**WEB VIEW IN ANDROID**

If you want to deliver a web application (or just a web page) as a part of a client application, you can do it using [WebView](https://developer.android.com/reference/android/webkit/WebView). The WebView class is an extension of Android's [View](https://developer.android.com/reference/android/view/View) class that allows you to display web pages as a part of your activity layout. It does *not* include any features of a fully developed web browser, such as navigation controls or an address bar. All that WebView does, by default, is show a web page.

To add a WebView to your app, you can either include the <WebView> element in your activity layout or set the entire Activity window as a WebView in [onCreate()](https://developer.android.com/reference/android/app/Activity#onCreate(android.os.Bundle,%20android.os.PersistableBundle)).

Below is a sample code for you to understand the application of Webview in your project. In this code, we tried to make a web browser using Google’s Url.

**activity\_main.xml:**

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<WebView

android:id="@+id/webView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:layout\_editor\_absoluteX="171dp"

tools:layout\_editor\_absoluteY="499dp" />

</androidx.constraintlayout.widget.ConstraintLayout>

**MainActivity.java:**

public class MainActivity extends AppCompatActivity {

private WebView webView;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_main*);

webView=findViewById(R.id.*webView*);

webView.loadUrl("https://www.google.co.in");

}

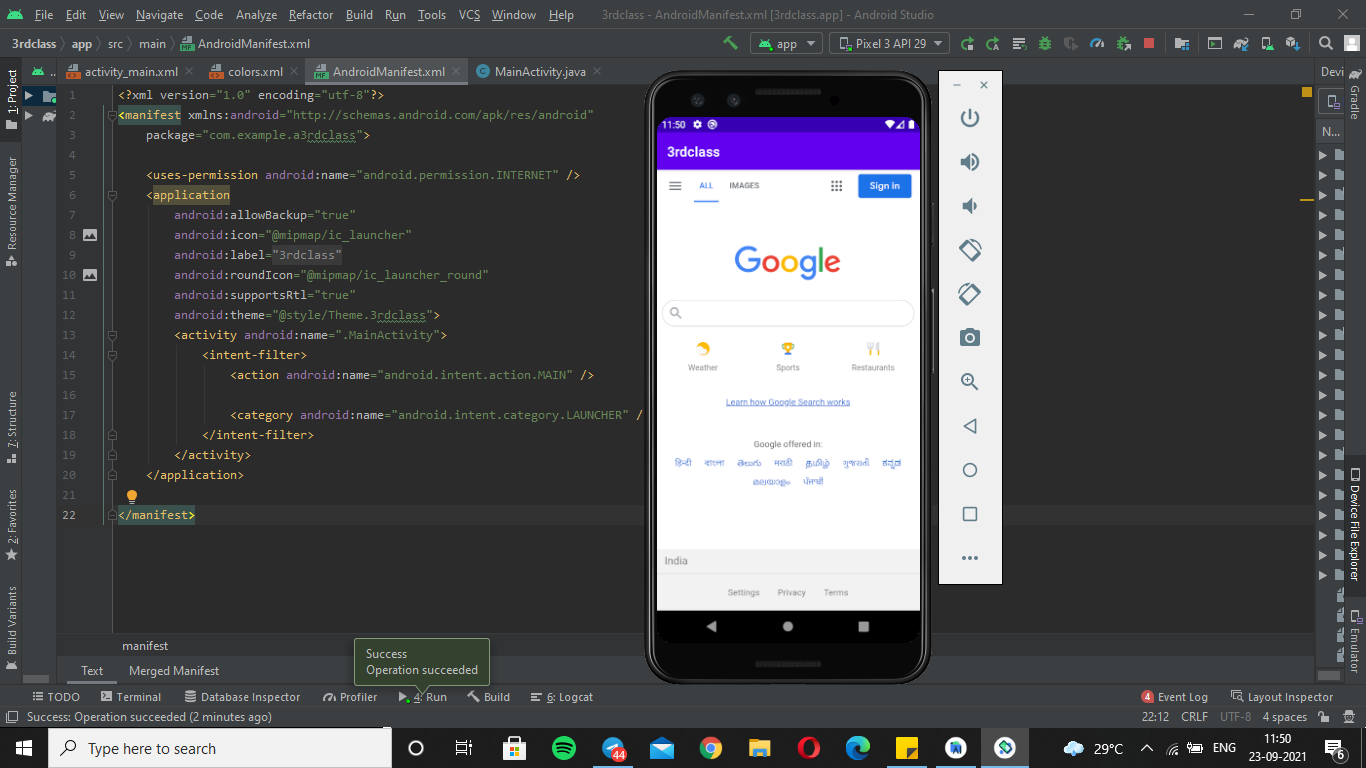
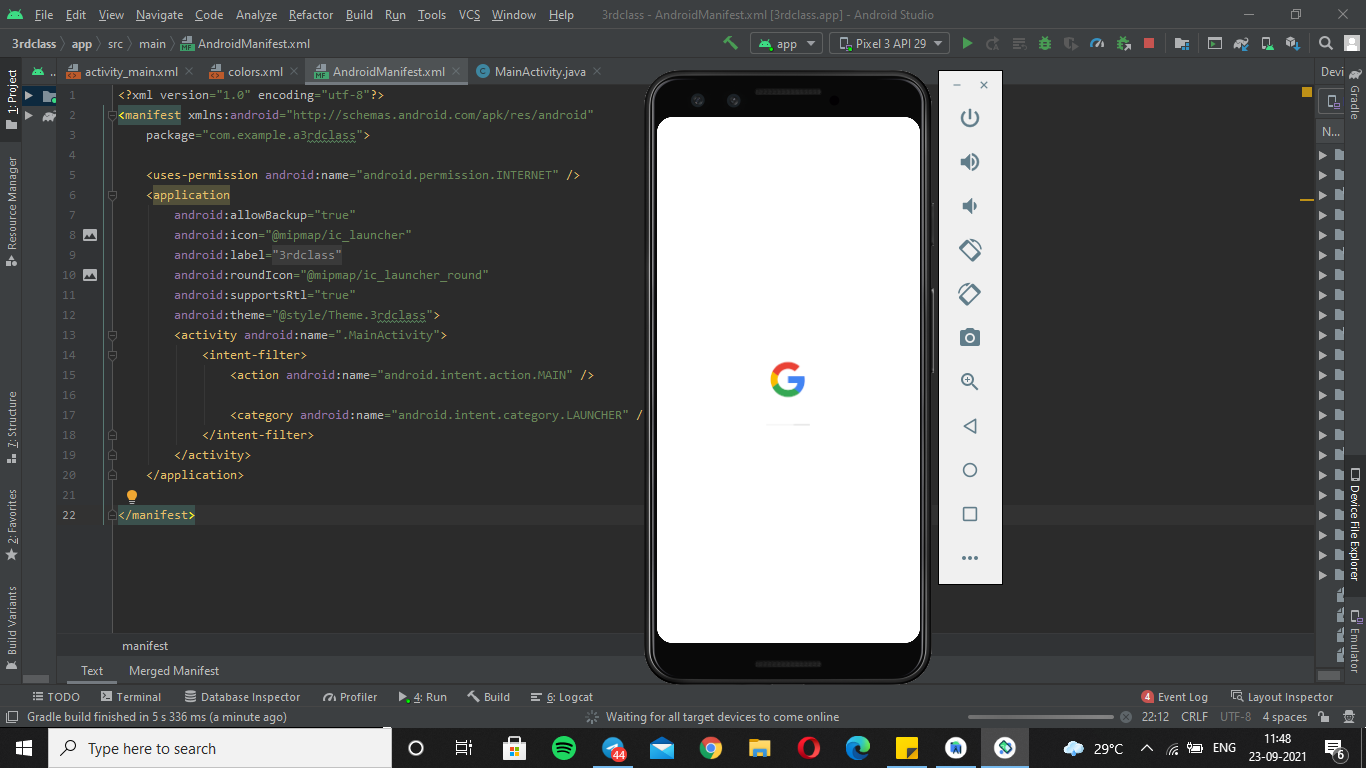
}

**Android manifest:**

Add permission to access the Internet.

<uses-permission android:name="android.permission.INTERNET" />

Our App will look like this:



Additionally, you can customize your WebView by modifying the following:

* Enabling fullscreen support with [WebChromeClient](https://developer.android.com/reference/android/webkit/WebChromeClient). This class is also called when a WebView needs permission to alter the host app's UI, such as creating or closing windows and sending JavaScript dialogs to the user. To learn more about debugging in this context, read [Debugging Web Apps](https://developer.android.com/guide/webapps/debugging).
* Handling events that impact content rendering, such as errors on form submissions or navigation with [WebViewClient](https://developer.android.com/reference/android/webkit/WebViewClient). You can also use this subclass to intercept URL loading.
* Enabling JavaScript by modifying [WebSettings](https://developer.android.com/reference/android/webkit/WebSettings).
* Using JavaScript to access Android framework objects that you have injected into a WebView.