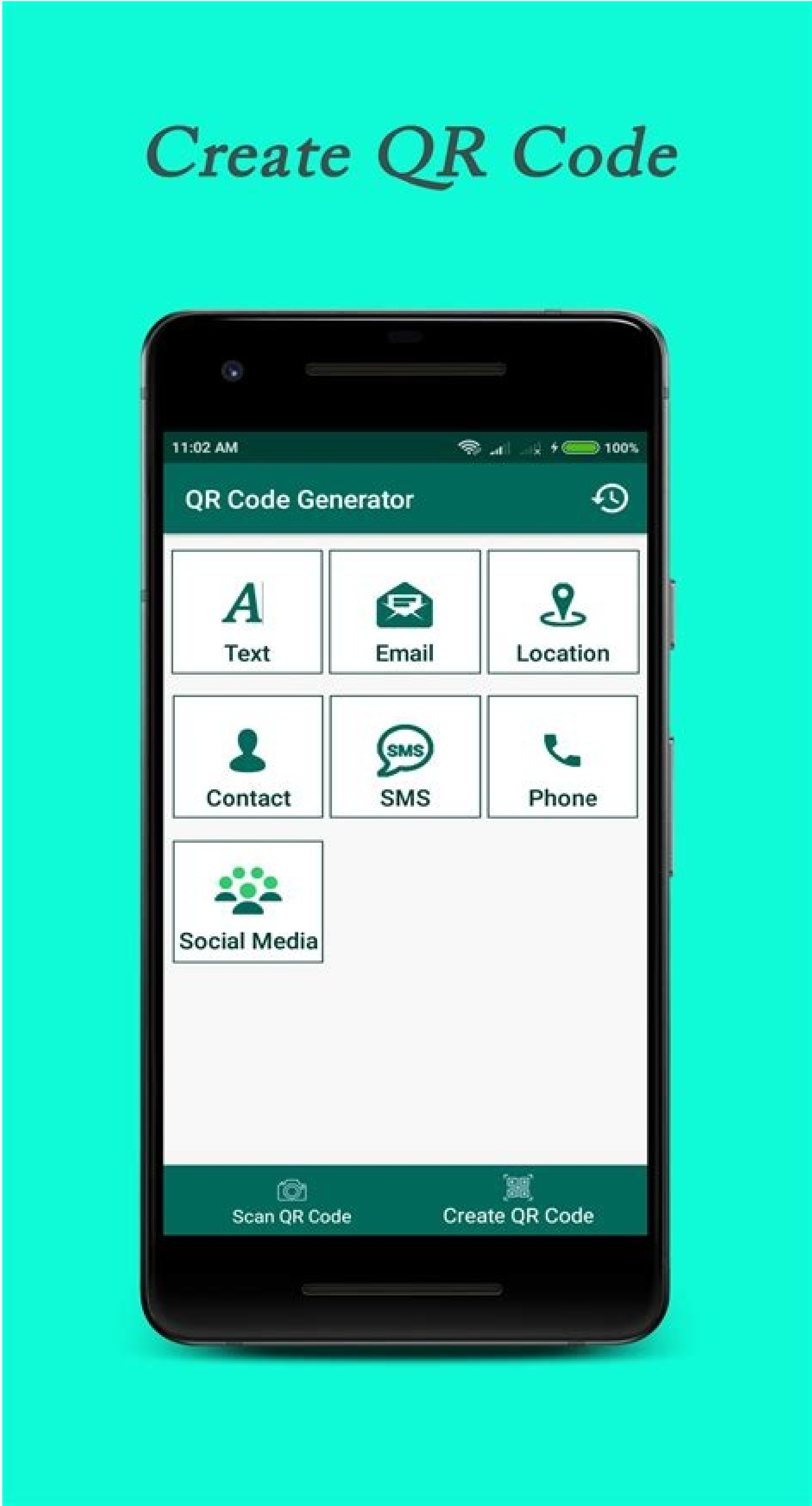
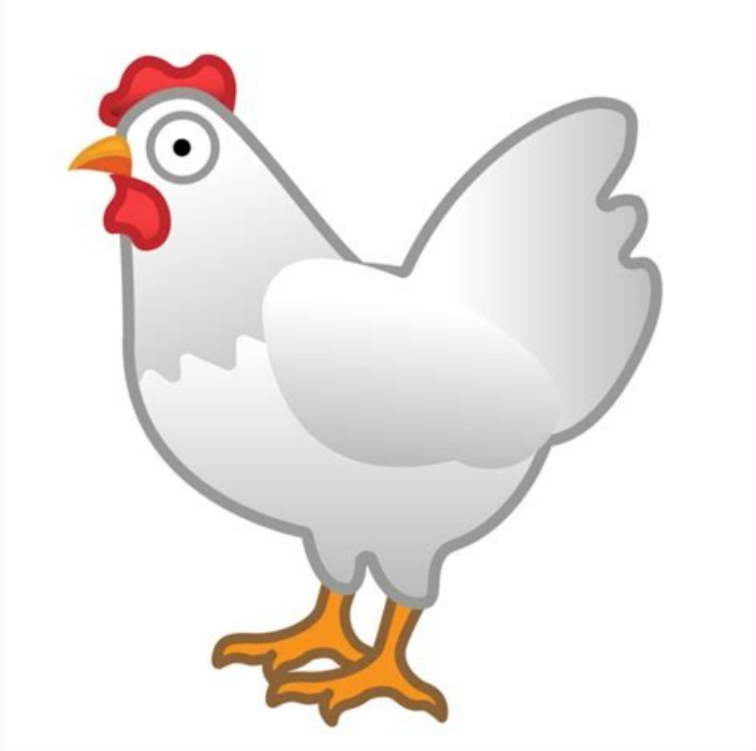


