

# AndroidJSon

Advanced Android Development Tutorials for beginners

HOME ANDROID

## Android Capture Image From Camera Upload to Server Using PHP MySQL

 Admin  April 26, 2017  Android

How to Take a photo from camera and upload online and save image path into MySQL database with image name project with Runtime Camera permission.

In this tutorial we would going to create an android application which would capture image from android mobile phone camera and show that image inside ImageView. After capturing we would set image name into EditText and upload image online to our hosting server. While uploading we would also show ProgressBar dialog on app screen. The image would store on our server inside a folder created by myself and the whole image path save into

Search Here



Archives

June 2017

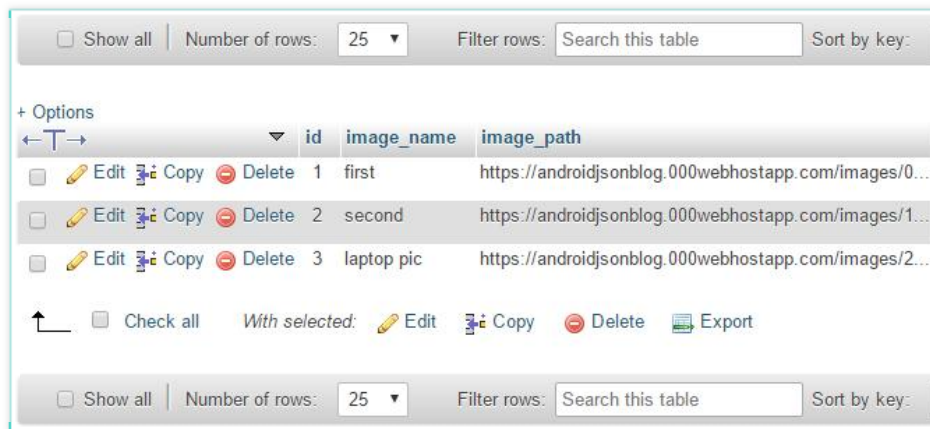
MySQL database along with image name. We are using PHP script to receive and store image on server. So here is the complete step by step tutorial for Android Capture Image From Camera Upload to Server Using PHP MySQL.

## Contents in this project Android Capture Image From Camera Upload to Server Using PHP MySQL :-

1. Create Database including table on your server.
2. Create folder named as **images** on your server.
3. Create PHP Script to receive send image from android phone.
4. Start a fresh android app development project.
5. Add internet and camera permission inside the project Manifest.xml file.
6. Start coding.

### 1. Create Database including table on your server :

Create a fresh database on your server and inside that database create a fresh table which will used to store image name and image path inside MySQL database like i did in below screenshot.



The screenshot shows a MySQL database interface with a table named 'images'. The table has three columns: 'id', 'image\_name', and 'image\_path'. There are three rows of data. Each row has action buttons for Edit, Copy, and Delete. The table is displayed with 25 rows per page, and the filter is set to 'Search this table'.

	id	image_name	image_path
<input type="checkbox"/> Edit Copy Delete	1	first	https://androidjsonblog.000webhostapp.com/images/0...
<input type="checkbox"/> Edit Copy Delete	2	second	https://androidjsonblog.000webhostapp.com/images/1...
<input type="checkbox"/> Edit Copy Delete	3	laptop pic	https://androidjsonblog.000webhostapp.com/images/2...

### 2. Create folder named as **images** on your server :

May 2017

April 2017

March 2017

February 2017

January 2017

December 2016

After creating table we have to create a folder on our hosting server which will used to store images.

### 3. Create PHP Script to receive send image from android phone :

Now we would have to create 2 php files first one is capture\_img\_upload\_to\_server.php file and DatabaseConfig.php file and upload that file into our server. These file will used to receive and store image details, image itself, image name on MySQL database.

