**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WEEK\_7\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Topics:**

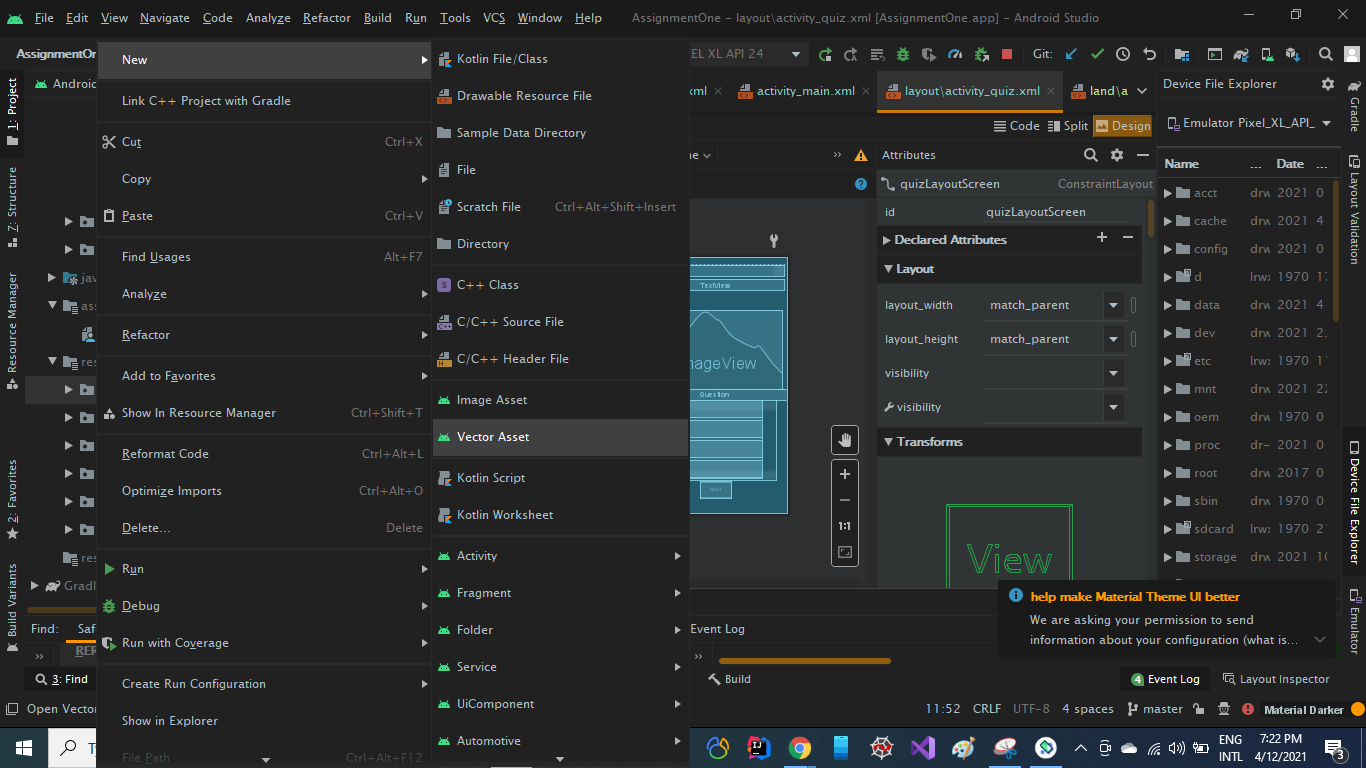
1. **Drawer Navigation**
2. **Recycler List View**

**1. Drawer Navigation**

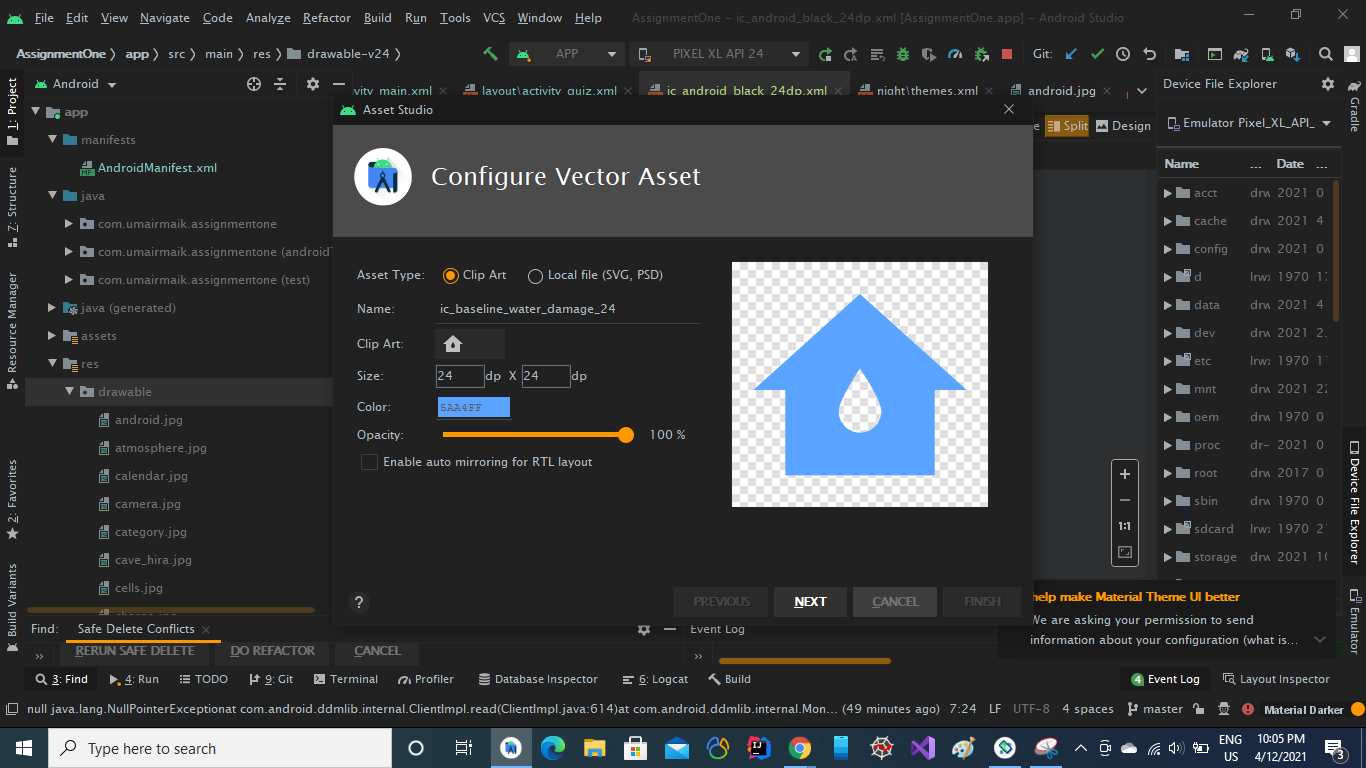
I’ll make a drawer navigation bar in my developed android App i.e., Quiz\_Box (our first assignment).

First create an assets folder in the project inside drawable.

