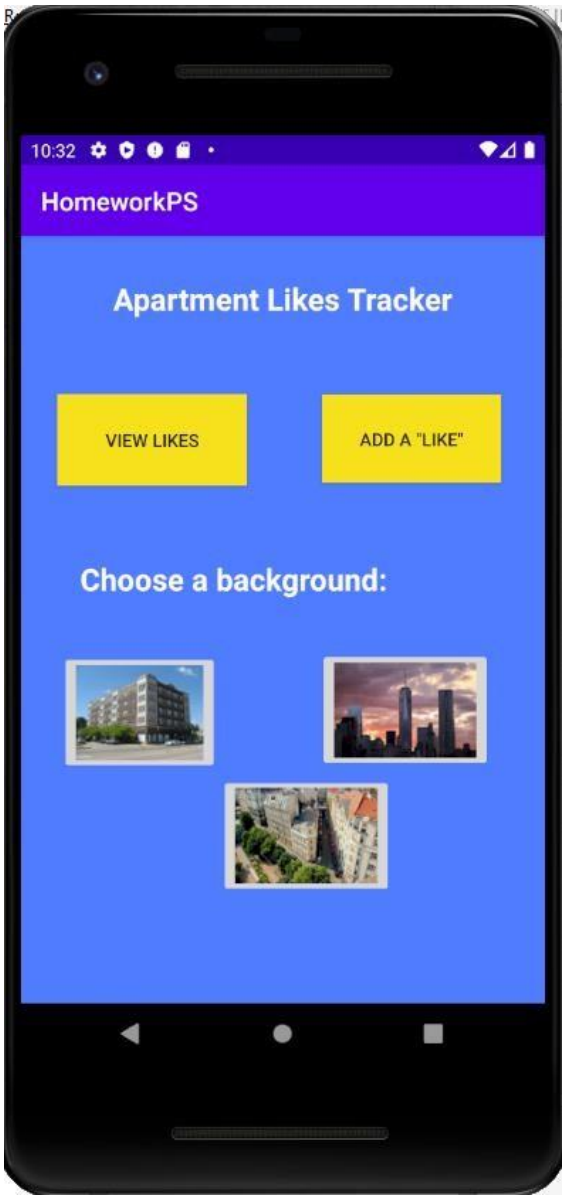
Topics: Persistent Storage (Shared Preferences, SQLite database)

Your task is to create an "Apartment Likes" mobile phone app. The app will be used in portrait orientation. There are three screen/activity sets in this app. The **initial screen** is shown here.



The default background color is #507CFF. The color of the text is white.

There are two buttons. The button on the left entitled "VIEW LIKES" when clicked, transfers the user to the screen termed "Main2Activity" below. The button on the right entitled "ADD A LIKE" when clicked, transfers the user to the screen termed "Main3Activity" below.

The user also has the opportunity to set a background image for the other two screens. This choice of background can be specified by clicking on one of the image buttons at the bottom of the screen. If the user clicks one of these image buttons, **save** the choice as a *Shared Preferences* key/value pair. **These button images are available for download from our course Canvas site.**

Main2Activity.java and activity_main2.xml

If the user has a saved preference for the background image, saved as a Shared Preference as described above, then this screen needs its background to be programmatically set to the preferred image. You can determine if the preference key/value pair has been set using the **.contains** method. So for

example, if you are using the key “backimg”, then you can determine if this shared preferences key has been set, using

```
if (sp.contains("backimg")) {
```

where sp has been defined as a SharedPreferences object, as described in the lectures.

There are at least two ways that you can set the background of a layout to a drawable resource. Here is a hint as to one way for a ConstraintLayout that has been given an id of “screenlikes”:

```
int xid = getResources().getIdentifier(imgref,null, null);
ConstraintLayout setl = (ConstraintLayout) findViewById(R.id.screenlikes);
setl.setBackgroundResource(xid);
```

where “imgref” is a *drawble* resource, similar to those we previously used for ImageView components (for example in the *TimeTravelInc* app).

Here is a less “slick way”:

```
if (sp.getString(“backimg”, “”).equals("apt1"))
setl.setBackgroundResource(R.drawable.apt1);
```

These background images are available for download from our course Canvas site.

