

**Department Of Computer Science**

**National University Of Computer And Emerging Sciences.**

THE DECORUM

Team Members:

Syed Muhammad Aqeel Abbas

17k-3701

Muneeb Ul Hasan

17k-3717

Muhammad Furqan

17k-3836

Dr. Fahad Samad - Signature

**ABSTRACT:**

This app enables customers to search, detect and position 3D objects via AR technology in the real environment. This app will address the existing problem by which consumers had to go to showrooms and try to imagine how each product  would fit into their place. However, this our app will allow customers to insert their e-commerce products that looks lifelike into their places before buying the actual furniture.

1. **INTRODUCTION:**
2. **Need For Product**

Purchasing interior design products often has a problem that consumers may not be satisfied with the products they have bought because they cannot arrange them in their own place before buying them. The aim of this project is to develop an android application called 'DECORUM' using Augmented Reality technology for interior decoration that will help customers visualize how furniture and interior design products will look and fit (to scale) in their home and can also provide details of products to support customer decision-making.

1. **Associated CS Problems**

Instead of going to the physical store to purchase the items, the customer would be able to display the products virtually in their home structure. The customer's basic issue is that they need to measure and search for the correct size of the product that suits their home environment because our application only enables users to see the product they want to buy according to their specifications.

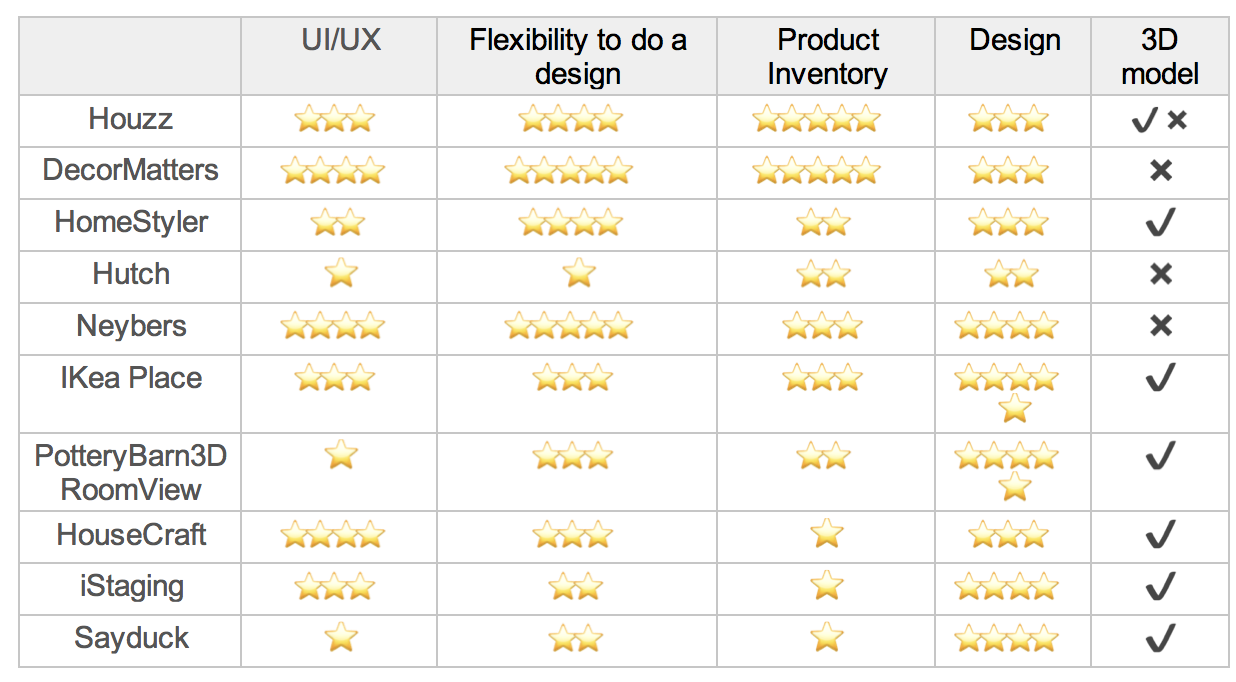
1. **Benefits to Users**

In the field of interior designing, the increased augmented reality could not fully take control. Today, people are familiar with the technology and operate smartphones that support AR. Thus, the idea of developing an interior design application leads the designer to be technologically advanced. Moreover, our app ‘DECORUM’ aims to achieve several other objectives which are as follows:

* **Improve customer’s experience in online shopping:** The project aims at providing the use of their smartphones to customers with a realistic picture of the furniture. Therefore, the user can then open the camera on the smartphone to see the selected product in AR from the list. Users can therefore immediately see how well the product is and whether it meets their needs and preferences, without the hassle of going to showrooms or travelling out.
* **Provide the featured product with realistic 3D view:** It is not enough to encourage users to display the products only from an angle so the application should be able to calibrate the 3D model with information provided for correct positioning of the products. Users will also switch around the products to see how it feels from different viewpoints. In addition, anchors should be positioned on the plane surface to the object on which the camera is aimed. This stops the movement of products from shifting with the smartphone thus making it impossible to watch from a certain perspective.
* **Provide the real-world atmosphere with correct live size:** The project must be able to view the items in a live format so that customers can ensure that the product blends into the actual space available. Therefore, having a live scale view helps create a practical environment for customers as if they had bought the items, however it disappears automatically when the application is closed.
* **Allow the application to be easily maintained:** Since the project is used on the online platform, developers need to be able to effectively perform software maintenance. Developers should also be able to add models to the application's database, which should automatically represent the application data retrieval. Therefore, users can instantly view new products and take a look at the product with AR.

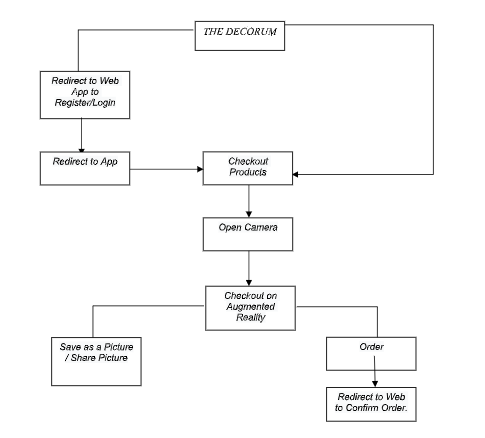
1. **Gap Analysis With Existing Solution**

The application currently in use is slower to capture images, offers low resolution, further degrades user experience and issues and has to wait before an image is properly processed, even the graphics of the interior decor items are lower in resolution. However there are some applications with strong UI and UX, but 3d modelling and inventory aren't really good.



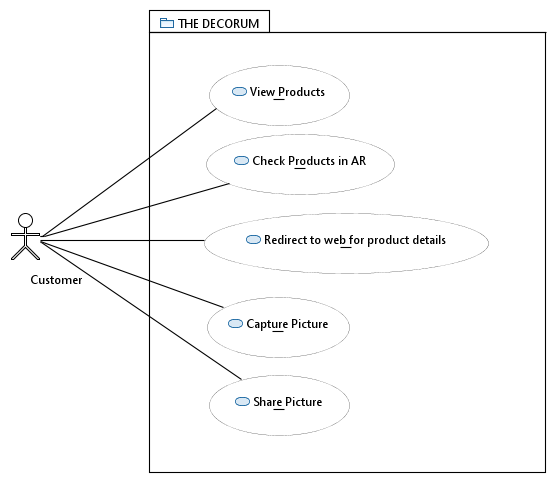
[fig. 1] Comparison of existing applications

1. **REQUIREMENTS ANALYSIS**
2. **Functional Requirements:**
   1. Functional Hierarchy



[fig. 2] Work Breakdown Structure of our Application

* 1. Use Cases



[fig. 3] Use Case Diagram of our App

|  |  |  |  |
| --- | --- | --- | --- |
| **<Use case Id: View Products>** | | | |
| **Use case Id:** | | 1 | |
| **Actors:**  Customer initiates the use case | | | |
| **Feature:** Customer can view the products | | | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Customer opens the app. | | App will show the available products. |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1.** | If customer wants to check product details he will be redirected to web. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **<Use case Id: Check product in AR>** | | | |
| **Use case Id:** | | 2 | |
| **Actors:**  Customer initiates the use case | | | |
| **Feature:** Customer can check the product in AR. | | | |
| **Pre-condition:** | | Customer should have clicked on a product he wishes to check in AR. | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Customer clicks on the product. | | App will open camera and user will be able to drag and place the product anywhere. |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1.** | If customer wants to checkout he will be redirected to web. | | |
| **2.** | If customer wants to check another product he can go to back to products list. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **<Use case Id: Redirect to web for product Details>** | | | |
| **Use case Id:** | | 3 | |
| **Actors:**  Customer initiates the use case | | | |
| **Feature:** Customer can view the products details | | | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Customer clicks on the product details button | | App will redirect customer to web. |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1.** | Customer can checkout from web. | | |
| **2.** | Comeback to web and view other products in AR. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **<Use case Id: Capture Pictures>** | | | |
| **Use case Id:** | | 4 | |
| **Actors:**  Customer initiates the use case | | | |
| **Feature:** Customer can capture picture. | | | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Customer will click the capture button. | | App will click a picture. |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1.** | Customer can share the picture. | | |
| **2.** | Save the picture and check it afterwards. | | |
| **3.** | Cancel and click another picture. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<Use case Id: Share Picture>** | | | | |
| **Use case Id:** | | 5 | | |
| **Actors:**  Customer initiates the use case | | | | |
| **Feature:** Customer can share the picture | | | | |
| **Pre-condition:** | | Customer should have captured a picture in order to share it. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Customer will click the share button. | | | App will show share options. |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.** | Click another picture. | | | |
| **2.** | Cancel this activity and go to products list. | | | |
| **Use Case Cross referenced** | | | Use Case: Capture Picture should be done before doing this use case. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **<Use case Id: Search Products by name>** | | | |
| **Use case Id:** | | *6* | |
| **Actors:**  Customer initiates the use case | | | |
| **Feature:** Customer can search the product by typing name of the product | | | |
| **Pre-condition:** | | Customer should have captured a picture in order to share it. | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Customer types the name | | All products containing input of the user will appear. |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1.** | Customer can then check the product in AR. | | |
| **2.** | Customer can also see details of product by clicking on view details, by which customer will be redirected to web. | | |

1. **Non-Functional Requirements:**
2. **Performance Requirements**

Performance will depend on the models that are loaded in the app. If the model is heavy it will take a little time to load.

1. User Documentation

FYP-I does not include this requirement however in case of time then we might implement this feature as well.

**In scope features of our Application:**

* 3d Object Tracking.
* Able to select angle of product.
* Take pictures while using AR.
* Share pictures.
* Select available color.
* Select available size.
* Search items.
* Click on product to see details of it on web.