***drawable>>new>>vector asset***



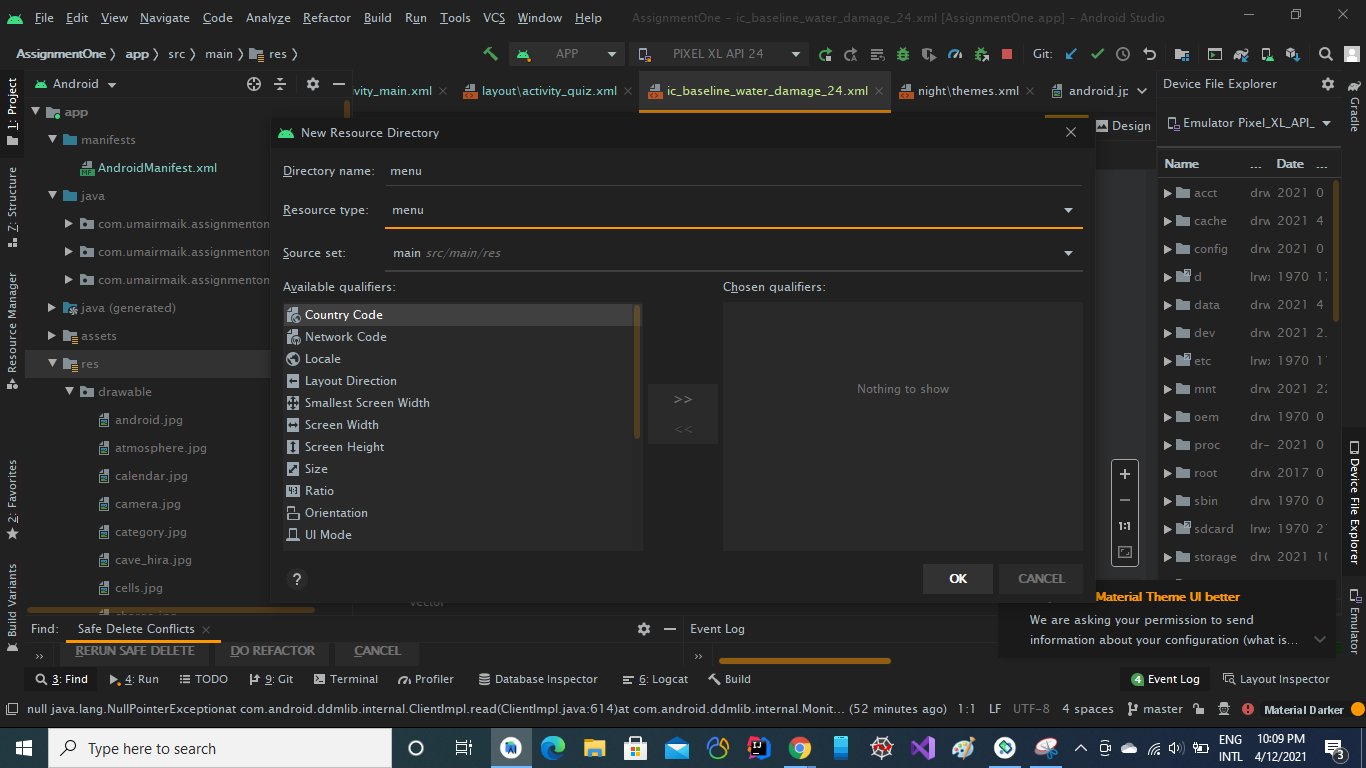
Then create a resource directory as  
 res>>new>>resource directory



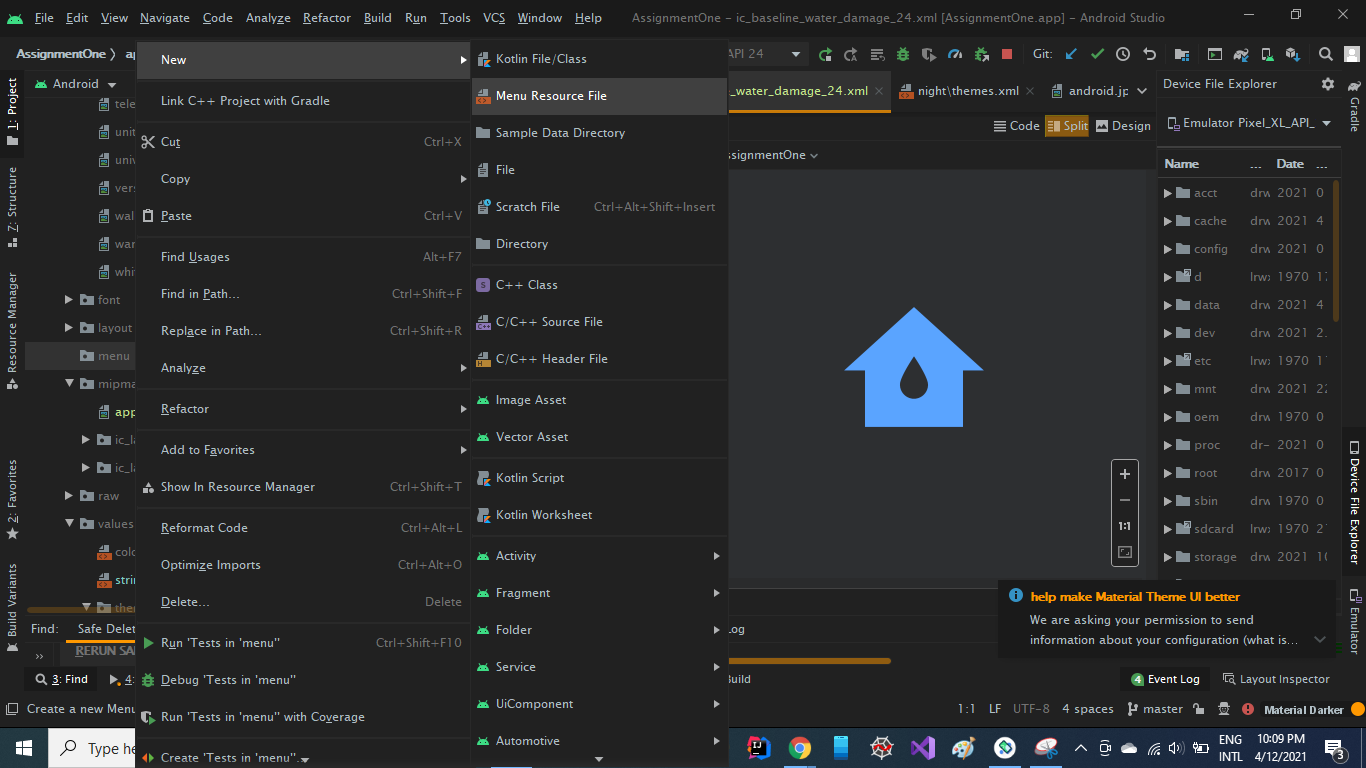
Select an icon from **clip art and press Next button.**

Now add a new resource directory for Menu of drawer.

