



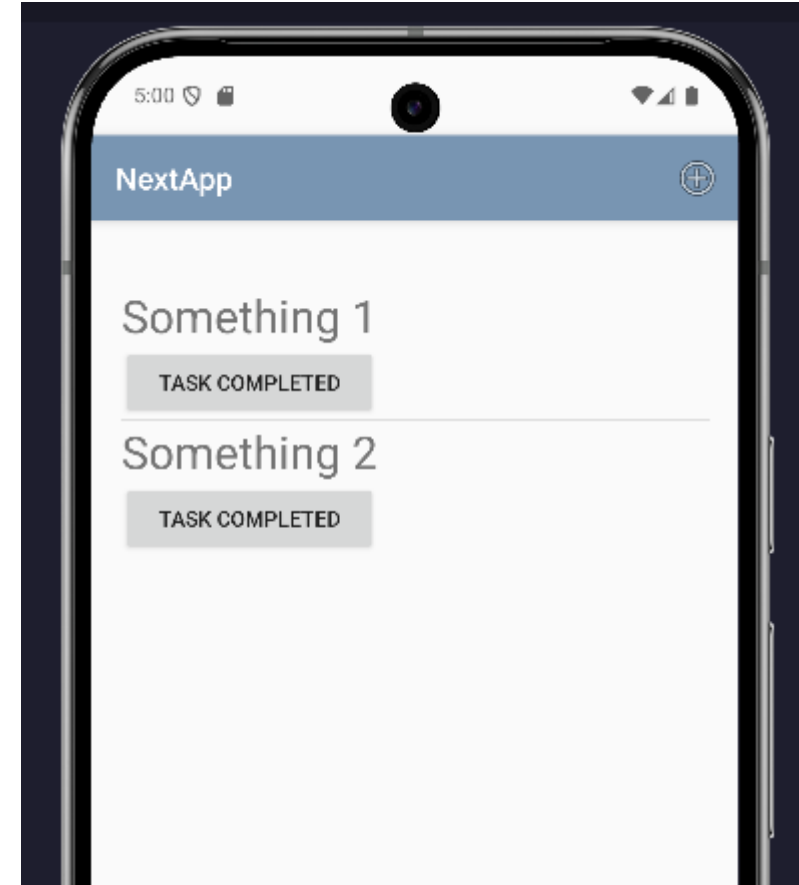
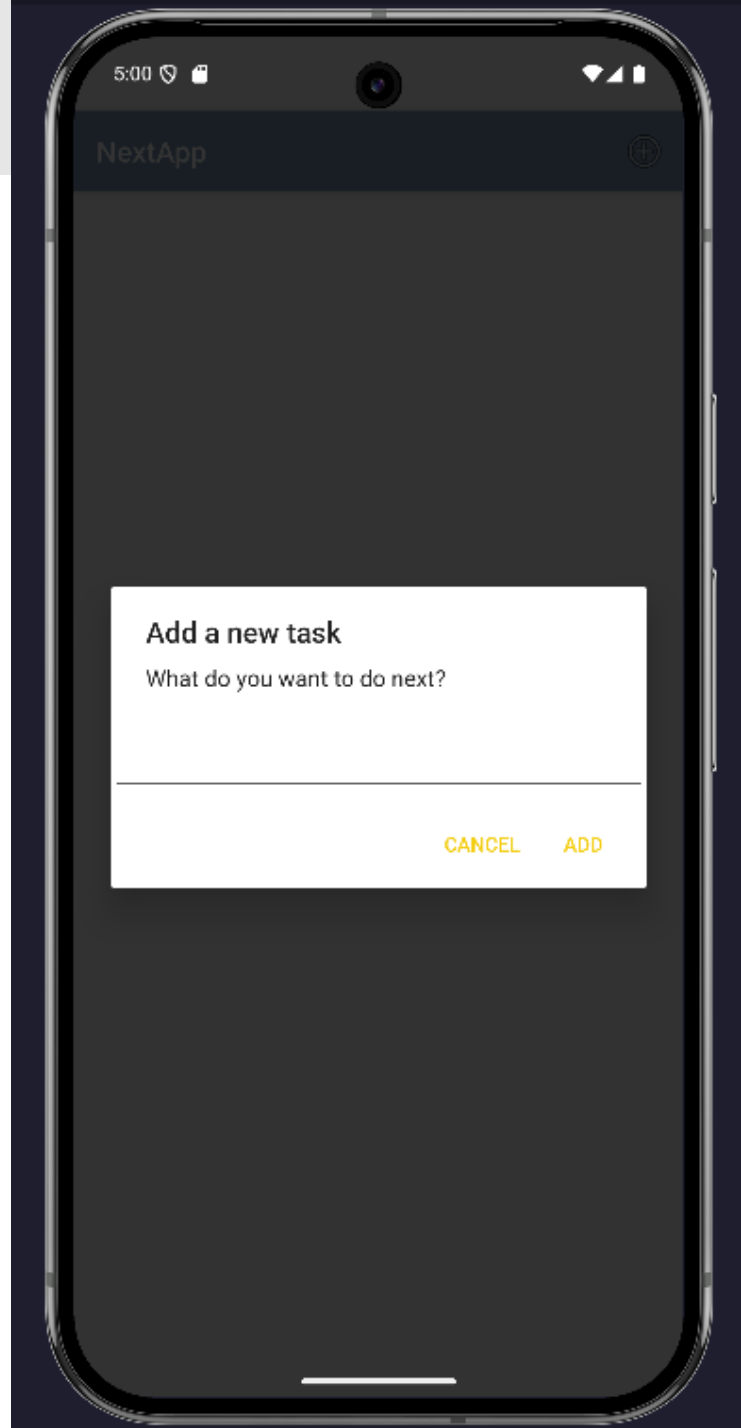
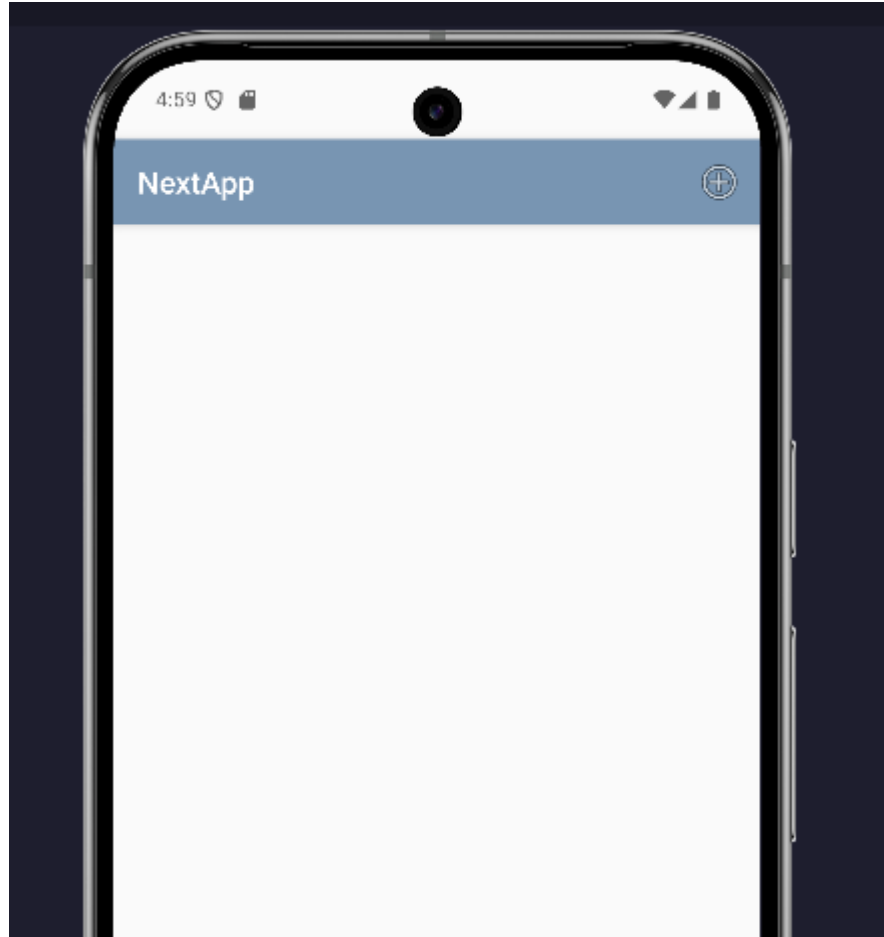
Todo-Tasks App

(With SQLiteDatabase)

Objective

- Create a todo-tasks app that stores data into SQLiteDatabase.