Not In Scope Features:

* Product details.
* Add to cart products.
* Signup/Login

1. **DESIGN DETAILS**
   1. **Covering Process And Data Models**
   2. **ULS**
   3. **Code in the Database**
2. **IMPLEMENTATION DETAILS**
   1. **Development Tools**

* **Frontend :** Android Studio
* **Backend:** Firebase
  1. **Design Trade-Offs During Implementation**

1. **TESTING**
   1. **Test Data**
   2. **Unit Test**

|  |  |
| --- | --- |
| Test Case Description | App shows all the required information of a product. |
| Test Procedure | 1. Open the app. 2. App will display all the products and their info. |
| Test Data | Item Name  Price  Size |
| Expected Result | Users will be able to view all the products and their information. |
| Actual Result | Customers are able to view all the products. |
| Status | PASS |
| Priority | Major |
| Remarks | The test result contains the desired output and shows all the information of the products. |

|  |  |
| --- | --- |
| Test Case Description | Customer can check the product in Augmented Reality. |
| Test Procedure | 1. Open the app. 2. Long tap on the product that you wish to check. 3. Click on “Check in AR” button. |
| Test Data | 3d model |
| Expected Result | Users will be able to view the product in AR. |
| Actual Result | Customer is able to view the product in AR. |
| Status | PASS |
| Priority | Major |
| Remarks | The test result contains the desired output and shows the desired product in AR. |

|  |  |
| --- | --- |
| Test Case Description | Customer can download the product model and place it on the plane. |
| Test Procedure | 1. Choose the product that you want to see in AR. 2. Camera will open. 3. Click on “Download Model” button to download the model. 4. Place it on the plane. |
| Test Data | 3d model |
| Expected Result | Customers will be able to download the correct 3d model and model will be displayed. |
| Actual Result | Customers are able to download the model and view it. |
| Status | PASS |
| Priority | Major |
| Remarks | The test result contains the desired output and shows the 3d model. |

|  |  |
| --- | --- |
| Test Case Description | Customer click on the screenshot button. |
| Test Procedure | 1. Place the product where the hit point is. 2. Click on “Take Screenshot” button. 3. Screenshot will be taken. |
| Test Data | - |
| Expected Result | Users will be able to view the screenshot that they had taken. |
| Actual Result | Black screen appearing without the camera contents. |
| Status | FAIL |
| Priority | Major |
| Remarks | The test result does not contain the desired output. |

4-

|  |  |
| --- | --- |
| Test Case Description | Customer search products. |
| Test Procedure | 1. Open the app. 2. Click on the search bar. 3. Type keyword to search. |
| Test Data | Item |
| Expected Result | Users will be able to all the products that he searched. |
| Actual Result | Customers are able to search and view all the products he searched. |
| Status | Pass |
| Priority | Major |
| Remarks | The test result contains the desired output and shows all the products. |

* 1. **System Test**

|  |  |
| --- | --- |
| Test Case Description | If the App launches properly with all relevant pages and logos. |
| Test Procedure | User launches the app. |
| Test Data | - |
| Expected Result | App should launch with all the working features. |
| Actual Result | App is launching with all features working fine. |
| Status | PASS |
| Priority | Major |
| Remarks | App is launching properly with all the features that are in working. |

**2)**

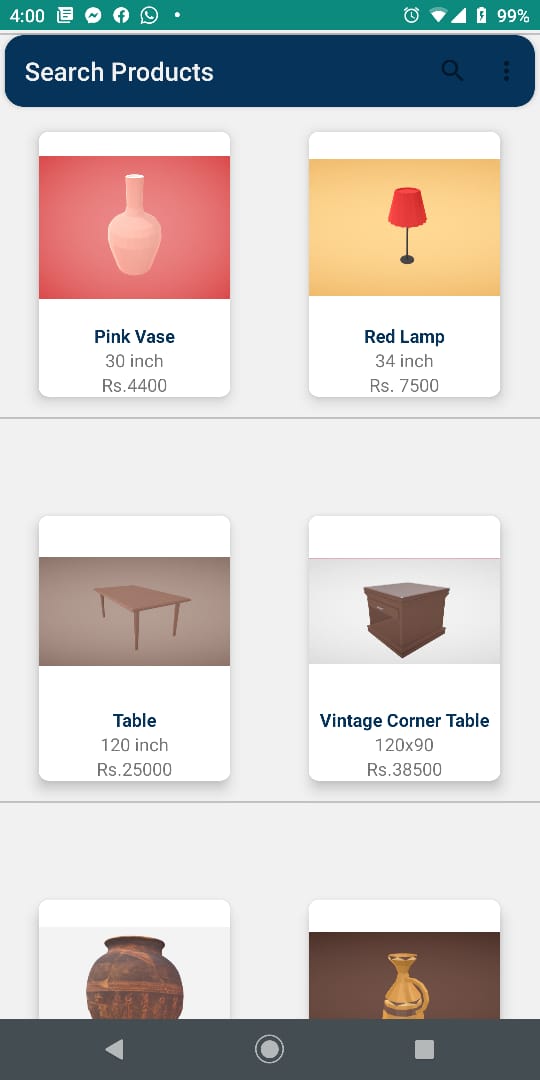
|  |  |
| --- | --- |
| Test Case Description | If Customers can see all the products. |
| Test Procedure | 1. Users first opens the app. 2. App will show all the products. |
| Test Data | - |
| Expected Result | Customers will be able to see all the products. |
| Actual Result | Customers can view all the products. |
| Status | PASS |
| Priority | Major |
| Remarks | This test is a success and shows the desired output. |

**3)**

|  |  |
| --- | --- |
| Test Case Description | If Customers can view all the products they can their desired product in AR. |
| Test Procedure | 1. Customer will open the app. 2. App will show all the products. 3. Customer can long tap on a product and click on “Check in AR” button. 4. App will open camera and show the product in AR. |
| Test Data | - |
| Expected Result | Customers will be able to check the product in AR. |
| Actual Result | Customers can view the product in AR. |
| Status | PASS |
| Priority | Major |
| Remarks | This test is a success and shows the desired output. |

1. **OUTPUTS OF THE PRODUCT**
   1. **Screenshots**

**Logo, company name

Description automatically generated **

[fig. 4]Splash Screen [fig. 5] Home Screen

**Graphical user interface, application

Description automatically generated Graphical user interface

Description automatically generated**

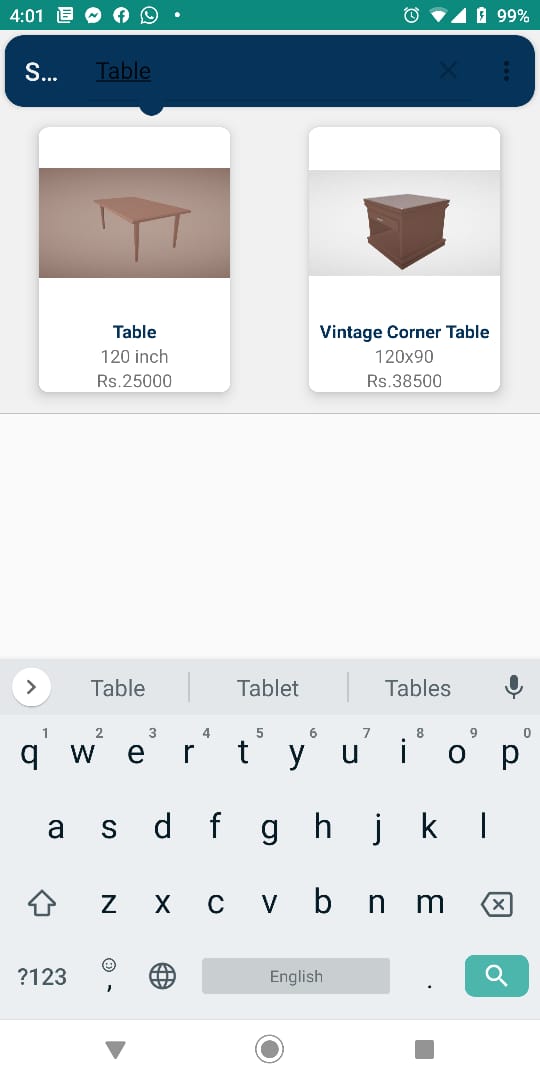
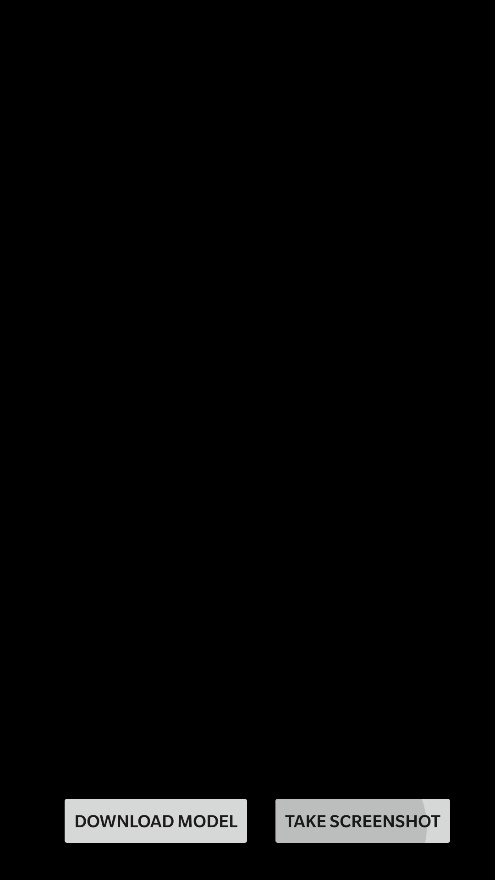
[fig. 6]Navigation Drawer [fig. 7] Price Filter

**Graphical user interface, application, PowerPoint

Description automatically generated** **Diagram

Description automatically generated**

[fig. 8] Product Category (Vase) [fig. 9] Product Category (Lamp)

