1. Create a new project in Android Studio as shown below by clicking on 'New Proj



2. Select 'Empty Views Activity' under the 'Phone and Tablet' tab. Click Next.

	<ul> <li>←</li> </ul>	PROJECT
[]		
		<b>•</b>
·*		
No Activity	Empty Activity	Gemini API Starter
¢ :	¢ I	¢
Basic Views Activity	Bottom Navigation Views Activity	Empty Views Activity
		<b>P</b>
	Cancel Previous	Next Finish
	No Activity	No Activity Empty Activity   Image: Straight of the stra

- 3. Configure your project.
  - a. Give the project a name.
  - b. Make sure the language is Java
  - c. The minimum API level should be enough so that the application runs on most devices.

Click on finish and wait f	or the project to l	build itself.
----------------------------	---------------------	---------------

	······,···
Empty Views Activity	
Creates a new empty activity	
Name	ListyCity
Package name	com.example.listycity
Save location	/Users/work/Documents/Cmput 301 Labs/WK 2 Demo
Language	Java 🗸
Minimum SDK	API 24 ("Nougat"; Android 7.0)
	Your app will run on approximately 97.4% of devices. Help me choose
Build configuration language (	Kotlin DSL (build.gradle.kts) [Recommended]
🔺 Save location should not c	contain whitespace, as this can cause problems with the NDK tools.
	Cancel Previous Next Finish

4. Navigate to app/res/layout directory in the 'Project' pane. Then right click on the 'layout' directory and then click on 'New' and then 'Layout Resource file'



5. Give a name to the layout resource file with the extension '.xml'. Then rename the "Root element" to "TextView", and then press 'OK'.

ile name:	content.xml		
Root element:	TextView		
Source set:	main src/main/res		
Directory name:	layout		
vailable qualifie	ers:	Choser	n qualifiers:
Country Coo Ketwork Co Country Coo Country Country Coo Country Co	de de stion reen Width th	>>	Nothing to show
Size Size Ratio			Cancel

6. Now go to 'content.xml' under res/layout and then add a TextView with the following properties. Make sure the 'Text' tab is selected. Make sure you add an 'id' to the TextView.

	ListyCity Version control	ř.	tê Medium Phone API35 ∨ 🛆 app ∨ ▷ 13: : To CA Ξ5 1,5 <i>#2</i> ,7 C 16	97 (S
	Android ~	> activ	ity_main.xml © MainActivity.java 🚸 content.xml ×	Ξ
<u>₽</u> 4 	<ul> <li>C app</li> <li>manifests</li> <li>java</li> <li>res</li> <li>reframe</li> <li>reframe</li> <li>layout</li> <li>solution</li> <li>solution</li> <li>solution</li> </ul>	1 2 3 4 5	<pre><?xml version="1.0" encoding="utf-8"?></pre>	× &
		6 7 8 9 10	<pre>android:padding="20dp" android:text="sample_text" android:textSize="40sp"&gt;          </pre>	

7. Go to activity\_main.xml under the res/layout directory and add a 'ListView' inside the 'constraintlayout' viewgroup. Make sure you add an 'id' attribute to the 'ListView'.

••	LC ListyCity      Version control	~	🖺 Medium Phone API 35 🗸 🖂 app 🗸 🖒 🏦 🏹 여숙	≣ ¢	<i>e</i> ? Q	\$	þ
	Android $\sim$ $\bigcirc$ $\diamond$ $\times$ $\vdots$ $-$	🚸 activi	ty_main.xml × (i) MainActivity.java 4 content.xml		10 🖾	: <i>L</i>	7
20	Chapp	1	xml version="1.0" encoding="utf-8"?		<u>@</u> 1 <u>@</u> 1 ~	· ~ ~	9
00	> 🗀 manifests	2 🔘	<androidx.constraintlayout.widget.constraintlayout <="" td="" xmlns:android="http://schemas.android.com/apk/res&lt;/td&gt;&lt;td&gt;/android"><td></td><td></td><td>ĺ</td></androidx.constraintlayout.widget.constraintlayout>			ĺ	
	> 🗀 java	3	xmlns:app="http://schemas.android.com/apk/res-auto"			3	3
	res	4	<pre>xmlns:tools="http://schemas.android.com/tools"</pre>			-	1
	> 🖻 drawable	5	android:id="@+id/main"			le	2
	<ul> <li>Iayout</li> </ul>	6	android:layout_width="match_parent"			4	
	activity_main.xml	7	android:layout_height="match_parent"				
	>content.xml	8	<pre>tools:context=".MainActivity"&gt;</pre>			[[	j
	> 🖻 mipmap	9					
	> in values	10	<listview< td=""><td></td><td></td><td></td><td></td></listview<>				
	> 🗈 xml	11	android:id="@+id/city_list"				
	> @ Gradle Scripts	12	android:layout_width="match_parent"				
		13	android:layout_height="match_parent">				
		14					
		15					
		16					
		17					
10							
$\bigcirc$							
0							
>_							
0.0							
р		androidx.	constraintlayout.widget.ConstraintLayout > ListView				
O WK	2 Demo > □ app > src > □ main > r	es > layou	ıt > � activity_main.xml 12:44 LF	UTF-8 🛄	4 spaces	ď ()	
							đ

- 8. Now let's head over to the MainActivity.java file which will contain the logic for mapping the data to the 'ListView' so that it can be shown as a scrolling list.
- 9. Create References for the 'ListView' along with a reference for 'ArrayAdapter' and an 'ArrayList'.

Android $\sim$ $\bigcirc$ $\diamond$ $\stackrel{\scriptstyle \times}{}$ $\stackrel{\scriptstyle -}{}$	> activity_main	.xml 💿 MainActivity.java 🛛 💠 content.xml	
🗸 🕞 app	2		4
> 🗅 manifests	3 >	import	
🗸 🗀 java	14		
<ul> <li>com.example.listycity</li> </ul>	15 > < >	nublic class MainActivity extends AnnComnatActivity {	
C MainActivity	10 0 11		
> com.example.listycity (and	16		
> com.example.listycity (tes	17	// Declare the variables so that you will be able to reference it later.	
✓ □ res	18	ListView cityList: no usages	
> 🖻 drawable			
<ul> <li>Iayout</li> </ul>	19	ArrayAdapter <string> cityAdapter; no usages</string>	
>activity_main.xml	20	ArrayList <string> dataList; no usages</string>	
ch content und			

- 10. Now inside the onCreate() method, we write the logic that will help to bind the data to the 'ListView'.
  - a. First find the reference to the 'ListView' using findViewById() and assign it to the reference 'cityList'.
  - b. Then declare a string array consisting of cities which can be fed into the 'ListView' later.
  - c. Create a new 'ArrayList' and assign it to the reference 'dataList'. This will contain the data (the string array of cities).
  - d. Add the data(string array containing city names) to the 'dataList' as shown in the picture below.
  - e. Now we have to link the content.xml to the 'dataList' so that each element will be displayed in a separate row in the list.

f. Finally, we connect the 'ListView' to the 'ArrayAdapter' (cityAdapter) which will show each 'TextView' in the form of a scrolling list.