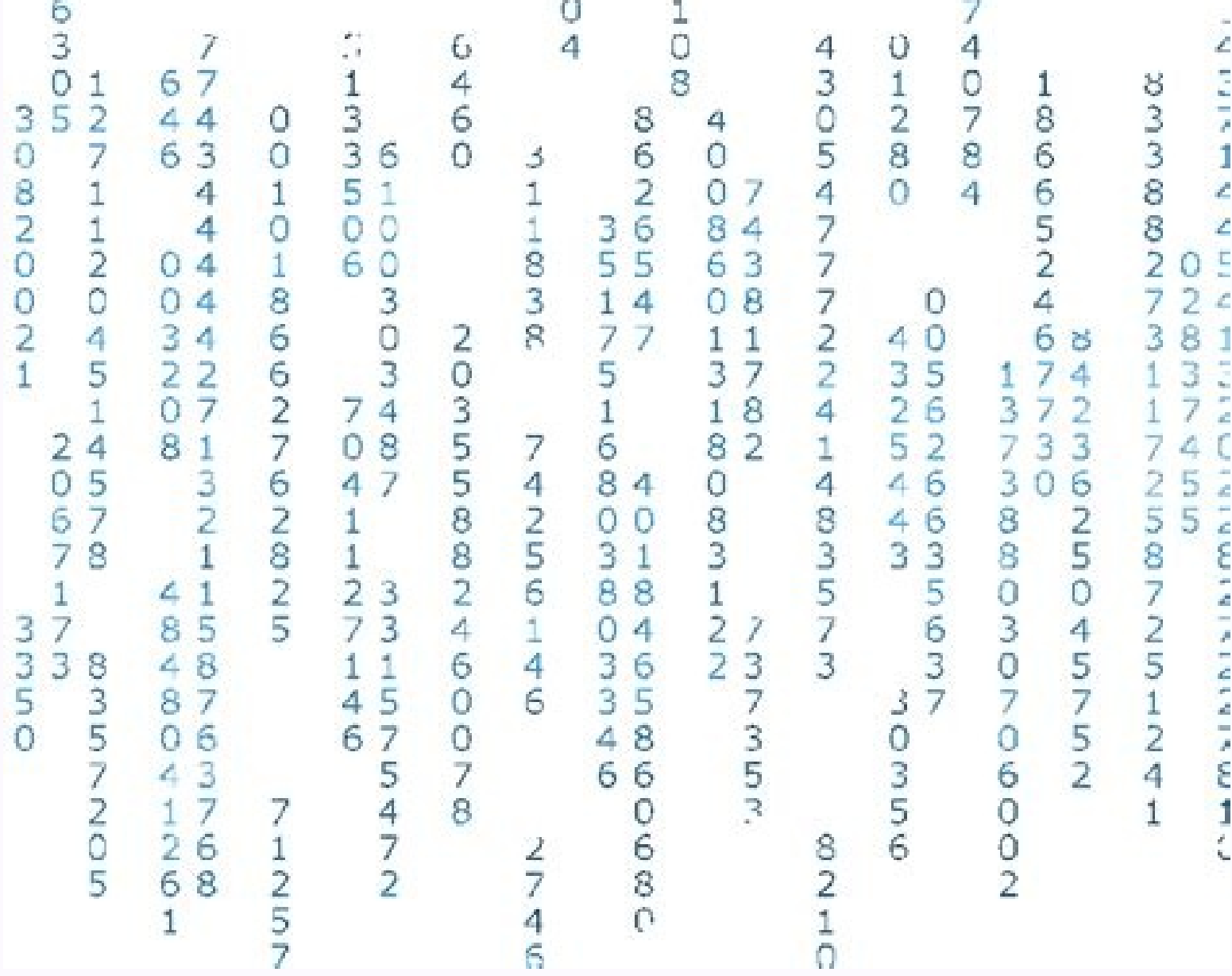
☐

I'm not robot


reCAPTCHA

Open





Android code to play video from url. Android code to download video from url.

Below is the code for the activity_main.xml file. Step 4: Working with the MainActivity.javaGo to the MainActivity.java file and refer to the following code. Note that we are going to implement this project using the Java language. Now let's see the step-by-step implementation of the above application shown in the gif.**Step By Step Implementation**Step 1: Create a new projectSo the first step is to create a new project. On the Welcome screen of Android Studio, click on Create New Project and If you have a project already opened, Go to File > New > New Project. In this article, you will see how to play a video from URL on Android. Then select a Project Template window, select Empty Activity and click Next. Comments are added inside the code to understand the code in more detail.`import android.net.Uri;import android.os.Bundle;import android.widget.MediaController;import android.widget.VideoView;import androidx.appcompat.app.AppCompatActivity;public class MainActivity extends AppCompatActivity { String videoUrl = " void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity_main); VideoView videoView = findViewById(R.id.videoView); Uri uri = Uri.parse(videoUrl); videoView.setVideoURI(uri); MediaController mediaController = new MediaController(this); mediaController.setAnchorView(videoView); mediaController.setMediaPlayer(videoView); videoView.start(); } }Output: Skip to contentHere you will learn how to play video from url in android using videoview.We can using VideoView widget to play offline video as well as video from url.In below example I have displayed a progress dialog until the video is buffering.Also Read: Picasso Android Tutorial - Load Image from URLCreate an android studio project with package name com.videoviewAdd internet access permission in AndroidManifest.xml file.Now add following code in respective files.activity_main.xml android:layout_width="match parent" android:layout_height="wrap content" android:id="@+id/video" />MainActivity.javapackage com.videoview; import android.app.AlertDialog; import android.media.MediaPlayer; import android.net.Uri; import