**** ****

[fig. 10] Searching Products [fig. 11] AR Screen

**A picture containing text, indoor

Description automatically generated A picture containing text

Description automatically generated**

[fig. 12] 3D Model (Vase) [fig. 13] 3D Model (Three Set Vase

* 1. **Use-Case Data**

1. **CODE**

**JAVA CODE**

**MAIN ACTIVITY:**

package com.company.thedecorum;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.ActionBarDrawerToggle;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.core.view.GravityCompat;  
import androidx.core.view.MenuItemCompat;  
import androidx.drawerlayout.widget.DrawerLayout;  
import androidx.fragment.app.Fragment;  
import androidx.fragment.app.FragmentTransaction;  
import androidx.recyclerview.widget.DividerItemDecoration;  
import androidx.recyclerview.widget.GridLayoutManager;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.app.AlertDialog;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.widget.Filter;  
import android.widget.RelativeLayout;  
import android.widget.SearchView;  
import android.widget.Toast;  
  
import com.company.thedecorum.Fargments.HomeFragment;  
import com.company.thedecorum.Fargments.LampFragment;  
import com.company.thedecorum.Fargments.RecommendationFragment;  
import com.company.thedecorum.Fargments.TableFragment;  
import com.company.thedecorum.Fargments.VaseFragment;  
import com.google.android.material.navigation.NavigationView;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
  
import java.util.ArrayList;  
  
public class MainActivity extends AppCompatActivity implements NavigationView.OnNavigationItemSelectedListener, ProductAdapter.AbcOnClickListener {  
  
 private RecyclerView view;  
  
 private ProductAdapter adapter;  
 private ArrayList<Product> list;  
 private DatabaseReference reference;  
 SharedPreferences mSharedPref;  
  
  
 Toolbar toolbar;  
 DrawerLayout mDrawerLayout;  
 ActionBarDrawerToggle mDrawerToggle;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 // setContentView(R.layout.activity\_drawer);  
  
 //setContentView(R.layout.product\_layout);  
  
 //startActivity(new Intent(MainActivity.this,DrawerActivity.class));  
  
  
 view = (RecyclerView) findViewById(R.id.*recycleview*);  
 reference = FirebaseDatabase.*getInstance*().getReference().child("product");  
 // view.setHasFixedSize(true);  
  
  
  
 list = new ArrayList<>();  
 adapter = new ProductAdapter(this,list);  
 // adapter.startListening();  
 view.setAdapter(adapter);  
 adapter.setabcOnClickListener(this);  
 // reference.addListenerForSingleValueEvent(listener);  
 //view.setLayoutManager(new GridLayoutManager(getApplicationContext(),2));  
  
 // GridLayoutManager mGrid = new GridLayoutManager(this,2);  
 // view.setLayoutManager(mGrid);  
  
 // view.setLayoutManager(new GridLayoutManager(this,2));  
  
 //adapter.setabcOnClickListener(this);  
 reference.addListenerForSingleValueEvent(listener);  
  
 view.addItemDecoration(new DividerItemDecoration(  
 view.getContext(),DividerItemDecoration.*VERTICAL* ));  
  
 toolbar = findViewById(R.id.*toolBar*);  
 setSupportActionBar(toolbar);  
 getSupportActionBar().setTitle("Search Products");  
  
 mDrawerLayout = findViewById(R.id.*drawerLayout*);  
 mDrawerToggle = new ActionBarDrawerToggle(this,mDrawerLayout, R.string.*open*, R.string.*close*);  
  
 mDrawerLayout.addDrawerListener(mDrawerToggle);  
 mDrawerToggle.syncState();  
  
 NavigationView navigationView = findViewById(R.id.*nav\_view*);  
 navigationView.setNavigationItemSelectedListener(this);  
  
 showFragments(new HomeFragment());  
 }  
  
 ValueEventListener listener = new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.exists()){  
 list.clear();  
 for(DataSnapshot snapshot:dataSnapshot.getChildren()){  
 Product p = snapshot.getValue(Product.class);  
 list.add(p);  
 }  
 adapter.notifyDataSetChanged();  
 }  
 else{  
 Toast.*makeText*(MainActivity.this,"No record found",Toast.*LENGTH\_LONG*).show();  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 Toast.*makeText*(MainActivity.this,databaseError.toString(),Toast.*LENGTH\_LONG*).show();  
 }  
 };  
  
  
 public void clickme(int pos) {  
 Toast.*makeText*(this,pos + "" , Toast.*LENGTH\_SHORT*).show();  
 }  
  
  
 public void viewdetail(int pos) {  
 Product p = list.get(pos);  
 Toast.*makeText*(this,p.size +" "+ p.name,Toast.*LENGTH\_SHORT*).show();  
 }  
  
  
 public void ar(int pos) {  
 Product p = list.get(pos);  
  
 Intent intent = new Intent(this,ArActivity.class);  
 intent.putExtra("id", String.*valueOf*(pos));  
 startActivity(intent);  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*searchmenu*,menu);  
 MenuItem item = menu.findItem(R.id.*search*);  
 SearchView searchView = (SearchView) MenuItemCompat.*getActionView*(item);  
 searchView.setQueryHint("Search...");  
 searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {  
 @Override  
 public boolean onQueryTextSubmit(String query) {  
// searchData(s);  
 Log.*e*("query 1",query.toString());  
 adapter.getFilter().filter(query);  
 return false;  
 }  
  
 @Override  
 public boolean onQueryTextChange(String newText) {  
  
 if(adapter != null)  
 {  
 adapter.getFilter().filter(newText);  
 }  
 //adapter.getFilter().filter(newText);  
 return false;  
 }  
 });  
// return super.onCreateOptionsMenu(menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(@NonNull MenuItem item) {  
  
 if(item.getItemId() == R.id.*action\_settings*){  
 Toast.*makeText*(this,"Settings",Toast.*LENGTH\_SHORT*).show();  
 return true;  
 }  
 if(item.getItemId() == R.id.*mSort*){  
 showSortDialog();  
 return true;  
 }  
 return super.onOptionsItemSelected(item);  
 }  
  
  
 private void showSortDialog() {  
 String[] sortOptions = {"Price: Low To High","Price: High To Low"};  
 AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this) ;  
 builder.setTitle("Sort By")  
 .setIcon(R.drawable.*ic\_action\_sort*)  
 .setItems(sortOptions, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
  
 if(i==0)  
 {  
 SharedPreferences.Editor editor = mSharedPref.edit();  
 editor.putString("Sort","Price: Low To High");  
 editor.apply();  
  
 view.setAdapter(adapter);  
  
 }  
 else if(i==1)  
 {  
 SharedPreferences.Editor editor = mSharedPref.edit();  
 editor.putString("Sort","Price: High To Low");  
 editor.apply();  
  
 view.setAdapter(adapter);  
 }  
  
 }  
 });  
 builder.create().show();  
 }  
  
 public boolean onNavigationItemSelected(@NonNull MenuItem item) {  
  
 int id = item.getItemId();  
  
 if(id == R.id.*nav\_home*){  
 showFragments(new HomeFragment());  
 mDrawerLayout.closeDrawer(GravityCompat.*START*);  
 }  
  
 if(id == R.id.*nav\_recommendation*){  
 showFragments(new RecommendationFragment());  
 mDrawerLayout.closeDrawer(GravityCompat.*START*);  
 }  
  
 if(id == R.id.*nav\_lamp*){  
 showFragments(new LampFragment());  
 mDrawerLayout.closeDrawer(GravityCompat.*START*);  
 }  
  
 if(id == R.id.*nav\_vase*){  
 showFragments(new VaseFragment());  
 mDrawerLayout.closeDrawer(GravityCompat.*START*);  
 }  
  
 if(id == R.id.*nav\_table*){  
 showFragments(new TableFragment());  
 mDrawerLayout.closeDrawer(GravityCompat.*START*);  
 }  
  
 return false;  
 }  
  
 private void showFragments(Fragment fragment){  
  
 FragmentTransaction ft = getSupportFragmentManager().beginTransaction();  
 ft.replace(R.id.*frame\_layout*, fragment);  
 ft.commit();  
 }  
  
 public void onBackPressed(){  
  
 if (mDrawerLayout.isDrawerOpen(GravityCompat.*START*)) {  
 mDrawerLayout.closeDrawer(GravityCompat.*START*);  
 }  
 else {  
 super.onBackPressed();  
 }  
  
 }  
  
 @Override  
 public void onPointerCaptureChanged(boolean hasCapture) {  
  
 }  
}

**AR ACTIVITY:**

package com.company.thedecorum;  
  
import androidx.annotation.RequiresApi;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
  
import android.content.Intent;  
import android.graphics.Bitmap;  
import android.net.Uri;  
import android.os.Build;  
import android.os.Bundle;  
import android.os.Environment;  
import android.os.StrictMode;  
import android.util.Log;  
import android.view.View;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.ar.sceneform.AnchorNode;  
import com.google.ar.sceneform.assets.RenderableSource;  
import com.google.ar.sceneform.rendering.ModelRenderable;  
import com.google.ar.sceneform.ux.ArFragment;  
import com.google.firebase.FirebaseApp;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
import com.google.firebase.storage.FileDownloadTask;  
import com.google.firebase.storage.FirebaseStorage;  
import com.google.firebase.storage.StorageReference;  
  
import java.io.File;  
import java.io.FileOutputStream;  
import java.io.IOException;  
import java.util.Calendar;  
  
import static android.Manifest.permission.*READ\_EXTERNAL\_STORAGE*;  
import static android.Manifest.permission.*WRITE\_EXTERNAL\_STORAGE*;  
import static android.content.Context.*BIND\_ADJUST\_WITH\_ACTIVITY*;  
import static android.content.pm.PackageManager.*PERMISSION\_GRANTED*;  
  
public class ArActivity extends AppCompatActivity {  
 TextView textName;  
 String productid;  
 private DatabaseReference reference;  
 private FirebaseStorage storage;  
 private StorageReference modelRef;  
 private ArFragment arFragment;  
 Product product;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_ar*);  
  
 productid = getIntent().getStringExtra("id");  
 Log.*d*("Furqan", "onCreate:" + productid);  
  
 FirebaseApp.*initializeApp*(this);  
 reference = FirebaseDatabase.*getInstance*().getReference().child("product").child(productid);  
  
 reference.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(DataSnapshot dataSnapshot) {  
 product = dataSnapshot.getValue(Product.class);  
 System.*out*.println(product);  
 Log.*d*("product",product.path);  
  
 storage = FirebaseStorage.*getInstance*();  
 modelRef = storage.getReference().child(product.path);  
  
 arFragment = (ArFragment) getSupportFragmentManager()  
 .findFragmentById(R.id.*arFragment*);  
  
 reference.get();  
  
 findViewById(R.id.*downloadBtn*)  
 .setOnClickListener(v -> {  
 try {  
 File file = File.*createTempFile*(product.name,product.extension);  
 modelRef.getFile(file).addOnSuccessListener(new OnSuccessListener<FileDownloadTask.TaskSnapshot>() {  
 @RequiresApi(api = Build.VERSION\_CODES.*N*)  
 @Override  
 public void onSuccess(FileDownloadTask.TaskSnapshot taskSnapshot) {  
 buildModel(file);  
 }  
 });  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 });  
  
 arFragment.setOnTapArPlaneListener((hitResult, plane, motionEvent) -> {  
 AnchorNode anchorNode = new AnchorNode(hitResult.createAnchor());  
 anchorNode.setRenderable(renderable);  
 arFragment.getArSceneView().getScene().addChild(anchorNode);  
 });  
  
