package com.shruti.lofo.ui.Found;  
  
import android.os.Bundle;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.Spinner;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.firebase.ui.firestore.FirestoreRecyclerOptions;  
import com.google.android.material.floatingactionbutton.FloatingActionButton;  
  
import com.google.firebase.firestore.QueryDocumentSnapshot;  
import com.shruti.lofo.R;  
import com.shruti.lofo.Utility;  
import com.shruti.lofo.databinding.FragmentFoundBinding;  
import com.google.firebase.firestore.Query;  
  
public class FoundFragment extends Fragment {  
  
 private FragmentFoundBinding binding;  
 FloatingActionButton addBtn;  
 RecyclerView recyclerView;  
 FoundItemsAdapter adapter;  
 TextView filter;  
 String selectedCategory = "";  
  
 public View onCreateView(@NonNull LayoutInflater inflater,  
 ViewGroup container, Bundle savedInstanceState) {  
  
 binding = FragmentFoundBinding.*inflate*(inflater, container, false);  
 View root = binding.getRoot();  
  
 recyclerView = root.findViewById(R.id.*foundRecyclerView*);  
  
 addBtn = root.findViewById(R.id.*add\_found*);  
  
 addBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 FoundItemsFragment dialogFragment = new FoundItemsFragment();  
 dialogFragment.show(getParentFragmentManager(), "form\_dialog");  
 }  
 });  
  
 filter = root.findViewById(R.id.*filterButton*);  
 Spinner categorySpinner = root.findViewById(R.id.*categorySpinner*);  
  
 filter.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (categorySpinner.getVisibility() == View.*VISIBLE*) {  
 categorySpinner.setVisibility(View.*GONE*);  
 } else {  
 categorySpinner.setVisibility(View.*VISIBLE*);  
 }  
 }  
 });  
  
 ArrayAdapter<CharSequence> spinnerAdapter = ArrayAdapter.*createFromResource*(getContext(),  
 R.array.*categories\_array*, android.R.layout.*simple\_spinner\_item*);  
 spinnerAdapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*);  
 categorySpinner.setAdapter(spinnerAdapter);  
  
 categorySpinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {  
 @Override  
 public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {  
 selectedCategory = parent.getItemAtPosition(position).toString();  
 setupRecyclerView();  
 }  
  
 @Override  
 public void onNothingSelected(AdapterView<?> parent) {  
 selectedCategory = "";  
 setupRecyclerView();  
 }  
 });  
  
 setupRecyclerView();  
  
 return root;  
 }  
  
 void setupRecyclerView() {  
 Query query;  
 if (selectedCategory.isEmpty()) {  
 query = Utility.*getCollectionReferrenceForFound*().orderBy("dateFound", Query.Direction.*DESCENDING*);  
 } else {  
 query = Utility.*getCollectionReferrenceForFound*().whereEqualTo("category", selectedCategory)  
 .orderBy("dateFound", Query.Direction.*DESCENDING*);  
 }  
 query.get().addOnSuccessListener(queryDocumentSnapshots -> {  
 for (QueryDocumentSnapshot documentSnapshot : queryDocumentSnapshots) {  
 Log.*d*("FoundFragment", "Document ID: " + documentSnapshot.getId());  
 // Log other fields to ensure that the data is retrieved correctly  
 }  
 }).addOnFailureListener(e -> {  
 Log.*e*("FoundFragment", "Error fetching data: " + e.getMessage());  
 });  
  
 FirestoreRecyclerOptions<FoundItems> options = new FirestoreRecyclerOptions.Builder<FoundItems>()  
 .setQuery(query, FoundItems.class).build();  
  
 // Reinitialize the adapter only if it is null or the category has changed  
 if (adapter == null || !adapter.getCategory().equals(selectedCategory)) {  
 if (adapter != null) {  
 adapter.stopListening();  
 }  
  
 adapter = new FoundItemsAdapter(options, requireContext(), selectedCategory,false);  
 adapter.setCategory(selectedCategory);  
  
 recyclerView.setLayoutManager(new LinearLayoutManager(requireContext()));  
 recyclerView.setAdapter(adapter);  
 adapter.startListening();  
 } else {  
 adapter.updateOptions(options);  
 }  
 }  
  
 @Override  
 public void onStart() {  
 super.onStart();  
 if (adapter != null) {  
 adapter.startListening();  
 }  
 }  
  
 @Override  
 public void onStop() {  
 super.onStop();  
 if (adapter != null) {  
 adapter.stopListening();  
 }  
 }  
  
 @Override  
 public void onResume() {  
 super.onResume();  
 if (adapter != null) {  
 adapter.notifyDataSetChanged();  
 }  
 }  
}