**NIMMI INTEGRATED VOICE ASSISTANT**

**SOURCE CODE:**

package com.example.nimmifinal.login;

//import static com.example.login.Functions.fetchName;

import static com.example.nimmifinal.R.id.profile\_nimmi;

import static com.example.nimmifinal.R.id.rec\_but;

import static com.example.nimmifinal.login.functions.wishMe;

import android.annotation.SuppressLint;

import android.app.SearchManager;

import android.content.Context;

import android.content.Intent;

import android.graphics.Typeface;

import android.media.MediaPlayer;

import android.net.Uri;

import android.os.Build;

import android.os.Bundle;

import android.provider.MediaStore;

import android.provider.Settings;

import android.speech.RecognitionListener;

import android.speech.RecognizerIntent;

import android.speech.SpeechRecognizer;

import android.speech.tts.TextToSpeech;

import android.text.format.DateUtils;

import android.view.View;

import android.view.WindowManager;

import android.widget.TextClock;

import android.widget.TextView;

import android.widget.Toast;

import android.widget.VideoView;

import androidx.annotation.NonNull;

import androidx.annotation.RequiresPermission;

import androidx.appcompat.app.AppCompatActivity;

import com.example.nimmifinal.R;

import com.example.nimmifinal.login.MainActivity;

import com.example.nimmifinal.login.MainActivity3;

import com.firebase.client.Firebase;

import com.google.android.material.floatingactionbutton.FloatingActionButton;

import java.util.ArrayList;

import java.util.Date;

import kotlinx.coroutines.BuildersKt;

import android.Manifest;

import android.content.pm.PackageManager;

import android.os.Bundle;

import android.speech.RecognitionListener;

import android.speech.RecognizerIntent;

import android.speech.SpeechRecognizer;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import java.util.ArrayList;

import java.util.Locale;

public class MainActivity2 extends AppCompatActivity {

VideoView videoView;

private SpeechRecognizer recognizer;

private TextView tvResult;

private TextToSpeech tts;

private BuildersKt Dexter;

static int PERMISSION\_CODE= 100;

FloatingActionButton callbtn;

Firebase firebase;

private static final int REQUEST\_RECORD\_AUDIO\_PERMISSION = 200;

private SpeechRecognizer speechRecognizer;

@SuppressLint("MissingInflatedId")

@Override

public void onCreate(Bundle savedInstanceState) {

TextClock textClock, textampm, textdate;

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main2);

getWindow().setFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN, WindowManager.LayoutParams.FLAG\_FULLSCREEN);

getWindow().addFlags(WindowManager.LayoutParams.FLAG\_KEEP\_SCREEN\_ON);

textClock = findViewById(R.id.clock);

textampm = findViewById(R.id.ampm);

textdate = findViewById(R.id.date);

callbtn = findViewById(R.id.callbtn);

Typeface myfornt = Typeface.createFromAsset(getApplicationContext().getAssets(), "fornt/digital\_fornt.ttf");

textClock.setTypeface(myfornt);

textdate.setTypeface(myfornt);

textampm.setTypeface(myfornt);

firebase.setAndroidContext(this);

String uniqueId = Settings.Secure.getString(getApplicationContext().getContentResolver(), Settings.Secure.ANDROID\_ID);

firebase = new Firebase("https://nimmi-1d18d-default-rtdb.firebaseio.com/User" + uniqueId);

FloatingActionButton rec = (FloatingActionButton) findViewById(rec\_but);

tvResult = findViewById(R.id.tv\_result);

tvResult.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

final String name = tvResult.getText().toString();

Firebase child\_name = firebase.child("Speak");

child\_name.setValue(name);

if (name.isEmpty()) {

tvResult.setError("this is an required field");

tvResult.setEnabled(false);

} else {

tvResult.setError(null);

tvResult.setEnabled(true);

}

}

});

// Dexter.withContext(this)

