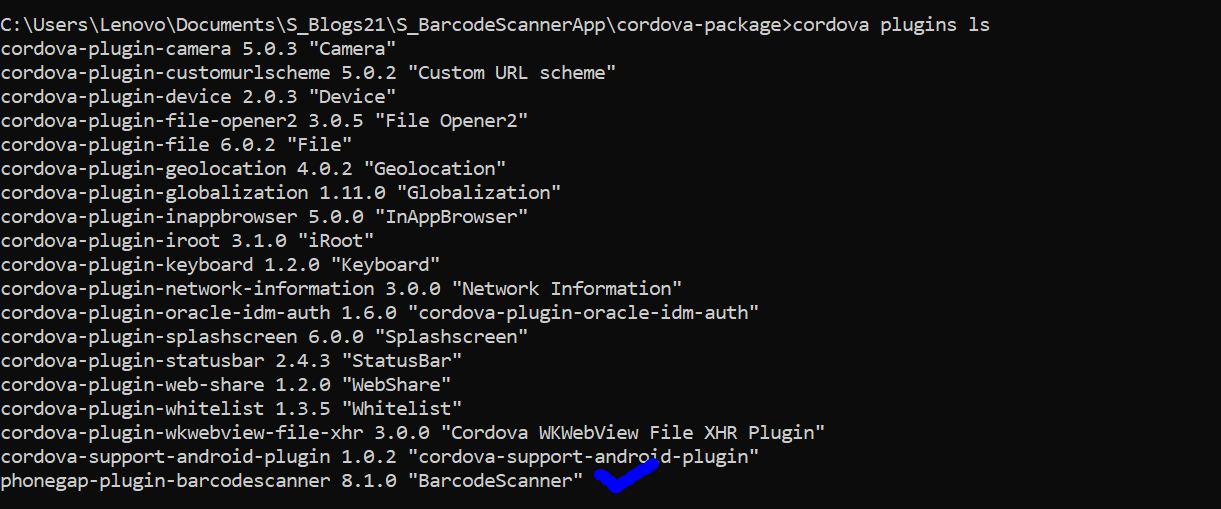
Please check in the command prompt that, the below command will be executed in the “cordova-package” folder only.

***cordova plugin add phonegap-plugin-barcodescanner***



To confirm if the plugin is added successfully, we can execute the below command

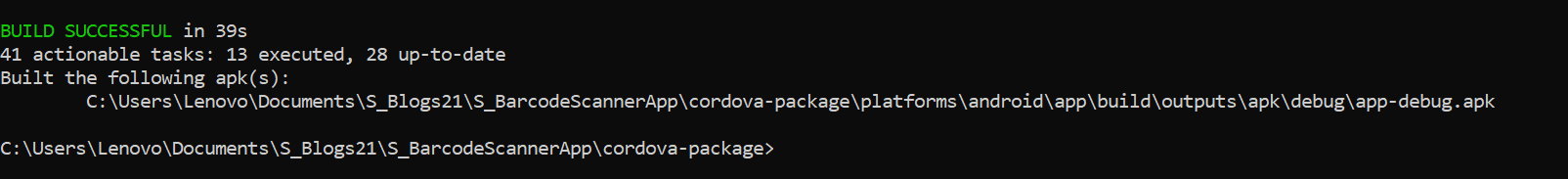
***cordova plugins ls***



Now we need to rebuild and recreate the APK file, using the same command, as we did above.

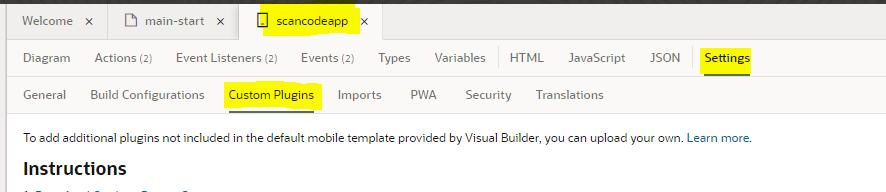
***cordova build android --device –debug***