  
 }  
  
 @Override  
 public void onCancelled(DatabaseError databaseError) {  
 System.*out*.println("The read failed: " + databaseError.getCode());  
 }  
 });  
  
 ActivityCompat.*requestPermissions*(this,new String[]{*WRITE\_EXTERNAL\_STORAGE*,*READ\_EXTERNAL\_STORAGE*}, *PERMISSION\_GRANTED*);  
 StrictMode.VmPolicy.Builder builder = new StrictMode.VmPolicy.Builder();  
 StrictMode.*setVmPolicy*(builder.build());  
  
 findViewById(R.id.*ssButton*).setOnClickListener(v -> {  
 View view1 = getWindow().getDecorView().getRootView();  
 view1.setDrawingCacheEnabled(true);  
  
 Bitmap bitmap = Bitmap.*createBitmap*(view1.getDrawingCache());  
 view1.setDrawingCacheEnabled(false);  
  
 String filepath = Environment.*getExternalStorageDirectory*()+"/Download/"+ Calendar.*getInstance*().getTime().toString()+".jpg";  
 File fileScreenshot = new File(filepath);  
  
 FileOutputStream fileOutputStream = null;  
  
 try {  
 fileOutputStream = new FileOutputStream(fileScreenshot);  
 bitmap.compress(Bitmap.CompressFormat.*JPEG*,100,fileOutputStream);  
 fileOutputStream.flush();  
 fileOutputStream.close();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 Intent intent = new Intent(Intent.*ACTION\_VIEW*);  
 Uri uri = Uri.*fromFile*(fileScreenshot);  
 intent.setDataAndType(uri,"image/jpeg");  
 intent.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 this.startActivity(intent);  
 });  
 }  
//  
// public void ScreenshotButton(View view){  
// View view1 = getWindow().getDecorView().getRootView();  
// view1.setDrawingCacheEnabled(true);  
//  
// Bitmap bitmap = Bitmap.createBitmap(view1.getDrawingCache());  
// view1.setDrawingCacheEnabled(false);  
//  
// String filepath = Environment.getExternalStorageDirectory()+"/Download/"+ Calendar.getInstance().getTime().toString()+".jpg";  
// File fileScreenshot = new File(filepath);  
//  
// FileOutputStream fileOutputStream = null;  
//  
// try {  
// fileOutputStream = new FileOutputStream(fileScreenshot);  
// bitmap.compress(Bitmap.CompressFormat.JPEG,100,fileOutputStream);  
// fileOutputStream.flush();  
// fileOutputStream.close();  
// } catch (Exception e) {  
// e.printStackTrace();  
// }  
//  
// Intent intent = new Intent(Intent.ACTION\_VIEW);  
// Uri uri = Uri.fromFile(fileScreenshot);  
// intent.setDataAndType(uri,"image/jpeg");  
// intent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);  
// this.startActivity(intent);  
//  
// }  
  
 private ModelRenderable renderable;  
  
 @RequiresApi(api = Build.VERSION\_CODES.*N*)  
 private void buildModel(File file) {  
 RenderableSource renderableSource = RenderableSource  
 .*builder*()  
 .setSource(this, Uri.*parse*(file.getPath()),RenderableSource.SourceType.*GLB*)  
 .setRecenterMode(RenderableSource.RecenterMode.*ROOT*)  
 .build();  
  
 ModelRenderable  
 .*builder*()  
 .setSource(this, renderableSource)  
 .setRegistryId(file.getPath())  
 .build()  
 .thenAccept(modelRenderable -> {  
 Toast.*makeText*(this,"Model Build",Toast.*LENGTH\_SHORT*).show();  
 renderable = modelRenderable;  
 });  
 }  
  
}

**DRAWER ACTIVITY:**

package com.company.thedecorum;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.ActionBarDrawerToggle;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.core.view.GravityCompat;  
import androidx.drawerlayout.widget.DrawerLayout;  
import androidx.fragment.app.Fragment;  
import androidx.fragment.app.FragmentTransaction;  
  
import android.os.Bundle;  
import android.view.MenuItem;  
  
import com.company.thedecorum.Fargments.HomeFragment;  
import com.company.thedecorum.Fargments.LampFragment;  
import com.company.thedecorum.Fargments.RecommendationFragment;  
import com.company.thedecorum.Fargments.TableFragment;  
import com.company.thedecorum.Fargments.VaseFragment;  
import com.google.android.material.navigation.NavigationView;  
  
public class DrawerActivity extends AppCompatActivity implements NavigationView.OnNavigationItemSelectedListener{  
  
 Toolbar toolbar;  
 DrawerLayout mDrawerLayout;  
 ActionBarDrawerToggle mDrawerToggle;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_drawer*);  
  
 // toolbar = findViewById(R.id.toolBar);  
 // setSupportActionBar(toolbar);  
  
 mDrawerLayout = findViewById(R.id.*drawerLayout*);  
 mDrawerToggle = new ActionBarDrawerToggle(this,mDrawerLayout, R.string.*open*, R.string.*close*);  
  
 mDrawerLayout.addDrawerListener(mDrawerToggle);  
 mDrawerToggle.syncState();  
  
 NavigationView navigationView = findViewById(R.id.*nav\_view*);  
 navigationView.setNavigationItemSelectedListener(this);  
  
 showFragments(new HomeFragment());  
 }  
  
 public boolean onNavigationItemSelected(@NonNull MenuItem item) {  
  
 int id = item.getItemId();  
  
 if(id == R.id.*nav\_home*){  
 showFragments(new HomeFragment());  
 mDrawerLayout.closeDrawer(GravityCompat.*START*);  
 }  
  
 if(id == R.id.*nav\_recommendation*){  
 showFragments(new RecommendationFragment());  
 mDrawerLayout.closeDrawer(GravityCompat.*START*);  
 }  
  
 if(id == R.id.*nav\_lamp*){  
 showFragments(new LampFragment());  
 mDrawerLayout.closeDrawer(GravityCompat.*START*);  
 }  
  
 if(id == R.id.*nav\_vase*){  
 showFragments(new VaseFragment());  
 mDrawerLayout.closeDrawer(GravityCompat.*START*);  
 }  
  
 if(id == R.id.*nav\_table*){  
 showFragments(new TableFragment());  
 mDrawerLayout.closeDrawer(GravityCompat.*START*);  
 }  
  
 return false;  
 }  
  
 private void showFragments(Fragment fragment){  
  
 FragmentTransaction ft = getSupportFragmentManager().beginTransaction();  
 ft.replace(R.id.*frame\_layout*, fragment);  
 ft.commit();  
 }  
  
 public void onBackPressed(){  
  
 if (mDrawerLayout.isDrawerOpen(GravityCompat.*START*)) {  
 mDrawerLayout.closeDrawer(GravityCompat.*START*);  
 }  
 else {  
 super.onBackPressed();  
 }  
  
 }  
}

**CUSTOM FILTER ACTIVITY:**

package com.company.thedecorum;  
  
import android.util.Log;  
import android.widget.Filter;  
  
import java.util.ArrayList;  
  
public class CustomFilter extends Filter {  
 ProductAdapter adapter;  
 ArrayList<Product> filterList;  
  
 public CustomFilter(ProductAdapter adapter, ArrayList<Product> filterList) {  
 this.adapter = adapter;  
 this.filterList = filterList;  
 }  
  
 @Override  
 protected FilterResults performFiltering(CharSequence constraint) {  
 FilterResults results = new FilterResults();  
 if(constraint != null && constraint.length()>0){  
 constraint = constraint.toString().toUpperCase();  
 ArrayList<Product> filteredModel = new ArrayList<>();  
 for(int i=0; i<filterList.size();i++){  
 if(filterList.get(i).getName().toUpperCase().contains(constraint)){  
 filteredModel.add(filterList.get(i));  
 }  
 }  
 results.count = filteredModel.size();  
 results.values = filteredModel;  
 }  
 else  
 {  
 results.count = filterList.size();  
 results.values = filterList;  
  
 }  
 Log.*e*("result", ""+results.count);  
 return results;  
 }  
  
 @Override  
 protected void publishResults(CharSequence constraint, FilterResults results) {  
 adapter.products = (ArrayList<Product>) results.values;  
 adapter.notifyDataSetChanged();  
 }  
}

**PRODUCT ADAPTER:**

package com.company.thedecorum;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.content.Context;  
import android.view.ContextMenu;  
import android.view.LayoutInflater;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Filter;  
import android.widget.Filterable;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import com.company.thedecorum.Fargments.HomeFragment;  
import com.firebase.ui.database.FirebaseRecyclerOptions;  
import com.google.ar.sceneform.ux.ArFragment;  
import com.squareup.picasso.Picasso;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class ProductAdapter extends RecyclerView.Adapter<ProductAdapter.MyHolder> implements Filterable {  
 public ArrayList<Product> products;  
 private Context context;  
 private List<Product> list,filterList;  
 private AbcOnClickListener listener;  
 CustomFilter filter;  
  
  
 public ProductAdapter(Context context, ArrayList<Product> list) {  
 this.context = context;  
 this.list = list;  
 this.filterList = list;  
 }  
  
 public ProductAdapter(FirebaseRecyclerOptions<Product> options) {  
 super();  
 }  
  
 @NonNull  
 @Override  
 public ProductAdapter.MyHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int viewType) {  
 View v = LayoutInflater.*from*(context).inflate(R.layout.*product\_layout*,viewGroup,false);  
  
   
 return new MyHolder(v);  
   
   
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull ProductAdapter.MyHolder abc, int i) {  
 Product p = filterList.get(i);  
  
 abc.tvshowname.setText(p.name);  
 abc.tvshowsize.setText(p.size);  
 abc.tvshowprice.setText(p.price);  
  
  
  
 Picasso.*get*().load(p.imageurl).into(abc.imageView);  
  
// abc.imageView.setImageURI(p.imageurl);  
 }  
  
 @Override  
 public int getItemCount() {  
 return filterList.size();  
 }  
  
  
 public void startListening() {  
 }  
  
 @Override  
 public Filter getFilter() {  
  
 return new Filter() {  
 List<Product> filteredList;  
 @Override  
 protected FilterResults performFiltering(CharSequence charSequence) {  
 String charString = charSequence.toString();  
 filteredList = new ArrayList<>();  
 if (charString.isEmpty()) {  
 filteredList = list;  
 } else {  
  
 for (Product row : list) {  
  
 // name match condition. this might differ depending on your requirement  
 // here we are looking for name or phone number match  
 if (row.getName().toLowerCase().contains(charString.toLowerCase())) {  
 filteredList.add(row);  
 }  
 }  
  