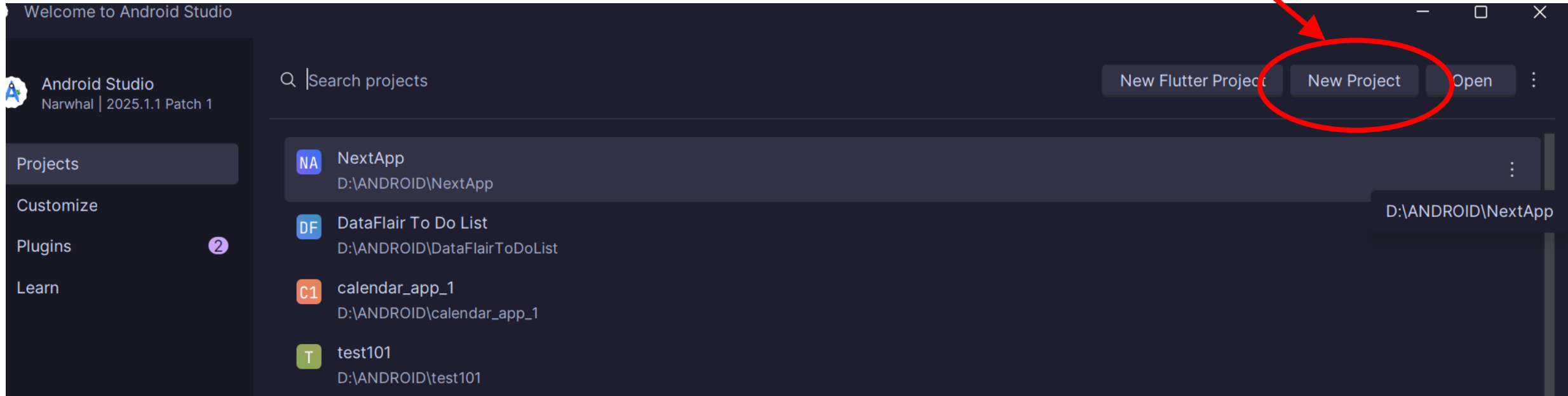
Final Output:



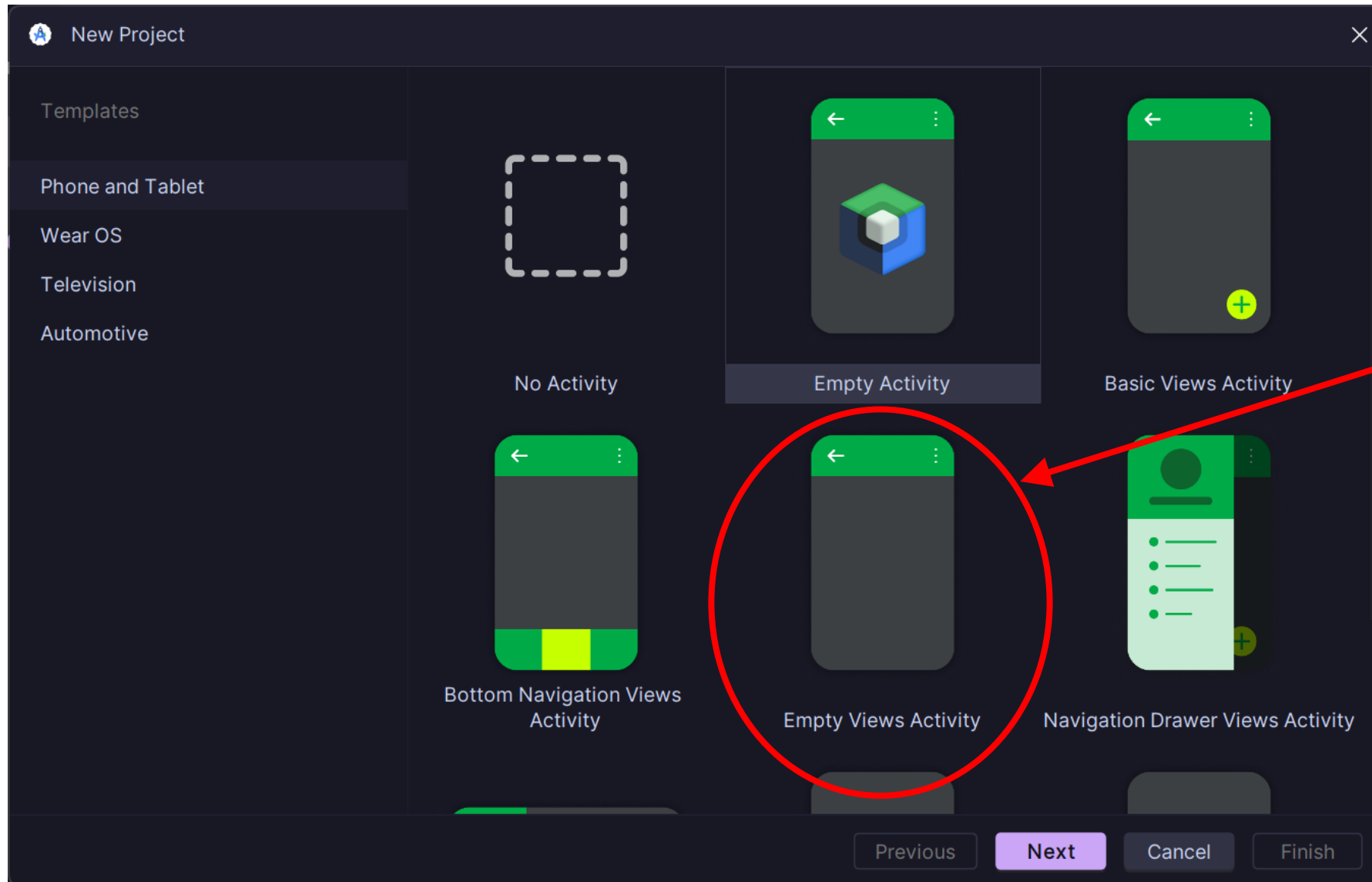
Section 1

Create a new project

Select
This



Select Empty Views Activity



Enter a Project Name

New Project

Empty Views Activity

Creates a new empty activity

Name: TodoTaskApp

Package name: inti.edu

Save location: D:\ANDROID\TodoTaskApp

Language: Java

Minimum SDK: API 24 ("Nougat"; Android 7.0)

Build configuration language: Kotlin DSL (build.gradle.kts) [Recommended]

