package com.project.womensafety.presentationLayer.user

import android.Manifest

import android.annotation.SuppressLint

import android.content.Intent

import android.content.pm.PackageManager

import android.net.Uri

import android.os.Build

import android.os.Bundle

import android.provider.MediaStore

import android.telephony.SmsManager

import android.util.Log

import androidx.activity.result.contract.ActivityResultContracts

import androidx.appcompat.app.AppCompatActivity

import androidx.core.app.ActivityCompat

import com.project.womensafety.databinding.ActivityTheMainBinding

import com.project.womensafety.presentationLayer.commonView.CommonClass

import com.project.womensafety.presentationLayer.commonView.showToast

import com.project.womensafety.responsiveLayer.RetrofitC

import com.project.womensafety.responsiveLayer.models.User

import kotlinx.coroutines.CoroutineScope

import kotlinx.coroutines.Dispatchers.IO

import kotlinx.coroutines.Dispatchers.Main

import kotlinx.coroutines.async

import kotlinx.coroutines.withContext

import okhttp3.MediaType.Companion.toMediaTypeOrNull

import okhttp3.MultipartBody

import okhttp3.RequestBody.Companion.toRequestBody

class TheMainActivity : AppCompatActivity() {

private var audioUri: Uri? = null

private var videoUri: Uri? = null

private val p by lazy {

CommonClass(this).p

}

private val register =

registerForActivityResult(ActivityResultContracts.StartActivityForResult()) {

it.data?.data?.let {

videoUri = it

}

}

private val audio = registerForActivityResult(ActivityResultContracts.StartActivityForResult()) {

it.data?.data?.let {

audioUri = it

}

}

private val bind by lazy {

ActivityTheMainBinding.inflate(layoutInflater)

}

@SuppressLint("Recycle")

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(bind.root)

bind.videoCapture.setOnClickListener {

register.launch(Intent(MediaStore.ACTION\_VIDEO\_CAPTURE).apply {

putExtra(MediaStore.EXTRA\_DURATION\_LIMIT, 60)

})

}

bind.audio.setOnClickListener {

audio.launch(

Intent(Intent.ACTION\_GET\_CONTENT).apply {

setType("audio/\*")

}

)

}

bind.sendBtn.setOnClickListener {

if (videoUri == null) {

showToast("Capture a video from your gallery")

} else if (audioUri == null) {

showToast("Capture a audio from your gallery")

} else {

if(check()){

requestPermissions(arrayOf(Manifest.permission.SEND\_SMS),100)

return@setOnClickListener

}

p.show()

val readBytes = contentResolver.openInputStream(audioUri!!)

val videoBytes = contentResolver.openInputStream(videoUri!!)

CoroutineScope(IO).async {

async {

try {

var audio: MultipartBody.Part? = null

var video: MultipartBody.Part? = null

readBytes?.readBytes()

?.toRequestBody("multipart/from-data".toMediaTypeOrNull())?.let {

audio = MultipartBody.Part.createFormData(

"audio", "Audio" + System.currentTimeMillis().toString(), it

)

}

videoBytes?.readBytes()

?.toRequestBody("multipart/from-data".toMediaTypeOrNull())?.let {

video = MultipartBody.Part.createFormData(

"video",

"Video" + System.currentTimeMillis().toString(), it)

}

RetrofitC.api.uploadFile(audio = audio!!, video = video!!)

} catch (e: Exception) {

withContext(Main) {

p.dismiss()

showToast(e.message)

}

null

}

}.await().let {

withContext(Main) {

it?.body()?.message?.let {

Log.i("ViewPartView", it)

if(it=="Success") {

finish()

}else{

setMyFun(it)

}

}

p.dismiss()

}

}

}.start()

}

}

}

private fun check() = ActivityCompat.checkSelfPermission(this,Manifest.permission.SEND\_SMS)!=PackageManager.PERMISSION\_GRANTED

private fun setMyFun(s: String) {

if("Fail"==s){

showToast("Some thing Went Wrong")

}else{

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

intent.getParcelableArrayListExtra("data",User::class.java)

}else{

intent.getParcelableArrayListExtra("data")

}?.let {

var num=0

it.forEach {

sendSms("${it.mobile}",s)

num++

}

if(num==it.size){

showToast("Message sent to near by your Area Peoples")

}

}

}

}

private fun sendSms(mobile: String, s1: String) {

SmsManager.getDefault().let {

it?.apply {

sendTextMessage(mobile,null,s1,null,null)

}

}

}

}