 // searchListFiltered = filteredList;  
 }  
  
  
 return new FilterResults();  
 }  
  
 @Override  
 protected void publishResults(CharSequence charSequence, FilterResults filterResults) {  
  
 filterList = new ArrayList<>();  
 filterList = filteredList;  
  
 notifyDataSetChanged();  
 }  
 };  
 }  
  
//  
// @Override  
// public Filter getFilter() {  
// if(filter == null){  
// Log.e("1","1");  
// filter = new CustomFilter(this, filterList);  
// }  
// Log.e("2","2");  
//  
// return filter;  
// }  
  
 public class MyHolder extends RecyclerView.ViewHolder implements View.OnClickListener,View.OnCreateContextMenuListener  
 , MenuItem.OnMenuItemClickListener {  
  
 private TextView tvshowname,tvshowsize, tvshowprice;  
 private ImageView imageView;  
 private ArFragment arFragment;  
  
 public MyHolder(@NonNull View itemView) {  
 super(itemView);  
  
 tvshowname = itemView.findViewById(R.id.*tvshowname*);  
 tvshowsize = itemView.findViewById(R.id.*tvshowsize*);  
 tvshowprice = itemView.findViewById(R.id.*tvshowprice*);  
 imageView = itemView.findViewById(R.id.*imageshowrecord*);  
  
 itemView.setOnClickListener(this);  
 itemView.setOnCreateContextMenuListener(this);  
 }  
  
 @Override  
 public void onClick(View v) {  
 int p = getAdapterPosition();  
  
 if(p != RecyclerView.*NO\_POSITION*){  
 listener.clickme(p);  
 }  
// Intent intent = new Intent(itemView.getContext(),ArActivity.class);  
// intent.putExtra("name", (Parcelable) list.get(getAdapterPosition()));  
// context.startActivity(intent);  
 }  
  
 @Override  
 public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {  
 menu.setHeaderTitle("Select Option");  
 MenuItem cart = menu.add(Menu.*NONE*,1,1,"View Details");  
 MenuItem ar = menu.add(Menu.*NONE*,2,2,"Check In AR");  
  
 cart.setOnMenuItemClickListener(this);  
 ar.setOnMenuItemClickListener(this);  
 }  
  
 @Override  
 public boolean onMenuItemClick(MenuItem item) {  
  
 int pos = getAdapterPosition();  
  
 if(pos != RecyclerView.*NO\_POSITION*){  
 switch (item.getItemId()){  
 case 1:  
 listener.viewdetail(pos);  
 return true;  
 case 2:  
 listener.ar(pos);  
 return true;  
 }  
 }  
 return false;  
 }  
 }  
  
 public interface AbcOnClickListener{  
 void clickme(int pos);  
 void viewdetail(int pos);  
 void ar(int pos);  
 }  
  
 public void setabcOnClickListener(AbcOnClickListener mlistener){  
 listener = mlistener;  
 }  
}

**LAMP ADAPTER:**

package com.company.thedecorum;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.content.Context;  
import android.view.ContextMenu;  
import android.view.LayoutInflater;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Filter;  
import android.widget.Filterable;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import com.company.thedecorum.Fargments.HomeFragment;  
import com.firebase.ui.database.FirebaseRecyclerOptions;  
import com.google.ar.sceneform.ux.ArFragment;  
import com.squareup.picasso.Picasso;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class LampAdapter extends RecyclerView.Adapter<LampAdapter.MyHolder> implements Filterable {  
 public ArrayList<Lamp> lamps;  
 private Context context;  
 private List<Lamp> list,filterList;  
 private AbcOnClickListener listener;  
 CustomFilter filter;  
  
  
 public LampAdapter(Context context, ArrayList<Lamp> list) {  
 this.context = context;  
 this.list = list;  
 this.filterList = list;  
 }  
  
 public LampAdapter(FirebaseRecyclerOptions<Lamp> options) {  
 super();  
 }  
  
 @NonNull  
 @Override  
 public LampAdapter.MyHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int viewType) {  
 View v = LayoutInflater.*from*(context).inflate(R.layout.*product\_layout*,viewGroup,false);  
  
  
 return new MyHolder(v);  
  
  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull LampAdapter.MyHolder abc, int i) {  
 Lamp p = filterList.get(i);  
  
 abc.tvshowname.setText(p.name);  
 abc.tvshowsize.setText(p.size);  
 abc.tvshowprice.setText(p.price);  
  
  
  
 Picasso.*get*().load(p.imageurl).into(abc.imageView);  
  
// abc.imageView.setImageURI(p.imageurl);  
 }  
  
 @Override  
 public int getItemCount() {  
 return filterList.size();  
 }  
  
  
 public void startListening() {  
 }  
  
 @Override  
 public Filter getFilter() {  
  
 return new Filter() {  
 List<Lamp> filteredList;  
 @Override  
 protected FilterResults performFiltering(CharSequence charSequence) {  
 String charString = charSequence.toString();  
 filteredList = new ArrayList<>();  
 if (charString.isEmpty()) {  
 filteredList = list;  
 } else {  
  
 for (Lamp row : list) {  
  
 // name match condition. this might differ depending on your requirement  
 // here we are looking for name or phone number match  
 if (row.getName().toLowerCase().contains(charString.toLowerCase())) {  
 filteredList.add(row);  
 }  
 }  
  
 // searchListFiltered = filteredList;  
 }  
  
  
 return new FilterResults();  
 }  
  
 @Override  
 protected void publishResults(CharSequence charSequence, FilterResults filterResults) {  
  
 filterList = new ArrayList<>();  
 filterList = filteredList;  
  
 notifyDataSetChanged();  
 }  
 };  
 }  
  
//  
// @Override  
// public Filter getFilter() {  
// if(filter == null){  
// Log.e("1","1");  
// filter = new CustomFilter(this, filterList);  
// }  
// Log.e("2","2");  
//  
// return filter;  
// }  
  
 public class MyHolder extends RecyclerView.ViewHolder implements View.OnClickListener,View.OnCreateContextMenuListener  
 , MenuItem.OnMenuItemClickListener {  
  
 private TextView tvshowname,tvshowsize, tvshowprice;  
 private ImageView imageView;  
 private ArFragment arFragment;  
  
 public MyHolder(@NonNull View itemView) {  
 super(itemView);  
  
 tvshowname = itemView.findViewById(R.id.*tvshowname*);  
 tvshowsize = itemView.findViewById(R.id.*tvshowsize*);  
 tvshowprice = itemView.findViewById(R.id.*tvshowprice*);  
 imageView = itemView.findViewById(R.id.*imageshowrecord*);  
  
 itemView.setOnClickListener(this);  
 itemView.setOnCreateContextMenuListener(this);  
 }  
  
 @Override  
 public void onClick(View v) {  
 int p = getAdapterPosition();  
  
 if(p != RecyclerView.*NO\_POSITION*){  
 listener.clickme(p);  
 }  
// Intent intent = new Intent(itemView.getContext(),ArActivity.class);  
// intent.putExtra("name", (Parcelable) list.get(getAdapterPosition()));  
// context.startActivity(intent);  
 }  
  
 @Override  
 public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {  
 menu.setHeaderTitle("Select Option");  
 MenuItem cart = menu.add(Menu.*NONE*,1,1,"View Details");  
 MenuItem ar = menu.add(Menu.*NONE*,2,2,"Check In AR");  
  
 cart.setOnMenuItemClickListener(this);  
 ar.setOnMenuItemClickListener(this);  
 }  
  
 @Override  
 public boolean onMenuItemClick(MenuItem item) {  
  
 int pos = getAdapterPosition();  
  
 if(pos != RecyclerView.*NO\_POSITION*){  
 switch (item.getItemId()){  
 case 1:  
 listener.viewdetail(pos);  
 return true;  
 case 2:  
 listener.ar(pos);  
 return true;  
 }  
 }  
 return false;  
 }  
 }  
  
 public interface AbcOnClickListener{  
 void clickme(int pos);  
 void viewdetail(int pos);  
 void ar(int pos);  
 }  
  
 public void setabcOnClickListener(AbcOnClickListener mlistener){  
 listener = mlistener;  
 }  
}

**TABLE ADAPTER:**

package com.company.thedecorum;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.content.Context;  
import android.view.ContextMenu;  
import android.view.LayoutInflater;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Filter;  
import android.widget.Filterable;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import com.company.thedecorum.Fargments.HomeFragment;  
import com.firebase.ui.database.FirebaseRecyclerOptions;  
import com.google.ar.sceneform.ux.ArFragment;  
import com.squareup.picasso.Picasso;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class TableAdapter extends RecyclerView.Adapter<TableAdapter.MyHolder> implements Filterable {  
 public ArrayList<Table> tables;  
 private Context context;  
 private List<Table> list,filterList;  
 private AbcOnClickListener listener;  
 CustomFilter filter;  
  
  
 public TableAdapter(Context context, ArrayList<Table> list) {  
 this.context = context;  
 this.list = list;  
 this.filterList = list;  
 }  
  
 public TableAdapter(FirebaseRecyclerOptions<Table> options) {  
 super();  
 }  
  
 @NonNull  
 @Override  
 public TableAdapter.MyHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int viewType) {  
 View v = LayoutInflater.*from*(context).inflate(R.layout.*product\_layout*,viewGroup,false);  
  
  
 return new MyHolder(v);  
  
  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull TableAdapter.MyHolder abc, int i) {  
 Table p = filterList.get(i);  
  
 abc.tvshowname.setText(p.name);  
 abc.tvshowsize.setText(p.size);  
 abc.tvshowprice.setText(p.price);  
  
  
  
 Picasso.*get*().load(p.imageurl).into(abc.imageView);  
  
// abc.imageView.setImageURI(p.imageurl);  
 }  
  
 @Override  
 public int getItemCount() {  
 return filterList.size();  
 }  
  
  
 public void startListening() {  
 }  
  
 @Override  
 public Filter getFilter() {  
  
 return new Filter() {  
 List<Table> filteredList;  
 @Override  
 protected FilterResults performFiltering(CharSequence charSequence) {  
 String charString = charSequence.toString();  
 filteredList = new ArrayList<>();  
 if (charString.isEmpty()) {  
 filteredList = list;  
 } else {  
  
 for (Table row : list) {  
  
 // name match condition. this might differ depending on your requirement  
 // here we are looking for name or phone number match  
 if (row.getName().toLowerCase().contains(charString.toLowerCase())) {  
 filteredList.add(row);  
 }  
 }  
  