// .withPermission(Manifest.permission.RECORD\_AUDIO)

// .withListener(new PermissionListener() {

// @Override public void onPermissionGranted(PermissionGrantedResponse response) {/\* ... \*/}

// @Override public void onPermissionDenied(PermissionDeniedResponse response) {/\* ... \*/}

// @Override public void onPermissionRationaleShouldBeShown(PermissionRequest permission, PermissionToken token) {/\* ... \*/}

//}).check();

// findById();

initializeTextToSpeech();

initializeResult();

// fetchName("call Amma and say hello to her");

videoView = findViewById(R.id.videoview);

Uri uri = Uri.parse("android.resource://" + getPackageName() + "/" + R.raw.nimmi\_bk);

videoView.setVideoURI(uri);

videoView.start();

videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {

@Override

public void onPrepared(MediaPlayer mp) {

// mp.setLooping(true);

}

});

TextView txtButton = findViewById(profile\_nimmi);

// imgButton.setSize(imgButton.SIZE\_STANDARD);

txtButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

openpro();

}

});

setContentView(R.layout.activity\_main2);

// Check for microphone permission

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.M) {

if (checkSelfPermission(Manifest.permission.RECORD\_AUDIO) != PackageManager.PERMISSION\_GRANTED) {

requestPermissions(new String[]{Manifest.permission.RECORD\_AUDIO}, REQUEST\_RECORD\_AUDIO\_PERMISSION);

}

}

// Initialize SpeechRecognizer

speechRecognizer = SpeechRecognizer.createSpeechRecognizer(this);

speechRecognizer.setRecognitionListener(new RecognitionListener() {

@Override

public void onReadyForSpeech(Bundle params) {

}

@Override

public void onBeginningOfSpeech() {

}

@Override

public void onRmsChanged(float rmsdB) {

}

@Override

public void onBufferReceived(byte[] buffer) {

}

@Override

public void onEndOfSpeech() {

}

@Override

public void onError(int error) {

}

@Override

public void onResults(Bundle results) {

// Process the recognized speech

ArrayList<String> matches = results.getStringArrayList(SpeechRecognizer.RESULTS\_RECOGNITION);

if (matches != null) {

for (String result : matches) {

if (result.toLowerCase(Locale.getDefault()).contains("wake up")) {

// Detected wake-up word, trigger action

Toast.makeText(MainActivity2.this, "Wake-up word detected!", Toast.LENGTH\_SHORT).show();

response("hi");

{

Intent intent = new Intent(RecognizerIntent.ACTION\_RECOGNIZE\_SPEECH);

intent.putExtra(RecognizerIntent.EXTRA\_LANGUAGE\_MODEL, RecognizerIntent.LANGUAGE\_MODEL\_FREE\_FORM);

intent.putExtra(RecognizerIntent.EXTRA\_LANGUAGE, Locale.getDefault());

speechRecognizer.startListening(intent);

}

}

}

}

}

@Override

public void onPartialResults(Bundle partialResults) {

}

@Override

public void onEvent(int eventType, Bundle params) {

}

});

// Start listening for speech

}

//@Override

private void openpro() {

Intent intent = new Intent(this, MainActivity9.class);

startActivity(intent);

}

@Override

protected void onResume() {

// videoView.resume();

super.onResume();

}

@Override

protected void onRestart() {

videoView.start();

super.onRestart();

}

@Override

protected void onPause() {

videoView.suspend();

super.onPause();

}

@Override

protected void onDestroy() {

videoView.stopPlayback();

super.onDestroy();

}

public void openActivity3() {

Intent intent = new Intent(this, MainActivity3.class);

startActivity(intent);

}

public void openActivity5() {

Intent intent = new Intent(this, MainActivity.class);

startActivity(intent);