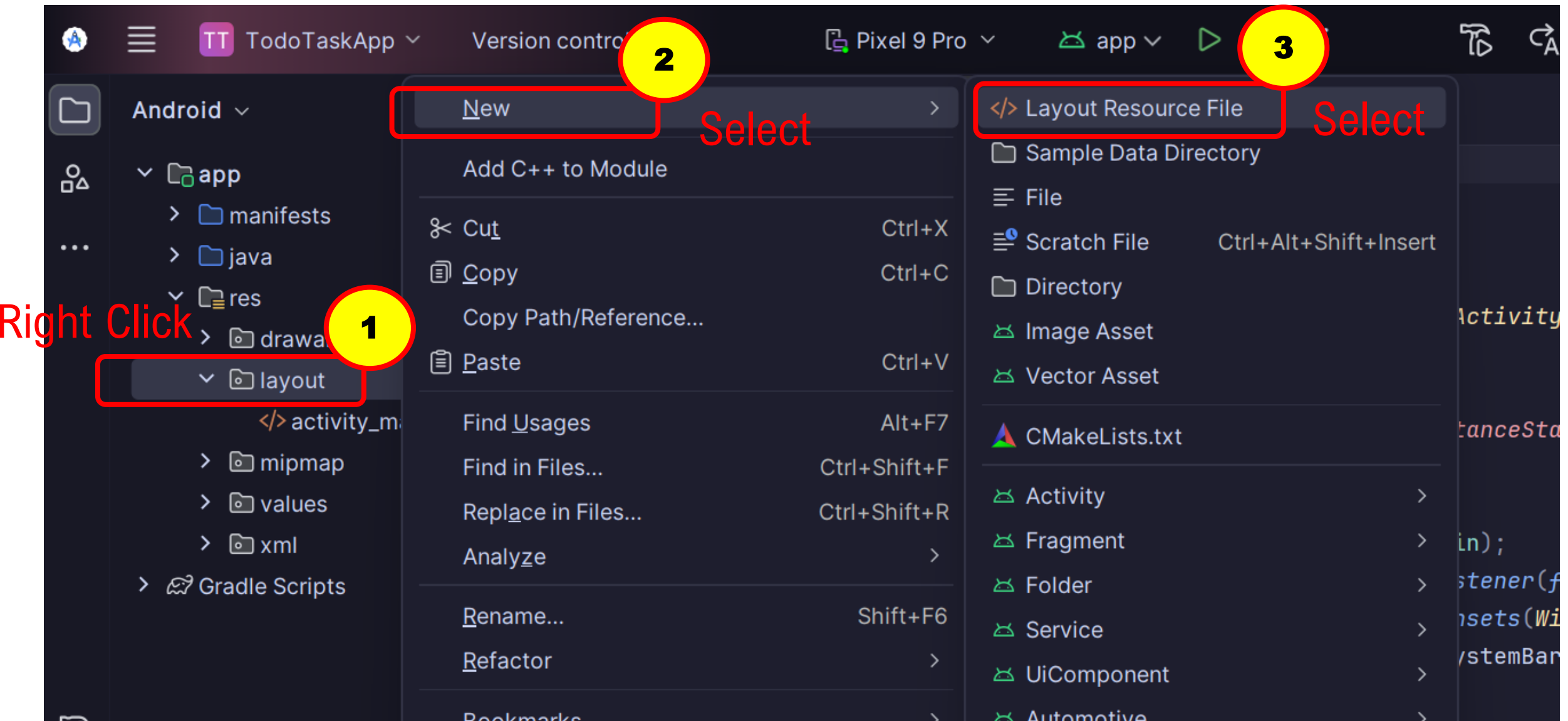
Previous Next Cancel Finish

Enter a new Project Name

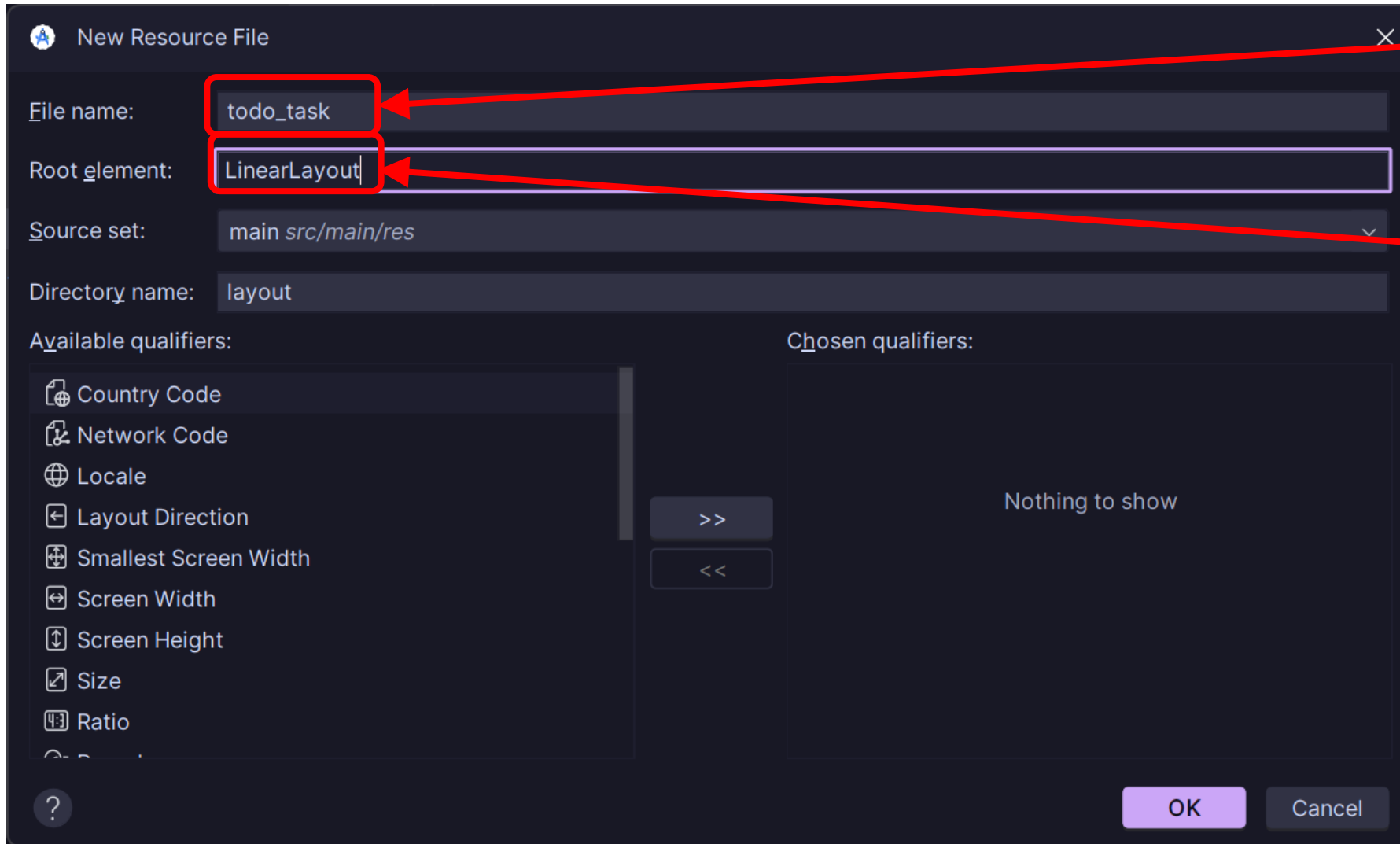
Enter a package name

Leave the rest as default and click "Finish"

Create a new Layout Resource File

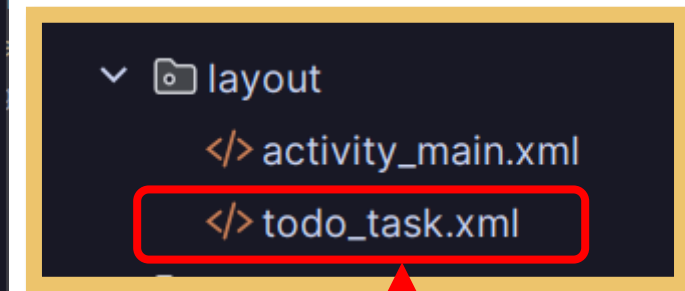


Enter “todo_task” as the file name and enter LinearLayout



File Name:
“todo_task”

Root Element:
“LinearLayout”



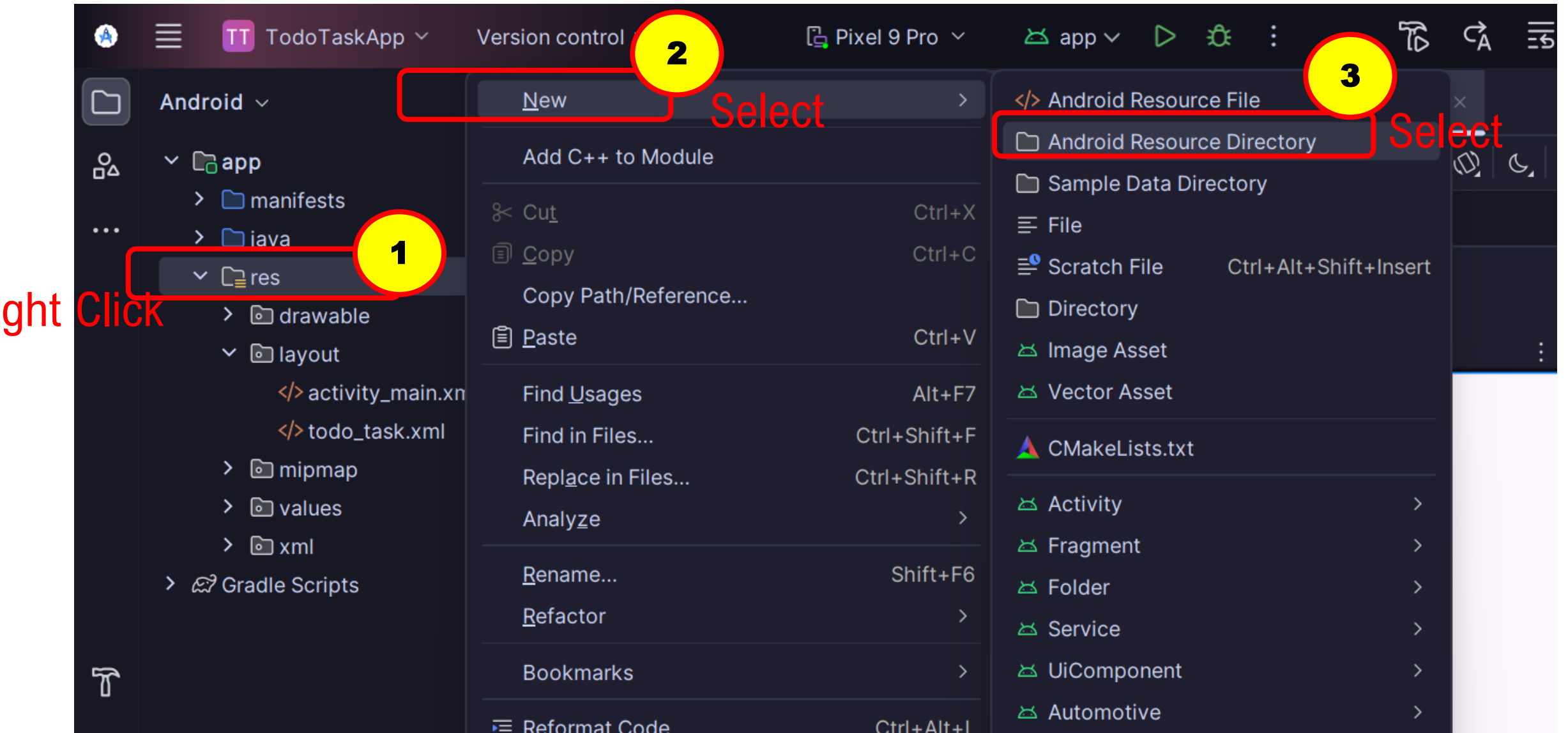
New layout File
Created

Update the todo_task.xml file as below:

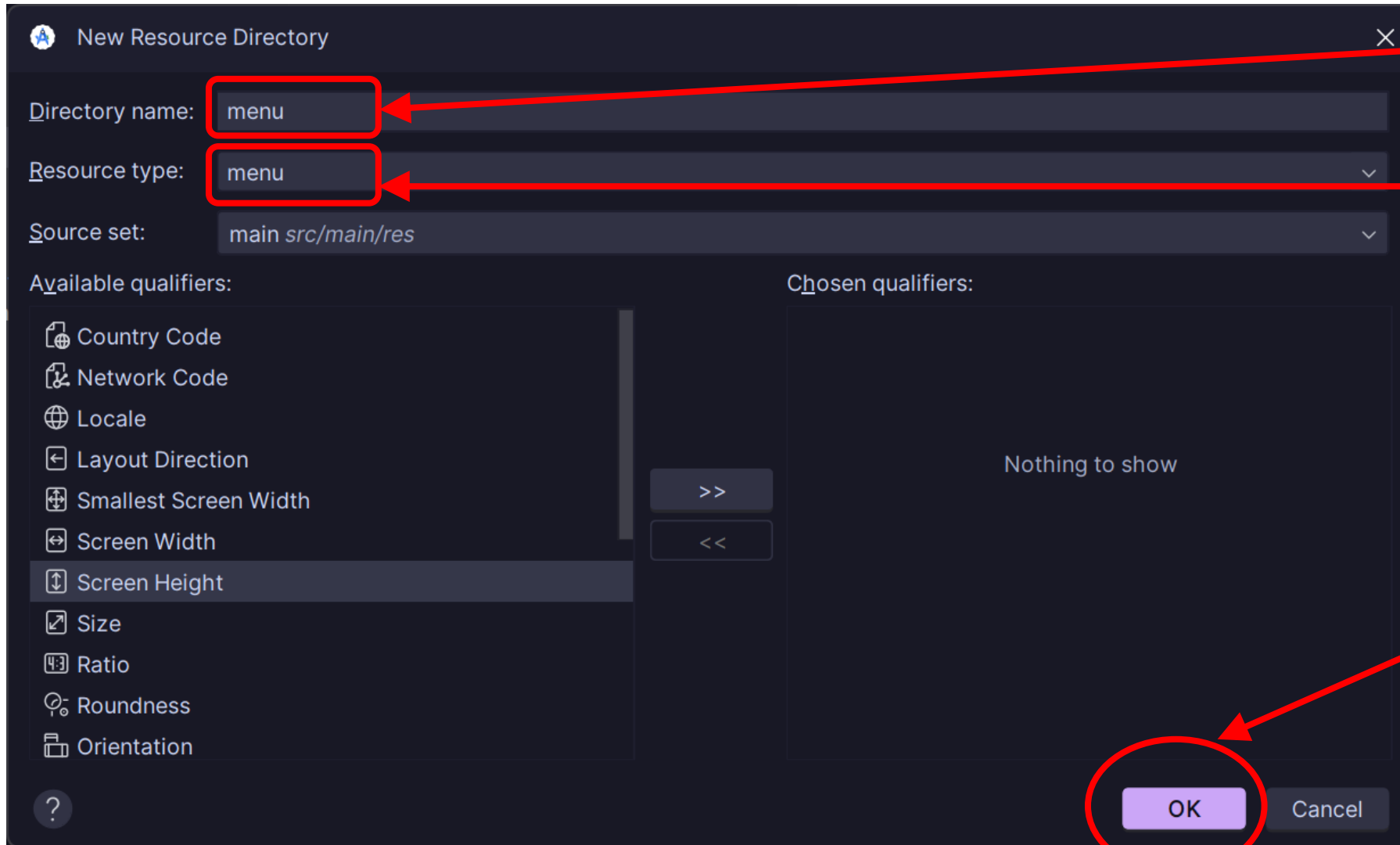
```
</> todo_task.xml ×
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:orientation="vertical"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      >
7
8      <TextView
9          android:id="@+id/title_task"
10         android:layout_width="match_parent"
11         android:layout_height="wrap_content"
12         android:textSize="30sp"
13         />
14
15     <Button
16         android:id="@+id/task_delete"
17         android:layout_width="match_parent"
18         android:layout_height="wrap_content"
19         android:text="Task Completed"
20         android:onClick="deleteTask" />
21
22 </LinearLayout>
```

Section 2

Create a new Android Resource Directory



Enter “menu” for the Directory name and select “menu” as the Resource Type



The screenshot shows the 'New Resource Directory' dialog box. The 'Directory name' field contains 'menu' and is highlighted with a red box. The 'Resource type' dropdown menu is also highlighted with a red box and shows 'menu' selected. The 'Source set' dropdown menu shows 'main src/main/res'. Below these fields are two lists: 'Available qualifiers' and 'Chosen qualifiers'. The 'Available qualifiers' list includes 'Country Code', 'Network Code', 'Locale', 'Layout Direction', 'Smallest Screen Width', 'Screen Width', 'Screen Height', 'Size', 'Ratio', 'Roundness', and 'Orientation'. The 'Chosen qualifiers' list is empty and contains the text 'Nothing to show'. At the bottom right, the 'OK' button is highlighted with a red circle. Red arrows point from the text annotations on the right to the 'Directory name' field, the 'Resource type' dropdown, and the 'OK' button.

New Resource Directory

Directory name: menu

Resource type: menu

Source set: main src/main/res

Available qualifiers:

- Country Code
- Network Code
- Locale
- Layout Direction
- Smallest Screen Width
- Screen Width
- Screen Height
- Size
- Ratio
- Roundness
- Orientation

Chosen qualifiers:

Nothing to show

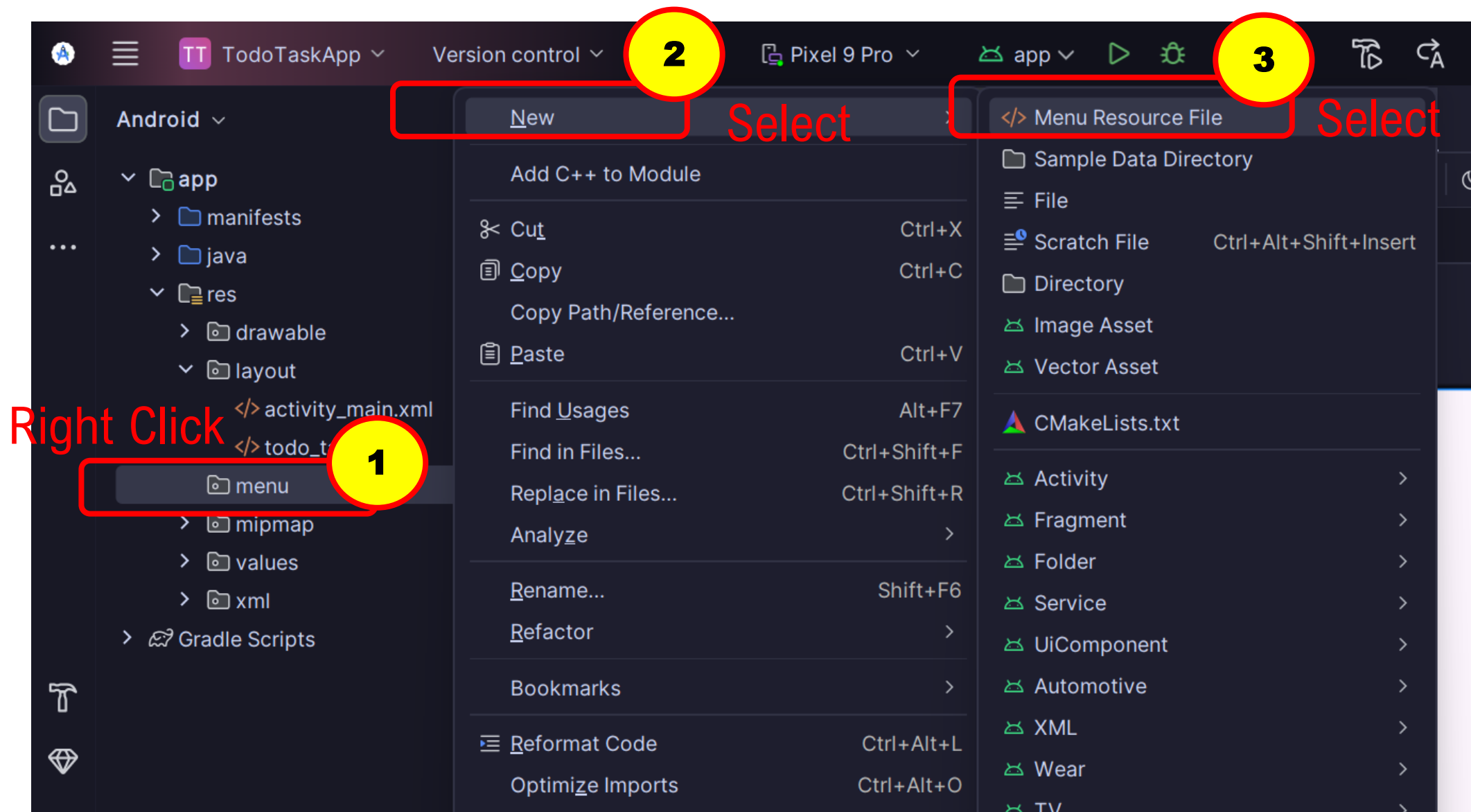
OK Cancel

Directory Name:
“menu”