android.support.v7.app.AppCompatActivity; import android.os.Bundle; import android.widget.VideoView; public class MainActivity extends AppCompatActivity { VideoView video; String video_url = " 20Days%20of%20Future%20Past%20Trailer%20-%20[Webmusic.IN].3gp"; ProgressDialog pd; @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity_main); video = (VideoView)findViewById(R.id.video); pd = new ProgressDialog(MainActivity.this); pd.setMessage("Buffering video please wait..."); pd.show(); Uri uri = Uri.parse(video_url); video.setVideoURI(uri); video.start(); video.setOnPreparedListener(new MediaPlayer.OnPreparedListener() { @Override public void onPrepared(MediaPlayer mp) { //close the progress dialog when buffering is done pd.dismiss(); } }); } }import android.app.AlertDialog;import android.media.MediaPlayer;import android.support.v7.app.AppCompatActivity;import android.os.Bundle;import android.widget.VideoView;public class MainActivity extends AppCompatActivity { String video_url = " 20Days%20of%20Future%20Past%20Trailer%20-%20[Webmusic.IN].3gp"; protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity_main); video = (VideoView)findViewById(R.id.video); pd = new ProgressDialog(MainActivity.this); pd.setMessage("Buffering video please wait..."); Uri uri = Uri.parse(video_url); video.setOnPreparedListener(new MediaPlayer.OnPreparedListener() { public void onPrepared(MediaPlayer mp) { //close the progress dialog when buffering is doneSave and run the project.Screenshotreport this ad Below is the code for the MainActivity.java file. A sample GIF is given below to get an idea about what we are going to do in this article. The VideoView widget is capable of playing media files, and the formats supported by the VideoView are 3gp and MP4. Enter your App Name in the Name field and select Java from the Language drop-down menu.Step 2: Add the Internet PermissionNavigate to app > manifest > AndroidManifest.xml and the internet permission to that file as shown below.Step 3: Working with the activity_main.xmlNavigate to the app > res > layout > activity_main.xml and add the below code to that file. For showing the video in our android application we will use the VideoView widget. By using VideoView you can play media files from the local storage and also from the internet.`