Confirming the build is successful

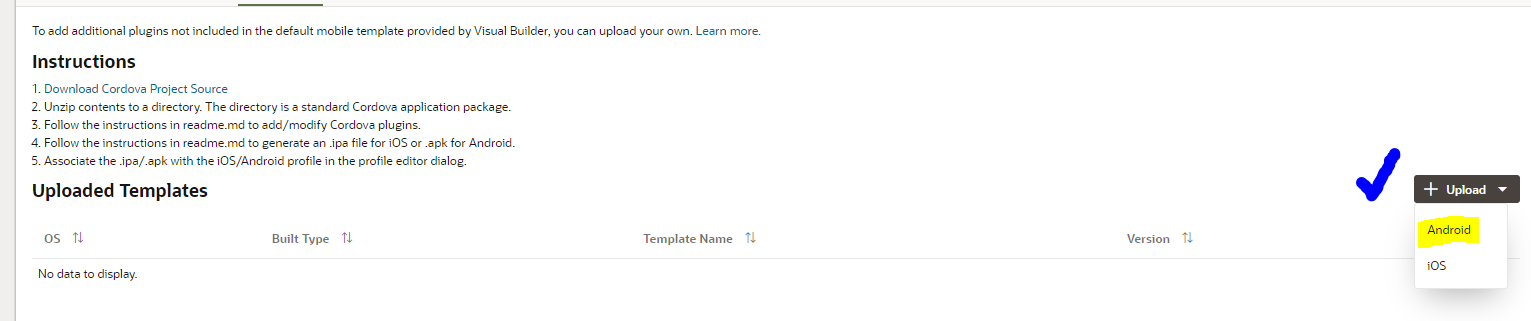


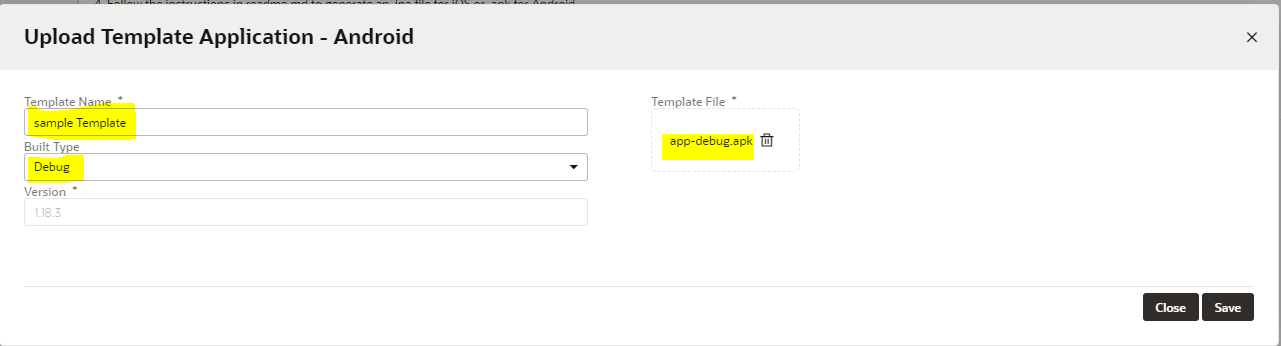
Once the APK file is ready, with barcode scanner plugin added into it, we need to upload it in our VBCS application

We need to navigate to the application “sacncodeapp” -> Settings -> Custom Plugins



Then we have to click on the Upload button, and choose “Android”

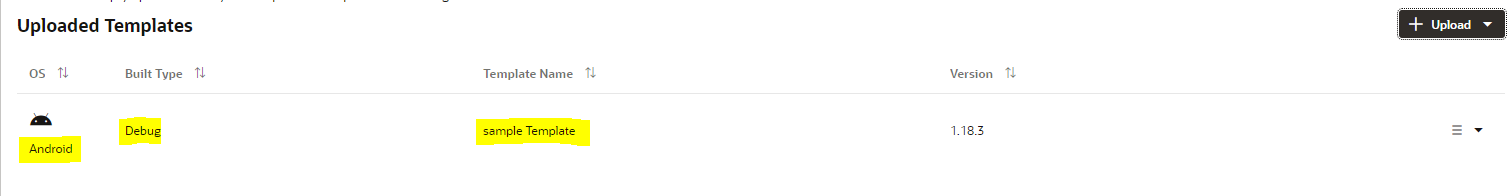




Please note the highlighted sections above. The Template Name, Built Type and the Template File.

