# ANDROID STUDIO – 2

## Multiple select item RecyclerView

1. Gradle file

// Recycler view  
implementation 'com.android.support:appcompat-v7:29.0.0'  
implementation 'com.android.support:design:29.0.0'

1. Create new class Adapter.java
2. Create new layout – recycler\_layout.xml
3. Create new Model class
4. MainActivity.java

package com.example.selectmultipleitems;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.content.res.Resources;  
import android.content.res.TypedArray;  
import android.os.Bundle;  
import android.view.View;  
  
import java.util.ArrayList;  
  
public class MainActivity extends AppCompatActivity implements Adapter.ChnageStatusListener {  
 private RecyclerView recyclerView=null;  
 private Adapter adapter=null;  
 private ArrayList<Model> models;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 buildResources();  
 recyclerView=(RecyclerView)findViewById(R.id.*recyclerView*);  
 RecyclerView.LayoutManager manager=new LinearLayoutManager(this,LinearLayoutManager.*VERTICAL*,false);  
 recyclerView.setLayoutManager(manager);  
 adapter=new Adapter(models,MainActivity.this,this);  
 recyclerView.setAdapter(adapter);  
 }  
  
 private void buildResources() {  
 Resources resources=getResources();  
 if(models==null&& resources!=null){  
 String[] text=resources.getStringArray(R.array.*text*);  
 TypedArray image=resources.obtainTypedArray(R.array.*image*);  
 models=new ArrayList<>();  
 for(int i=0;i<text.length;i++){  
 Model model=new Model();  
 model.setText(text[i]);  
 model.setImage(image.getResourceId(i,R.mipmap.*ic\_launcher*));  
 model.setSelect(false);  
 models.add(model);  
 }  
 }  
 }  
  
 @Override  
 public void onItemChangeListener(int position, Model model) {  
 try{  
 models.set(position,model);  
  
 }catch (Exception e){  
  
 }  
 }  
  
  
 public void setBtnDelete(View v) {  
 updateList();  
 }  
  
 private void updateList() {  
 ArrayList<Model> temp=new ArrayList<>();  
 try{  
 for(int i=0;i<models.size();i++){  
 if(!models.get(i).isSelect()){  
 temp.add(models.get(i));  
 }  
 }  
  
 }catch (Exception e){  
  
 }  
 models=temp;  
 if(models.size()==0){  
 recyclerView.setVisibility(View.*GONE*);  
 }  
 adapter.setModel(models);  
 adapter.notifyDataSetChanged();  
 }  
}

1. Activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".MainActivity">  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recyclerView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="400dp"  
 />  
 <Button  
 android:id="@+id/delete"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Delete"  
 android:onClick="setBtnDelete"  
 />  
</LinearLayout>

1. Adapter.java

package com.example.selectmultipleitems;  
  
import android.content.Context;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
  
import java.util.ArrayList;  
  
public class Adapter extends RecyclerView.Adapter<Adapter.ViewHolder> {  
 public interface ChnageStatusListener{  
 void onItemChangeListener(int position,Model model);  
 }  
  
 ArrayList<Model> models;  
 Context mContext;  
 ChnageStatusListener chnageStatusListener;  
  
 public void setModel(ArrayList<Model> models){  
 this.models=models;  
 }  
  
 public Adapter(ArrayList<Model> models, Context mContext, ChnageStatusListener chnageStatusListener) {  
 this.models = models;  
 this.mContext = mContext;  
 this.chnageStatusListener = chnageStatusListener;  
 }  
  
  
  
 @NonNull  
 @Override  
 public ViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {  
 View v=LayoutInflater.*from*(viewGroup.getContext()).inflate(R.layout.*recycler\_layout*,viewGroup,false);  
 ViewHolder viewHolder=new ViewHolder(v);  
 return viewHolder;  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull final ViewHolder viewHolder, final int i) {  
  
  
  
 Model model=models.get(i);  
 if(model!=null){  
 viewHolder.text.setText(model.getText());  
 viewHolder.position=i;  
 viewHolder.image.setImageResource(model.getImage());  
 if(model.isSelect()){  
 viewHolder.view.setBackgroundColor(mContext.getResources().getColor(R.color.*colorAccent*));  
 }  
 else{  
 viewHolder.view.setBackgroundColor(mContext.getResources().getColor(R.color.*colorPrimary*));  
 }  
 }  
 viewHolder.view.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Model model1=models.get(i);  
 if(model1.isSelect()){  
 model1.setSelect(false);  
 }else{  
 model1.setSelect(true);  
 }  
 models.set(viewHolder.position,model1);  
 if(chnageStatusListener!=null){  
 chnageStatusListener.onItemChangeListener(viewHolder.position,model1);  
 }  
 notifyItemChanged(viewHolder.position);  
  
