Introduction

QR Code Generator

QR (Quick Response) codes have become ubiquitous in modern society, offering a convenient way to share information quickly and efficiently. In Android development, integrating a QR code generator into your application can enhance its functionality and provide users with a seamless experience. This tutorial will introduce you to the fundamentals of QR code generation in Android development, guiding you through the process of creating a basic QR code generator app.

- ❖ QR code generators are used to create QR codes for various purposes, including

1.Sharing URLs: Websites often use QR codes to make it easy for users to access their sites on mobile devices.
2.Contact Information: QR codes can encode contact details, making it convenient to share business cards digitally.
3.Product Information: QR codes on product packaging can provide consumers with additional information or promotions.
4.Wi-Fi Access: QR codes can contain Wi-Fi network credentials, allowing users to connect to a network quickly.

- ❖ QR codes can encode different types of data, including:

1.Text: Plain text, such as a message or note.
2.URLs: Web addresses that direct users to specific websites.
3.Contact Information: vCard format containing name, phone number, email, etc.
4.WiFi Credentials: SSID and password for connecting to a wireless network.
Business Information: Company details, addresses, and contact information.

- ❖ Usage:

QR code generators are widely used across various industries and applications, including marketing, advertising, retail, logistics, and more. They provide a convenient way to share information digitally, especially in situations where typing or manual input is cumbersome or impractical

❖ Software Technology

1)Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, offering a suite of powerful tools tailored specifically for building Android applications.

❖ Language

1)xml

XML, or Extensible Markup Language, is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. It is widely used for storing and exchanging structured data between different systems and platforms.

2) java
Java is a widely-used programming language known for its versatility, platform independence, and extensive ecosystem. Originally developed by Sun Microsystems and later acquired by Oracle Corporation, Java has become one of the most popular languages for building a wide range of software applications, including desktop, web, mobile, and enterprise-level systems

❖ ZXing Library Dependency:

One of the most commonly used libraries is the ZXing ("Zebra Crossing") library, which is open-source and widely adopted. Here's a basic guide on how to use ZXing to generate QR codes in an Android application.

Code

Build Gradle

Add 3 dependencies

1)implementation 'com.journeyapps:zxing-android-embedded:4.3.0'

2) coreLibraryDesugaring 'com.android.tools:desugar_jdk_libs:2.0.2'

3) implementation 'androidx.multidex:multidex:2.0.1'

Color added

```
<?xml version="1.0" encoding="utf-8"?>

<resources>

   <color name="purple_200">#ffa500</color>

   <color name="purple_500">#ffa500</color>

   <color name="purple_700">#292D36</color>

   <color name="teal_200">#FF03DAC5</color>

   <color name="teal_700">#FF018786</color>

   <color name="black">#FF000000</color>

   <color name="white">#FFFFFFFF</color>

   <color name="black_shade_1">#292D36</color>

   <color name="yellow">#ffa500</color>

</resources>
```

New color add for Project.

1st page

Activity_main.xml Code :

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:background="@color/black_shade_1"
    tools:context=".MainActivity">
        <ImageView
            android:layout_width="200dp"
            android:layout_height="200dp"
            android:src="@drawable/qrcode"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="50dp"
            android:id="@+id/idIVLogo"
            />
        <TextView
            android:id="@+id/idTVHeading"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@+id/idIVLogo"
            android:layout_marginTop="28dp"
            android:gravity="center"
            android:padding="4dp"
            android:text="Welcome to QR CODE Generator"
            android:textAlignment="center"
            android:textColor="@color/yellow"
            android:textSize="20sp"
            android:textStyle="bold" />
        <Button
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginStart="50dp"
            android:layout_marginEnd="50dp"
            android:layout_marginTop="60dp"
            android:text="Generate QR CODE"
            android:layout_below="@+id/idTVHeading"
            android:id="@+id/idBtnGenerateQR"/>
    </RelativeLayout>
```

MainActivity.java code :