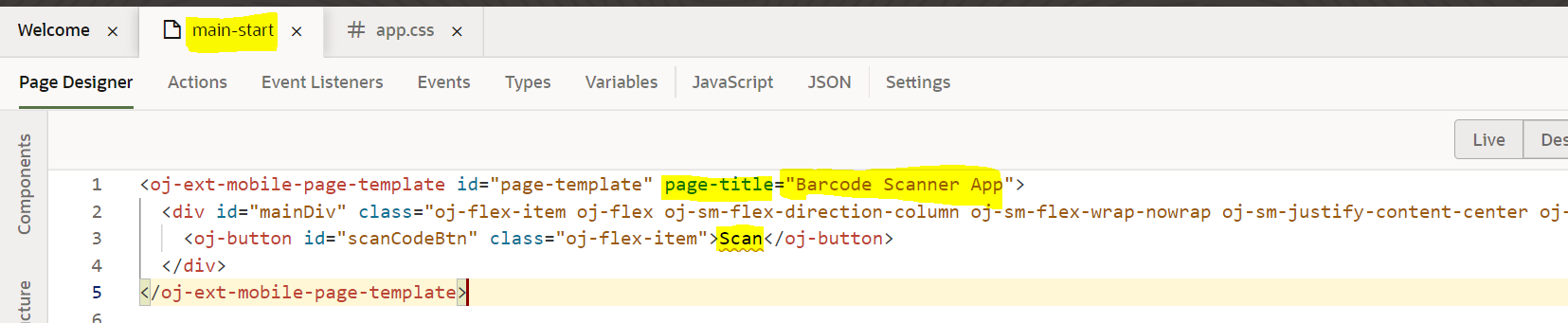
The template file will be the APK file generated after we added the barcode scanner plugin and rebuilt the package.

We proceed by Clicking on Save



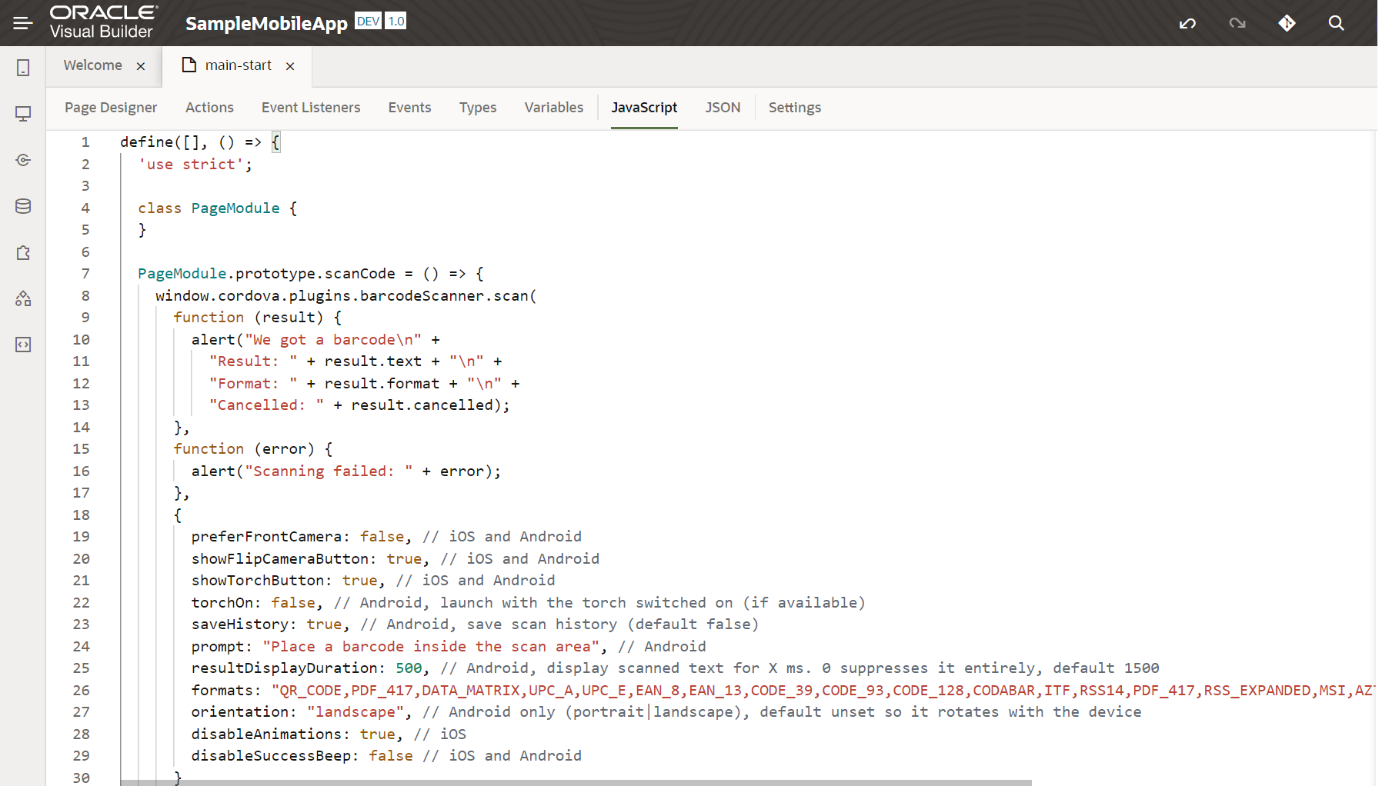
Completing the above step, we are now ready to create the remaining part in VBCS