 }  
 });  
  
  
  
 }  
  
 @Override  
 public int getItemCount() {  
 if(models!=null){  
 return models.size();  
 }  
 return 0;  
 }  
  
 public class ViewHolder extends RecyclerView.ViewHolder {  
 public ImageView image;  
 public TextView text;  
 public View view;  
 public int position;  
  
 public ViewHolder(@NonNull View itemView) {  
 super(itemView);  
 view=itemView;  
 image=itemView.findViewById(R.id.*image*);  
 text=itemView.findViewById(R.id.*text*);  
 }  
 }  
}

1. Model.java

package com.example.selectmultipleitems;  
  
public class Model {  
 private String text;  
 private int image;  
 private boolean isSelect;  
  
 public String getText() {  
 return text;  
 }  
  
 public void setText(String text) {  
 this.text = text;  
 }  
  
 public int getImage() {  
 return image;  
 }  
  
 public void setImage(int image) {  
 this.image = image;  
 }  
  
 public boolean isSelect() {  
 return isSelect;  
 }  
  
 public void setSelect(boolean select) {  
 isSelect = select;  
 }  
}

1. Strings.xml

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <string name="app\_name">SelectMultipleItems</string>  
 <string-array name="text" >  
 <item > India1 </item>  
 <item > India2 </item>  
 <item > India3 </item>  
 <item > India4 </item>  
 <item > India5 </item>  
 </string-array>  
  
 <integer-array name="image">  
 <item>@drawable/ic\_launcher\_foreground</item>  
 <item>@drawable/ic\_launcher\_foreground</item>  
 <item>@drawable/ic\_launcher\_foreground</item>  
 <item>@drawable/ic\_launcher\_foreground</item>  
 <item>@drawable/ic\_launcher\_foreground</item>  
 </integer-array>  
</resources>

1. Recycler\_layout.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="horizontal" android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <ImageView  
 android:id="@+id/image"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 />  
 <TextView  
 android:id="@+id/text"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Hello"  
 android:textColor="@color/colorPrimaryDark"  
 android:textStyle="bold"  
 android:textSize="15dp"  
 />  
  
</LinearLayout>

# Custom Search Bar

[Youtube Video](https://www.youtube.com/watch?v=f1nC8HcbMOw)

1. Activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".MainActivity">  
  
 <SearchView  
 android:id="@+id/searchView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:queryHint="Search here"  
 android:iconifiedByDefault="false"  
 />  
 <ListView  
 android:id="@+id/myList"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 ></ListView>  
  
</LinearLayout>

1. MainActivity.java

package com.example.customsearchbar;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
import android.widget.SearchView;  
import android.widget.Toast;  
  
import java.util.ArrayList;  
  
public class MainActivity extends AppCompatActivity {  
 SearchView searchView;  
 ListView myList;  
  
 ArrayList<String> list;  
 ArrayAdapter<String> adapter;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 searchView = (SearchView) findViewById(R.id.*searchView*);  
 myList = (ListView) findViewById(R.id.*myList*);  
  
 list = new ArrayList<String>();  
 list.add("Monday");  
 list.add("Tuesday");  
 list.add("Wednesday");  
 list.add("Thursday");  
 list.add("Friday");  
 list.add("Saturday");  
 list.add("Sunday");  
  
 adapter = new ArrayAdapter<>(this, android.R.layout.*simple\_list\_item\_1*, list);  
 myList.setAdapter(adapter);  
  
 searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {  
 @Override  
 public boolean onQueryTextSubmit(String s) {  
 return false;  
 }  
  
 @Override  
 public boolean onQueryTextChange(String s) {  
 Toast.*makeText*(MainActivity.this, s, Toast.*LENGTH\_SHORT*).show();  
 adapter.getFilter().filter(s);  
 return false;  
 }  
 });  
 }  
}

# ESP8266 connection – NOT Working

[Youtube video](https://www.youtube.com/watch?v=li4bN1N6ifA)

[Git code](http://allaboutee.com/2015/01/20/esp8266-android-application-for-arduino-pin-control/)