**Code for capture\_img\_upload\_to\_server.php file.**

```
<?php

include 'DatabaseConfig.php';

// Create connection
$conn = new mysqli($HostName, $HostUser, $HostPass, $DatabaseName);

if($_SERVER['REQUEST_METHOD'] == 'POST')
{
    $DefaultId = 0;

    $ImageData = $_POST['image_path'];

    $ImageName = $_POST['image_name'];

    $GetOldIdSQL = "SELECT id FROM ImageToServerTable ORDER BY id ASC";

    $Query = mysqli_query($conn,$GetOldIdSQL);

    while($row = mysqli_fetch_array($Query)){

        $DefaultId = $row['id'];
    }

    $ImagePath = "images/$DefaultId.png";
```

```
$ServerURL = "https://androidjsonblog.000webhostapp.com/$ImagePath";

$insertSQL = "insert into ImageToServerTable (image_path,image_name) values ('$ServerURI

if(mysqli_query($conn, $insertSQL)){

file_put_contents($ImagePath,base64_decode($ImageData));

echo "Your Image Has Been Uploaded.";
}

mysqli_close($conn);
}else{
echo "Not Uploaded";
}

?>
```

**Code for DatabaseConfig.php file.**

```
<?php

//Define your host here.
$HostName = "localhost";

//Define your database username here.
$HostUser = "id632449_androidjson";

//Define your database password here.
$HostPass = "kdfjdfdskljomew9ry3873";

//Define your database name here.
$DatabaseName = "id632449_androidjson";

?>
```

**5. Add internet and camera permission inside the project Manifest.xml file :**

Open your project's AndroidManifest.xml file and add camera permission and internet permission inside it.

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

## 6. Start coding For Project :-

**Code for MainActivity.java file.**

```
package com.androidjson.captureimageupload_androidjsoncom;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;
import java.net.HttpURLConnection;
import android.graphics.Bitmap;
import android.os.Bundle;
import android.view.View;
import android.app.AlertDialog;
import android.os.AsyncTask;
import android.widget.EditText;
import android.net.Uri;
import java.io.InputStreamReader;
import java.io.OutputStream;
import javax.net.ssl.HttpURLConnection;
import java.io.BufferedWriter;
import java.util.Map;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.util.HashMap;
import java.io.OutputStreamWriter;
```

```
import java.net.URL;
import android.provider.MediaStore;
import java.io.BufferedReader;
import java.net.URLEncoder;
import java.io.UnsupportedEncodingException;
import android.util.Base64;

public class MainActivity extends AppCompatActivity {

    Button CaptureImageFromCamera, UploadImageToServer;

    ImageView ImageViewHolder;

    EditText imageName;

    ProgressDialog progressDialog ;

    Intent intent ;

    public static final int RequestPermissionCode = 1 ;

    Bitmap bitmap;

    boolean check = true;

    String GetImageNameFromEditText;

    String ImageNameFieldOnServer = "image_name" ;

    String ImagePathFieldOnServer = "image_path" ;

    String ImageUploadPathOnSever = "https://androidjsonblog.000webhostapp.com/capture\_image.php" ;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

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

```
        ImageViewHolder = (ImageView)findViewById(R.id.imageView);
        UploadImageToServer = (Button) findViewById(R.id.button2);
        imageName = (EditText)findViewById(R.id.editText);

        EnableRuntimePermissionToAccessCamera();

        CaptureImageFromCamera.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                intent = new Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);

                startActivityForResult(intent, 7);

            }
        });

        UploadImageToServer.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                GetImageNameFromEditText = imageName.getText().toString();

                ImageUploadToServerFunction();

            }
        });
    }

    // Star activity for result method to Set captured image on image view after click.
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {

        super.onActivityResult(requestCode, resultCode, data);

        if (requestCode == 7 && resultCode == RESULT_OK && data != null && data.getData() != null) {

            Uri uri = data.getData();

            try {
```

```
        // Adding captured image in bitmap.
        bitmap = MediaStore.Images.Media.getBitmap(getContentResolver(), uri);

        // adding captured image in imageview.
        ImageViewHolder.setImageBitmap(bitmap);

    } catch (IOException e) {

        e.printStackTrace();
    }
}

// Requesting runtime permission to access camera.
public void EnableRuntimePermissionToAccessCamera(){

    if (ActivityCompat.shouldShowRequestPermissionRationale(MainActivity.this,
        Manifest.permission.CAMERA))
    {