 // searchListFiltered = filteredList;  
 }  
  
  
 return new FilterResults();  
 }  
  
 @Override  
 protected void publishResults(CharSequence charSequence, FilterResults filterResults) {  
  
 filterList = new ArrayList<>();  
 filterList = filteredList;  
  
 notifyDataSetChanged();  
 }  
 };  
 }  
  
//  
// @Override  
// public Filter getFilter() {  
// if(filter == null){  
// Log.e("1","1");  
// filter = new CustomFilter(this, filterList);  
// }  
// Log.e("2","2");  
//  
// return filter;  
// }  
  
 public class MyHolder extends RecyclerView.ViewHolder implements View.OnClickListener,View.OnCreateContextMenuListener  
 , MenuItem.OnMenuItemClickListener {  
  
 private TextView tvshowname,tvshowsize, tvshowprice;  
 private ImageView imageView;  
 private ArFragment arFragment;  
  
 public MyHolder(@NonNull View itemView) {  
 super(itemView);  
  
 tvshowname = itemView.findViewById(R.id.*tvshowname*);  
 tvshowsize = itemView.findViewById(R.id.*tvshowsize*);  
 tvshowprice = itemView.findViewById(R.id.*tvshowprice*);  
 imageView = itemView.findViewById(R.id.*imageshowrecord*);  
  
 itemView.setOnClickListener(this);  
 itemView.setOnCreateContextMenuListener(this);  
 }  
  
 @Override  
 public void onClick(View v) {  
 int p = getAdapterPosition();  
  
 if(p != RecyclerView.*NO\_POSITION*){  
 listener.clickme(p);  
 }  
// Intent intent = new Intent(itemView.getContext(),ArActivity.class);  
// intent.putExtra("name", (Parcelable) list.get(getAdapterPosition()));  
// context.startActivity(intent);  
 }  
  
 @Override  
 public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {  
 menu.setHeaderTitle("Select Option");  
 MenuItem cart = menu.add(Menu.*NONE*,1,1,"View Details");  
 MenuItem ar = menu.add(Menu.*NONE*,2,2,"Check In AR");  
  
 cart.setOnMenuItemClickListener(this);  
 ar.setOnMenuItemClickListener(this);  
 }  
  
 @Override  
 public boolean onMenuItemClick(MenuItem item) {  
  
 int pos = getAdapterPosition();  
  
 if(pos != RecyclerView.*NO\_POSITION*){  
 switch (item.getItemId()){  
 case 1:  
 listener.viewdetail(pos);  
 return true;  
 case 2:  
 listener.ar(pos);  
 return true;  
 }  
 }  
 return false;  
 }  
 }  
  
 public interface AbcOnClickListener{  
 void clickme(int pos);  
 void viewdetail(int pos);  
 void ar(int pos);  
 }  
  
 public void setabcOnClickListener(AbcOnClickListener mlistener){  
 listener = mlistener;  
 }  
}

**VASE ADAPTER:**

package com.company.thedecorum;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.content.Context;  
import android.view.ContextMenu;  
import android.view.LayoutInflater;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Filter;  
import android.widget.Filterable;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import com.company.thedecorum.Fargments.HomeFragment;  
import com.firebase.ui.database.FirebaseRecyclerOptions;  
import com.google.ar.sceneform.ux.ArFragment;  
import com.squareup.picasso.Picasso;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class VaseAdapter extends RecyclerView.Adapter<VaseAdapter.MyHolder> implements Filterable {  
 public ArrayList<Vase> vases;  
 private Context context;  
 private List<Vase> list,filterList;  
 private AbcOnClickListener listener;  
 CustomFilter filter;  
  
  
 public VaseAdapter(Context context, ArrayList<Vase> list) {  
 this.context = context;  
 this.list = list;  
 this.filterList = list;  
 }  
  
 public VaseAdapter(FirebaseRecyclerOptions<Vase> options) {  
 super();  
 }  
  
 @NonNull  
 @Override  
 public VaseAdapter.MyHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int viewType) {  
 View v = LayoutInflater.*from*(context).inflate(R.layout.*product\_layout*,viewGroup,false);  
  
  
 return new MyHolder(v);  
  
  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull VaseAdapter.MyHolder abc, int i) {  
 Vase p = filterList.get(i);  
  
 abc.tvshowname.setText(p.name);  
 abc.tvshowsize.setText(p.size);  
 abc.tvshowprice.setText(p.price);  
  
  
  
 Picasso.*get*().load(p.imageurl).into(abc.imageView);  
  
// abc.imageView.setImageURI(p.imageurl);  
 }  
  
 @Override  
 public int getItemCount() {  
 return filterList.size();  
 }  
  
  
 public void startListening() {  
 }  
  
 @Override  
 public Filter getFilter() {  
  
 return new Filter() {  
 List<Vase> filteredList;  
 @Override  
 protected FilterResults performFiltering(CharSequence charSequence) {  
 String charString = charSequence.toString();  
 filteredList = new ArrayList<>();  
 if (charString.isEmpty()) {  
 filteredList = list;  
 } else {  
  
 for (Vase row : list) {  
  
 // name match condition. this might differ depending on your requirement  
 // here we are looking for name or phone number match  
 if (row.getName().toLowerCase().contains(charString.toLowerCase())) {  
 filteredList.add(row);  
 }  
 }  
  
 // searchListFiltered = filteredList;  
 }  
  
  
 return new FilterResults();  
 }  
  
 @Override  
 protected void publishResults(CharSequence charSequence, FilterResults filterResults) {  
  
 filterList = new ArrayList<>();  
 filterList = filteredList;  
  
 notifyDataSetChanged();  
 }  
 };  
 }  
  
//  
// @Override  
// public Filter getFilter() {  
// if(filter == null){  
// Log.e("1","1");  
// filter = new CustomFilter(this, filterList);  
// }  
// Log.e("2","2");  
//  
// return filter;  
// }  
  
 public class MyHolder extends RecyclerView.ViewHolder implements View.OnClickListener,View.OnCreateContextMenuListener  
 , MenuItem.OnMenuItemClickListener {  
  
 private TextView tvshowname,tvshowsize, tvshowprice;  
 private ImageView imageView;  
 private ArFragment arFragment;  
  
 public MyHolder(@NonNull View itemView) {  
 super(itemView);  
  
 tvshowname = itemView.findViewById(R.id.*tvshowname*);  
 tvshowsize = itemView.findViewById(R.id.*tvshowsize*);  
 tvshowprice = itemView.findViewById(R.id.*tvshowprice*);  
 imageView = itemView.findViewById(R.id.*imageshowrecord*);  
  
 itemView.setOnClickListener(this);  
 itemView.setOnCreateContextMenuListener(this);  
 }  
  
 @Override  
 public void onClick(View v) {  
 int p = getAdapterPosition();  
  
 if(p != RecyclerView.*NO\_POSITION*){  
 listener.clickme(p);  
 }  
// Intent intent = new Intent(itemView.getContext(),ArActivity.class);  
// intent.putExtra("name", (Parcelable) list.get(getAdapterPosition()));  
// context.startActivity(intent);  
 }  
  
 @Override  
 public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {  
 menu.setHeaderTitle("Select Option");  
 MenuItem cart = menu.add(Menu.*NONE*,1,1,"View Details");  
 MenuItem ar = menu.add(Menu.*NONE*,2,2,"Check In AR");  
  
 cart.setOnMenuItemClickListener(this);  
 ar.setOnMenuItemClickListener(this);  
 }  
  
 @Override  
 public boolean onMenuItemClick(MenuItem item) {  
  
 int pos = getAdapterPosition();  
  
 if(pos != RecyclerView.*NO\_POSITION*){  
 switch (item.getItemId()){  
 case 1:  
 listener.viewdetail(pos);  
 return true;  
 case 2:  
 listener.ar(pos);  
 return true;  
 }  
 }  
 return false;  
 }  
 }  
  
 public interface AbcOnClickListener{  
 void clickme(int pos);  
 void viewdetail(int pos);  
 void ar(int pos);  
 }  
  
 public void setabcOnClickListener(AbcOnClickListener mlistener){  
 listener = mlistener;  
 }  
}

**LAMP:**

package com.company.thedecorum;  
  
public class Lamp {  
 public String name;  
 public String size;  
 public String imageurl;  
 public String path;  
 public String extension;  
 public String price;  
  
  
 public Lamp(String name,String size,String imageurl){  
 this.name = name;  
 this.size = size;  
 this.imageurl = imageurl;  
 this.path = path;  
 this.extension = extension;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getSize() {  
 return size;  
 }  
  
 public void setSize(String size) {  
 this.size = size;  
 }  
  
 public String getExtension() {  
 return extension;  
 }  
  
 public void setExtension(String extension) {  
 this.extension = extension;  
 }  
  
 public String getModel() {  
 return path;  
 }  
  
 public void setModel(String model) {  
 this.path = model;  
 }  
  
 public Lamp(){}  
  
 public String getImageurl() {  
 return imageurl;  
 }  
  
 public void setImageurl(String imageurl) {  
 this.imageurl = imageurl;  
 }  
  
 public String getPrice() {  
 return price;  
 }  
  
 public void setPrice(String price) {  
 this.price = price;  
 }  
}

**TABLE:**

package com.company.thedecorum;  
  
public class Table {  
 public String name;  
 public String size;  
 public String imageurl;  
 public String path;  
 public String extension;  
 public String price;  
  
  
 public Table(String name,String size,String imageurl){  
 this.name = name;  
 this.size = size;  
 this.imageurl = imageurl;  
 this.path = path;  
 this.extension = extension;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getSize() {  
 return size;  
 }  
  
 public void setSize(String size) {  
 this.size = size;  
 }  
  
 public String getExtension() {  
 return extension;  
 }  
  
 public void setExtension(String extension) {  
 this.extension = extension;  
 }  
  
 public String getModel() {  
 return path;  
 }  
  
 public void setModel(String model) {  
 this.path = model;  
 }  
  
 public Table(){}  
  
 public String getImageurl() {  
 return imageurl;  
 }  
  
 public void setImageurl(String imageurl) {  
 this.imageurl = imageurl;  
 }  
  
 public String getPrice() {  
 return price;  
 }  
  
 public void setPrice(String price) {  
 this.price = price;  
 }  
}

**VASE:**

package com.company.thedecorum;  
  
public class Vase {  
 public String name;  
 public String size;  
 public String imageurl;  
 public String path;  
 public String extension;  
 public String price;  
  
  
 public Vase(String name,String size,String imageurl){  
 this.name = name;  
 this.size = size;  
 this.imageurl = imageurl;  
 this.path = path;  
 this.extension = extension;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getSize() {  
 return size;  
 }  
  
