package com.shruti.lofo.ui.DashBoard;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.GridLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.denzcoskun.imageslider.ImageSlider;  
import com.denzcoskun.imageslider.constants.ScaleTypes;  
import com.denzcoskun.imageslider.models.SlideModel;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.firestore.CollectionReference;  
import com.google.firebase.firestore.DocumentSnapshot;  
import com.google.firebase.firestore.FirebaseFirestore;  
import com.google.firebase.firestore.Query;  
import com.google.firebase.firestore.QueryDocumentSnapshot;  
import com.shruti.lofo.R;  
import com.shruti.lofo.databinding.FragmentDashboardBinding;  
import com.shruti.lofo.ui.Found.FoundDetails;  
import com.shruti.lofo.ui.Lost.LostDetails;  
  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.Date;  
import java.util.List;  
  
public class DashBoardFragment extends Fragment {  
 private FragmentDashboardBinding binding;  
 private ArrayList<DashBoardViewModel> arr\_recent\_lofo;  
 private RecyclerRecentLoFoAdapter adapter;  
 private FirebaseFirestore db;  
  
 public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 binding = FragmentDashboardBinding.*inflate*(inflater, container, false);  
 View root = binding.getRoot();  
  
 //image sliderc  
 ImageSlider imageSlider = root.findViewById(R.id.*imageSlider*);  
 ArrayList<SlideModel> slideModels = new ArrayList<>();  
  
 slideModels.add(new SlideModel(R.drawable.*dashboard\_img1*, ScaleTypes.*FIT*));  
 slideModels.add(new SlideModel(R.drawable.*dashboard\_img2*, ScaleTypes.*FIT*));  
// slideModels.add(new SlideModel(R.drawable.dashboard\_img3, ScaleTypes.FIT));  
// slideModels.add(new SlideModel(R.drawable.dashboard\_img4, ScaleTypes.FIT));  
  
 imageSlider.setImageList(slideModels, ScaleTypes.*FIT*);  
  
  
  
 RecyclerView recentLostFoundList = root.findViewById(R.id.*recent\_lost\_found\_list*);  
// recentLostFoundList.setLayoutManager(new LinearLayoutManager(requireContext()));  
  
 arr\_recent\_lofo = new ArrayList<>();  
  
 GridLayoutManager gridLayoutManager = new GridLayoutManager(requireContext(),2,GridLayoutManager.*VERTICAL*,false);  
 recentLostFoundList.setLayoutManager(gridLayoutManager);  
 adapter = new RecyclerRecentLoFoAdapter(requireContext(), arr\_recent\_lofo);  
 recentLostFoundList.setAdapter(adapter);  
  
 db = FirebaseFirestore.*getInstance*();  
  
 // Query the 'lostItems' collection  
 Query lostItemsQuery = db.collection("lostItems");  
  
 // Query the 'foundItems' collection  
 Query foundItemsQuery = db.collection("foundItems");  
  
 // Execute the queries for both lost and found items  
 lostItemsQuery.get().addOnSuccessListener(lostItemsSnapshot -> {  
 foundItemsQuery.get().addOnSuccessListener(foundItemsSnapshot -> {  
 // Create a list to store the combined items  
 List<DocumentSnapshot> mergedItems = new ArrayList<>();  
  
 // Add all lost and found items to the mergedItems list  
 mergedItems.addAll(lostItemsSnapshot.getDocuments());  
 mergedItems.addAll(foundItemsSnapshot.getDocuments());  
  
 // Sort the merged items by date in descending order  
 Collections.*sort*(mergedItems, (o1, o2) -> {  
 String dateLostString1 = o1.getString("dateLost");  
 String dateFoundString1 = o1.getString("dateFound");  
 String dateLostString2 = o2.getString("dateLost");  
 String dateFoundString2 = o2.getString("dateFound");  
  
 SimpleDateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy");  
 Date date1 = null;  
 Date date2 = null;  
  
 if (dateLostString1 != null) {  
 try {  
 date1 = dateFormat.parse(dateLostString1);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 } else if (dateFoundString1 != null) {  
 try {  
 date1 = dateFormat.parse(dateFoundString1);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 }  
  
 if (dateLostString2 != null) {  
 try {  
 date2 = dateFormat.parse(dateLostString2);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 } else if (dateFoundString2 != null) {  
 try {  
 date2 = dateFormat.parse(dateFoundString2);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 }  
  
 if (date1 != null && date2 != null) {  
 return date2.compareTo(date1);  
 } else if (date1 != null) {  
 return -1;  
 } else if (date2 != null) {  
 return 1;  
 }  
 return 0;  
 });  
  
 // Limit the list to 5 items  
 if (mergedItems.size() > 10) {  
 mergedItems = mergedItems.subList(0, 10);  
 }  
  
 // Now, you have the top 5 most recent items from both collections in descending order  
 for (DocumentSnapshot item : mergedItems) {  
 DashBoardViewModel lofo = item.toObject(DashBoardViewModel.class);  
 if (lofo != null) {  
 arr\_recent\_lofo.add(lofo);  
 }  
 }  
 adapter.notifyDataSetChanged();  
 });  
 });  
  
  
  
 adapter.setOnItemClickListener(new RecyclerRecentLoFoAdapter.OnItemClickListener() {  
 @Override  
 public void onItemClick(DashBoardViewModel item) {  
 // Handle the item click here  
 String selectedItemName = item.getItemName();  
 // Create an Intent and navigate to LostDetails activity with the selected item name  
 Intent intent;  
 if(item.getTag().equalsIgnoreCase("lost")) {  
 intent = new Intent(requireContext(), LostDetails.class);  
 intent.putExtra("itemId", selectedItemName);  
 }  
 else{  
 intent = new Intent(requireContext(), FoundDetails.class);  
 intent.putExtra("itemId", selectedItemName);  
 }  
 startActivity(intent);  
 }  
 });  
  
// recentLostFoundList.setAdapter(adapter);  
 // Get the currently logged in user  
 FirebaseUser user = FirebaseAuth.*getInstance*().getCurrentUser();  
 TextView userName = root.findViewById(R.id.*userName*); // Replace with your TextView's ID  
  
 if (user != null) {  
 String email = user.getEmail(); // Get the user's email  
  
 FirebaseFirestore db = FirebaseFirestore.*getInstance*();  
 CollectionReference usersCollectionRef = db.collection("users");  
  
 Query query = usersCollectionRef.whereEqualTo("email", email);  
  
 query.get().addOnCompleteListener(task -> {  
 if (task.isSuccessful()) {  
 for (QueryDocumentSnapshot document : task.getResult()) {  
 String name = document.getString("name");  
 if (name != null) {  
 userName.setText(name); // Set the user's name in the TextView  
 }  
 }  
 } else {  
 Log.*d*("FirebaseDebug", "Error getting documents: ", task.getException());  
 }  
 });  
 } else {  
 Log.*d*("FirebaseDebug", "No user currently logged in.");  
 }  
  
  
 return root;  
 }  
  
 @Override  
 public void onDestroyView() {  
 super.onDestroyView();  
 binding = null;  
 }  
}