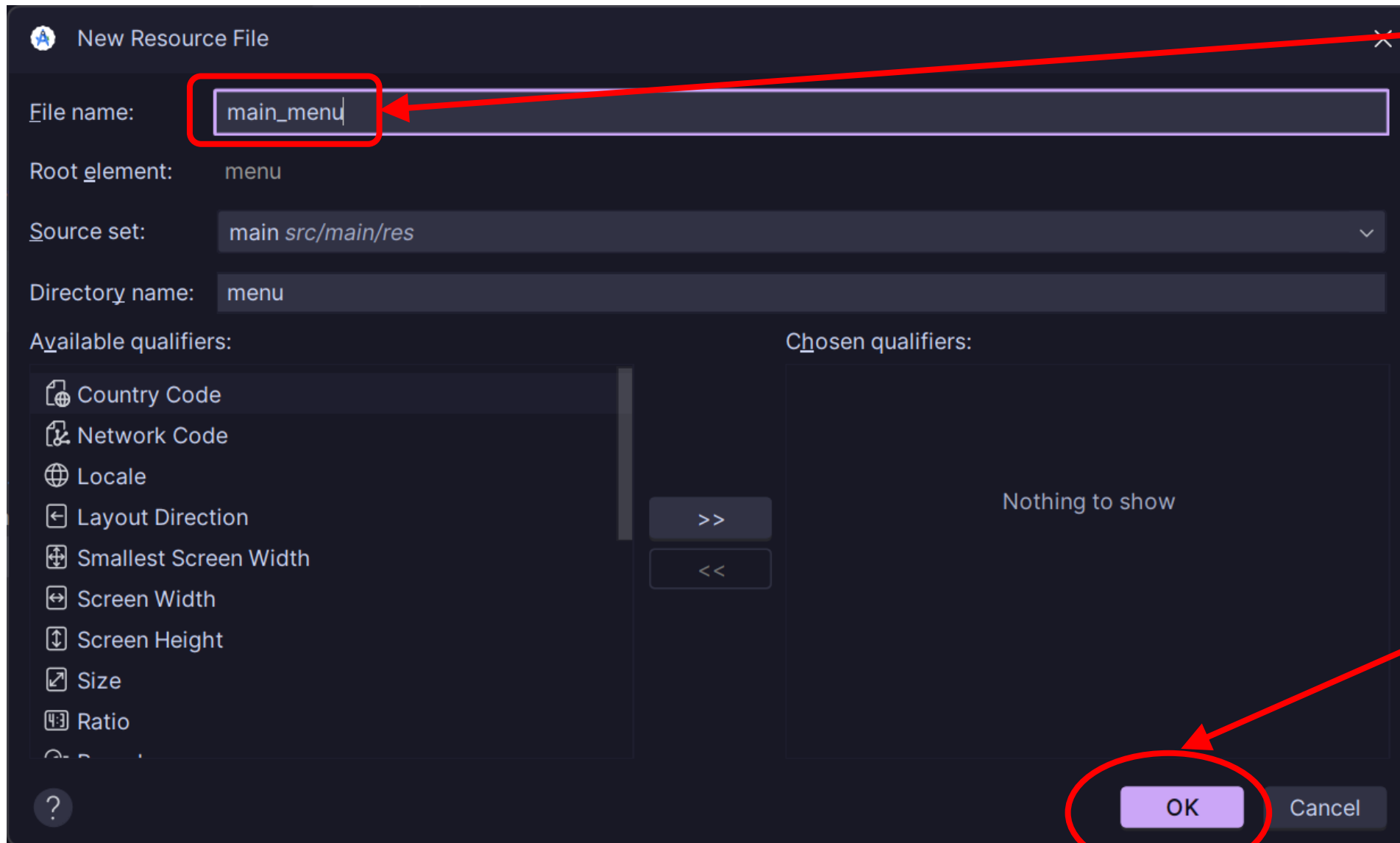
Resource Type:
“menu”

Click “OK”

Create a new Menu Resource File



Enter “main_menu” as the file name.



The screenshot shows the 'New Resource File' dialog box with the following fields and options:

- File name:** A text input field containing 'main_menu', which is highlighted by a red rectangle and a red arrow pointing to it from the text 'File Name: “main_menu”'.
- Root element:** A dropdown menu set to 'menu'.
- Source set:** A dropdown menu set to 'main src/main/res'.
- Directory name:** A text input field containing 'menu'.
- Available qualifiers:** A list of qualifiers including Country Code, Network Code, Locale, Layout Direction, Smallest Screen Width, Screen Width, Screen Height, Size, and Ratio.
- Chosen qualifiers:** A list that is currently empty, displaying 'Nothing to show'.
- Buttons:** '>>' and '<<' buttons are located between the two lists. At the bottom right, the 'OK' button is circled in red, with a red arrow pointing to it from the text 'Click “OK”'.

File Name:
“main_menu”

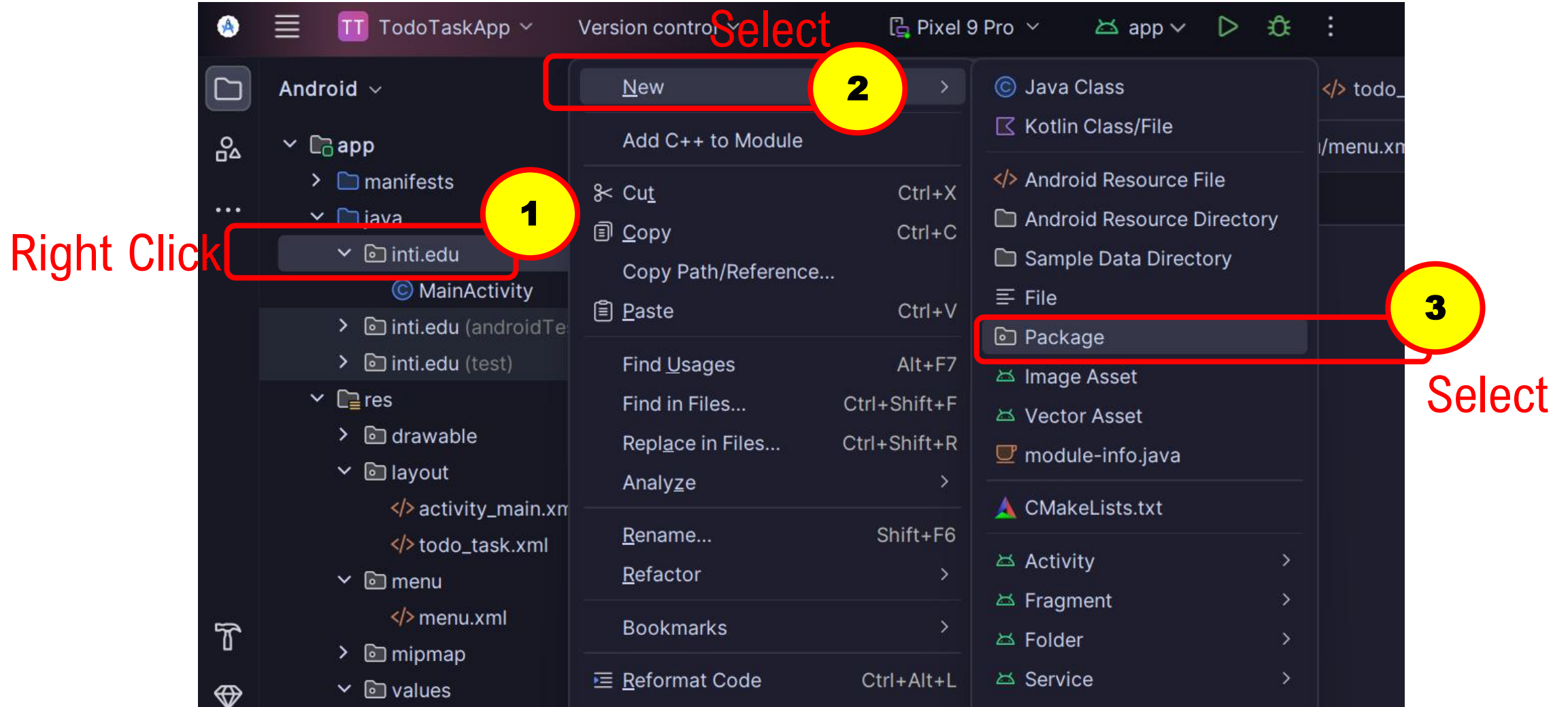
Click “OK”

Update the main_menu.xml file as below:

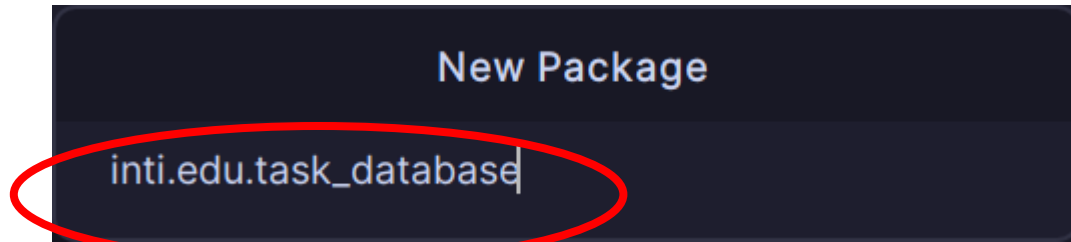
```
</> main_menu.xml ×
1  <?xml version="1.0" encoding="utf-8"?>
2  <menu xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto">
4
5      <item
6          android:id="@+id/action_add_task"
7          android:icon="@android:drawable/ic_menu_add"
8          android:title="Add Task"
9          app:showAsAction="always"
10
11      />
12  </menu>
```


Section 3

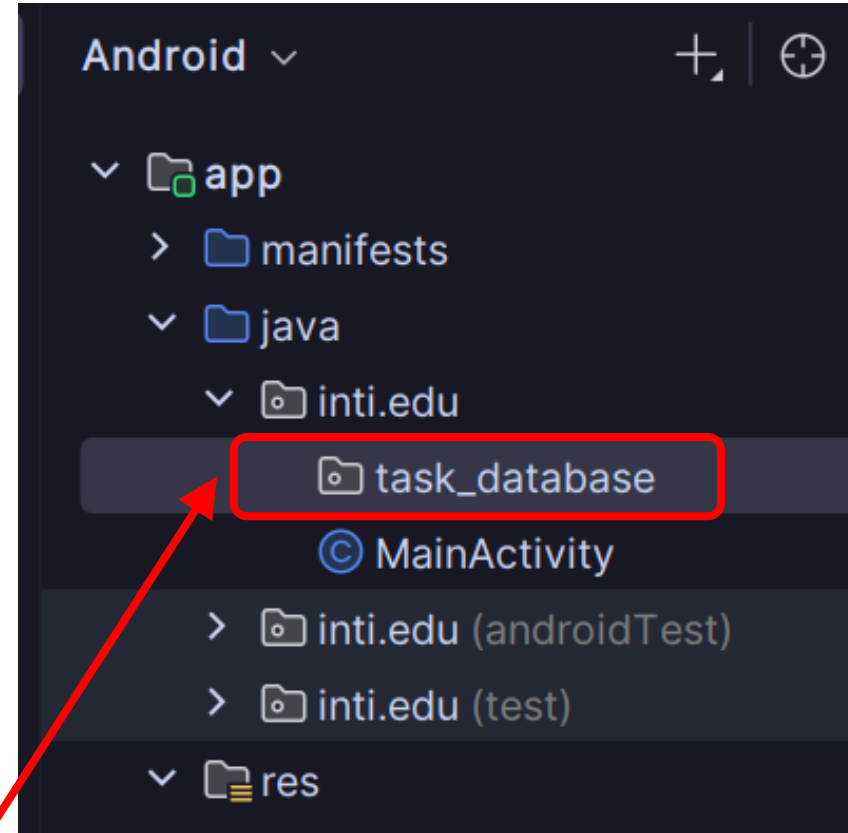
Create a new Java Package under the inti.edu package.