First, we add a button called “Scan” on the HTML page, as shown below



Please note the above highlighted sections. The button is added in the “main-start” HTML page. The page-title is changed to “Barcode Scanner App”, and we have added “Scan” button too. Also, please make sure to set the height to the button, else it will occupy the complete available screen size.

Now lets add the JS function, as shown below, which gets invoked when the button is clicked.



PageModule.prototype.scanCode = () => {

window.cordova.plugins.barcodeScanner.scan(

function (result) {

alert("We got a barcode\n" +

"Result: " + result.text + "\n" +

"Format: " + result.format + "\n" +

"Cancelled: " + result.cancelled);

},

function (error) {

alert("Scanning failed: " + error);

},

{

preferFrontCamera: false, // iOS and Android

showFlipCameraButton: true, // iOS and Android

showTorchButton: true, // iOS and Android

torchOn: false, // Android, launch with the torch switched on (if available)

saveHistory: true, // Android, save scan history (default false)

prompt: "Place a barcode inside the scan area", // Android

resultDisplayDuration: 500, // Android, display scanned text for X ms. 0 suppresses it entirely, default 1500

formats: "QR\_CODE,PDF\_417,DATA\_MATRIX,UPC\_A,UPC\_E,EAN\_8,EAN\_13,CODE\_39,CODE\_93,CODE\_128,CODABAR,ITF,RSS14,PDF\_417,RSS\_EXPANDED,MSI,AZTEC", // default: all but PDF\_417 and RSS\_EXPANDED

orientation: "landscape", // Android only (portrait|landscape), default unset so it rotates with the device

disableAnimations: true, // iOS

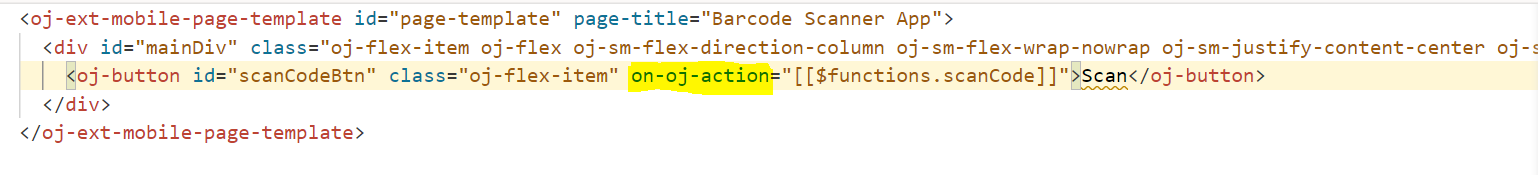
disableSuccessBeep: false // iOS and Android

}

);

};

The above JS function should now be linked to the “on-oj-action” property of the oj-button



Also we can add the below CSS in “app.css” under the Resources folder of the Mobil Application

#mainDiv{

  background: linear-gradient(90deg, rgba(0,191,255,1) 0%, rgba(100,149,237,1) 100%) !important;

}

#scanCodeBtn{

  max-height: 4em;

  height:4em;

  background: #000;

  max-width: 25em;

  width: 15em;

  border-radius: 0.4em;

}

#scanCodeBtn .oj-button-text{

  color:#fff !important;

  font-size: 1.75em;

