package com.appdroid.tourstravelsproject.User;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.GridLayoutManager;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
import androidx.viewpager2.widget.ViewPager2;  
  
import android.content.Intent;  
import android.graphics.Color;  
import android.os.Bundle;  
import android.os.Handler;  
import android.text.Html;  
import android.util.Log;  
import android.view.View;  
import android.widget.Adapter;  
import android.widget.ImageView;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.appdroid.tourstravelsproject.Adapter.DestinationAdapter;  
import com.appdroid.tourstravelsproject.Adapter.HotelAdapter;  
import com.appdroid.tourstravelsproject.Adapter.PackageAdapter;  
import com.appdroid.tourstravelsproject.Adapter.SliderAdepter;  
import com.appdroid.tourstravelsproject.Admin.ViewHotels;  
import com.appdroid.tourstravelsproject.Holder.DestinationHolder;  
import com.appdroid.tourstravelsproject.Holder.HotelHolder;  
import com.appdroid.tourstravelsproject.Holder.PackageHolder;  
import com.appdroid.tourstravelsproject.R;  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.OnFailureListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.firestore.DocumentSnapshot;  
import com.google.firebase.firestore.FirebaseFirestore;  
import com.google.firebase.firestore.Query;  
import com.google.firebase.firestore.QuerySnapshot;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class UserDashboard extends AppCompatActivity {  
  
 List<String> sliderList;  
 SliderAdepter sliderAdepter;  
 List<DestinationHolder> holder;  
 ViewPager2 viewPager2;  
 Handler handler;  
  
 RecyclerView destinationRecyclerView;  
 LinearLayoutManager manager;  
 DestinationAdapter adapter;  
  
 RecyclerView packageRecyclerView;  
 LinearLayoutManager packageManager;  
 PackageAdapter packageAdapter;  
 List<PackageHolder> packageList;  
  
 RecyclerView hotelRecyclerView;  
 LinearLayoutManager hotelManager;  
 HotelAdapter hotelAdapter;  
 List<HotelHolder> hotelList;  
  
 ImageView bookingHistory;  
 LinearLayout sendQuery;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_user\_dashboard*);  
  
 sendQuery = findViewById(R.id.*sendQuery*);  
 bookingHistory = findViewById(R.id.*bookingHistory*);  
  
 handler = new Handler();  
 viewPager2 = findViewById(R.id.*viewPager*);  
  
 destinationRecyclerView = findViewById(R.id.*destinationRecyclerView*);  
 holder = new ArrayList<>();  
 manager = new LinearLayoutManager(this);  
 manager.setOrientation(RecyclerView.*HORIZONTAL*);  
 destinationRecyclerView.hasFixedSize();  
 destinationRecyclerView.setLayoutManager(manager);  
  
 packageRecyclerView = findViewById(R.id.*packageRecyclerView*);  
 packageList = new ArrayList<>();  
 packageManager = new LinearLayoutManager(this);  
 packageManager.setOrientation(RecyclerView.*HORIZONTAL*);  
 packageRecyclerView.hasFixedSize();  
 packageRecyclerView.setLayoutManager(packageManager);  
  
 hotelRecyclerView = findViewById(R.id.*hotelRecyclerView*);  
 hotelList = new ArrayList<>();  
 hotelManager = new LinearLayoutManager(this);  
 hotelManager.setOrientation(RecyclerView.*HORIZONTAL*);  
 hotelRecyclerView.hasFixedSize();  
 hotelRecyclerView.setLayoutManager(hotelManager);  
  
 sliderList = new ArrayList<>();  
  
 viewPager2.registerOnPageChangeCallback(new ViewPager2.OnPageChangeCallback() {  
 @Override  
 public void onPageSelected(int position) {  
 super.onPageSelected(position);  
 handler.removeCallbacks(runnable);  
 handler.postDelayed(runnable,3000);  
 }  
 });  
  