Enter “task_database” as the package name

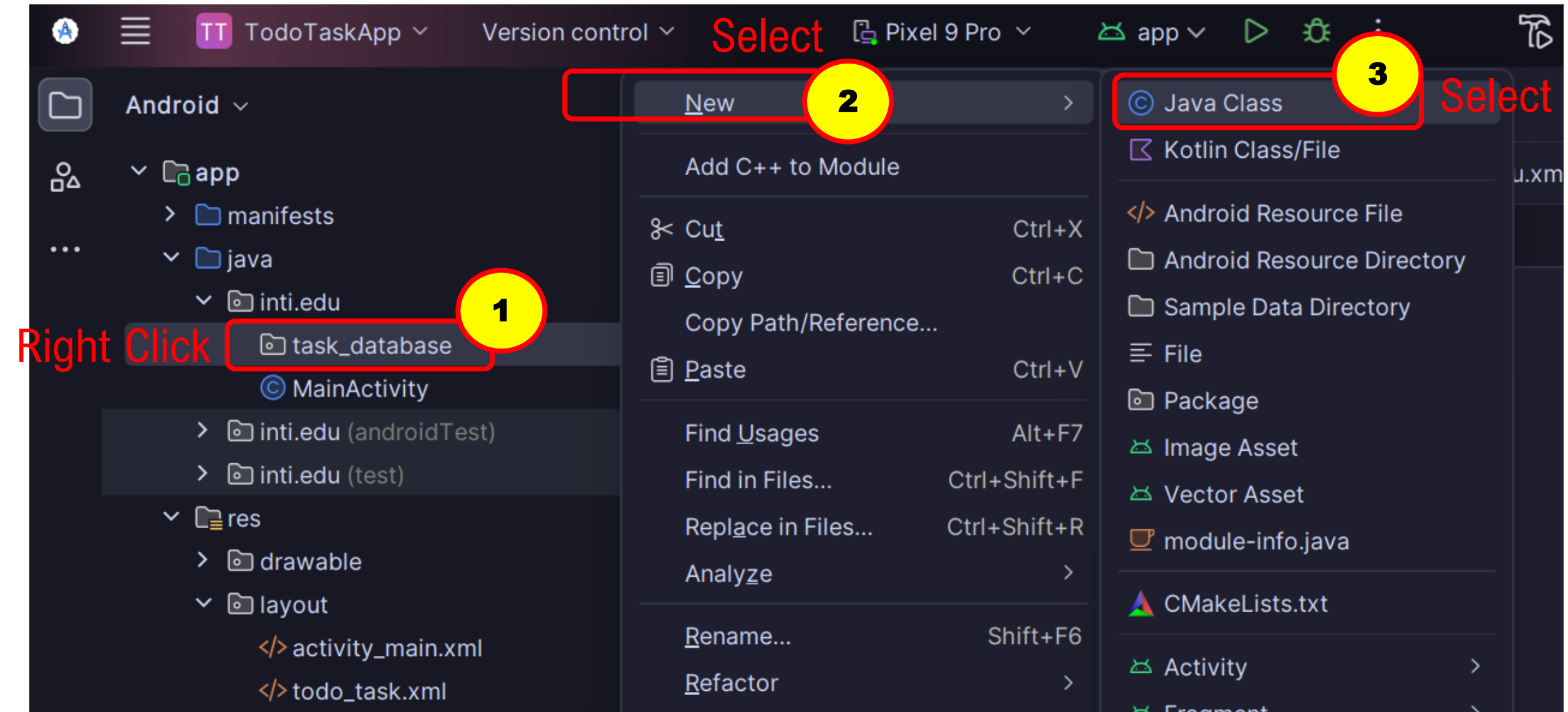


Package Name



New Package
Created

Create a new Java Class



Name the new Java Class “TaskContract”



Class Name:
“TaskContract”

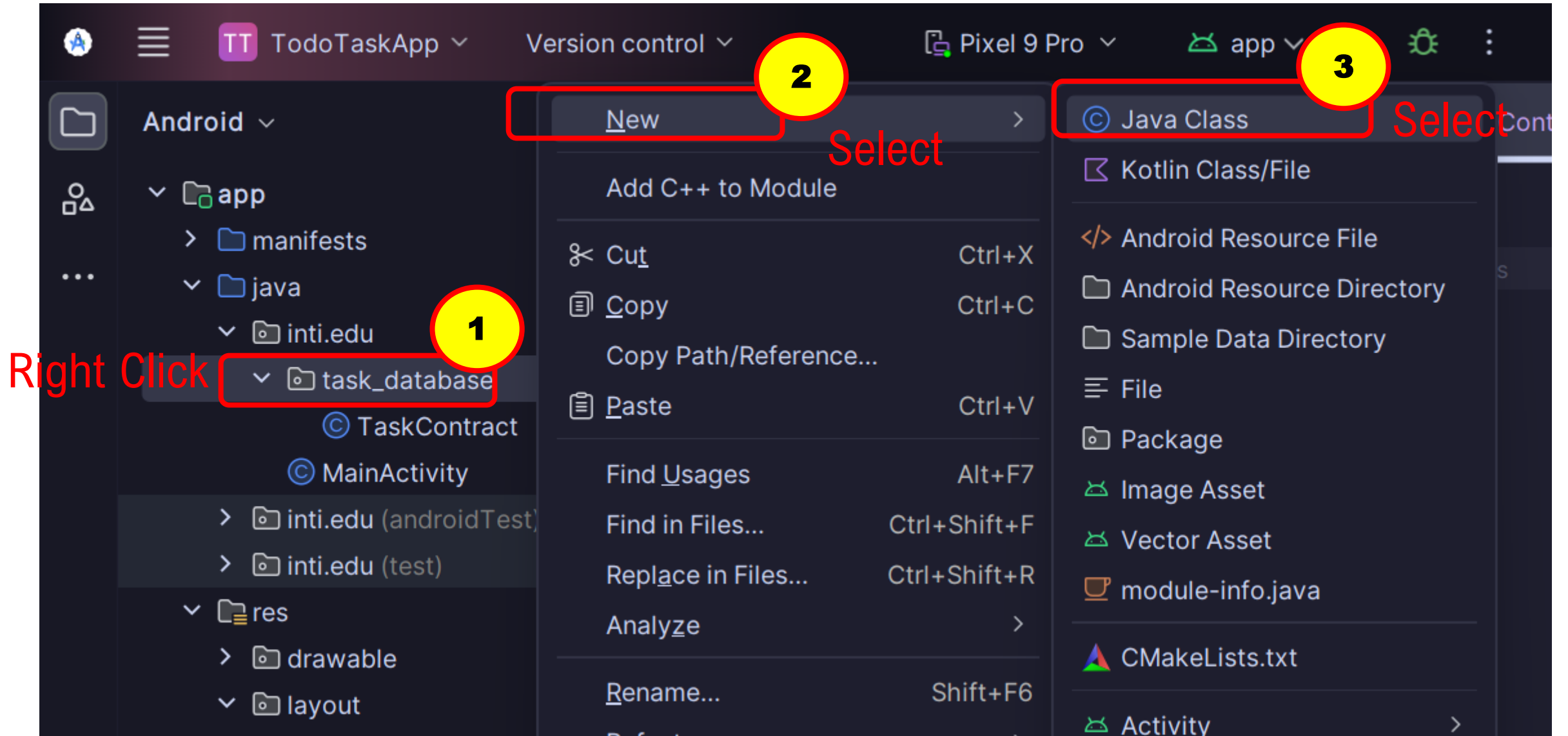


New Java Class
created

Enter the code below:

```
© TaskContract.java ×
1  package inti.edu.task_database;
2
3  import android.provider.BaseColumns;
4
5  public class TaskContract { 15 usages
6      public static final String DB_NAME = "todolist.db"; 1 usage
7      public static final int DB_VERSION = 1; 1 usage
8
9      public class TaskEntry implements BaseColumns{ 12 usages
10         public static final String TABLE = "tasks"; 5 usages
11         public static final String COL_TASK_TITLE = "title"; 5 usages
12     }
13 }
14
```