        // Printing toast message after enabling runtime permission.
        Toast.makeText(MainActivity.this, "CAMERA permission allows us to Access CAMERA app",
            Toast.LENGTH_SHORT).show();

    } else {

        ActivityCompat.requestPermissions(MainActivity.this, new String[]{Manifest.permission.CAMERA},
            REQUEST_CAMERA_PERMISSION);

    }
}

// Upload captured image online on server function.
public void ImageUploadToServerFunction(){

    ByteArrayOutputStream byteArrayOutputStreamObject ;

    byteArrayOutputStreamObject = new ByteArrayOutputStream();

    // Converting bitmap image to jpeg format, so by default image will upload in jpeg format.
    bitmap.compress(Bitmap.CompressFormat.JPEG, 100, byteArrayOutputStreamObject);
```



```
byte[] byteArrayVar = byteArrayOutputStreamObject.toByteArray();

final String ConvertImage = Base64.encodeToString(byteArrayVar, Base64.DEFAULT);

class AsyncTaskUploadClass extends AsyncTask<Void,Void,String> {

    @Override
    protected void onPreExecute() {

        super.onPreExecute();

        // Showing progress dialog at image upload time.
        progressDialog = ProgressDialog.show(MainActivity.this,"Image is Uploading",null,true,false);
    }

    @Override
    protected void onPostExecute(String string1) {

        super.onPostExecute(string1);

        // Dismiss the progress dialog after done uploading.
        progressDialog.dismiss();

        // Printing uploading success message coming from server on android app.
        Toast.makeText(MainActivity.this,string1,Toast.LENGTH_LONG).show();

        // Setting image as transparent after done uploading.
        ImageViewHolder.setImageResource(android.R.color.transparent);

    }

    @Override
    protected String doInBackground(Void... params) {

        ImageProcessClass imageProcessClass = new ImageProcessClass();

        HashMap<String,String> HashMapParams = new HashMap<String,String>();
```

```
        HashMapParams.put(ImageNameFieldOnServer, GetImageNameFromEditText);

        HashMapParams.put(ImagePathFieldOnServer, ConvertImage);

        String FinalData = imageProcessClass.ImageHttpRequest(ImageUploadPathOnSe

        return FinalData;
    }
}
AsyncTaskUploadClass AsyncTaskUploadClassOBJ = new AsyncTaskUploadClass();

AsyncTaskUploadClassOBJ.execute();
}

public class ImageProcessClass{

    public String ImageHttpRequest(String requestURL,HashMap<String, String> PData) {

        StringBuilder stringBuilder = new StringBuilder();

        try {

            URL url;
            HttpURLConnection httpURLConnectionObject ;
            OutputStream OutPutStream;
            BufferedWriter bufferedWriterObject ;
            BufferedReader bufferedReaderObject ;
            int RC ;

            url = new URL(requestURL);

            httpURLConnectionObject = (HttpURLConnection) url.openConnection();

            httpURLConnectionObject.setReadTimeout(19000);

            httpURLConnectionObject.setConnectTimeout(19000);

            httpURLConnectionObject.setRequestMethod("POST");

            httpURLConnectionObject.setDoInput(true);
```

```
        httpURLConnectionObject.setDoOutput(true);

        OutPutStream = httpURLConnectionObject.getOutputStream();

        bufferedWriterObject = new BufferedWriter(

            new OutputStreamWriter(OutPutStream, "UTF-8"));

        bufferedWriterObject.write(bufferedWriterDataFN(PData));

        bufferedWriterObject.flush();

        bufferedWriterObject.close();

        OutPutStream.close();

        RC = httpURLConnectionObject.getResponseCode();

        if (RC == HttpsURLConnection.HTTP_OK) {

            bufferedReaderObject = new BufferedReader(new InputStreamReader(httpURLConnectionObject.getInputStream()));

            stringBuilder = new StringBuilder();

            String RC2;

            while ((RC2 = bufferedReaderObject.readLine()) != null){

                stringBuilder.append(RC2);