1. ActivityMain.java

package com.example.nodemcuconnection;  
  
import android.app.Activity;  
import android.app.AlertDialog;  
import android.content.Context;  
import android.content.SharedPreferences;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
  
import org.apache.http.HttpResponse;  
import org.apache.http.client.ClientProtocolException;  
import org.apache.http.client.HttpClient;  
import org.apache.http.client.methods.HttpGet;  
import org.apache.http.impl.client.DefaultHttpClient;  
  
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.net.URI;  
import java.net.URISyntaxException;  
  
  
public class MainActivity extends Activity implements View.OnClickListener {  
  
 public final static String *PREF\_IP* = "PREF\_IP\_ADDRESS";  
 public final static String *PREF\_PORT* = "PREF\_PORT\_NUMBER";  
 // declare buttons and text inputs  
 private Button buttonPin11,buttonPin12,buttonPin13;  
 private EditText editTextIPAddress, editTextPortNumber;  
 // shared preferences objects used to save the IP address and port so that the user doesn't have to  
 // type them next time he/she opens the app.  
 SharedPreferences.Editor editor;  
 SharedPreferences sharedPreferences;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 sharedPreferences = getSharedPreferences("HTTP\_HELPER\_PREFS", Context.*MODE\_PRIVATE*);  
 editor = sharedPreferences.edit();  
  
 // assign buttons  
 buttonPin11 = (Button)findViewById(R.id.*buttonPin11*);  
 buttonPin12 = (Button)findViewById(R.id.*buttonPin12*);  
 buttonPin13 = (Button)findViewById(R.id.*buttonPin13*);  
  
 // assign text inputs  
 editTextIPAddress = (EditText)findViewById(R.id.*editTextIPAddress*);  
 editTextPortNumber = (EditText)findViewById(R.id.*editTextPortNumber*);  
  
 // set button listener (this class)  
 buttonPin11.setOnClickListener(this);  
 buttonPin12.setOnClickListener(this);  
 buttonPin13.setOnClickListener(this);  
  
 // get the IP address and port number from the last time the user used the app,  
 // put an empty string "" is this is the first time.  
 editTextIPAddress.setText(sharedPreferences.getString(*PREF\_IP*,""));  
 editTextPortNumber.setText(sharedPreferences.getString(*PREF\_PORT*,""));  
 }  
  
  
 @Override  
 public void onClick(View view) {  
  
 // get the pin number  
 String parameterValue = "";  
 // get the ip address  
 String ipAddress = editTextIPAddress.getText().toString().trim();  
 // get the port number  
 String portNumber = editTextPortNumber.getText().toString().trim();  
  
  
 // save the IP address and port for the next time the app is used  
 editor.putString(*PREF\_IP*,ipAddress); // set the ip address value to save  
 editor.putString(*PREF\_PORT*,portNumber); // set the port number to save  
 editor.commit(); // save the IP and PORT  
  
 // get the pin number from the button that was clicked  
 if(view.getId()==buttonPin11.getId())  
 {  
 parameterValue = "11";  
 }  
 else if(view.getId()==buttonPin12.getId())  
 {  
 parameterValue = "12";  
 }  
 else  
 {  
 parameterValue = "13";  
 }  
  
  
  
 // execute HTTP request  
 if(ipAddress.length()>0 && portNumber.length()>0) {  
 new HttpRequestAsyncTask(  
 view.getContext(), parameterValue, ipAddress, portNumber, "pin"  
 ).execute();  
 }  
 }  
  
 */\*\*  
 \* Description: Send an HTTP Get request to a specified ip address and port.  
 \* Also send a parameter "parameterName" with the value of "parameterValue".  
 \** ***@param*** *parameterValue the pin number to toggle  
 \** ***@param*** *ipAddress the ip address to send the request to  
 \** ***@param*** *portNumber the port number of the ip address  
 \** ***@param*** *parameterName  
 \** ***@return*** *The ip address' reply text, or an ERROR message is it fails to receive one  
 \*/* public String sendRequest(String parameterValue, String ipAddress, String portNumber, String parameterName) {  
 String serverResponse = "ERROR";  
  