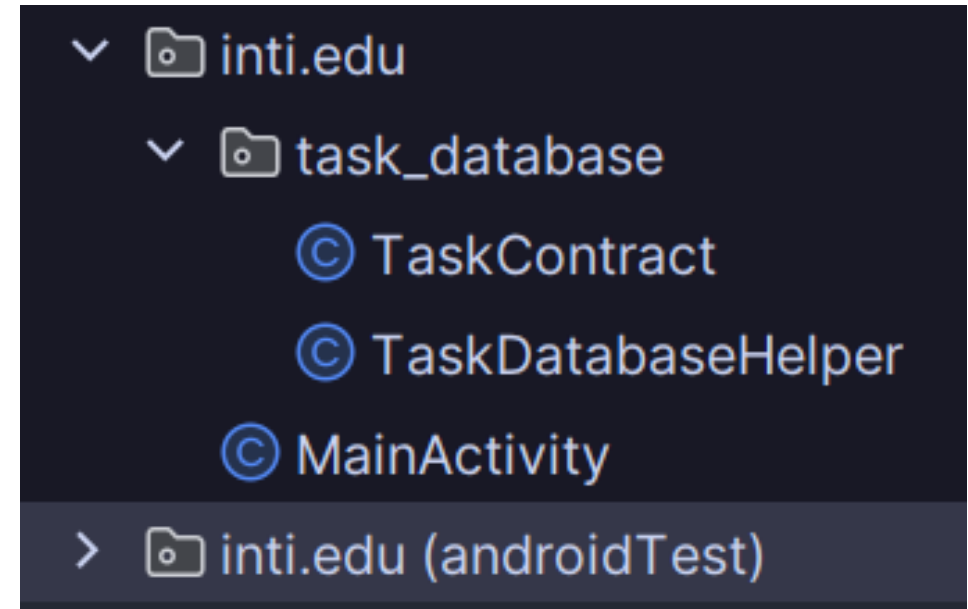
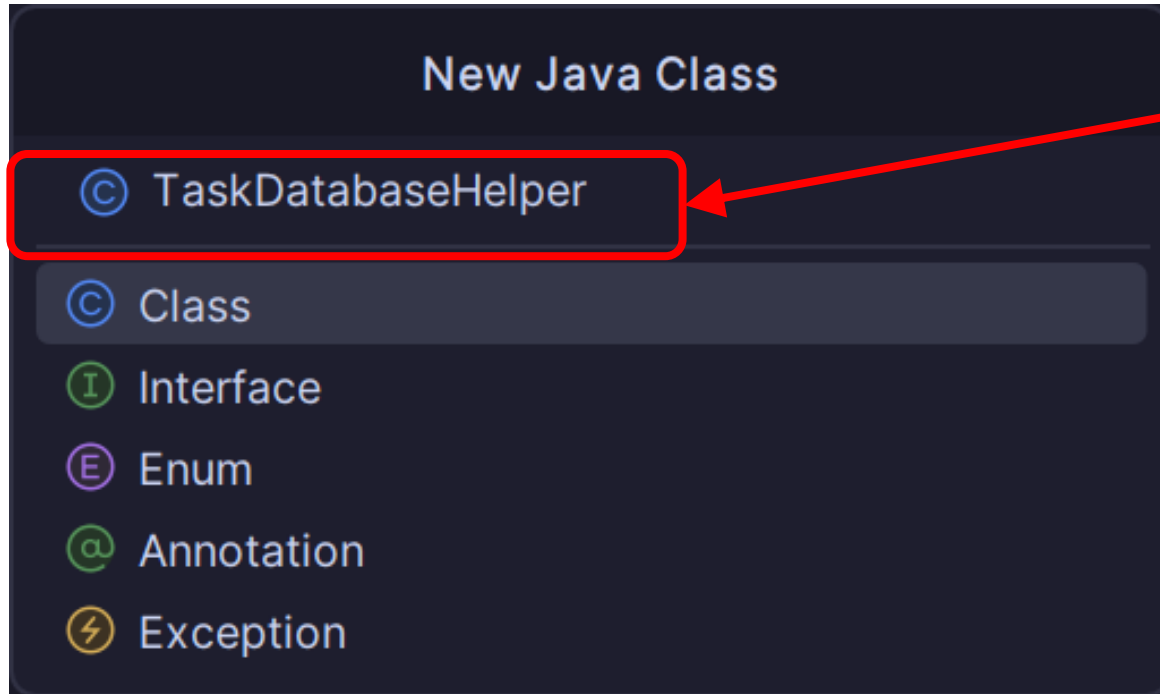
Create a new Java Class



Name the new Java Class “TaskDatabaseHelper”

Class Name:

“TaskDatabaseHelper”



New Java Class Created

Enter the code below:

```
TaskDatabaseHelper.java x
1 package inti.edu.task_database;
2
3 import android.content.Context;
4 import android.database.sqlite.SQLiteDatabase;
5 import android.database.sqlite.SQLiteOpenHelper;
6
7 public class TaskDatabaseHelper extends SQLiteOpenHelper{ 3 usages
8     public TaskDatabaseHelper(Context context){ 1 usage
9         super(context, TaskContract.DB_NAME, null, TaskContract.DB_VERSION);
10    }
11
12    @Override
13    public void onCreate(SQLiteDatabase db){
14        String createTable = "CREATE TABLE " + TaskContract.TaskEntry.TABLE + " ( " +
15            TaskContract.TaskEntry._ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
16            TaskContract.TaskEntry.COL_TASK_TITLE + " TEXT NOT NULL);";
17        db.execSQL(createTable);
18    }
19
20    @Override no usages
21    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion){
22        db.execSQL("DROP TABLE IF EXISTS " + TaskContract.TaskEntry.TABLE);
23        onCreate(db);
24    }
25 }
```

Section 4

Add these 3 colors

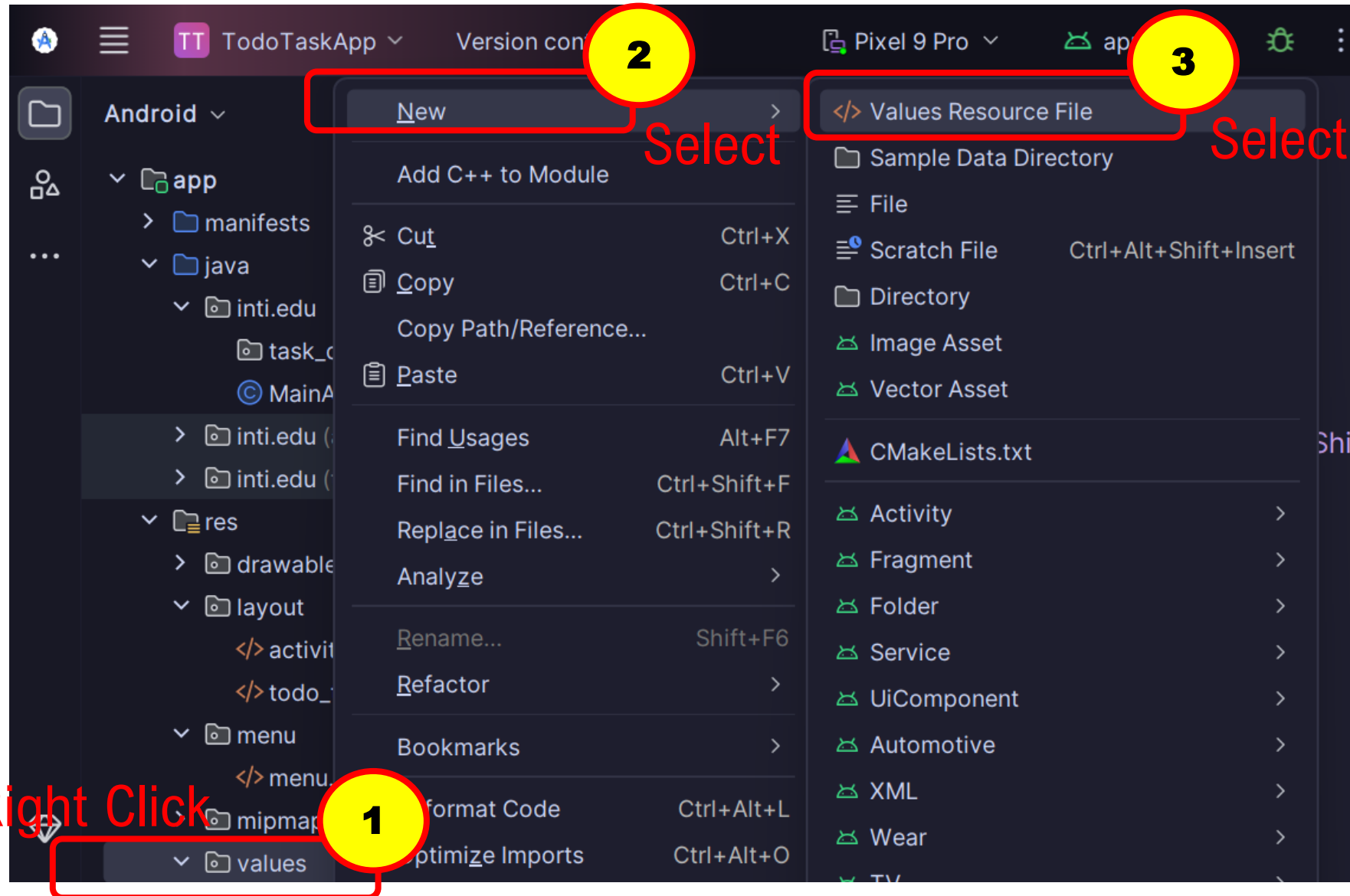
- Add these 3 colours to **colors.xml**

```
<color name="colorPrimary">#7895B2</color>  
<color name="colorPrimaryDark">#C3BEF0</color>  
<color name="colorAccent">#F4CE14</color>
```