            }

        } catch (Exception e) {

            e.printStackTrace();

        }

        return stringBuilder.toString();

    }

    private String bufferedWriterDataFN(HashMap<String, String> HashMapParams) throws
```

```
        StringBuilder stringBuilderObject;

        stringBuilderObject = new StringBuilder();

        for (Map.Entry<String, String> KEY : HashMapParams.entrySet()) {

            if (check)

                check = false;
            else
                stringBuilderObject.append("&");

            stringBuilderObject.append(URLEncoder.encode(KEY.getKey(), "UTF-8"));

            stringBuilderObject.append("=");

            stringBuilderObject.append(URLEncoder.encode(KEY.getValue(), "UTF-8"));
        }

        return stringBuilderObject.toString();
    }

}

@Override
public void onRequestPermissionsResult(int RC, String per[], int[] PResult) {

    switch (RC) {

        case RequestPermissionCode:

            if (PResult.length > 0 && PResult[0] == PackageManager.PERMISSION_GRANTED)

                Toast.makeText(MainActivity.this, "Permission Granted, Now your applica

            } else {

                Toast.makeText(MainActivity.this, "Permission Canceled, Now your applica
```

```
        }  
        break;  
    }  
}
```

**Code for activity\_main.xml layout file.**

```
<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:context="com.androidjson.captureimageupload_androidjsoncom.MainActivity">  
  
    <ImageView  
        android:layout_width="fill_parent"  
        android:layout_height="270dp"  
        android:id="@+id/imageView"  
        android:layout_alignParentTop="true"  
        android:layout_alignParentLeft="true"  
        android:layout_alignParentStart="true" />
```

```
<Button
    android:text="Click here to capture image using camera"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/button"
    android:layout_below="@+id/imageView" />

<EditText
    android:id="@+id/editText"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/button"
    android:layout_centerHorizontal="true"
    android:ems="10"
    android:inputType="textPersonName"
    android:hint="Enter Image Name"
    android:gravity="center"
    android:layout_marginTop="10dp" />

<Button
    android:id="@+id/button2"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Upload Captured Image On Server"
    android:layout_below="@+id/editText"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

</RelativeLayout>
```

**Code for AndroidManifest.xml file.**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.androidjson.captureimageupload_androidjsoncom">

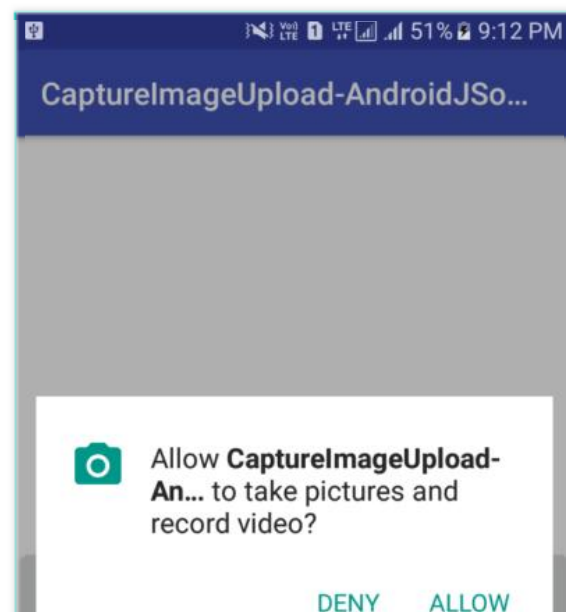
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.INTERNET" />
```