 try {  
  
 HttpClient httpclient = new DefaultHttpClient(); // create an HTTP client  
 // define the URL e.g. http://myIpaddress:myport/?pin=13 (to toggle pin 13 for example)  
 URI website = new URI("http://"+ipAddress+":"+portNumber+"/?"+parameterName+"="+parameterValue);  
 HttpGet getRequest = new HttpGet(); // create an HTTP GET object  
 getRequest.setURI(website); // set the URL of the GET request  
 HttpResponse response = httpclient.execute(getRequest); // execute the request  
 // get the ip address server's reply  
 InputStream content = null;  
 content = response.getEntity().getContent();  
 BufferedReader in = new BufferedReader(new InputStreamReader(  
 content  
 ));  
 serverResponse = in.readLine();  
 // Close the connection  
 content.close();  
 } catch (ClientProtocolException e) {  
 // HTTP error  
 serverResponse = e.getMessage();  
 e.printStackTrace();  
 } catch (IOException e) {  
 // IO error  
 serverResponse = e.getMessage();  
 e.printStackTrace();  
 } catch (URISyntaxException e) {  
 // URL syntax error  
 serverResponse = e.getMessage();  
 e.printStackTrace();  
 }  
 // return the server's reply/response text  
 return serverResponse;  
 }  
  
  
 */\*\*  
 \* An AsyncTask is needed to execute HTTP requests in the background so that they do not  
 \* block the user interface.  
 \*/* private class HttpRequestAsyncTask extends AsyncTask<Void, Void, Void> {  
  
 // declare variables needed  
 private String requestReply,ipAddress, portNumber;  
 private Context context;  
 private AlertDialog alertDialog;  
 private String parameter;  
 private String parameterValue;  
  
 */\*\*  
 \* Description: The asyncTask class constructor. Assigns the values used in its other methods.  
 \** ***@param*** *context the application context, needed to create the dialog  
 \** ***@param*** *parameterValue the pin number to toggle  
 \** ***@param*** *ipAddress the ip address to send the request to  
 \** ***@param*** *portNumber the port number of the ip address  
 \*/* public HttpRequestAsyncTask(Context context, String parameterValue, String ipAddress, String portNumber, String parameter)  
 {  
 this.context = context;  
  
 alertDialog = new AlertDialog.Builder(this.context)  
 .setTitle("HTTP Response From IP Address:")  
 .setCancelable(true)  
 .create();  
  
 this.ipAddress = ipAddress;  
 this.parameterValue = parameterValue;  
 this.portNumber = portNumber;  
 this.parameter = parameter;  
 }  
  
 */\*\*  
 \* Name: doInBackground  
 \* Description: Sends the request to the ip address  
 \** ***@param*** *voids  
 \** ***@return*** *\*/* @Override  
 protected Void doInBackground(Void... voids) {  
 alertDialog.setMessage("Data sent, waiting for reply from server...");  
 if(!alertDialog.isShowing())  
 {  
 alertDialog.show();  
 }  
 requestReply = sendRequest(parameterValue,ipAddress,portNumber, parameter);  
 return null;  
 }  
  
 */\*\*  
 \* Name: onPostExecute  
 \* Description: This function is executed after the HTTP request returns from the ip address.  
 \* The function sets the dialog's message with the reply text from the server and display the dialog  
 \* if it's not displayed already (in case it was closed by accident);  
 \** ***@param*** *aVoid void parameter  
 \*/* @Override  
 protected void onPostExecute(Void aVoid) {  
 alertDialog.setMessage(requestReply);  
 if(!alertDialog.isShowing())  
 {  
 alertDialog.show(); // show dialog  
 }  
 }  
  
 */\*\*  
 \* Name: onPreExecute  
 \* Description: This function is executed before the HTTP request is sent to ip address.  
 \* The function will set the dialog's message and display the dialog.  
 \*/* @Override  
 protected void onPreExecute() {  
 alertDialog.setMessage("Sending data to server, please wait...");  
 if(!alertDialog.isShowing())  
 {  
 alertDialog.show();  
 }  
 }  
  
 }  
}

1. Activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"  
 >  
  
 <LinearLayout  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"  
 android:orientation="vertical">  
  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:text="IP Address:"  
 android:id="@+id/textView" />  
  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="e.g. 192.168.0.10"  
 android:id="@+id/editTextIPAddress" />  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Port Number:"  
 android:id="@+id/textView2" />  
  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="number"  
 android:ems="10"  
 android:hint="e.g. 80"  
 android:id="@+id/editTextPortNumber" />  
  
 <Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Pin 11"  
 android:id="@+id/buttonPin11" />  
  
 <Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Pin 12"  
 android:id="@+id/buttonPin12" />  
  
 <Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Pin 13"  
 android:id="@+id/buttonPin13" />  
 </LinearLayout>  
  