}

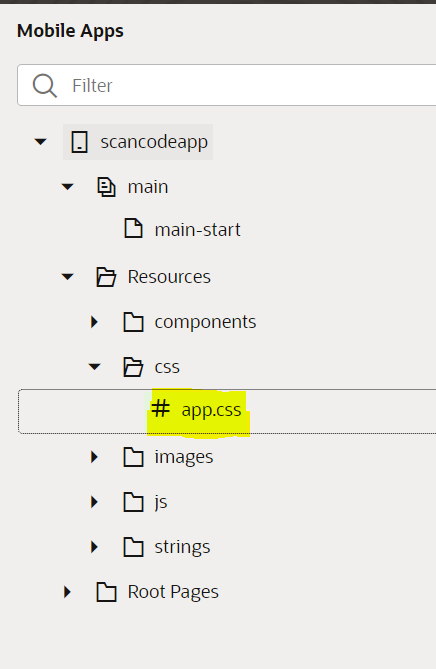
.oj-hybrid-applayout-header-title{

  margin: auto;

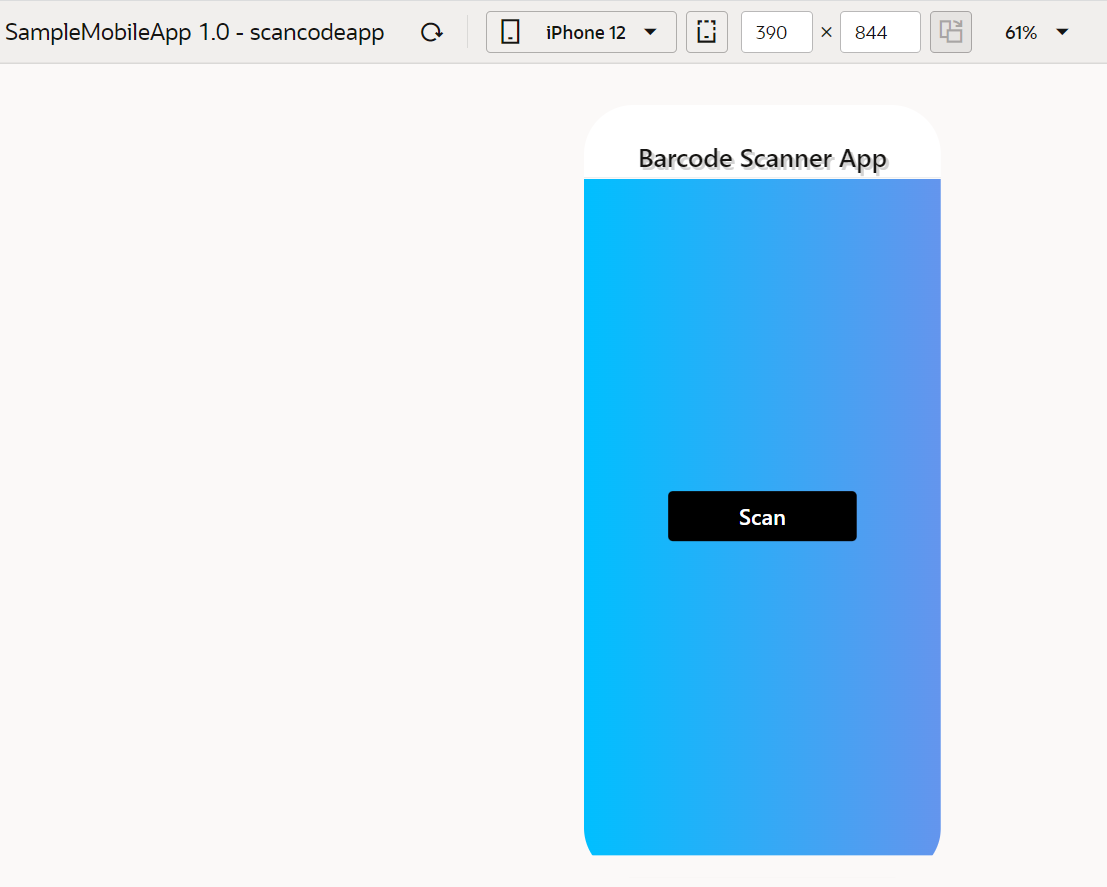
  font-size: 1.75em;

  text-shadow: 3px 3px lightgrey;

