Create a Hybrid App

Demo 2nd day of class- just walk through creating an application Create new 'Empty Activity' project, use all other defaults

These are the files to change

```
▲ Android 

▼
  ∨ 📭 app
   manifests
    AndroidManifest.xml
   ∨ ijava

com.example.myapplication

     © MainActivity
  > com.example.myapplication (androidTest)
   > 🖿 com.example.myapplication (test)
   ∨ IIII res
    > 🖿 drawable
    🗸 🖿 layout
     activity_main.xml
    > 🖿 mipmap
    values
      # colors.xml
       # strings.xml
     > 🖿 themes (2)
Use plain activity, put the following in layout/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/wv1"
  android:layout width="wrap content"
  android:layout height="wrap content"
  app:layout constraintBottom toTopOf="@+id/but1"
  app:layout constraintEnd toEndOf="parent"
  app:layout constraintStart toStartOf="parent"
  app:layout constraintTop toTopOf="parent" />
<Button
  android:id="@+id/but1"
  android:layout width="wrap content"
  android:layout height="wrap content"
  android:onClick="doClick"
  android:text="change site"
  app:layout constraintBottom toBottomOf="parent"
  app:layout constraintLeft toLeftOf="parent"
  app:layout constraintRight toRightOf="parent"
  app:layout_constraintTop_toTopOf="parent"
  app:layout constraintVertical bias="0.944" />
```

</androidx.constraintlayout.widget.ConstraintLayout>

In MainActivity add the following (replacing onCreate()), hover over any undefined classes and press alt-enter to fix imports WebView wv; String [] URLs={"https://www.washingtonpost.com","https://www.nytimes.com"}; int url number = 0; //start with frst string private void load_next_page() { wv.loadUrl(URLs[url number]); url number=(url number+1)%URLs.length; } @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity_main); //get a handle to the widget in the UI wv=(WebView)findViewById(R.id.wv1); //go to the first page load next page(); } //tied to the doClick event handler in the xml public void doClick(View view) { load_next_page(); }

In manifest- add permissions to access things (add just after manifest tag and before the application tag)

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

<uses-permission android:name="android.permission.READ EXTERNAL STORAGE" />

That's it you now have a working Hybrid app. Or a web browser thats wrapped with a native application that controls the browser. Put most logic in the web page you are accessing so you get ease of updating your app. Also since the native portion of the app is simple (all logic is in web page), it's easy to create a iOS version. You can also host your app on app stores (both Apple's and Google Playstore)