}

private void initializeTextToSpeech() {

tts=new TextToSpeech(this, new TextToSpeech.OnInitListener() {

@Override

public void onInit(int i) {

if (tts.getEngines().size()==0){

Toast.makeText(MainActivity2.this, "Engine is not available ", Toast.LENGTH\_SHORT).show();

}else{ String s = wishMe();

//speak(s);

}

}

});

}

public void name(){

videoView=findViewById(R.id.videoview);

Uri i= Uri.parse("android.resource://"+getPackageName()+"/"+R.raw.nim\_name);

videoView.setVideoURI(i);

videoView.start();

videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {

@Override

public void onPrepared(MediaPlayer mp) {

// mp.setLooping(true);

}

});

}

public void angry() {

videoView = findViewById(R.id.videoview);

Uri i = Uri.parse("android.resource://" + getPackageName() + "/" + R.raw.angry);

videoView.setVideoURI(i);

videoView.start();

videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {

@Override

public void onPrepared(MediaPlayer mp) {

// mp.setLooping(true);

}

});

}

public void sad(){

videoView=findViewById(R.id.videoview);

Uri i= Uri.parse("android.resource://"+getPackageName()+"/"+R.raw.sad);

videoView.setVideoURI(i);

videoView.start();

videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {

@Override

public void onPrepared(MediaPlayer mp) {

// mp.setLooping(true);

}

});

}

public void loop(){

videoView=findViewById(R.id.videoview);

Uri i= Uri.parse("android.resource://"+getPackageName()+"/"+R.raw.nimmi\_bk);

videoView.setVideoURI(i);

videoView.start();

videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {

@Override

public void onPrepared(MediaPlayer mp) {

// mp.setLooping(true);

}

});

}

public void hello(){

videoView=findViewById(R.id.videoview);

Uri i= Uri.parse("android.resource://"+getPackageName()+"/"+R.raw.nim\_hello);

videoView.setVideoURI(i);

videoView.start();

videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {

@Override

public void onPrepared(MediaPlayer mp) {

// mp.setLooping(true);

}

});

}

public void lust(){

videoView=findViewById(R.id.videoview);

Uri i= Uri.parse("android.resource://"+getPackageName()+"/"+R.raw.lre);

videoView.setVideoURI(i);

videoView.start();

videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {

@Override

public void onPrepared(MediaPlayer mp) {

// mp.setLooping(true);

}

});

}

public void happy(){

videoView=findViewById(R.id.videoview);

Uri i= Uri.parse("android.resource://"+getPackageName()+"/"+R.raw.happy);

videoView.setVideoURI(i);

videoView.start();

videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {

@Override

public void onPrepared(MediaPlayer mp) {

// mp.setLooping(true);

}

});

}

private void speak(String msg) {

tts.speak(msg, TextToSpeech.QUEUE\_FLUSH, null, null);

}

private void initializeResult() {

if (SpeechRecognizer.isRecognitionAvailable(this)) {

recognizer = SpeechRecognizer.createSpeechRecognizer(this);

recognizer.setRecognitionListener(new RecognitionListener() {

@Override

public void onReadyForSpeech(Bundle bundle) {

}

@Override

public void onBeginningOfSpeech() {

}

@Override

public void onRmsChanged(float v) {

}

@Override

public void onBufferReceived(byte[] bytes) {

}

@Override

public void onEndOfSpeech() {

tvResult = findViewById(R.id.tv\_result);

tvResult.setOnClickListener(new View.OnClickListener(){

@Override

public void onClick(View v) {

final String name=tvResult.getText().toString();

Firebase child\_name=firebase.child("Speak");

child\_name.setValue(name);

if (name.isEmpty()){

// tv\_result.setError("this is an required field");

tvResult.setEnabled(false);

}else

{

// tv\_result.setError(null);

tvResult.setEnabled(true);

}}});

}

@Override

public void onError(int i) {

try{

Thread.sleep(200);

} catch (InterruptedException e) {

speak("oop sorry, i can't understand.. but you may give a report to server");

e.printStackTrace();