</ScrollView>

1. Gradle file

If you need sdk 23, add this to your gradle:

android {

useLibrary 'org.apache.http.legacy'

}

1. androidManifest.xml

<uses-permission android:name="android.permission.INTERNET" />

# Node MCU connection

[Youtube Video](https://www.youtube.com/watch?v=QytOKNjZx_k)

[Android Studio code](https://github.com/parthao/Android-App-for-NodeMCU)

[Arduino code](https://gist.github.com/parthao/f02e862751ea723d422008dfa48c041b)

# Bar code and QR code scanning

Gradle dependency

implementation 'me.dm7.barcodescanner:zxing:1.9'

implementation 'com.android.support:appcompat-v7:29

MainActivity.java

package com.example.barcodescanner;  
  
import android.app.AlertDialog;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.hardware.Camera;  
import android.net.Uri;  
import android.os.Build;  
import android.os.Bundle;  
import android.util.Log;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import androidx.core.content.ContextCompat;  
  
import com.google.zxing.Result;  
  
import me.dm7.barcodescanner.zxing.ZXingScannerView;  
  
import static android.Manifest.permission.*CAMERA*;  
  
public class MainActivity extends AppCompatActivity implements ZXingScannerView.ResultHandler {  
  
 private static final int *REQUEST\_CAMERA* = 1;  
 private ZXingScannerView scannerView;  
 private static int *camId* = Camera.CameraInfo.*CAMERA\_FACING\_BACK*;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
 scannerView = new ZXingScannerView(this);  
 setContentView(scannerView);  
 int currentApiVersion = Build.VERSION.*SDK\_INT*;  
  
 if(currentApiVersion >= Build.VERSION\_CODES.*M*)  
 {  
 if(checkPermission())  
 {  
 Toast.*makeText*(getApplicationContext(), "Permission already granted!", Toast.*LENGTH\_LONG*).show();  
 }  
 else  
 {  
 requestPermission();  
 }  
 }  
 }  
  
 private boolean checkPermission()  
 {  
 return (ContextCompat.*checkSelfPermission*(getApplicationContext(), *CAMERA*) == PackageManager.*PERMISSION\_GRANTED*);  
 }  
  
 private void requestPermission()  
 {  
 ActivityCompat.*requestPermissions*(this, new String[]{*CAMERA*}, *REQUEST\_CAMERA*);  
 }  
  
 @Override  
 public void onResume() {  
 super.onResume();  
  
 int currentapiVersion = android.os.Build.VERSION.*SDK\_INT*;  
 if (currentapiVersion >= android.os.Build.VERSION\_CODES.*M*) {  
 if (checkPermission()) {  
 if(scannerView == null) {  
 scannerView = new ZXingScannerView(this);  
 setContentView(scannerView);  
 }  
 scannerView.setResultHandler(this);  
 scannerView.startCamera();  
 } else {  
 requestPermission();  
 }  
 }  
 }  
  
 @Override  
 public void onDestroy() {  
 super.onDestroy();  
 scannerView.stopCamera();  
 }  
  
 public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {  
 switch (requestCode) {  
 case *REQUEST\_CAMERA*:  
 if (grantResults.length > 0) {  
  
 boolean cameraAccepted = grantResults[0] == PackageManager.*PERMISSION\_GRANTED*;  
 if (cameraAccepted){  
 Toast.*makeText*(getApplicationContext(), "Permission Granted, Now you can access camera", Toast.*LENGTH\_LONG*).show();  
 }else {  
 Toast.*makeText*(getApplicationContext(), "Permission Denied, You cannot access and camera", Toast.*LENGTH\_LONG*).show();  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*M*) {  
 if (shouldShowRequestPermissionRationale(*CAMERA*)) {  
 showMessageOKCancel("You need to allow access to both the permissions",  
 new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*M*) {  
 requestPermissions(new String[]{*CAMERA*},  
 *REQUEST\_CAMERA*);  
 }  
 }  
 });  
 return;  
 }  
 }  
 }  
 }  
 break;  
 }  
 }  
  
 private void showMessageOKCancel(String message, DialogInterface.OnClickListener okListener) {  
 new AlertDialog.Builder(MainActivity.this)  
 .setMessage(message)  
 .setPositiveButton("OK", okListener)  
 .setNegativeButton("Cancel", null)  
 .create()  
 .show();  
 }  
  
 @Override  
 public void handleResult(Result result) {  
 final String myResult = result.getText();  
 Log.*d*("QRCodeScanner", result.getText());  
 Log.*d*("QRCodeScanner", result.getBarcodeFormat().toString());  
  