**res>>new>>android resource directory**

****

Now add new resource file in the above created directory

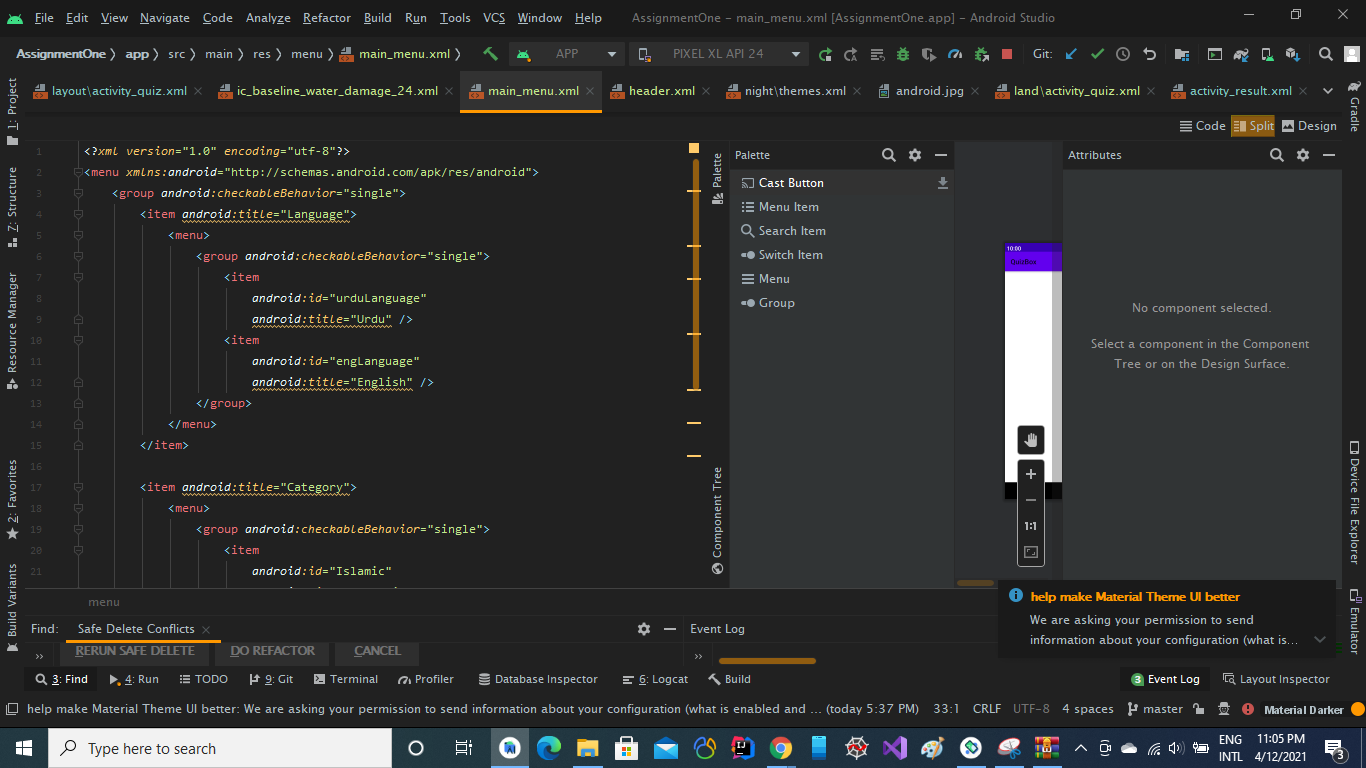


**Now we have following files available**

1. menu>>main\_menu.xml

2. layout>>header.xml

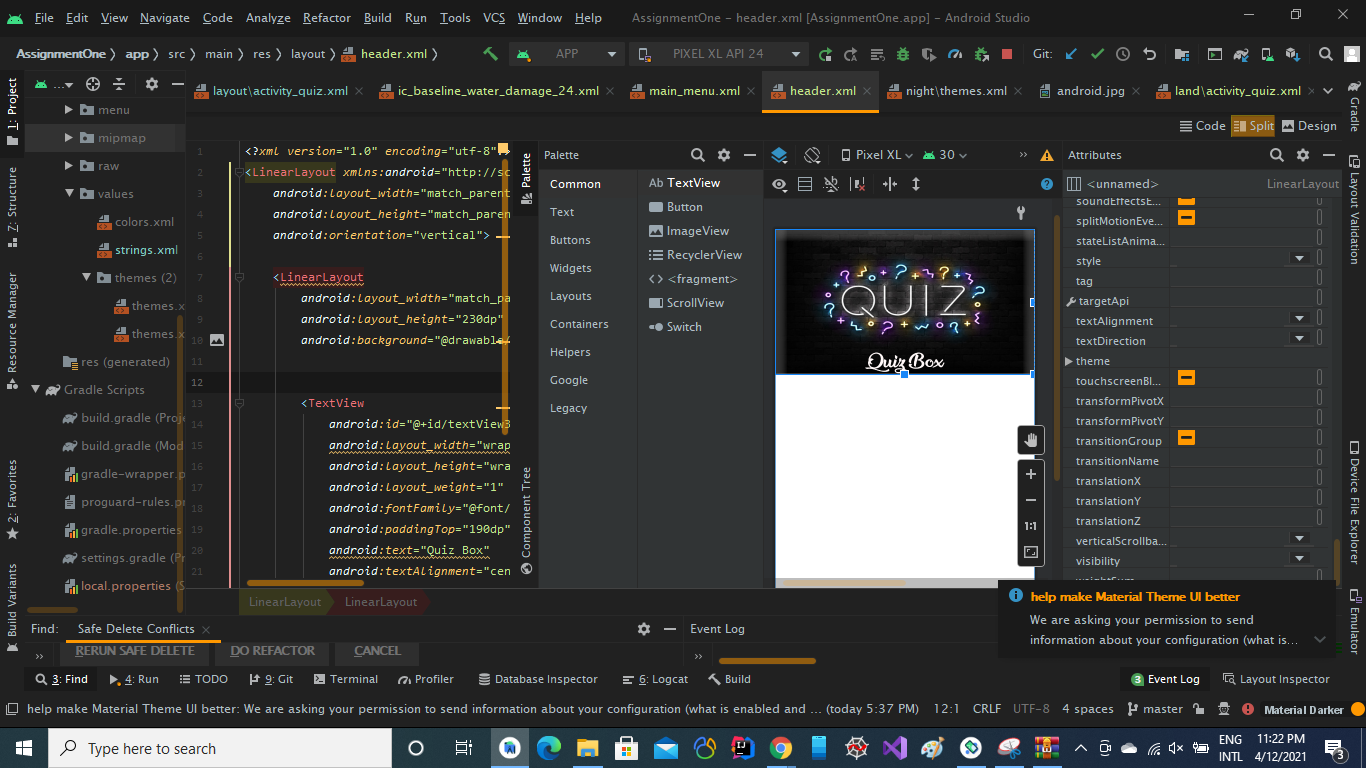
Now we’ll add items in our main\_menu.xml



Here we have added two items which has further sub-items  
First is Language of App i.e., **English or Urdu**

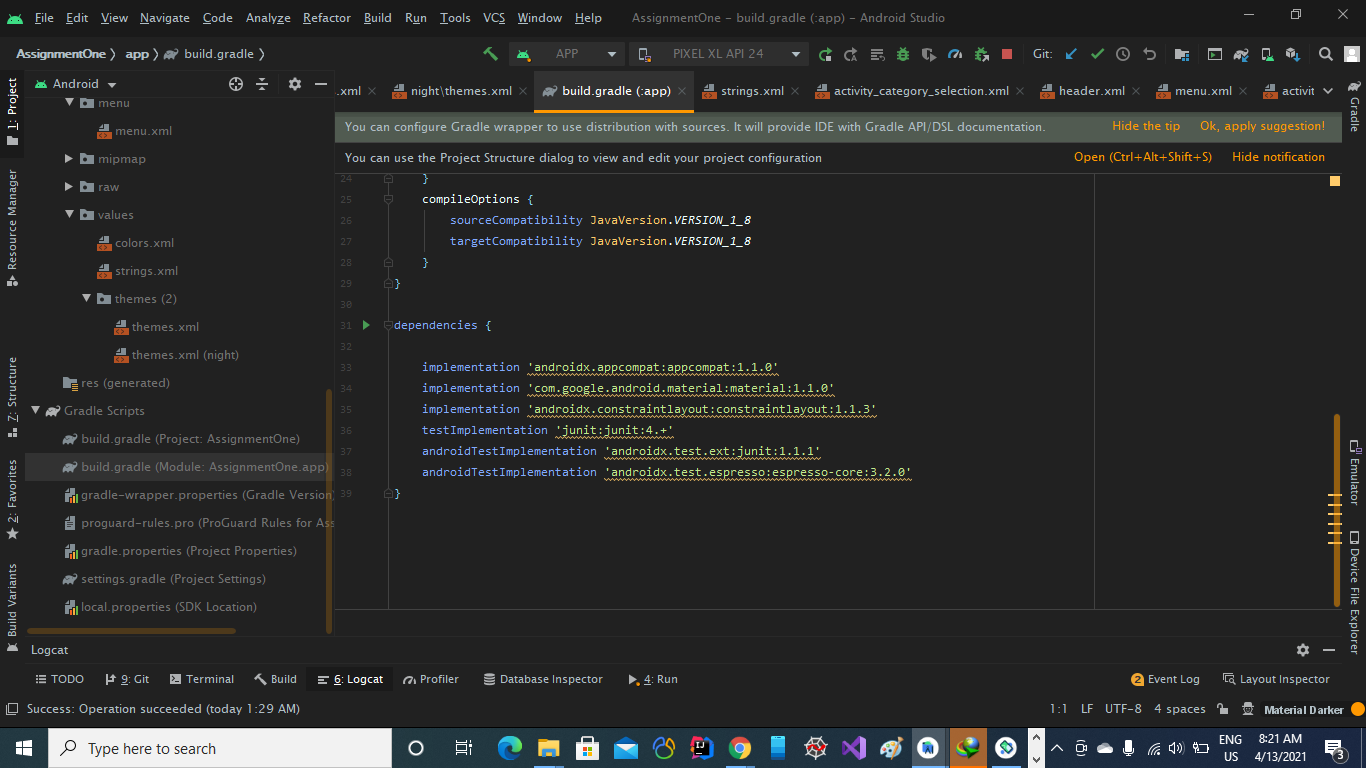
Secondly, we have a menu of Quiz Category from where user will select what type of quiz he/she wants?