 public void setSize(String size) {  
 this.size = size;  
 }  
  
 public String getExtension() {  
 return extension;  
 }  
  
 public void setExtension(String extension) {  
 this.extension = extension;  
 }  
  
 public String getModel() {  
 return path;  
 }  
  
 public void setModel(String model) {  
 this.path = model;  
 }  
  
 public Vase(){}  
  
 public String getImageurl() {  
 return imageurl;  
 }  
  
 public void setImageurl(String imageurl) {  
 this.imageurl = imageurl;  
 }  
  
 public String getPrice() {  
 return price;  
 }  
  
 public void setPrice(String price) {  
 this.price = price;  
 }  
}

**PRODUCT:**

package com.company.thedecorum;  
  
public class Product {  
 public String name;  
 public String size;  
 public String imageurl;  
 public String path;  
 public String extension;  
 public String price;  
  
  
 public Product(String name,String size,String imageurl){  
 this.name = name;  
 this.size = size;  
 this.imageurl = imageurl;  
 this.path = path;  
 this.extension = extension;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getSize() {  
 return size;  
 }  
  
 public void setSize(String size) {  
 this.size = size;  
 }  
  
 public String getExtension() {  
 return extension;  
 }  
  
 public void setExtension(String extension) {  
 this.extension = extension;  
 }  
  
 public String getModel() {  
 return path;  
 }  
  
 public void setModel(String model) {  
 this.path = model;  
 }  
  
 public Product(){}  
  
 public String getImageurl() {  
 return imageurl;  
 }  
  
 public void setImageurl(String imageurl) {  
 this.imageurl = imageurl;  
 }  
  
 public String getPrice() {  
 return price;  
 }  
  
 public void setPrice(String price) {  
 this.price = price;  
 }  
}

**SPLASH ACTIVITY:**

package com.company.thedecorum;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.os.Handler;  
  
public class SplashActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_splash*);  
  
 new Handler().postDelayed(() -> {  
 Intent i = new Intent(SplashActivity.this,MainActivity.class);  
 startActivity(i);  
  
 finish();  
 },3000);  
 }  
}

**HOME FRAGMENT:**

package com.company.thedecorum.Fargments;  
  
import android.content.Context;  
import android.content.Intent;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.ActionBarDrawerToggle;  
import androidx.appcompat.widget.Toolbar;  
import androidx.drawerlayout.widget.DrawerLayout;  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.DividerItemDecoration;  
import androidx.recyclerview.widget.GridLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.company.thedecorum.ArActivity;  
import com.company.thedecorum.MainActivity;  
import com.company.thedecorum.Product;  
import com.company.thedecorum.ProductAdapter;  
import com.company.thedecorum.R;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
  
import java.util.ArrayList;  
  
  
public class HomeFragment extends Fragment implements ProductAdapter.AbcOnClickListener {  
  
 private RecyclerView Recyclerview;  
 private MainActivity mainActivity;  
 private ProductAdapter adapter;  
 private ArrayList<Product> list;  
 private DatabaseReference reference;  
  
  
 Toolbar toolbar;  
 DrawerLayout mDrawerLayout;  
 ActionBarDrawerToggle mDrawerToggle;  
  
  
 TextView textView;  
  
 public HomeFragment() {  
 // Required empty public constructor  
 }  
  
 @Override  
 public void onAttach(Context activity) {  
 super.onAttach(activity);  
 mainActivity = (MainActivity) activity;  
 }  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 View view= inflater.inflate(R.layout.*fragment\_home*, container, false);  
  
 Recyclerview = view.findViewById(R.id.*recycleview*);  
 list = new ArrayList<>();  
  
 toolbar = view.findViewById(R.id.*toolBar*);  
 // setSupportActionBar(toolbar);  
  
 reference = FirebaseDatabase.*getInstance*().getReference().child("product");  
 Recyclerview.setHasFixedSize(true);  
 Recyclerview.addItemDecoration(new DividerItemDecoration(  
 Recyclerview.getContext(),DividerItemDecoration.*VERTICAL* ));  
 Recyclerview.setLayoutManager(new GridLayoutManager(getActivity(),2));  
 adapter = new ProductAdapter(getActivity(),list);  
 Recyclerview.setAdapter(adapter);  
 adapter.setabcOnClickListener(this);  
 reference.addListenerForSingleValueEvent(listener);  
  
 return view;  
 }  
  
 ValueEventListener listener = new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.exists()){  
 list.clear();  
 for(DataSnapshot snapshot:dataSnapshot.getChildren()){  
 Product p = snapshot.getValue(Product.class);  
 list.add(p);  
 }  
 adapter.notifyDataSetChanged();  
 }  
 else{  
 Toast.*makeText*(getActivity(),"No record found",Toast.*LENGTH\_LONG*).show();  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 Toast.*makeText*(getActivity(),databaseError.toString(),Toast.*LENGTH\_LONG*).show();  
 }  
 };  
  
 @Override  
 public void clickme(int pos) {  
 Toast.*makeText*(getActivity(),pos + "" , Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public void viewdetail(int pos) {  
 Product p = list.get(pos);  
 Toast.*makeText*(getActivity(),p.size +" "+p.price +" "+ p.name,Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public void ar(int pos) {  
 Product p = list.get(pos);  
  
 Intent intent = new Intent(getActivity(), ArActivity.class);  
 intent.putExtra("id", String.*valueOf*(pos));  
 startActivity(intent);  
 }  
}

**LAMP FRAGMENT:**

package com.company.thedecorum.Fargments;  
  
import android.content.Context;  
import android.content.Intent;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.ActionBarDrawerToggle;  
import androidx.appcompat.widget.Toolbar;  
import androidx.drawerlayout.widget.DrawerLayout;  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.DividerItemDecoration;  
import androidx.recyclerview.widget.GridLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.company.thedecorum.ArActivity;  
import com.company.thedecorum.Lamp;  
import com.company.thedecorum.LampAdapter;  
import com.company.thedecorum.MainActivity;  
import com.company.thedecorum.Product;  
import com.company.thedecorum.ProductAdapter;  
import com.company.thedecorum.R;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
  
import java.util.ArrayList;  
  
  
public class LampFragment extends Fragment implements LampAdapter.AbcOnClickListener {  
  
 private RecyclerView Recyclerview;  
 private MainActivity mainActivity;  
 private LampAdapter adapter;  
 private ArrayList<Lamp> list;  
 private DatabaseReference reference;  
  
  
 Toolbar toolbar;  
 DrawerLayout mDrawerLayout;  
 ActionBarDrawerToggle mDrawerToggle;  
  
  
 TextView textView;  
  
 public LampFragment() {  
 // Required empty public constructor  
 }  
  
 @Override  
 public void onAttach(Context activity) {  
 super.onAttach(activity);  
 mainActivity = (MainActivity) activity;  
 }  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 View view= inflater.inflate(R.layout.*fragment\_lamp*, container, false);  
  
 Recyclerview = view.findViewById(R.id.*recycleview*);  
 list = new ArrayList<>();  
  
 //toolbar = view.findViewById(R.id.toolBar);  
 // setSupportActionBar(toolbar);  
  
 reference = FirebaseDatabase.*getInstance*().getReference().child("Lamp");  
 Recyclerview.setHasFixedSize(true);  
 Recyclerview.addItemDecoration(new DividerItemDecoration(  
 Recyclerview.getContext(),DividerItemDecoration.*VERTICAL* ));  
 Recyclerview.setLayoutManager(new GridLayoutManager(getActivity(),2));  
 adapter = new LampAdapter(getActivity(),list);  
 Recyclerview.setAdapter(adapter);  
 adapter.setabcOnClickListener(this);  
 reference.addListenerForSingleValueEvent(listener);  
  
 return view;  
 }  
  
 ValueEventListener listener = new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.exists()){  
 list.clear();  
 for(DataSnapshot snapshot:dataSnapshot.getChildren()){  
 Lamp p = snapshot.getValue(Lamp.class);  
 list.add(p);  
 }  
 adapter.notifyDataSetChanged();  
 }  
 else{  
 Toast.*makeText*(getActivity(),"No record found",Toast.*LENGTH\_LONG*).show();  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 Toast.*makeText*(getActivity(),databaseError.toString(),Toast.*LENGTH\_LONG*).show();  
 }  
 };  
  
 @Override  
 public void clickme(int pos) {  
 Toast.*makeText*(getActivity(),pos + "" , Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public void viewdetail(int pos) {  
 Lamp p = list.get(pos);  
 Toast.*makeText*(getActivity(),p.size +" "+p.price +" "+ p.name,Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public void ar(int pos) {  
 Lamp p = list.get(pos);  
  
 Intent intent = new Intent(getActivity(), ArActivity.class);  
 intent.putExtra("id", String.*valueOf*(pos));  
 startActivity(intent);  
 }  
}

**TABLE FRAGMENT:**

package com.company.thedecorum.Fargments;  
  
import android.content.Context;  
import android.content.Intent;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.ActionBarDrawerToggle;  
import androidx.appcompat.widget.Toolbar;  
import androidx.drawerlayout.widget.DrawerLayout;  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.DividerItemDecoration;  
import androidx.recyclerview.widget.GridLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.company.thedecorum.ArActivity;  
import com.company.thedecorum.Lamp;  
import com.company.thedecorum.LampAdapter;  
import com.company.thedecorum.MainActivity;  
import com.company.thedecorum.Product;  
import com.company.thedecorum.ProductAdapter;  
import com.company.thedecorum.R;  
import com.company.thedecorum.Table;  
import com.company.thedecorum.TableAdapter;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
  
import java.util.ArrayList;  
  
  
public class TableFragment extends Fragment implements TableAdapter.AbcOnClickListener {  
  
 private RecyclerView Recyclerview;  
 private MainActivity mainActivity;  
 private TableAdapter adapter;  
 private ArrayList<Table> list;  
 private DatabaseReference reference;  
  
  
 Toolbar toolbar;  
 DrawerLayout mDrawerLayout;  
 ActionBarDrawerToggle mDrawerToggle;  
  
  
 TextView textView;  
  
 public TableFragment() {  
 // Required empty public constructor  
 }  
  
 @Override  
 public void onAttach(Context activity) {  
 super.onAttach(activity);  
 mainActivity = (MainActivity) activity;  
 }  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 View view= inflater.inflate(R.layout.*fragment\_table*, container, false);  
  
 Recyclerview = view.findViewById(R.id.*recycleview*);  
 list = new ArrayList<>();  
  