 AlertDialog.Builder builder = new AlertDialog.Builder(this);  
 builder.setTitle("Scan Result");  
 builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 scannerView.resumeCameraPreview(MainActivity.this);  
 }  
 });  
 builder.setNeutralButton("Visit", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 Intent browserIntent = new Intent(Intent.*ACTION\_VIEW*, Uri.*parse*(myResult));  
 startActivity(browserIntent);  
 }  
 });  
 builder.setMessage(result.getText());  
 AlertDialog alert1 = builder.create();  
 alert1.show();  
 }  
  
 @Override  
 public void onPointerCaptureChanged(boolean hasCapture) {  
  
 }  
}

## Record Audio and upload to server

Youtube [Video](https://www.youtube.com/watch?v=-pkweD2s-Oo)

1. Add permission to androidmanifest.xml

<uses-permission android:name="android.permission.RECORD\_AUDIO" />  
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

1. MainActivity.java

package com.example.voicerecord;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
  
import android.Manifest;  
import android.app.ProgressDialog;  
import android.media.MediaRecorder;  
import android.net.Uri;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.MotionEvent;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
  
import java.io.File;  
import java.io.IOException;  
  
public class MainActivity extends AppCompatActivity {  
 private static final String *LOG\_TAG* = "AudioRecordTest";  
  
 private Button mRecordBtn;  
 private TextView mRecordLabel;  
 private MediaRecorder recorder = null;  
 private static String *fileName* = null;  
  
 // Requesting permission to RECORD\_AUDIO  
 private boolean permissionToRecordAccepted = false;  
 private String [] permissions = {Manifest.permission.*RECORD\_AUDIO*};  
 private static final int *REQUEST\_RECORD\_AUDIO\_PERMISSION* = 200;  
  
// private StorageReference mStorage;  
 private ProgressDialog mProgress;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 ActivityCompat.*requestPermissions*(this, permissions, *REQUEST\_RECORD\_AUDIO\_PERMISSION*);  
  
 // firebase  
// mStorage = FireSorage.getInstance.getReference();  
 mProgress = new ProgressDialog(this);  
 mRecordLabel = (TextView) findViewById(R.id.*recordLabel*);  
 mRecordBtn = (Button) findViewById(R.id.*recordBtn*);  
  
 // Record to the external cache directory for visibility  
 *fileName* = getExternalCacheDir().getAbsolutePath();  
 *fileName* += "/audiorecordtest.3gp";  
  
 mRecordBtn.setOnTouchListener(new View.OnTouchListener() {  
 @Override  
 public boolean onTouch(View view, MotionEvent motionEvent) {  
 if(motionEvent.getAction() == MotionEvent.*ACTION\_DOWN*){  
 startRecording();  
 mRecordLabel.setText("Recording started ...");  
 }  
 else if(motionEvent.getAction() == MotionEvent.*ACTION\_UP*){  
 stopRecording();  
  
 mRecordLabel.setText(*fileName*);  
 }  
 return true;  
 }  
 });  
 }  
  
 private void startRecording() {  
 recorder = new MediaRecorder();  
 recorder.setAudioSource(MediaRecorder.AudioSource.*MIC*);  
 recorder.setOutputFormat(MediaRecorder.OutputFormat.*THREE\_GPP*);  
 recorder.setOutputFile(*fileName*);  
 recorder.setAudioEncoder(MediaRecorder.AudioEncoder.*AMR\_NB*);  
  
 try {  
 recorder.prepare();  
 } catch (IOException e) {  
 Log.*e*(*LOG\_TAG*, "prepare() failed");  
 }  
  
 recorder.start();  
 }  
  
 private void stopRecording() {  
 recorder.stop();  
 recorder.release();  
 recorder = null;  
 uploadAudio();  
 }  
  
 private void uploadAudio() {  
// mProgress.setTitle("Uploading audio ...");  
// mProgress.show();  
// StorageReference filepath = mStorage.child("Audio").child("new\_audio.3gp");  
// Uri uri = Uri.fromFile(new File(fileName));  
// filepath.putfile.onsuccess{  
// mProgress.dismiss();  
// }  
 }  
  
  
}

1. Activitymain.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <Button  
 android:id="@+id/recordBtn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Tap and hold to record"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/recordLabel"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Tap and hold to record"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/recordBtn" />  
</androidx.constraintlayout.widget.ConstraintLayout>