Now we’ll add data in our header.xml

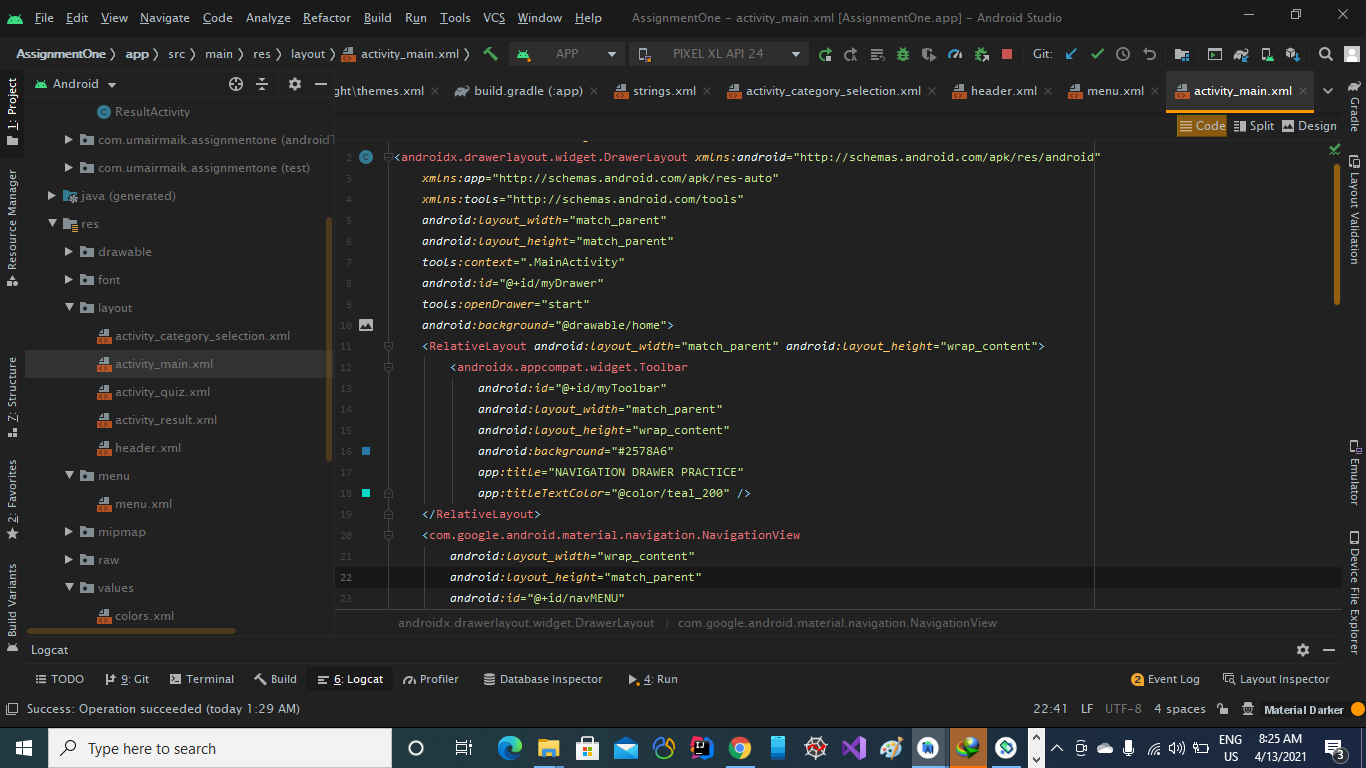


This is our haeder.xml file

Now we’ll add some project dependencies in our project.

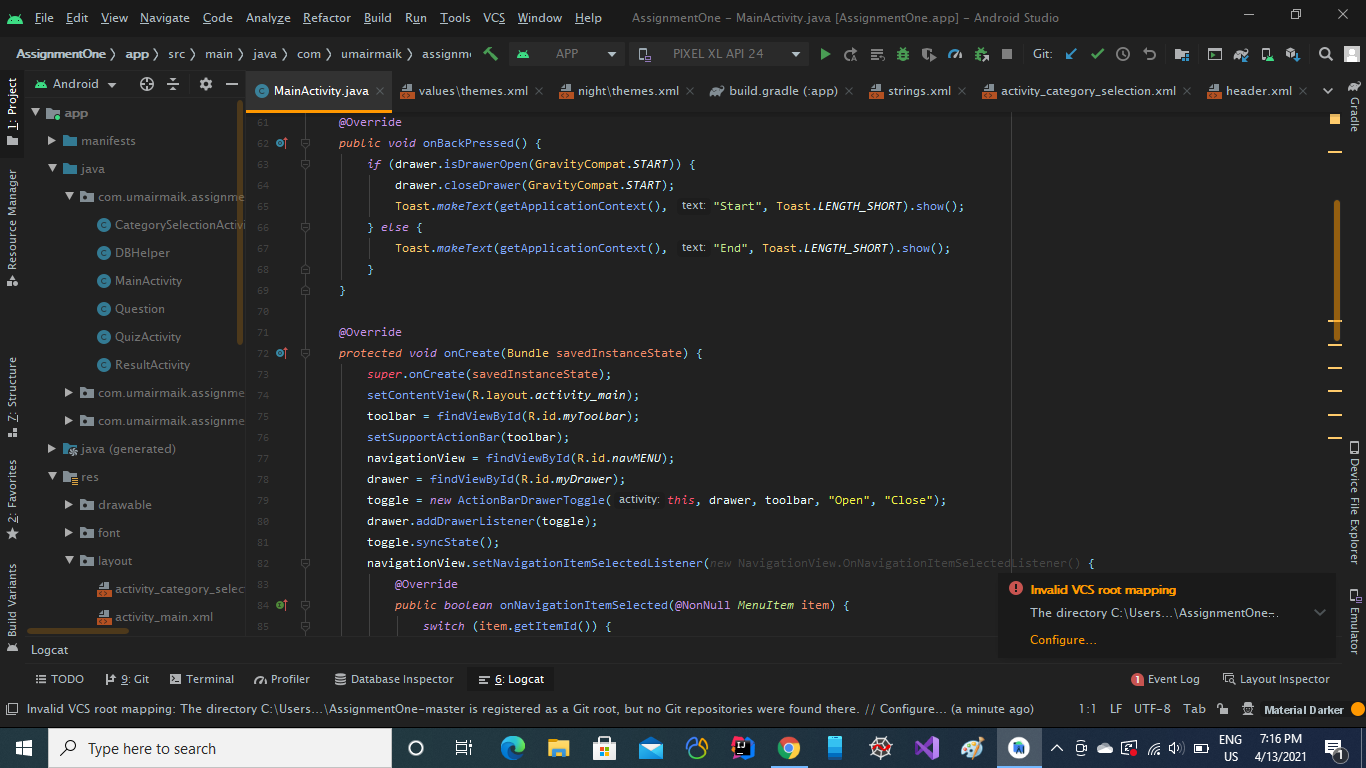


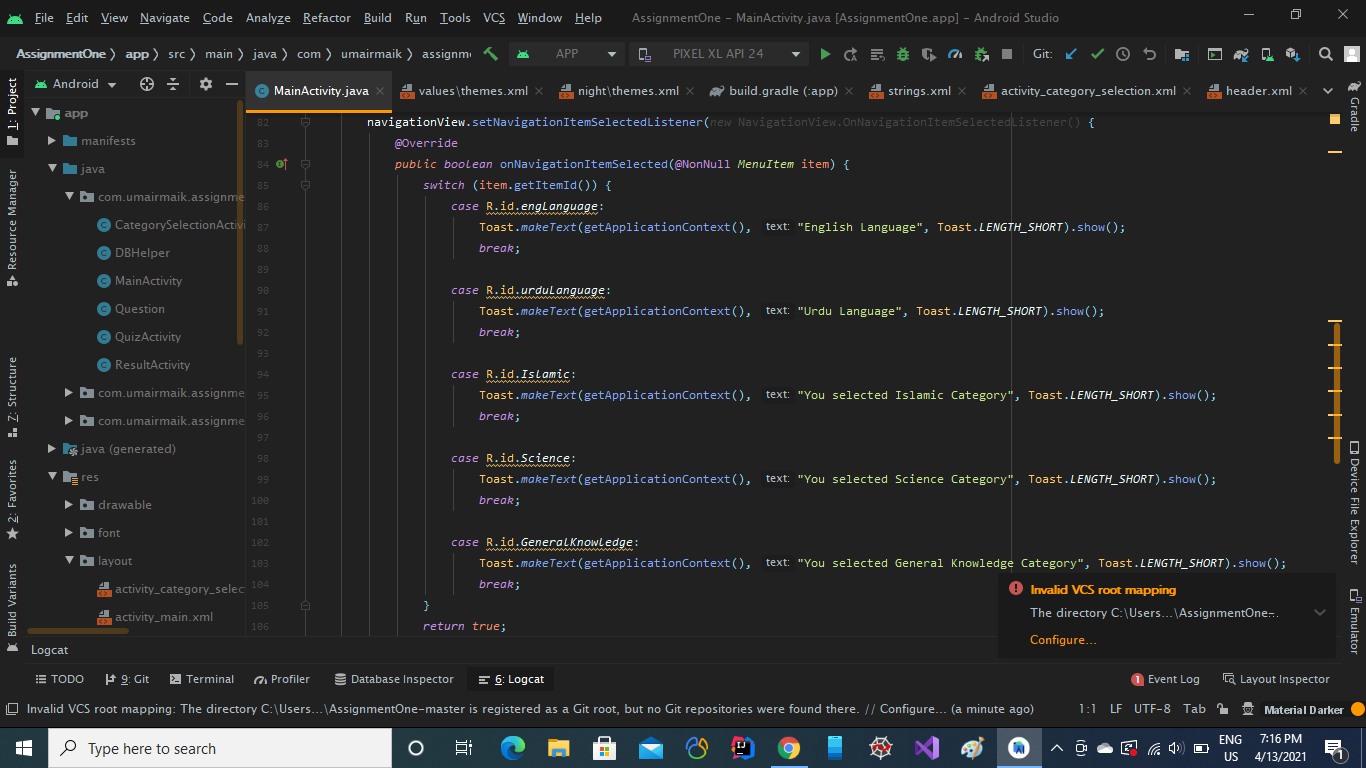
Now we have to merge our android drawer navigation to an activity



Now it is connected to our activity.