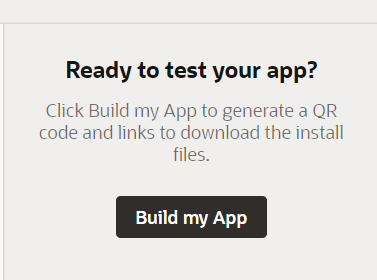
}



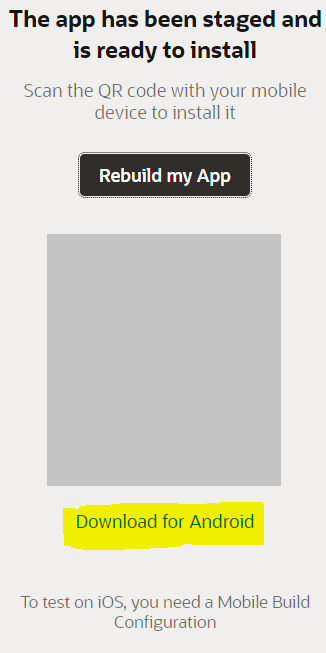
Now we can run the application, and we see below is how the application will look



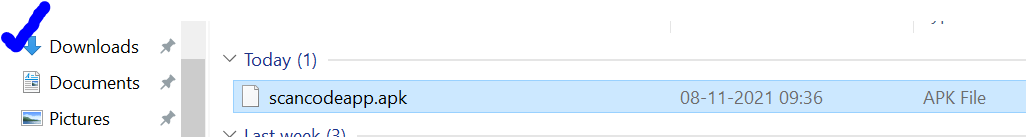
Next we continue to “Build” the application.



Once the application is staged, next we continue to Download the APK installable file for our Android phones



The downloaded file will be in the “Downloads” folder

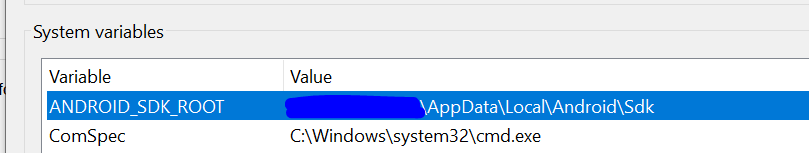


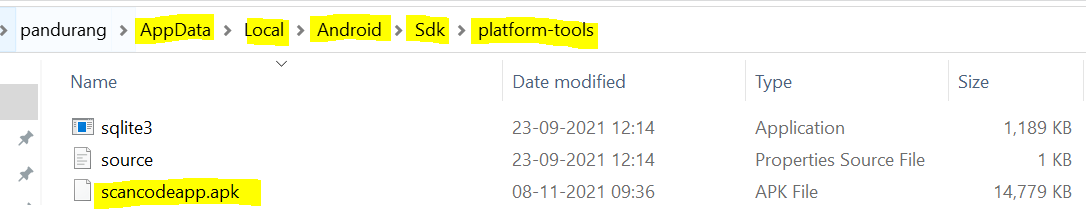
Next we need to install this in our Android mobile. For that we need Android Studio to be installed from <https://developer.android.com/studio>

Once installed, follow the instructions in this video to get the APK installed in our Android phones

<https://www.youtube.com/watch?v=kmg9RY53pFc>

The downloaded APK should be placed in ANDROID\_SDK\_ROOT folder as shown below

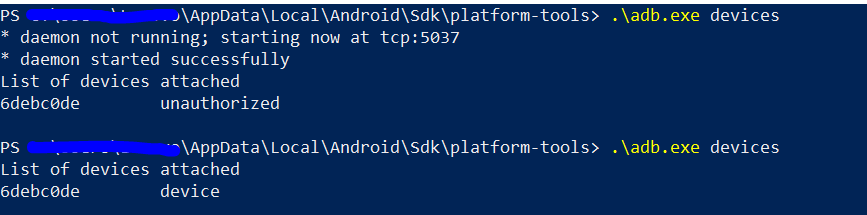




Lets connect our Android phone to the Laptop/Desktop.

Next we open the powershell prompt in the above folder. We run the below command to check the connected devices.

***.\adb.exe devices***



We need to enable USB Debugging on our phones else we can see from the above image, that before enabling USB Debugging, the device 6d….. was ***unauthorized***, and once enabled and rerunning the same command, now the device is listed correctly.

Next we proceed by installing the APK. We use the below command.

***.\adb.exe install .\scancodeapp.apk***