package com.shruti.lofo;  
  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.Toast;  
  
import com.google.android.material.bottomnavigation.BottomNavigationView;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.ActionBarDrawerToggle;  
import androidx.appcompat.app.AlertDialog;  
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.FragmentManager;  
import androidx.fragment.app.FragmentTransaction;  
  
import com.google.android.material.navigation.NavigationView;  
import com.google.firebase.firestore.FirebaseFirestore;  
import com.shruti.lofo.databinding.ActivityBindNavBinding;  
import com.shruti.lofo.ui.AboutUs.AboutUsFragment;  
import com.shruti.lofo.ui.DashBoard.DashBoardFragment;  
import com.shruti.lofo.ui.Found.FoundFragment;  
import com.shruti.lofo.ui.Help.HelpFragment;  
import com.shruti.lofo.ui.Lost.LostFragment;  
import com.shruti.lofo.ui.MyItems.MyItems;  
import com.shruti.lofo.ui.MyProfile.MyProfileFragment;  
import com.shruti.lofo.ui.PrivacyPolicy.PrivacyPolicyFragment;  
  
  
public class BindingNavigation extends AppCompatActivity implements NavigationView.OnNavigationItemSelectedListener {  
  
 DrawerLayout drawerlayout;  
 BottomNavigationView bottomNavigationView; // Change this to BottomNavigationView  
 FragmentManager fragmentManager;  
 Toolbar toolbar;  
 private ActivityBindNavBinding binding;  
  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_bind\_nav*);  
  
 FirebaseFirestore db = FirebaseFirestore.*getInstance*();  
  
 toolbar = findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
  
  
 drawerlayout = findViewById(R.id.*drawer\_layout*);  
 ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(this, drawerlayout, toolbar, R.string.*OpenDrawer*, R.string.*CloseDrawer*);  
 drawerlayout.addDrawerListener(toggle);  
 toggle.syncState();  
  
 NavigationView navigationView = findViewById(R.id.*nav\_drawer*);  
 navigationView.setNavigationItemSelectedListener(this);  
  
 View headerView = navigationView.getHeaderView(0);  
  
 // Call the method from DrawerManipulator class to update the drawer header  
 DrawerManipulator.*updateDrawerHeader*(this, headerView);  
 // Fix the ID for your BottomNavigationView  
 bottomNavigationView = findViewById(R.id.*nav\_view*);  
  
 bottomNavigationView.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {  
 @Override  
 public boolean onNavigationItemSelected(@NonNull MenuItem item) {  
 int itemId = item.getItemId();  
 if (itemId == R.id.*navigation\_dashboard*) {  
 openFragment(new DashBoardFragment());  
 return true;  
 } else if (itemId == R.id.*navigation\_lost*) {  
 openFragment(new LostFragment());  
 return true;  
 } else if (itemId == R.id.*navigation\_found*) {  
 openFragment(new FoundFragment());  
 return true;  
 } else if (itemId == R.id.*navigation\_help*) {  
 openFragment(new HelpFragment());  
 return true;  
 }  
 return false;  
 }  
 });  
  
 fragmentManager = getSupportFragmentManager();  
 openFragment(new DashBoardFragment());  
  
 }  
  
 @Override  
 public boolean onNavigationItemSelected(@NonNull MenuItem item) {  
 int itemId = item.getItemId();  
  
 if (itemId == R.id.*myprofile\_drawer*) {  
 openFragment(new MyProfileFragment());  
 }  
 else if (itemId == R.id.*lost\_drawer*) {  
 openFragment(new LostFragment());  
 highlightBottomNavigationItem(R.id.*navigation\_lost*);  
 }  
 else if (itemId == R.id.*found\_drawer*) {  
 openFragment(new FoundFragment());  
 highlightBottomNavigationItem(R.id.*navigation\_found*);  
 }  
 else if (itemId == R.id.*aboutus\_drawer*) {  
 openFragment(new AboutUsFragment());  
 }  
 else if (itemId == R.id.*privacy\_drawer*) {  
 openFragment(new PrivacyPolicyFragment());  
 }  
 else if (itemId == R.id.*myitems\_drawer*) {  
 openFragment(new MyItems());  
 }  
 else if (itemId == R.id.*logout\_drawer*) {  
  
 AlertDialog.Builder builder = new AlertDialog.Builder(this);  
 builder.setTitle("Logout");  
 builder.setMessage("Are you sure you want to log out?");  
 builder.setPositiveButton("Yes", (dialogInterface, i) -> {  
 // Update the shared preferences to indicate that the user is logged out  
 SharedPreferences sharedPreferences = getSharedPreferences("loginPrefs", *MODE\_PRIVATE*);  
 SharedPreferences.Editor editor = sharedPreferences.edit();  
 editor.putBoolean("isLoggedIn", false);  
 editor.apply();  
  
 Toast.*makeText*(this, "Logged out successfully!!", Toast.*LENGTH\_SHORT*).show();  
 Intent intent = new Intent(this, Login.class);  
 startActivity(intent);  
 finishAffinity(); // Close all activities in the stack  
 });  
 builder.setNegativeButton("No", (dialogInterface, i) -> {  
 dialogInterface.dismiss();  
 });  
 AlertDialog alertDialog = builder.create();  
 alertDialog.show();  
 }  
  
 drawerlayout.closeDrawer(GravityCompat.*START*);  
 return true;  
 }  
  
 @Override  
 public void onBackPressed() {  
 if (drawerlayout.isDrawerOpen(GravityCompat.*START*)) {  
 drawerlayout.closeDrawer(GravityCompat.*START*);  
 } else {  
 super.onBackPressed();  
 }  
 }  
  
 private void openFragment(Fragment fragment) {  
 FragmentTransaction transaction = fragmentManager.beginTransaction();  
 transaction.replace(R.id.*fragment\_container*, fragment);  
 transaction.commit();  
 }  
  
  
 private void highlightBottomNavigationItem(int itemId) {  
 bottomNavigationView.getMenu().findItem(itemId).setChecked(true);  
 }  
}