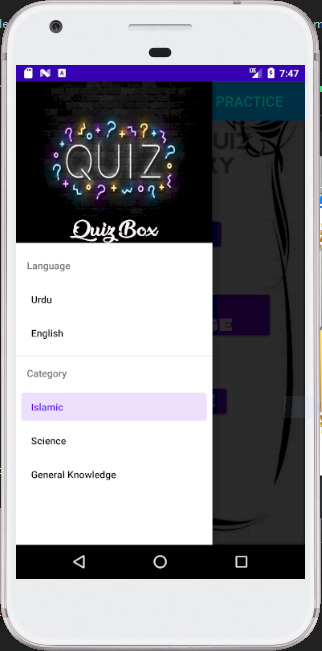
Now we have to set onClickLitsener.

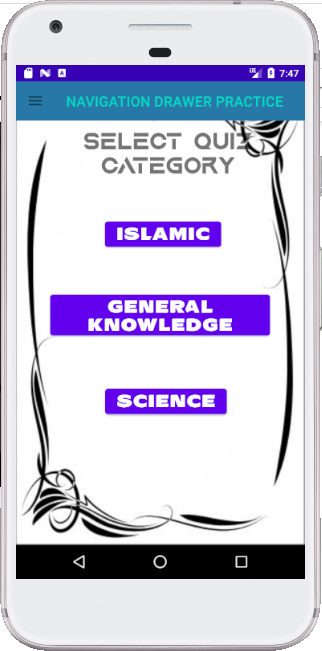




Now we are all set with our drawer navigation.

We can now do whatever we want in above case statements with our drawer items.





\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**2. Recycler View:**

**What?**

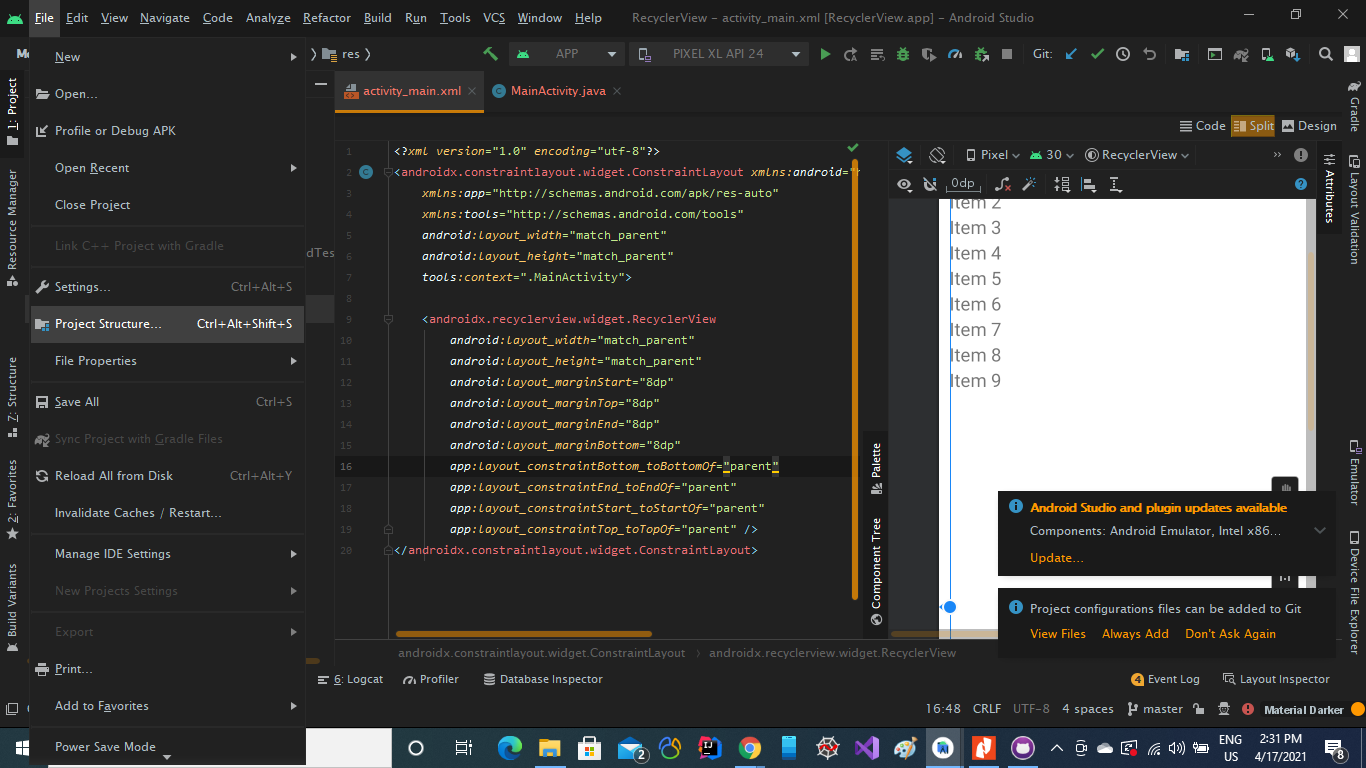
As the name implies, RecyclerView *recycles* those individual elements. When an item scrolls off the screen, RecyclerView doesn't destroy its view. Instead, RecyclerView reuses the view for new items that have scrolled onscreen. This reuse vastly improves performance, improving your app's responsiveness and reducing power consumption

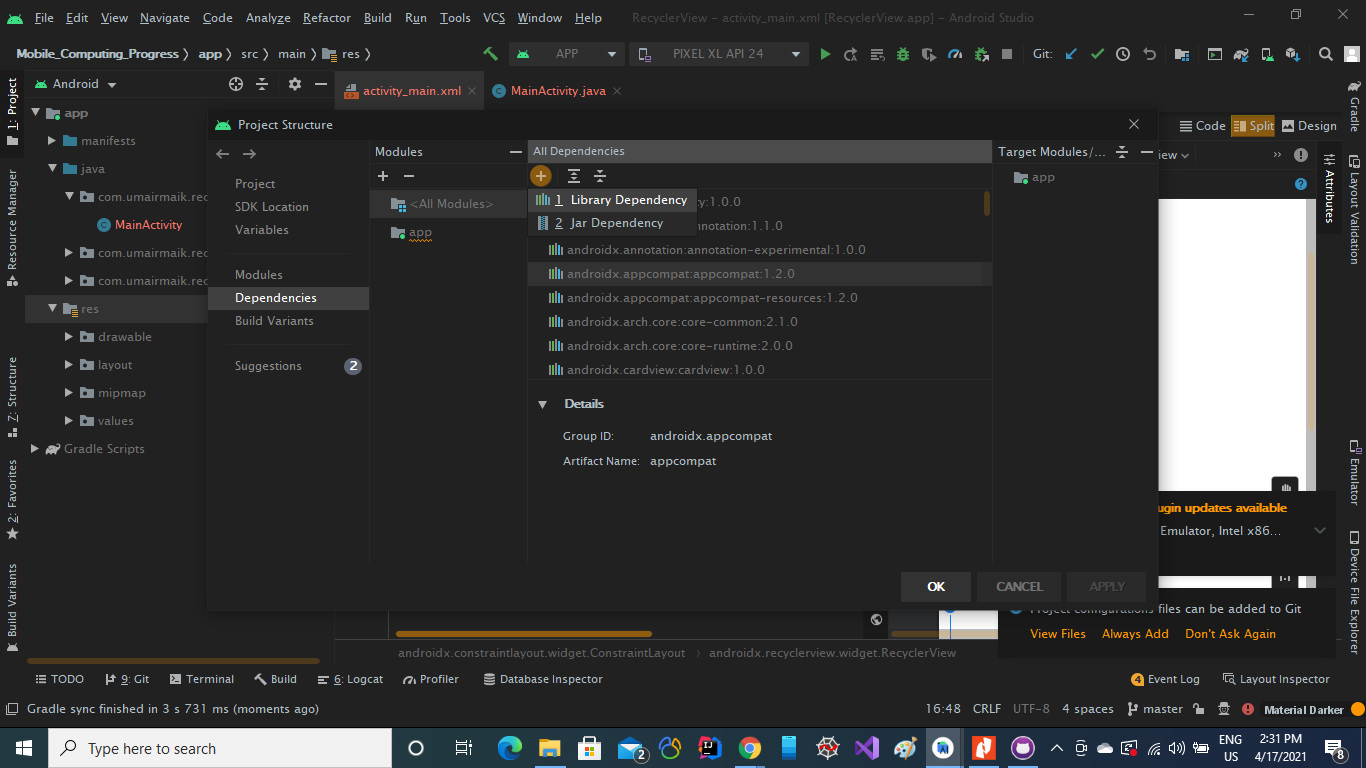
**Reference:** <https://developer.android.com/guide/topics/ui/layout/recyclerview>