 getSliderData();  
 getDestinations();  
 getTourPackages();  
 getHotels();  
  
 bookingHistory.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(getApplicationContext(), BookingHistory.class));  
 }  
 });  
  
 sendQuery.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(getApplicationContext(), SendQuery.class));  
 }  
 });  
 }  
  
 private void getHotels() {  
 Query query = FirebaseFirestore.*getInstance*().collection("Hotels");  
 query.get().addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {  
 @Override  
 public void onComplete(@NonNull Task<QuerySnapshot> task) {  
 if (task.isSuccessful()){  
 for (DocumentSnapshot snapshot : task.getResult()){  
 HotelHolder hotelHolder = snapshot.toObject(HotelHolder.class);  
 hotelHolder.setDocId(snapshot.getId());  
 hotelList.add(hotelHolder);  
 }  
  
 hotelAdapter = new HotelAdapter(UserDashboard.this, hotelList,"User");  
 hotelAdapter.notifyDataSetChanged();  
 hotelRecyclerView.setAdapter(hotelAdapter);  
 }  
 }  
 }).addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 Toast.*makeText*(UserDashboard.this, "Failed...", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
 private void getTourPackages() {  
 Query query = FirebaseFirestore.*getInstance*().collection("TourPackage");  
 query.get().addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {  
 @Override  
 public void onComplete(@NonNull Task<QuerySnapshot> task) {  
 if (task.isSuccessful()){  
 for (DocumentSnapshot snapshot : task.getResult()){  
 PackageHolder packageHolder = snapshot.toObject(PackageHolder.class);  
 packageHolder.setDocId(snapshot.getId());  
 packageList.add(packageHolder);  
 }  
  
 packageAdapter = new PackageAdapter(UserDashboard.this, packageList, "User");  
 packageRecyclerView.setAdapter(packageAdapter);  
 }  
 }  
 }).addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 Toast.*makeText*(UserDashboard.this, "Failed...", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
  
 private void getDestinations() {  
  
 Query query = FirebaseFirestore.*getInstance*().collection("Destination");  
 query.get().addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {  
 @Override  
 public void onComplete(@NonNull Task<QuerySnapshot> task) {  
 if (task.isSuccessful()){  
 for (DocumentSnapshot snapshot : task.getResult()){  
 DestinationHolder destinationHolder = snapshot.toObject(DestinationHolder.class);  
 holder.add(destinationHolder);  
 }  
  
 adapter = new DestinationAdapter(UserDashboard.this, holder);  
 destinationRecyclerView.setAdapter(adapter);  
 }  
 }  
 }).addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 Toast.*makeText*(UserDashboard.this, "Failed...", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
  
 private void getSliderData() {  
 Query query = FirebaseFirestore.*getInstance*().collection("Slider").orderBy("date", Query.Direction.*DESCENDING*);  
 query.get().addOnSuccessListener(new OnSuccessListener<QuerySnapshot>() {  
 @Override  
 public void onSuccess(QuerySnapshot queryDocumentSnapshots) {  
  
 for (DocumentSnapshot snapshot:queryDocumentSnapshots.getDocuments()){  
 sliderList.add(snapshot.getString("image"));  
 }  
  
 sliderAdepter = new SliderAdepter(getApplicationContext(),sliderList);  
 viewPager2.setAdapter(sliderAdepter);  
 sliderAdepter.notifyDataSetChanged();  
  
 }  
 }).addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 Toast.*makeText*(UserDashboard.this, "Failed...", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
  
 private Runnable runnable = new Runnable() {  
 @Override  
 public void run() {  
  
 Log.*d*("TAG", "run: insilder "+viewPager2.getCurrentItem() + " size "+sliderList.size());  
 if (viewPager2.getCurrentItem() != (sliderList.size()-1)){  
 viewPager2.setCurrentItem(viewPager2.getCurrentItem() + 1);  
 }else {  
 viewPager2.setCurrentItem(0);  
 }  
  
 }  
 };  
  
}