```
package com.example.qrcodeproject;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class MainActivity extends AppCompatActivity {
    private Button generateQRBtn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate (savedInstanceState);
        setContentView (R.layout.activity_main);

        generateQRBtn=findViewById (R.id.idBtnGenerateQR);
        generateQRBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i =new Intent (MainActivity.this,GenerateQRCode.class);
                startActivity (i);
            }
        });
    }
}
```

2nd page

Generate QR code Activity.xml code:

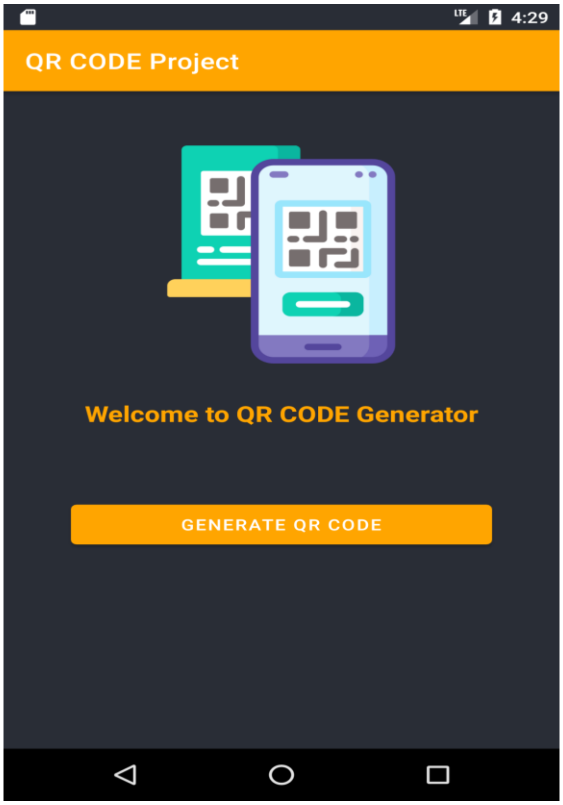
```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout 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=".GenerateQRCode">

  <EditText

      android:id="@+id/esit_text"

      android:layout_width="wrap_content"

      android:layout_height="wrap_content" />

  <Button

      android:id="@+id/button"

      android:layout_width="171dp"

      android:layout_height="wrap_content"

      android:layout_below="@+id/esit_text"

      android:layout_alignParentStart="true"

      android:layout_alignParentEnd="true"

      android:layout_marginStart="110dp"

      android:layout_marginTop="29dp"

      android:layout_marginEnd="112dp"

      android:text="Generated" />
```

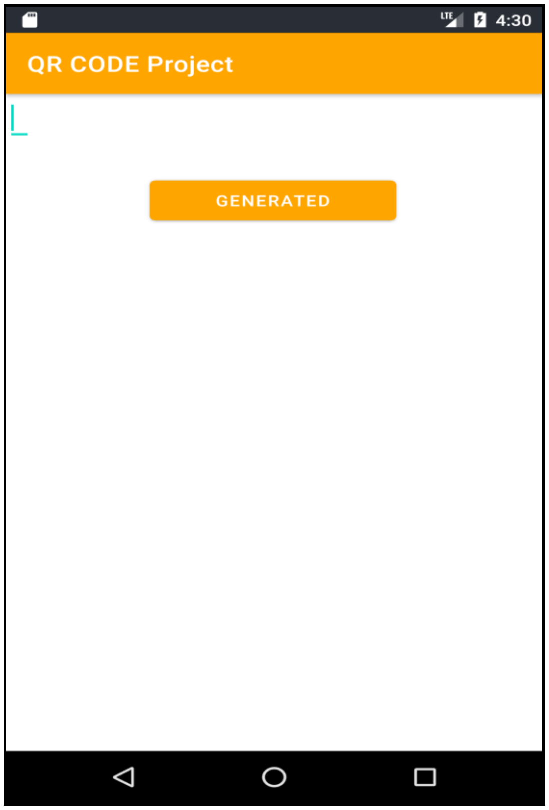
```
    <ImageView
        android:id="@+id/qr_code"
        android:layout_width="match_parent"
        android:layout_height="569dp"
        android:layout_below="@+id/button"
        android:layout_marginTop="15dp"
        android:scaleType="fitCenter"/>
</RelativeLayout>
```

Generate QR code Activity.java code:

```
package com.example.qrcodeproject;

import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Bitmap;

import android.os.Bundle;

import com.google.zxing.BarcodeFormat;

import com.google.zxing.MultiFormatWriter;

import com.google.zxing.WriterException;

import com.google.zxing.common.BitMatrix;

import com.journeyapps.barcodescanner.BarcodeEncoder;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageView;

import android.os.Bundle;



public class GenerateQRCode extends AppCompatActivity {

  @Override

  protected void onCreate(Bundle savedInstanceState) {

      super.onCreate (savedInstanceState);

      setContentView (R.layout.activity_generate_qrcode);

      EditText editText =findViewById(R.id.esit_text);

      Button button =findViewById(R.id.button);

      ImageView imageView =findViewById(R.id.qr_code);

      button.setOnClickListener (new View.OnClickListener () {
            @Override
            public void onClick(View v) {
                MultiFormatWriter multiFormatWriter = new MultiFormatWriter ();
                try {
                    BitMatrix bitMatrix =
      multiFormatWriter.encode(editText.getText().toString(), BarcodeFormat.QR_CODE,
      300, 300);


                    BarcodeEncoder barcodeEncoder = new BarcodeEncoder();
                    Bitmap bitmap = barcodeEncoder.createBitmap (bitMatrix);
                    imageView.setImageBitmap(bitmap);
                } catch (WriterException e) {
                    throw new RuntimeException (e);
                }
            }
        });
```

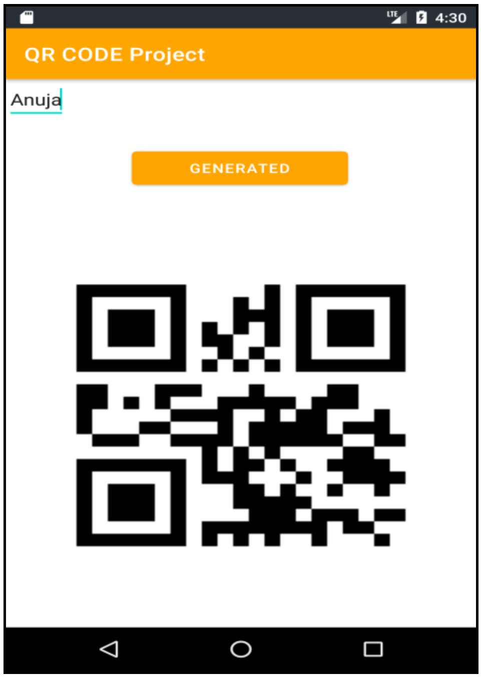
Output

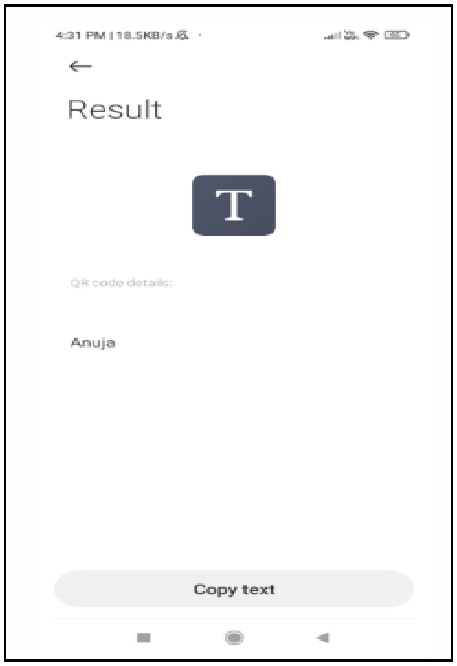




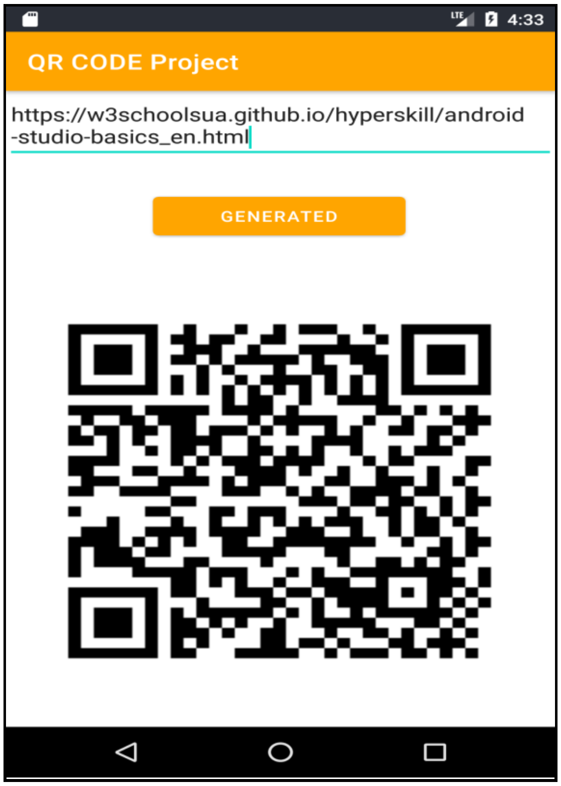
Click the button

This Result check form another phone

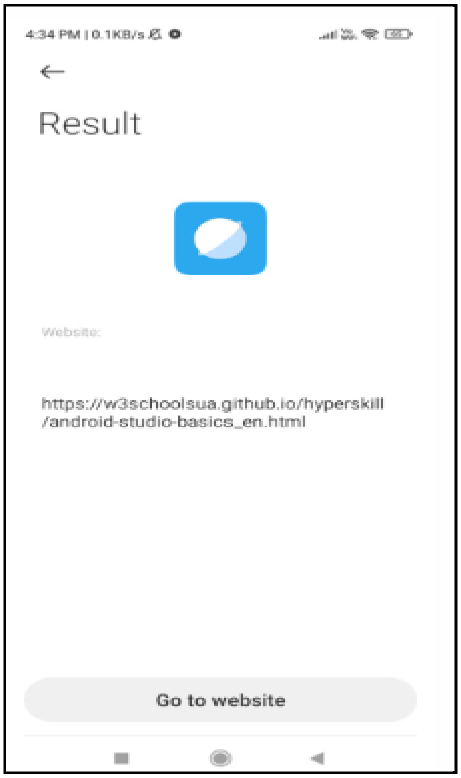




2nd URL link QR CODE

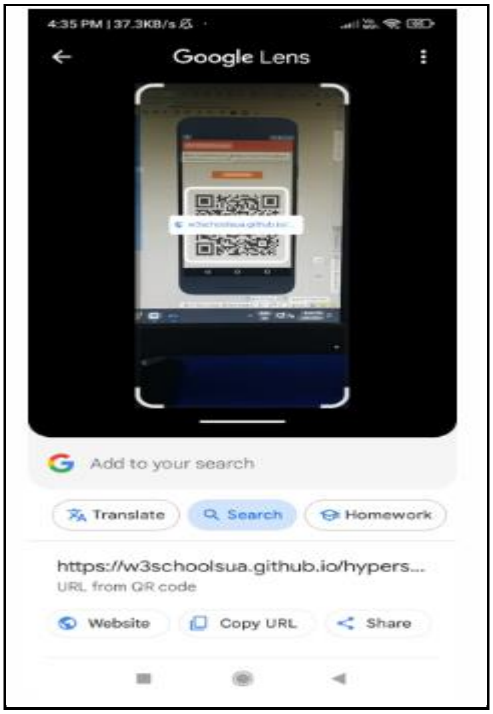


https://w3schoolsua.github.io/hyperskill/android-studio-basics_en.html



Website:

https://w3schoolsua.github.io/hyperskill/android-studio-basics_en.html



https://w3schoolsua.github.io/hypers...
URL from QR code

Thank you