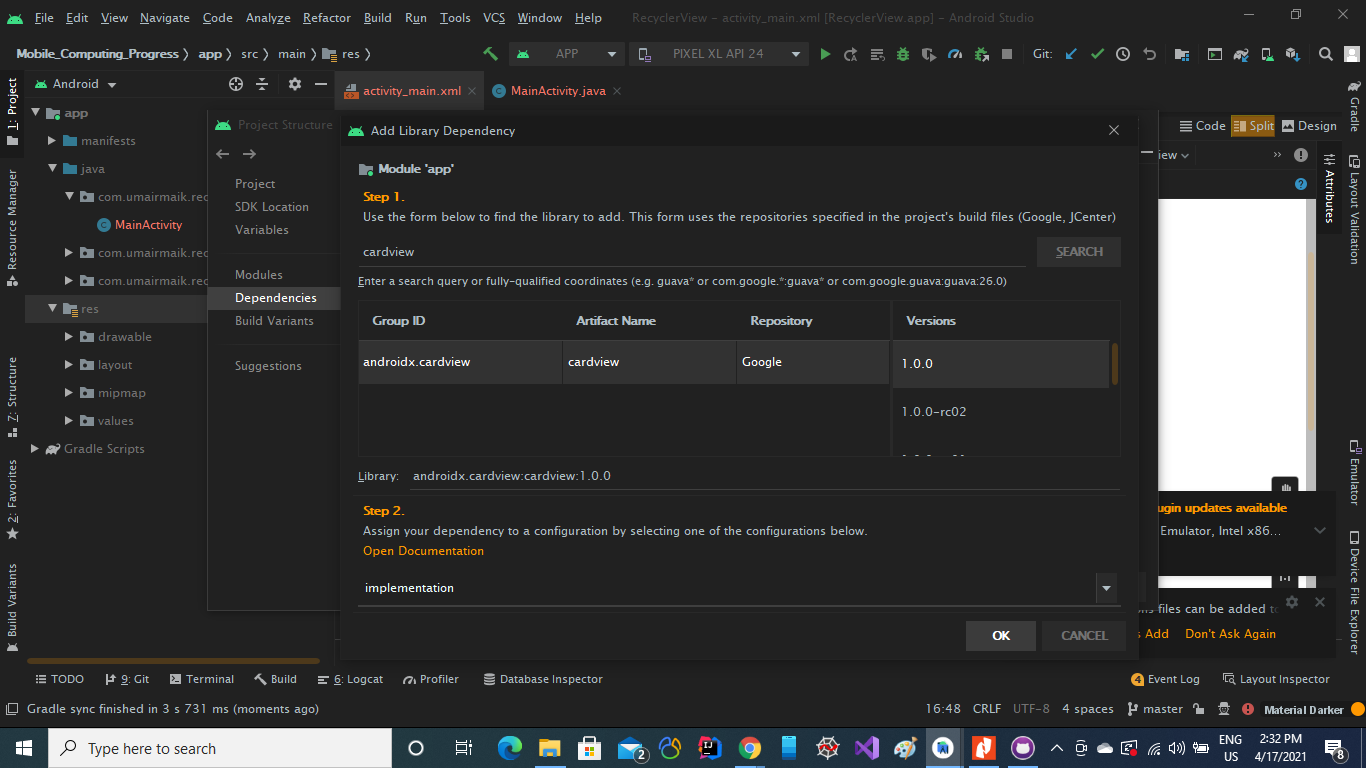
**Steps:**

1. First create a new project with empty activity.
2. Add the RecyclerView element in your main\_activity.xaml.
3. Now add recycler view dependency.
   1. File>>Project Structure>>Dependencies
   2. Click on ‘+’ icon and type recycler view in search field.
   3. Select library and type recyclerview in search field.
   4. Add the dependency