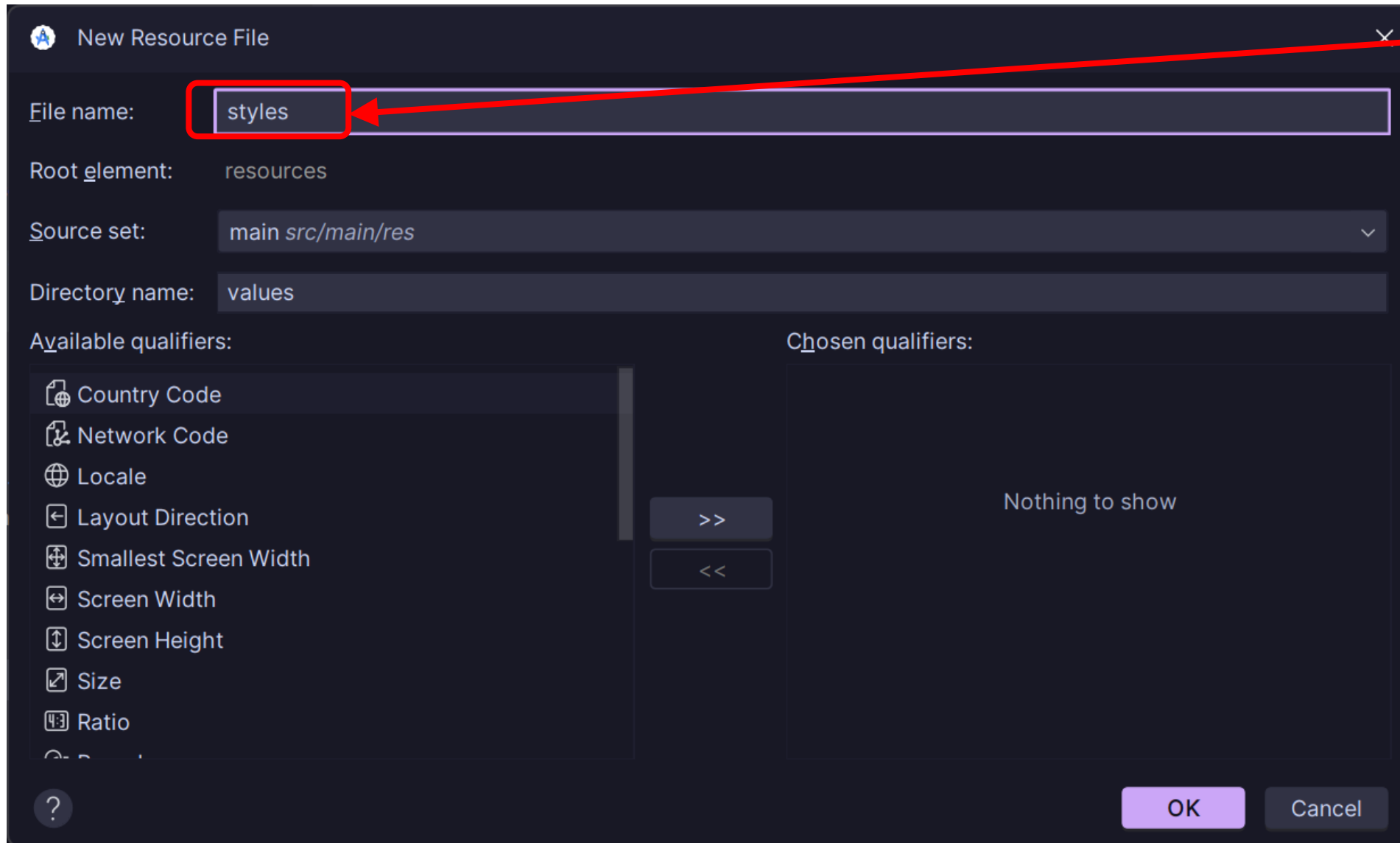
Explore more interesting colours from this website:

<https://colorhunt.co/>

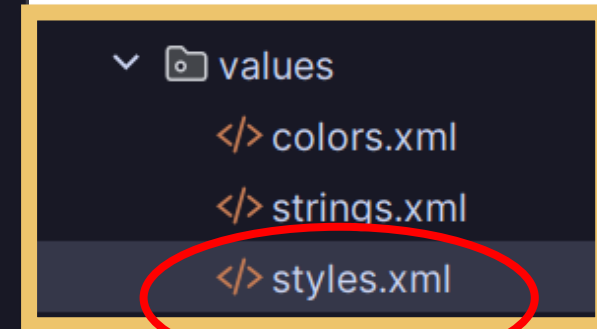
Create a new Values Resource File



Enter “styles” as the file name.



File Name:
“styles”



New value
resource file
created

Enter the code below into **styles.xml**:

```
</> styles.xml ×  
1  <?xml version="1.0" encoding="utf-8"?>  
2  <resources>  
3      <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">  
4          <item name="colorPrimary">@color/colorPrimary</item>  
5          <item name="colorPrimaryDark">@color/colorPrimaryDark</item>  
6          <item name="colorAccent">@color/colorAccent</item>  
7      </style>  
8  </resources>  
9  
10
```

Update the **activity_main.xml** file as below

```
</> activity_main.xml ×
1  <?xml version="1.0" encoding="utf-8"?>
2  ©  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3
4      xmlns:tools="http://schemas.android.com/tools"
5      android:id="@+id/main"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      ✨ android:paddingBottom="50dp"
9      android:paddingLeft="20dp"
10     android:paddingRight="20dp"
11     android:paddingTop="150dp"
12     android:orientation="vertical"
13     tools:context=".MainActivity">
14
15     <ListView
16         android:id="@+id/list_todo"
17         android:layout_width="wrap_content"
18         android:layout_height="wrap_content"
19         />
20
21 </LinearLayout>
```

Section 5

Import the following classes:

```
© MainActivity.java x
1  package inti.edu;
2
3  import android.os.Bundle;
4
5  import androidx.activity.EdgeToEdge;
6  import androidx.appcompat.app.AppCompatActivity;
7  import androidx.appcompat.app.AlertDialog;
8
9  import android.database.sqlite.SQLiteDatabase;
10 import android.database.Cursor;
11
12 import android.view.Menu;
13 import android.view.MenuItem;
14 import android.view.View;
15
16 import android.widget.EditText;
17 import android.widget.ListView;
18 import android.widget.TextView;
19 import android.widget.AdapterView;
20
21 import java.util.ArrayList;
22
23 import android.content.ContentValues;
24 import android.content.DialogInterface;
25
```

```
import inti.edu.task_database.TaskDatabaseHelper;
import inti.edu.task_database.TaskContract;
```

```
public class MainActivity extends AppCompatActivity {  
    private static final String TAG = "MainActivity";  
    private TaskDatabaseHelper taskHelper;  
    private ListView TaskList;  
    private ArrayAdapter<String> arrAdapter;
```

Add these code

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    EdgeToEdge.enable(this);  
    setContentView(R.layout.activity_main);
```

```
    taskHelper = new TaskDatabaseHelper(this);  
    TaskList = (ListView) findViewById(R.id.list_todo);  
    updateUI();  
}
```

Add these code

Add codes A, B, C, and D
shown in the following slides

Code A

```
@Override 6 usages  
public boolean onCreateOptionsMenu(Menu menu){  
    getMenuInflater().inflate(R.menu.main_menu, menu);  
    return super.onCreateOptionsMenu(menu);  
}
```

Code B

```
if(item.getItemId() == R.id.action_add_task){
    final EditText taskEdit = new EditText(this);
    AlertDialog dialog = new AlertDialog.Builder(this)
        .setTitle("Add a new task")
        .setMessage("What do you want to do next?")
        .setView(taskEdit)
        .setPositiveButton("Add", new DialogInterface.OnClickListener(){

            @Override
            public void onClick(DialogInterface dialog, int which){
                String task = String.valueOf(taskEdit.getText());
                SQLiteDatabase db = taskHelper.getWritableDatabase();
                ContentValues values = new ContentValues();
                values.put(TaskContract.TaskEntry.COL_TASK_TITLE, task);
                db.insertWithOnConflict(TaskContract.TaskEntry.TABLE, null, values, SQLiteDatabase.CONFLICT_REPLACE);
                db.close();
                updateUI();
            }
        })
        .setNegativeButton("Cancel", null).create();
    dialog.show();
    return true;
}
```

```
@Override 3 usages
public boolean onOptionsItemSelected(MenuItem item){
```

```
    }else{
        return super.onOptionsItemSelected(item);
    }
}
```

Code C

```
public void deleteTask(View view){ 1 usage
    View parent = (View) view.getParent();
    TextView taskTextView = (TextView) parent.findViewById(R.id.title_task);
    String task = String.valueOf(taskTextView.getText());
    SQLiteDatabase db = taskHelper.getWritableDatabase();
    db.delete(TaskContract.TaskEntry.TABLE, TaskContract.TaskEntry.COL_TASK_TITLE +
        " = ?", new String[]{task});
    db.close();
    updateUI();
}
```

Code D

```
public void updateUI(){ 3 usages
    ArrayList<String> taskList = new ArrayList<>();
    SQLiteDatabase db = taskHelper.getReadableDatabase();
    Cursor cursor = db.query(TaskContract.TaskEntry.TABLE,
        new String[]{TaskContract.TaskEntry._ID, TaskContract.TaskEntry.COL_TASK_TITLE},
        null, null, null, null, null);

    while(cursor.moveToNext()){
        int idx = cursor.getColumnIndex(TaskContract.TaskEntry.COL_TASK_TITLE);
        taskList.add(cursor.getString(idx));
    }

    if(arrAdapter == null){
        arrAdapter = new ArrayAdapter<>(this, R.layout.todo_task, R.id.title_task, taskList);
        TaskList.setAdapter(arrAdapter);
    }else{
        arrAdapter.clear();
        arrAdapter.addAll(taskList);
        arrAdapter.notifyDataSetChanged();
    }
    cursor.close();
    db.close();
}
```

Update AndroidManifest.xml File

```
android:theme="@style/AppTheme">
```

**Code is Complete.
Now Check the Output.**