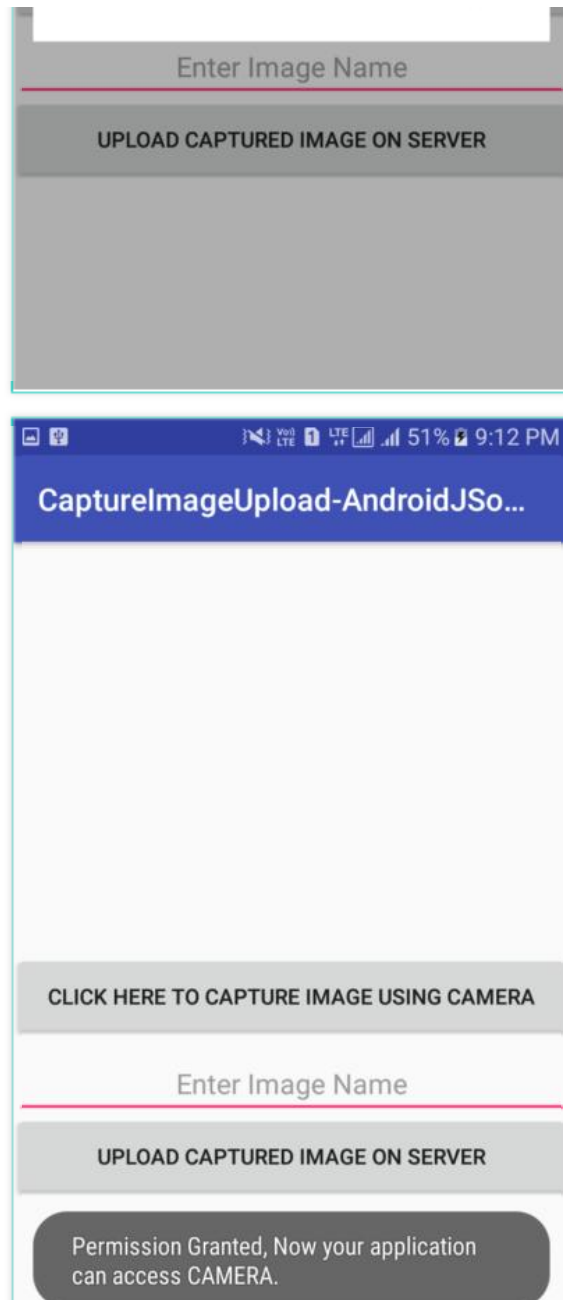
```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>

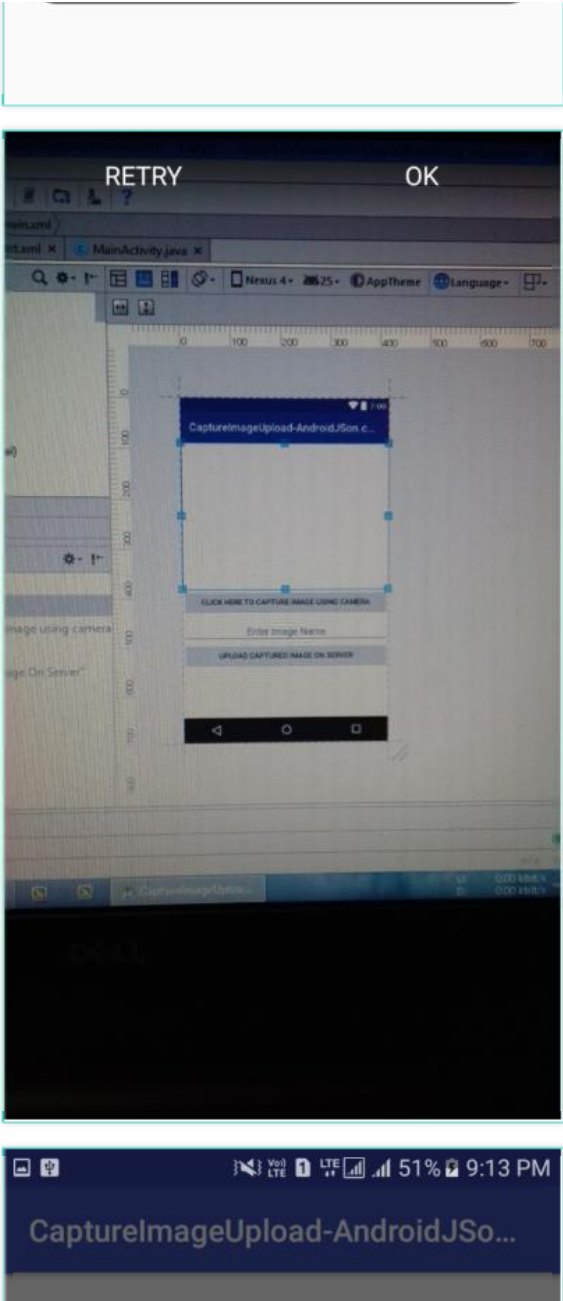
</manifest>
```