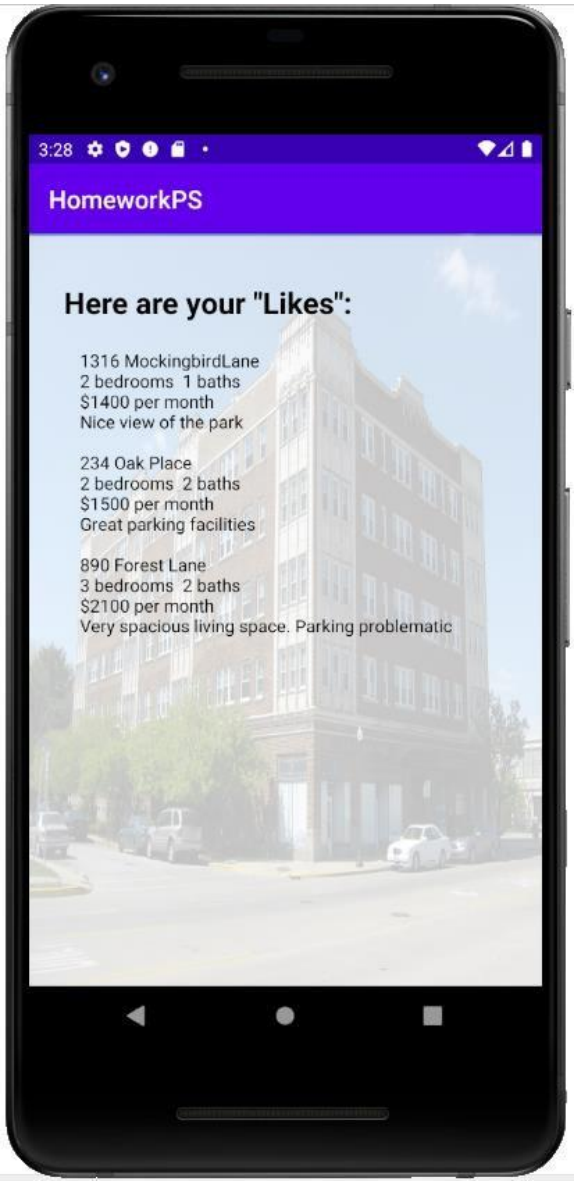
If no shared preferences has been set, the background should default to the same blue color as described for the opening screen.

This activity displays the database records on the screen.

Here is the information required for working with the database:		
Database name:		aptlikes.db **AVAILABLE FOR DOWNLOAD FROM COURSE CANVAS SITE**
Table name:		apts
Field names		and types:
	ADDRESS	String
	NUMBEDRMS	int
	NUMBATHS	int
	MONTHLYRENT	int
	NOTES	String

To display all records, the SQL statement is: SELECT * FROM apts;

Here is what this screen should look like, if the first image button or second image button, respectively, was selected for the background on the previous screen:

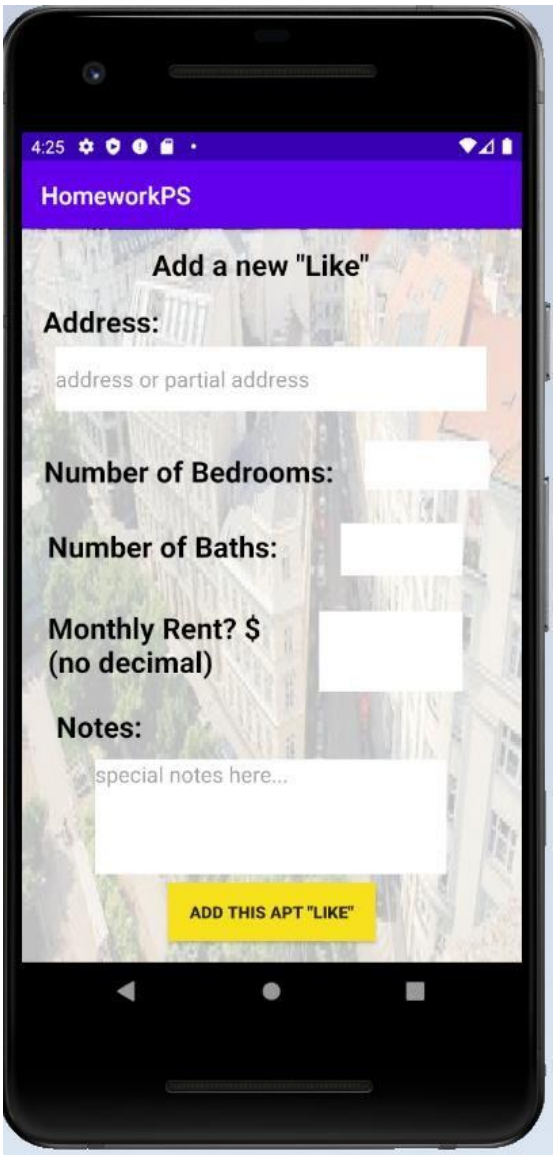


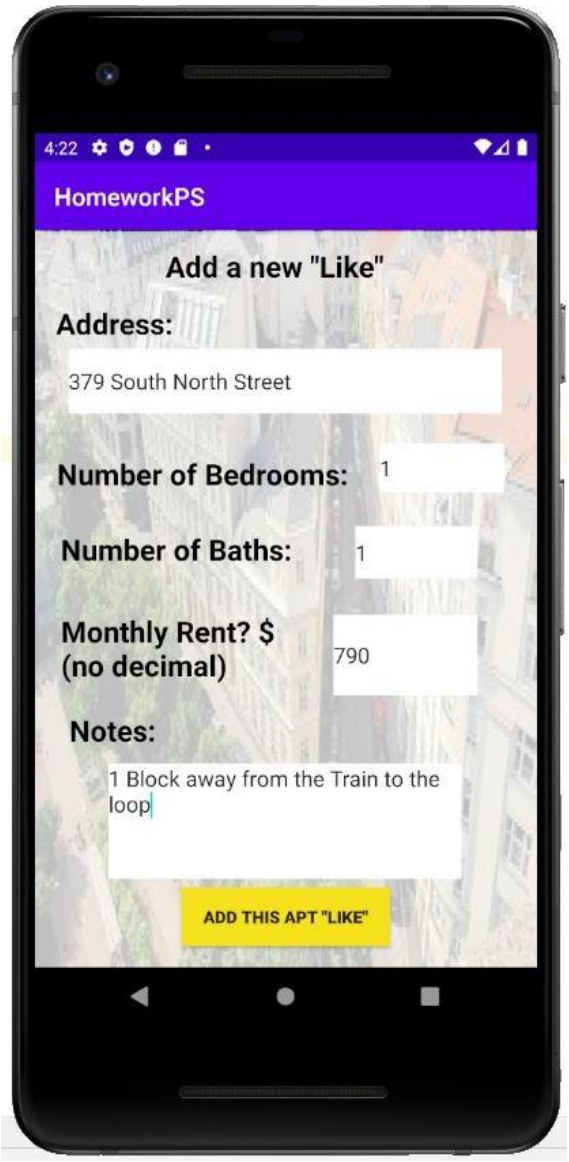
Main3Activity.java and activity_main3.xml

This activity and corresponding screen layout prompt the user for new information to add to the database. If the user has a saved preference for the background image, saved as a Shared Preference as described above, then this screen needs its background to be programmatically set to the preferred image, as was described for the Main2Activity and screen described above.

Here is how this screen should prompt the user:

….. with sample input:





When the user clicks the "Add Like" button, this information should be added to the database. Here is an example of the syntax to use for the INSERT INTO SQL statement:

```
String insertSQL = "INSERT INTO apts (ADDRESS, NUMBEDRMS, NUMBATHS, MONTHLYRENT,
NOTES) VALUES ('" +
        address + "', " + numBedRms + ", " + numBaths + ", " + rent + ", '"
        + notes + "');" ; mybooks.execSQL(insertSQL);
```

Note: Your variables may have different names than do mine!

GRADING RUBRIC:

TASK		MAX POINTS
MainActivity		
	Background color and text color	2
	Screen Layout	5
	Retrieval of selected background button image	2
	Saving of the user Shared Preference	6
	Button “View Likes” transfer to MainActivity2	3
	Button “Add a Like” transfer to MainActivity3	3
MainActivity2		
	Screen Layout	3
	Correct Background using shared preferences	4
	DB record listing	6
MainActivity3		
	Screen Layout	4
	Correct Background using shared preferences	3
	DB record addition	4
	TOTAL:	**45**