}

//openActivity3();

// speak("oop sorry, i can't understand.. but you may give a report to server");

}

@Override

public void onResults(Bundle bundle) {

ArrayList<String> result = bundle.getStringArrayList(SpeechRecognizer.RESULTS\_RECOGNITION);

Toast.makeText(MainActivity2.this, "" + result.get(0), Toast.LENGTH\_SHORT).show();

// tvResult.setText(result.get(0));

response(result.get(0));

}

@Override

public void onPartialResults(Bundle bundle) {

}

@Override

public void onEvent(int i, Bundle bundle) {

}

});

}

}

public void response(@NonNull String msg) {

String msgs = msg.toLowerCase();

if (msgs.contains("call")) {

openActivity5();

}

else if (msgs.startsWith("hi", 0)) {

// speak("Hello nice to see you");

hello();

} else if (msgs.contains("hello")) {

//speak(" Hello , Tell me how can I help you");

hello();

} else if (msgs.matches("sad||sad face||face of sad")) {

//speak(" Hello , Tell me how can I help you");

sad();

}

else if (msgs.matches("lust||lust face||")) {

//speak(" Hello , Tell me how can I help you");

lust();

}

else if (msgs.matches("anger||angry||angry face||face of angry")) {

//speak(" Hello , Tell me how can I help you");

angry();

}

else if (msgs.matches("happy||happy||happy face||smiling face")) {

//speak(" Hello , Tell me how can I help you");

happy();

}else if (msgs.endsWith("your name")) {

// speak("Okay, let me introduce myself I am NEMMI, An artificial intelligence, i can do many things when compare to other AI");

name();

}// else if (msgs.contains("nimmi")) {

// speak("yeah ,tell me ");

//hello();}

else if (msgs.contains("how are you")) {

speak("i am doing great,what about you");

} else if (msgs.endsWith("report")) {

speak(" you can send to the server");

openActivity3();

} else if (msgs.endsWith("time")) {

Date date = new Date();

String time = DateUtils.formatDateTime(this, date.getTime(), DateUtils.FORMAT\_SHOW\_TIME);

speak(("The time is" + time));

} else if (msgs.endsWith("open arlo")) {

speak(" you can send to the server");

// openActivity6();

} else if (msgs.endsWith("date")) {

Date date = new Date();

String todays\_date = DateUtils.formatDateTime(this, date.getDate(), DateUtils.FORMAT\_SHOW\_DATE);

speak(("Todays date is" + todays\_date));

}

if (msgs.endsWith("google")) {

speak("opening google chrome");

Intent intent = new Intent(Intent.ACTION\_VIEW, Uri.parse("https://www.google.com"));

startActivity(intent);

}

else if (msgs.contains("open")) {

if (msgs.contains("browser")) {

speak("opening google chrome");

Intent intent = new Intent(Intent.ACTION\_VIEW, Uri.parse("https://www.google.com"));

startActivity(intent);

}

}

if (msgs.endsWith("chrome")) {

speak("opening google chrome");

Intent intent = new Intent(Intent.ACTION\_VIEW, Uri.parse("https://www.google.com"));

startActivity(intent);

}

if (msgs.endsWith("open youtube")) {

speak("opening YouTube");

Intent intent = new Intent(Intent.ACTION\_VIEW, Uri.parse("https://www.youtube.com"));

startActivity(intent);

} else if (msgs.contains("open")) {

if (msgs.contains("facebook")) {

speak("opening Facebook");

Intent intent = new Intent(Intent.ACTION\_VIEW, Uri.parse("https://www.facebook.com"));

startActivity(intent);

}

} else if (msgs.endsWith("song")) {

//Intent intent=msgs.replace()

// Uri uri = Uri.parse("http://www.youtube.com/search?q=" + msgs);