Apply the same above procedure to add other dependencies

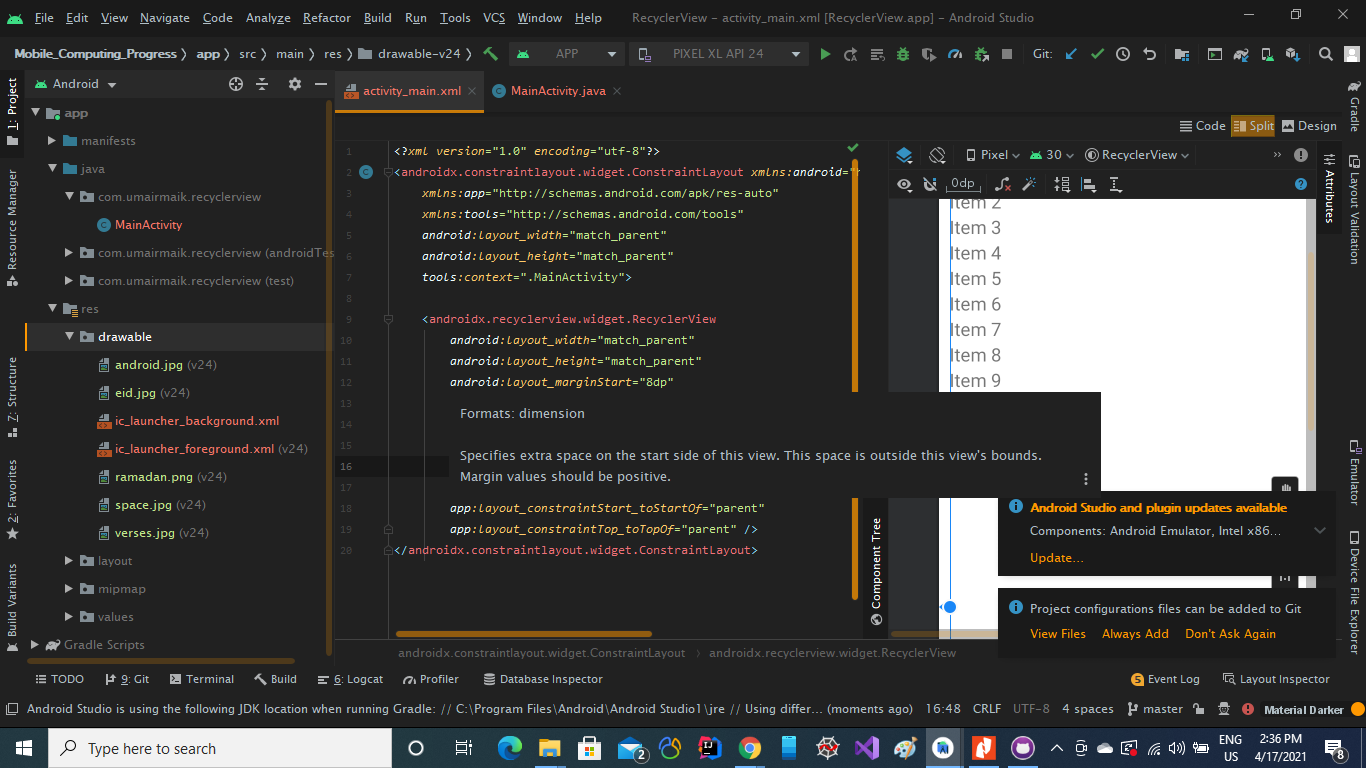


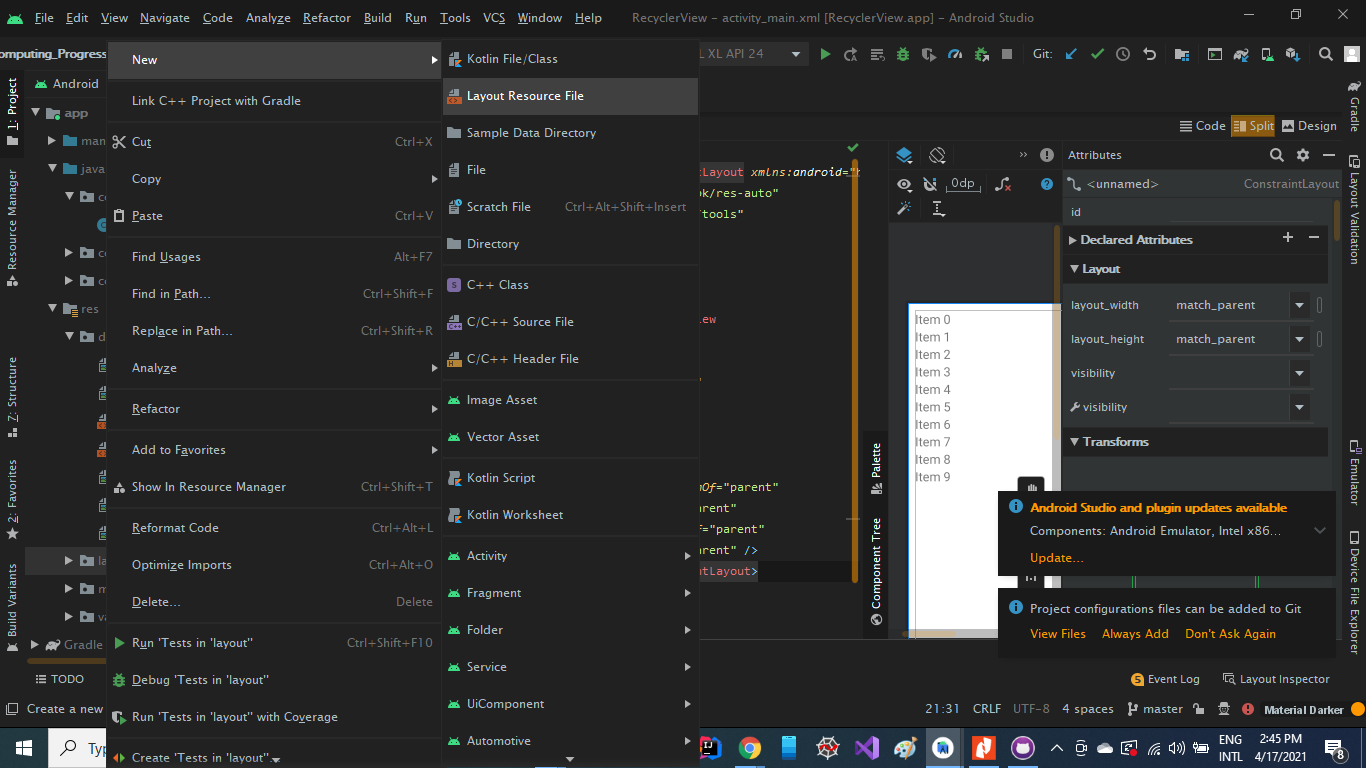


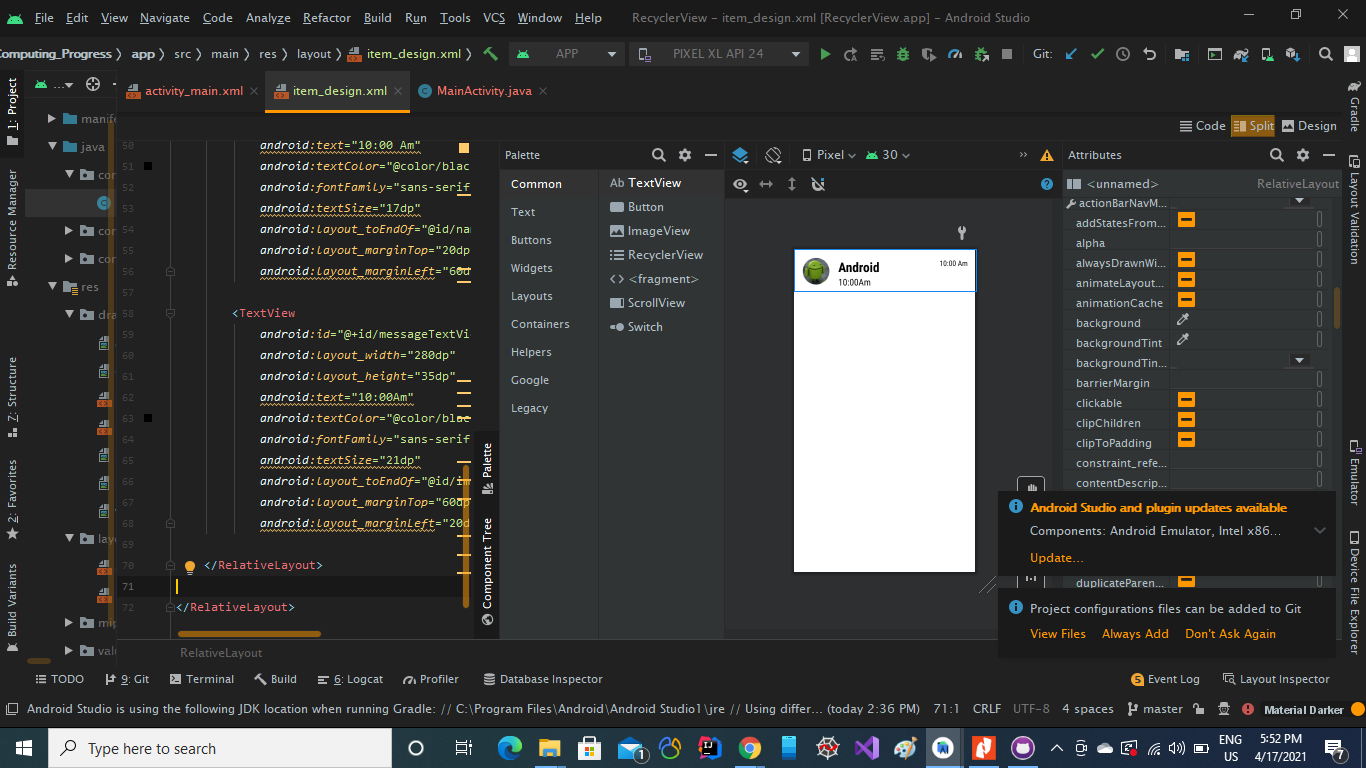


We applied the same above procedure to add card view because we are going to use rounded corner images in our recycler view.

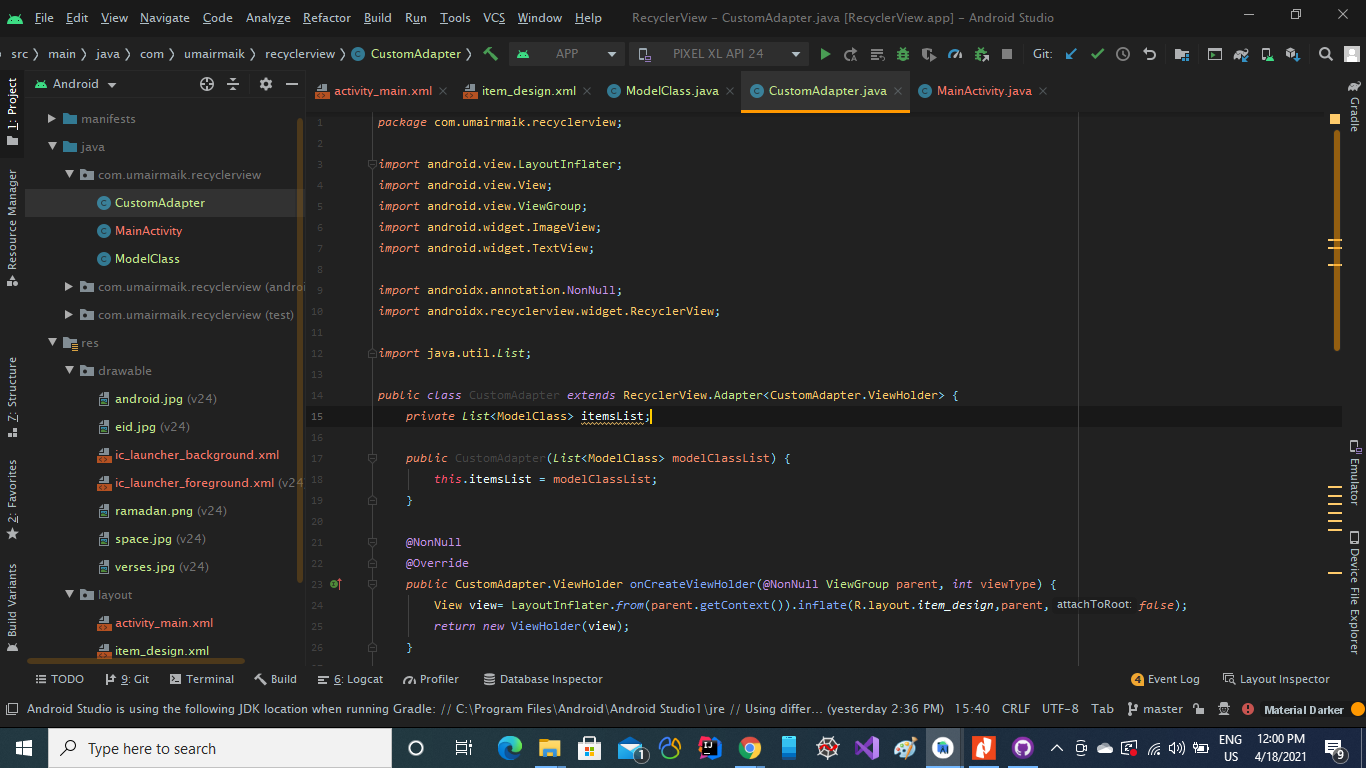
1. Now add images which we are going to use in our recycler view.
2. So we have added 5 images.
3. Now add the recycler view in activity\_main.xaml.

