**BildirimGonder.java**

package com.example.hayatsinyalim;  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.RadioButton;  
import android.widget.RadioGroup;  
  
import com.google.firebase.auth.FirebaseAuth;  
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 org.json.JSONException;  
import org.json.JSONObject;  
  
import java.io.IOException;  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
import okhttp3.Call;  
import okhttp3.Callback;  
import okhttp3.MediaType;  
import okhttp3.OkHttpClient;  
import okhttp3.Request;  
import okhttp3.RequestBody;  
import okhttp3.Response;  
public class BildirimGonder extends AppCompatActivity {  
 List<String> fcmTokenList = new ArrayList<>();  
 Button btnGonder;  
 EditText edtText;  
 RadioGroup radioGroup;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_bildirim\_gonder*);  
 btnGonder = findViewById(R.id.*gonder*);  
 edtText = findViewById(R.id.*mesaj*);  
 radioGroup = findViewById(R.id.*radioGroup*);  
 tokenal();  
 btnGonder.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String mesaj = edtText.getText().toString();  
 int selectedId = radioGroup.getCheckedRadioButtonId();  
  
 if (selectedId != -1) {  
 // Bir radio butonu seçilmiş  
 RadioButton selectedRadioButton = findViewById(selectedId);  
 String selectedText = selectedRadioButton.getText().toString();  
 sendNoti("Bulunduğunuz Bölgede Bir '" + selectedText + "' Felaketi Gerçekleşti!!!" + "\n" + mesaj + "\nLütfen geri bildirim vermek için bu bildirime tıklayınız veya programa giriş yapınız!!!");  
 } else {  
  
 }  
 AcilDurumBildir();  
 }  
 });  
 }  
 void AcilDurumBildir() {  
 FirebaseDatabase database = FirebaseDatabase.*getInstance*();  
 DatabaseReference myRef = database.getReference("AcilDurumDurumu");  
 Map<String, Object> userMap = new HashMap<>();  
 userMap.put("Durum", "AcilDurum");  
 myRef.setValue(userMap);  
 DatabaseReference databaseReference = FirebaseDatabase.*getInstance*().getReference().child("users");  
 databaseReference.addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 for (DataSnapshot snapshot : dataSnapshot.getChildren()) {  
 DatabaseReference userRef = snapshot.getRef().child("GeriBildirimdurumu");  
 userRef.child("GeriBildirimi").setValue("GeriBildirimVerilmedi")  
 .addOnSuccessListener(aVoid -> {  
 })  
 .addOnFailureListener(e -> {  
 });  
 DatabaseReference userRefs = snapshot.getRef().child("Gördümü");  
 userRefs.setValue("Görmedi")  
 .addOnSuccessListener(aVoid -> {  
 })  
 .addOnFailureListener(e -> {  
 });  
 }  
 }  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
 }  
 void tokenal()  
 {  
 DatabaseReference databaseReference = FirebaseDatabase.*getInstance*().getReference().child("users");  
 ValueEventListener tokenListener = new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 for (DataSnapshot snapshot : dataSnapshot.getChildren()) {  
 String fcmToken = snapshot.child("FcmToken").getValue(String.class);  
 if (fcmToken != null) {  
 fcmTokenList.add(fcmToken);  
 }  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 System.*out*.println("Veri okuma işlemi iptal edildi: " + databaseError.getMessage());  
 }  
 };  
 databaseReference.addListenerForSingleValueEvent(tokenListener);  
 }  
 void sendNoti(String mes) {  
 try {  
 for (String token : fcmTokenList)  
 {  
 JSONObject not = new JSONObject();  
 not.put("title","Hayat Sinyalim");  
 not.put("body", mes);  
 JSONObject nots = new JSONObject();  
 nots.put("notification",not);  
 nots.put("to",token);  
 callApi(nots);  
  
 }  
 }  
 catch (JSONException e)  
 {  
 throw new RuntimeException(e);  
 }  
 }  
 void callApi( JSONObject jsonObject)  
 {  
 MediaType JSON = MediaType.*get*("application/json; charset=utf-8");  
 OkHttpClient client = new OkHttpClient();  
 String Url = "https://fcm.googleapis.com/fcm/send";  
 RequestBody body = RequestBody.*create*(jsonObject.toString(),JSON);  
 Request request = new Request.Builder()  
 .url(Url)  
 .post(body)  
 .header("Authorization","Bearer AAAAU2qESqo:APA91bGdTOmZ9DPSEXDx\_9nVkzsVx\_1ZxUff4t-qJTNxzguo6gz7sR4o77ZSWXt96\_jdKvSQwAY13U3\_XI2iKsnKqpt-jsmEWYUl--XV1EnpkHfWe7k5n0caRlW8GykLsln564ELk7JL")  
 .build();  
 client.newCall(request).enqueue(new Callback() {  
 @Override  
 public void onFailure(@NonNull Call call, @NonNull IOException e) {  
 }  
 @Override  
 public void onResponse(@NonNull Call call, @NonNull Response response) throws IOException {  
 }  
 });  
 }  
}

**Açıklama ;**

1. **Firebase Bağlantısı ve Veri İşlemleri:**
   * Firebase üzerindeki veritabanından kullanıcıların FCM token'larını alarak bir listeye depoluyor. Bu token'lar bildirim gönderme işlemi için kullanılacak.
   * Acil durum bildirimi için bir Firebase veritabanı düğümüne "AcilDurum" durumu gönderiyor.
   * Kullanıcıların geri bildirim durumlarını ve "Gördümü" durumlarını güncelliyor.
2. **Bildirim Gönderme:**
   * **sendNoti** fonksiyonu, Firebase Cloud Messaging API'sini kullanarak aldığı mesajı (**title** ve **body**) FCM token'larına sahip kullanıcılara gönderiyor.
   * Bu işlem JSON formatında bildirim yapısını oluşturuyor ve **callApi** fonksiyonu aracılığıyla HTTP POST isteği yaparak bildirimi gönderiyor.