Intent intent = new Intent(MediaStore.INTENT\_ACTION\_MEDIA\_PLAY\_FROM\_SEARCH);

intent.putExtra(MediaStore.EXTRA\_MEDIA\_FOCUS, MediaStore.Audio.Playlists.ENTRY\_CONTENT\_TYPE);

intent.putExtra(SearchManager.QUERY,msgs);// The user's search query

intent.putExtra(MediaStore.EXTRA\_MEDIA\_TITLE,msgs); // Set the title of the media item you want to play

intent.putExtra(MediaStore.EXTRA\_MEDIA\_FOCUS, MediaStore.Audio.Playlists.ENTRY\_CONTENT\_TYPE);

intent.putExtra(MediaStore.EXTRA\_MEDIA\_TITLE,msgs); // Set the title of the media item you want to play

intent.putExtra(SearchManager.QUERY,msgs); // The user's search query

startActivity(intent);

} else if (msgs.startsWith("tell me ", 0)) {

Intent intent = new Intent(Intent.ACTION\_WEB\_SEARCH);

intent.putExtra (SearchManager.QUERY, msgs);

startActivity(intent);

}

if (msg.endsWith("photos")) {

speak("opening albums");

Intent i = new Intent(MediaStore.ACTION\_PICK\_IMAGES);

startActivity(i);

}

if (msg.endsWith("album")) {

speak("opening albums");

Intent i = new Intent(MediaStore.ACTION\_PICK\_IMAGES);

startActivity(i);

}

if (msg.endsWith("camera")) {

speak("opening camara");

Intent i = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);

startActivity(i);

} else if (msgs.contains("record")) {

if (msg.contains("video")) {

speak("recording video");

Intent i = new Intent(MediaStore.ACTION\_VIDEO\_CAPTURE);

startActivity(i);

}

}

if (msgs.endsWith("spotify")) {

speak("opening spotify");

Intent intent = new Intent(Intent.ACTION\_VIEW, Uri.parse("https://open.spotify.com/search/"+msgs));

startActivity(intent);

// Context ctx = this;

//Intent intent = ctx.getPackageManager().getLaunchIntentForPackage("com.spotify.android");

//startActivity(intent);

} else if (msgs.endsWith("youtube")){

Intent intent = new Intent(Intent.ACTION\_VIEW, Uri.parse("https://www.youtube.com/results?search\_query="+msgs));

startActivity(intent);

}

else if (msgs.contains("open")) {

if (msgs.contains("whatsapp")) {

speak("opening whatsapp");

Context ctx = this;

Intent intent = ctx.getPackageManager().getLaunchIntentForPackage("com.whatsapp.android");

startActivity(intent);

}

} else if (msgs.contains("open")) {

if (msgs.contains("twitter")) {

speak("opening twitter");

Intent intent = new Intent(Intent.ACTION\_VIEW, Uri.parse("https://www.twitter.com"));

startActivity(intent);

}

}

if (msgs.endsWith("9994 920 257")) {

Intent i = new Intent(Intent.ACTION\_CALL);

i.setData(Uri.parse(msgs));

startActivity(i);

}

/\* else{ {

Intent intent = new Intent(Intent.ACTION\_WEB\_SEARCH);

intent.putExtra(SearchManager.QUERY,msgs);

startActivity(intent);

}

/\*if (msgs.endsWith("9994 920 257" )) {

Intent i = new Intent(Intent.ACTION\_CALL);

i.setData(Uri.parse(msgs));

startActivity(i);}\*/

}

public void StartRecording(View view){

Intent intent = new Intent(RecognizerIntent.ACTION\_RECOGNIZE\_SPEECH);

intent.putExtra(RecognizerIntent.EXTRA\_LANGUAGE\_MODEL,RecognizerIntent.LANGUAGE\_MODEL\_FREE\_FORM);

intent.putExtra(RecognizerIntent.EXTRA\_MAX\_RESULTS,1);

recognizer.startListening(intent);

}

}