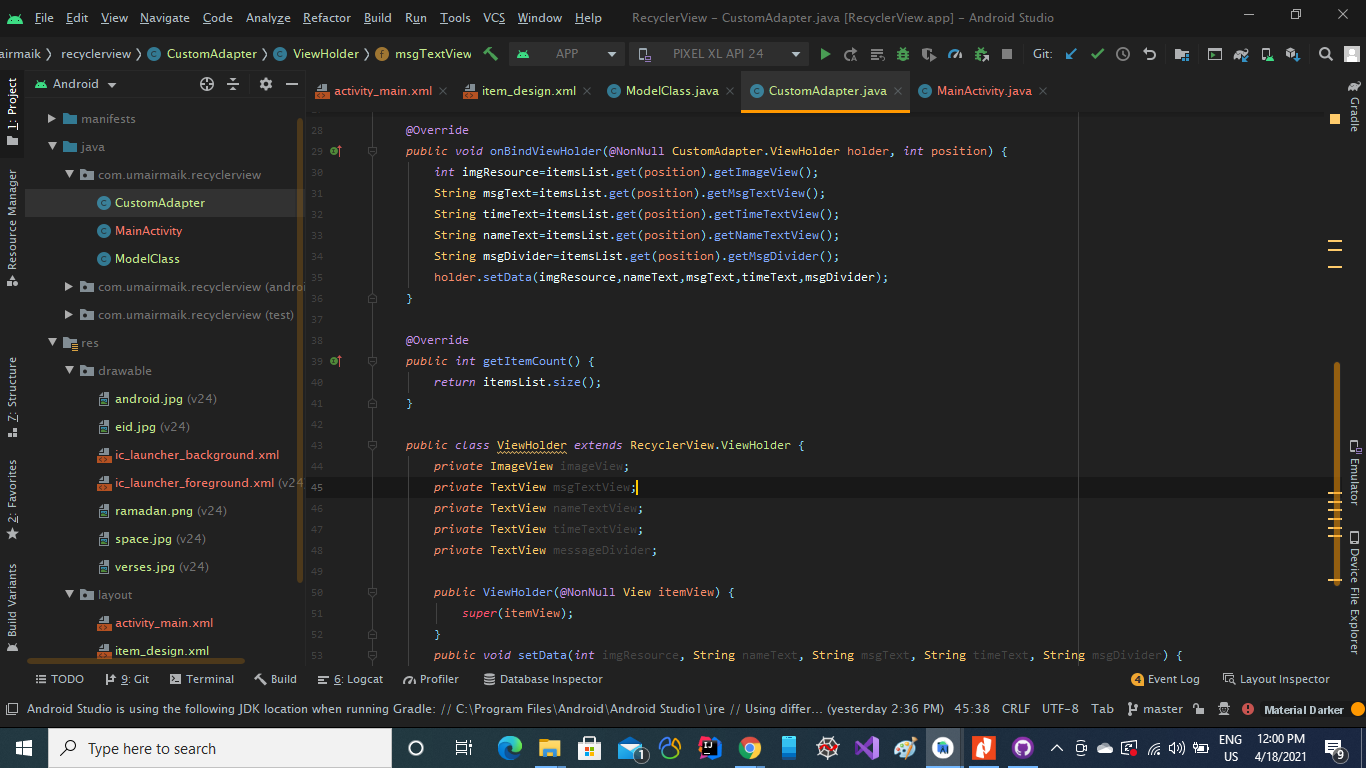


1. Now we have to define the layout of our **“Item”** in the recyclerView list.
2. For this add new xaml file in the layout folder.
3. 
4. Create a layout in your item.xaml class for your items in recyclerView class.



1. Add a new java class named as “ModelClass.java”.
2. This class will be responsible for providing informations related to our items in the recycler view.
3. Now we have to add a new class **customAdapter.java**  
   In this class we’ll get data and will put that data in our views using another ViewHolder.java which we’ll implement in this class also as **ViewHolder.java**.





|  |
| --- |
| package com.umairmaik.recyclerview;  import android.view.LayoutInflater; import android.view.View; import android.view.ViewGroup; import android.widget.ImageView; import android.widget.TextView;  import androidx.annotation.NonNull; import androidx.recyclerview.widget.RecyclerView;  import java.util.List;  public class CustomAdapter extends RecyclerView.Adapter<CustomAdapter.ViewHolder> {   private List<ModelClass> itemsList;    public CustomAdapter(List<ModelClass> modelClassList) {      this.itemsList = modelClassList;   }    @NonNull   @Override   public CustomAdapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {      View view= LayoutInflater.from(parent.getContext()).inflate(R.layout.item\_design,parent,false);      return new ViewHolder(view);   }    @Override   public void onBindViewHolder(@NonNull CustomAdapter.ViewHolder holder, int position) {      int imgResource=itemsList.get(position).getImageView();      String msgText=itemsList.get(position).getMsgTextView();      String timeText=itemsList.get(position).getTimeTextView();      String nameText=itemsList.get(position).getNameTextView();      String msgDivider=itemsList.get(position).getMsgDivider();      holder.setData(imgResource,nameText,msgText,timeText,msgDivider);   }    @Override   public int getItemCount() {      return itemsList.size();   }    public class ViewHolder extends RecyclerView.ViewHolder {      private ImageView imageView;      private TextView msgTextView;      private TextView nameTextView;      private TextView timeTextView;      private TextView messageDivider;       public ViewHolder(@NonNull View itemView) {         super(itemView);      }      public void setData(int imgResource, String nameText, String msgText, String timeText, String msgDivider) {      }   } } |

Now we are all done with our RecyclerView code, now we just have to write code in our mainActivity in which we’ll use this code we have implemented above.

In our mainactivity we’ll get our recyclerView element

Set a layout manager o.i.e, ***linear layout.***

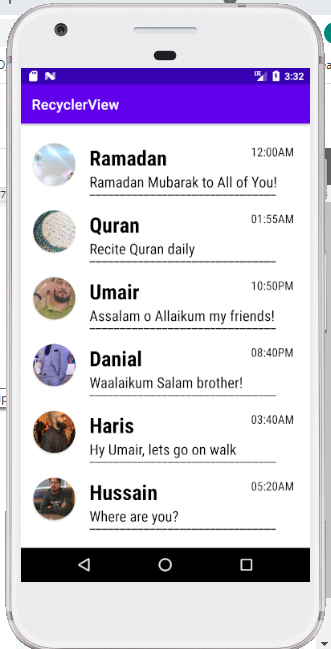
Then set our **CustomAdapter to RecyclerView.**

Set data (by creating objects of our model class and adding them in a list and then pass it to the CustomAdapter Class.

Here is the code.

|  |
| --- |
| package com.umairmaik.recyclerview;  import androidx.appcompat.app.AppCompatActivity; import androidx.recyclerview.widget.LinearLayoutManager; import androidx.recyclerview.widget.RecyclerView;  import android.os.Bundle;  import java.util.ArrayList; import java.util.List;  public class MainActivity extends AppCompatActivity {   RecyclerView recyclerView;   LinearLayoutManager layoutManager;   List<ModelClass> itemsList;   CustomAdapter customAdapter;     @Override   protected void onCreate(Bundle savedInstanceState) {      super.onCreate(savedInstanceState);      setContentView(R.layout.activity\_main);      initData();      initRecyclerView();   }    private void initData() {      itemsList = new ArrayList<>();      itemsList.add(new ModelClass(R.drawable.ramadan, "Ramadan", "12:00AM", "Ramadan Mubarak to All of You!", "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"));      itemsList.add(new ModelClass(R.drawable.verses, "Quran", "01:55AM", "Recite Quran daily", "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"));      itemsList.add(new ModelClass(R.drawable.umair, "Umair ", "10:50PM", "Assalam o Allaikum my friends!", "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"));      itemsList.add(new ModelClass(R.drawable.danial, "Danial ", "08:40PM", "Waalaikum Salam brother!", "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"));      itemsList.add(new ModelClass(R.drawable.haris, "Haris ", "03:40AM", "Hy Umair, lets go on walk", "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"));      itemsList.add(new ModelClass(R.drawable.hussain, "Hussain ", "05:20AM", "Where are you?", "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"));      itemsList.add(new ModelClass(R.drawable.sajid, "Sajid Bhai", "06:04AM", "Are you ok?", "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"));      itemsList.add(new ModelClass(R.drawable.papa, "Papa", "08:01PM", "Come home,I've been waiting for you...", "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"));   }    private void initRecyclerView() {      recyclerView = findViewById(R.id.myRecyclerView);      layoutManager = new LinearLayoutManager(this);      layoutManager.setOrientation(RecyclerView.VERTICAL);      customAdapter = new CustomAdapter(itemsList);      recyclerView.setAdapter(customAdapter);      recyclerView.setLayoutManager(layoutManager);      customAdapter.notifyDataSetChanged();    } } |

Now we’ll run it on device to see the output results.



Here are the results.