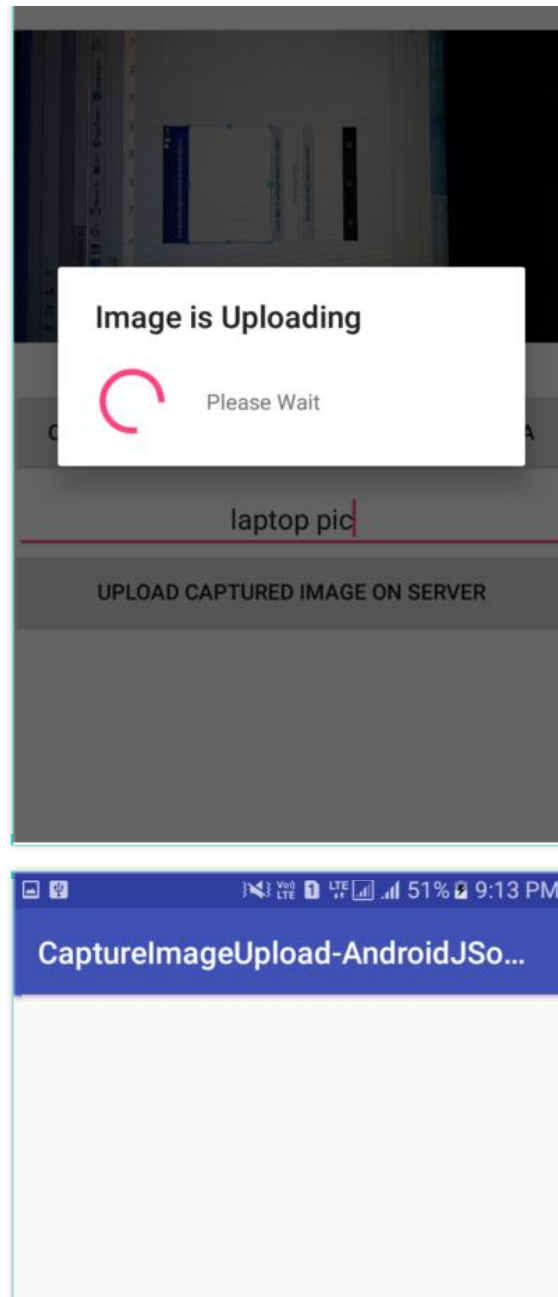
#### Screenshots:-

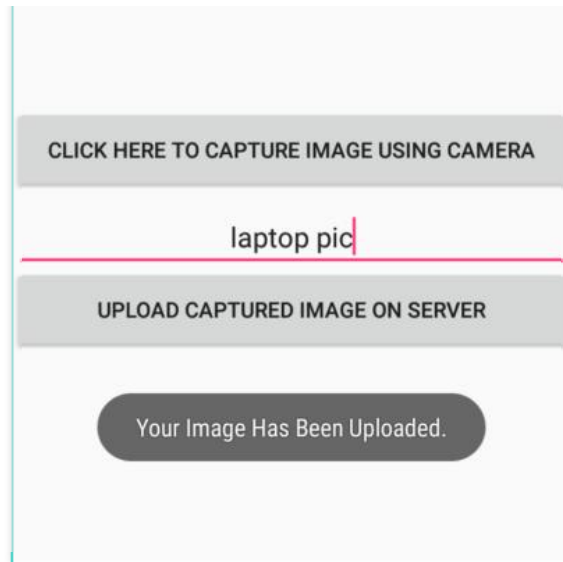












[Download Code](#)

## Related Posts

- [Android Upload Image To Server Using PHP MySQL](#)
- [Android JSon Parsing ListView Items Retrieve Via PHP MySQL Database](#)
- [Add Click Filter on JSon Data in Android App Using PHP MySQL](#)