 //toolbar = view.findViewById(R.id.toolBar);  
 // setSupportActionBar(toolbar);  
  
 reference = FirebaseDatabase.*getInstance*().getReference().child("Table");  
 Recyclerview.setHasFixedSize(true);  
 Recyclerview.addItemDecoration(new DividerItemDecoration(  
 Recyclerview.getContext(),DividerItemDecoration.*VERTICAL* ));  
 Recyclerview.setLayoutManager(new GridLayoutManager(getActivity(),2));  
 adapter = new TableAdapter(getActivity(),list);  
 Recyclerview.setAdapter(adapter);  
 adapter.setabcOnClickListener(this);  
 reference.addListenerForSingleValueEvent(listener);  
  
 return view;  
 }  
  
 ValueEventListener listener = new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.exists()){  
 list.clear();  
 for(DataSnapshot snapshot:dataSnapshot.getChildren()){  
 Table p = snapshot.getValue(Table.class);  
 list.add(p);  
 }  
 adapter.notifyDataSetChanged();  
 }  
 else{  
 Toast.*makeText*(getActivity(),"No record found",Toast.*LENGTH\_LONG*).show();  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 Toast.*makeText*(getActivity(),databaseError.toString(),Toast.*LENGTH\_LONG*).show();  
 }  
 };  
  
 @Override  
 public void clickme(int pos) {  
 Toast.*makeText*(getActivity(),pos + "" , Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public void viewdetail(int pos) {  
 Table p = list.get(pos);  
 Toast.*makeText*(getActivity(),p.size +" "+p.price +" "+ p.name,Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public void ar(int pos) {  
 Table p = list.get(pos);  
  
 Intent intent = new Intent(getActivity(), ArActivity.class);  
 intent.putExtra("id", String.*valueOf*(pos));  
 startActivity(intent);  
 }  
}

**VASE FRAGMENT:**

package com.company.thedecorum.Fargments;  
  
import android.content.Context;  
import android.content.Intent;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.ActionBarDrawerToggle;  
import androidx.appcompat.widget.Toolbar;  
import androidx.drawerlayout.widget.DrawerLayout;  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.DividerItemDecoration;  
import androidx.recyclerview.widget.GridLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.company.thedecorum.ArActivity;  
import com.company.thedecorum.Lamp;  
import com.company.thedecorum.LampAdapter;  
import com.company.thedecorum.MainActivity;  
import com.company.thedecorum.Product;  
import com.company.thedecorum.ProductAdapter;  
import com.company.thedecorum.R;  
import com.company.thedecorum.Vase;  
import com.company.thedecorum.VaseAdapter;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
  
import java.util.ArrayList;  
  
  
public class VaseFragment extends Fragment implements VaseAdapter.AbcOnClickListener {  
  
 private RecyclerView Recyclerview;  
 private MainActivity mainActivity;  
 private VaseAdapter adapter;  
 private ArrayList<Vase> list;  
 private DatabaseReference reference;  
  
  
 Toolbar toolbar;  
 DrawerLayout mDrawerLayout;  
 ActionBarDrawerToggle mDrawerToggle;  
  
  
 TextView textView;  
  
 public VaseFragment() {  
 // Required empty public constructor  
 }  
  
 @Override  
 public void onAttach(Context activity) {  
 super.onAttach(activity);  
 mainActivity = (MainActivity) activity;  
 }  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 View view= inflater.inflate(R.layout.*fragment\_vase*, container, false);  
  
 Recyclerview = view.findViewById(R.id.*recycleview*);  
 list = new ArrayList<>();  
  
 //toolbar = view.findViewById(R.id.toolBar);  
 // setSupportActionBar(toolbar);  
  
 reference = FirebaseDatabase.*getInstance*().getReference().child("Vase");  
 Recyclerview.setHasFixedSize(true);  
 Recyclerview.addItemDecoration(new DividerItemDecoration(  
 Recyclerview.getContext(),DividerItemDecoration.*VERTICAL* ));  
 Recyclerview.setLayoutManager(new GridLayoutManager(getActivity(),2));  
 adapter = new VaseAdapter(getActivity(),list);  
 Recyclerview.setAdapter(adapter);  
 adapter.setabcOnClickListener(this);  
 reference.addListenerForSingleValueEvent(listener);  
  
 return view;  
 }  
  
 ValueEventListener listener = new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if(dataSnapshot.exists()){  
 list.clear();  
 for(DataSnapshot snapshot:dataSnapshot.getChildren()){  
 Vase p = snapshot.getValue(Vase.class);  
 list.add(p);  
 }  
 adapter.notifyDataSetChanged();  
 }  
 else{  
 Toast.*makeText*(getActivity(),"No record found",Toast.*LENGTH\_LONG*).show();  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 Toast.*makeText*(getActivity(),databaseError.toString(),Toast.*LENGTH\_LONG*).show();  
 }  
 };  
  
 @Override  
 public void clickme(int pos) {  
 Toast.*makeText*(getActivity(),pos + "" , Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public void viewdetail(int pos) {  
 Vase p = list.get(pos);  
 Toast.*makeText*(getActivity(),p.size +" "+p.price +" "+ p.name,Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public void ar(int pos) {  
 Vase p = list.get(pos);  
  
 Intent intent = new Intent(getActivity(), ArActivity.class);  
 intent.putExtra("id", String.*valueOf*(pos));  
 startActivity(intent);  
 }  
}

**XML CODE**

**ACTIVITY\_AR.XML :**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout 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=".ArActivity">  
  
 <fragment  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/arFragment"  
 android:name="com.google.ar.sceneform.ux.ArFragment"/>  
  
 <Button  
 android:id="@+id/downloadBtn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginBottom="25dp"  
 android:text="Download Model" />  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"  
 android:layout\_marginLeft="225dp"  
 android:layout\_marginBottom="25dp"  
 android:text="Take Screenshot"  
 android:id="@+id/ssButton"/>  
  
</RelativeLayout>

**ACTIVITY\_DRAWER.XML :**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.drawerlayout.widget.DrawerLayout 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"  
 android:id="@+id/drawerLayout"  
 tools:context=".DrawerActivity">  
  
 <include  
 android:layout\_height="wrap\_content"  
 android:layout\_width="match\_parent"  
 layout="@layout/app\_bar"/>  
   
 <com.google.android.material.navigation.NavigationView  
 android:layout\_width= "wrap\_content"  
 android:layout\_height="match\_parent"  
 android:id="@+id/nav\_view"  
 android:background="@color/white"  
 android:fitsSystemWindows="true"  
 app:menu="@menu/drawer\_menu"  
 app:headerLayout="@layout/drawer\_header"  
 android:layout\_gravity="start"/>  
  
</androidx.drawerlayout.widget.DrawerLayout>

**ACTIVITY\_MAIN.XML :**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout 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"  
 android:id="@+id/relative"  
 tools:context=".MainActivity">  
  
 <include  
 layout="@layout/activity\_drawer"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 />  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/recycleview">  
 </androidx.recyclerview.widget.RecyclerView>  
  
  
</RelativeLayout>

**ACTIVITY\_SPLASH.XML :**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout 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"  
 android:background="#f1f1f1"  
 style="@style/Theme.AppCompat.NoActionBar">  
  
 <ImageView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerInParent="true"  
 android:src="@drawable/fyp\_logo"  
 android:id="@+id/logo"/>  
  
</RelativeLayout>

**APP\_BAR.XML :**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.coordinatorlayout.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
  
 <androidx.appcompat.widget.Toolbar  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 android:id="@+id/toolBar"  
  
 app:titleTextColor="#f1f1f1"  
 android:elevation="3dp"  
 android:background="@drawable/tool\_bar\_style"  
 android:layout\_margin="4dp"/>  
  
 <FrameLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/frame\_layout"/>  
  
  
  
</androidx.coordinatorlayout.widget.CoordinatorLayout>

**DRAWER\_HEADER.XML :**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="200dp"  
 android:orientation="vertical">  
  
 <ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:scaleType="center"  
 android:src="@drawable/fyp\_logo" />  
  
  
</RelativeLayout>

**PRODUCT\_LAYOUT.XML :**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:weightSum="10"  
  
 android:orientation="vertical"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:background="#f1f1f1">  
  
 <GridLayout  
 android:paddingTop="60dp"  
 android:id="@+id/gridView"  
 android:columnCount="2"  
 android:rowCount="3"  
 android:columnOrderPreserved="true"  
 android:layout\_centerInParent="true"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content">  
  
 <androidx.cardview.widget.CardView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_rowWeight="1"  
 android:layout\_columnWeight="1"  
 android:layout\_gravity="fill"  
 android:layout\_margin="16dp"  
 app:cardCornerRadius="8dp"  
 app:cardElevation="8dp">  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_vertical|center\_horizontal"  
 android:gravity="center"  
 android:orientation="vertical">  
  
 <ImageView  
 android:id="@+id/imageshowrecord"  
 android:layout\_width="150dp"  
 android:layout\_height="150dp"  
 tools:src="@drawable/ic\_android\_black" />  
  
 <TextView  
 android:id="@+id/tvshowname"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textColor="#053359"  
 android:textStyle="bold"  
 tools:text="Product Name" />  
  
 <TextView  
 android:id="@+id/tvshowsize"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/tvshowname"  
 android:text="Size" />  
  
 <TextView  
 android:id="@+id/tvshowprice"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/tvshowprice"  
 android:text="Price" />  
  
  
 </LinearLayout>  
  
  
 </androidx.cardview.widget.CardView>  
 </GridLayout>  
</RelativeLayout>

**SEARCHMENU.XML :**

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
  
  
  
 <item  
 android:id="@+id/search"  
 android:icon="@drawable/ic\_search"  
 android:iconTint="@color/white"  
 android:title="Search Data"  
 app:actionViewClass="android.widget.SearchView"  
 app:showAsAction="always" />  
  
 <item  
 android:id="@+id/action\_settings"  
 android:title="Settings" />  
 <item  
 android:title="Sort"  
 android:visible="true"  
 android:id="@+id/mSort"/>  
  
</menu>

**DRAWER\_MENU.XML :**

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android">  
 <group android:checkableBehavior="single">  
 <item  
 android:id="@+id/nav\_home"  
 android:icon="@drawable/house"  
 android:title="Home" />  
 <item  
 android:id="@+id/nav\_recommendation"  
 android:title="Recommendation"  
 android:icon="@drawable/recommendation" />  
 </group>  
 <item  
 android:title="Categories"  
 android:visible="true">  
 <menu>  
 <item  
 android:id="@+id/nav\_lamp"  
 android:icon="@drawable/lamp"  
 android:title="Lamp" />  
 <item  
 android:id="@+id/nav\_vase"  
 android:icon="@drawable/vase"  
 android:title="Vase" />  
 <item  
 android:id="@+id/nav\_table"  
 android:icon="@drawable/table"  
 android:title="Table" />  
 </menu>  
 </item>  
</menu>

* 1. **Limitation of the System:**
  2. **Future Work:**

In future our “DECORUM” dataset and scope will be scalable. The user might not only be able to try out different interior designing products but they can also try out this application by trying on other e-commerce products too like clothing, electronic items, watches etc. It can also be used in various industries like in auto mobile, medical science etc. New technology may come into existence in future that will help in developing 3D models automatically.

* 1. **Conclusion:**