**Login Form:**

```
import java.util.ArrayList;
import java.util.List;
import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONException;
import org.json.JSONObject;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import com.example.virtualclassroomproject.R;

public class StudentLogin extends Activity {
	public static final String STUDENT_PREFERENCE="student";
	private ProgressDialog pDialog;
	JSONParser jsonParser = new JSONParser();
	EditText username;
	EditText password;
	private static String url_checkValidStudent =
"http://10.0.2.2/VirtualClassroom/check_valid_student.php";
	private static final String TAG_SUCCESS = "success";

	@Override
	public void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
		setContentView(R.layout.student_login);
		username = (EditText) findViewById(R.id.studentLoginUsername);
		password = (EditText) findViewById(R.id.studentLoginPassword);

		Button btnNewStudentRegistration = (Button)
findViewById(R.id.btnLinkToNewStudentRegistration);
		btnNewStudentRegistration.setOnClickListener(new View.OnClickListener() {

			@Override
			public void onClick(View v) {
				Intent i = new
Intent(getApplicationContext(),StudentNewRegistration.class);
```

```
                                        startActivity(i);
                                }
                        });

                Button btnCheckStudent = (Button) findViewById(R.id.btnStudentLogin);
                btnCheckStudent.setOnClickListener(new View.OnClickListener() {
                        @Override
                        public void onClick(View v) {
                                new CheckValidStudent().execute();
                        }
                });

                Button
btnStudentForgetPassword=(Button)findViewById(R.id.btnLinkToStudentForgetPassword);
                btnStudentForgetPassword.setOnClickListener(new View.OnClickListener() {

                        @Override
                        public void onClick(View v) {
                                Intent i = new
Intent(getApplicationContext(),StudentForgetPassword.class);
                                startActivity(i);

                        }
                });
        }

        @Override
        protected void onActivityResult(int requestCode, int resultCode, Intent data) {
                super.onActivityResult(requestCode, resultCode, data);
                if (resultCode == 100) {
                        Intent intent = getIntent();
                        finish();
                        startActivity(intent);
                }
        }

        class CheckValidStudent extends AsyncTask<String, String, String> {
                @Override
                protected void onPreExecute() {
                        super.onPreExecute();
                        pDialog = new ProgressDialog(StudentLogin.this);
                        pDialog.setMessage("Checking credentials. Please wait...");
                        pDialog.setIndeterminate(false);
                        pDialog.setCancelable(false);
                        pDialog.show();
                }
```

```
		protected String doInBackground(String... args) {
			String adminUsername = username.getText().toString();
			String adminPassword = password.getText().toString();
			List<NameValuePair> params = new ArrayList<NameValuePair>();
			params.add(new BasicNameValuePair("username", adminUsername));
			params.add(new BasicNameValuePair("password", adminPassword));
			JSONObject json = jsonParser.makeHttpRequest(url_checkValidStudent,
	"POST", params);
			Log.d("Create Response", json.toString());

			try {
				int success = json.getInt(TAG_SUCCESS);
				if (success == 1) {

			Intent i = new
Intent(getApplicationContext(),StudentRegistrationSuccessful.class);

			SharedPreferences settings=
getSharedPreferences(STUDENT_PREFERENCE, 0);
	SharedPreferences.Editor editor= settings.edit();

	String editedStandard= username.getText().toString();
	String standard= editedStandard.substring(3, 5);

	String editedBranch= username.getText().toString();
	String branch= editedBranch.substring(5, 8);

	editor.putString("standard", standard);
	editor.putString("branch", branch);
	editor.putString("username", username.getText().toString());
	editor.commit();
			startActivity(i);
			finish();
			}
			else {
					runOnUiThread(new Runnable() {
						public void run() {
							Toast.makeText(getApplicationContext(),
									"Invalid Student. Please
check details",Toast.LENGTH_LONG).show();
						}
					});

				}
			} catch (JSONException e) {
				e.printStackTrace();
```

```
                }
                return null;
            }
            protected void onPostExecute(String file_url) {
                pDialog.dismiss();

            }
        }
}
```

**Registration Form:**

```
public class StudentNewRegistration extends Activity {

        private ProgressDialog pDialog;
        JSONParser jsonParser = new JSONParser();
        EditText firstName;
        EditText middleName;
        EditText lastName;
        EditText standard;
        EditText branch;
        EditText rollNumber;
        EditText password;
        EditText confirmPassword;
        EditText securityQuestion;
        EditText securityQuestionAnswer;
        private static String url_create_student =
"http://10.0.2.2/virtualClassroom/create_student.php";
        private static final String TAG_SUCCESS = "success";

        @Override
        public void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.student_registration);

            firstName = (EditText) findViewById(R.id.studentFirstName);
            middleName = (EditText) findViewById(R.id.studentMiddleName);
            lastName = (EditText) findViewById(R.id.studentLastName);
            standard = (EditText) findViewById(R.id.studentStandard);
            branch = (EditText) findViewById(R.id.studentBranch);
            rollNumber = (EditText) findViewById(R.id.studentRollNumber);
            password = (EditText) findViewById(R.id.studentPassword);
            confirmPassword = (EditText) findViewById(R.id.StudentConfirmPassword);
            securityQuestion = (EditText) findViewById(R.id.studentSecurityQuestion);
            securityQuestionAnswer = (EditText) findViewById(R.id.studentAnswer);
```

```
		Button btnCreateNewStudent = (Button) findViewById(R.id.btnStudentRegister);
		btnCreateNewStudent.setOnClickListener(new View.OnClickListener() {
			@Override
			public void onClick(View view) {
				new CreateNewStudent().execute();
			}
		});
	}

	class CreateNewStudent extends AsyncTask<String, String, String> {
		@Override
		protected void onPreExecute() {
			super.onPreExecute();
			pDialog = new ProgressDialog(StudentNewRegistration.this);
			pDialog.setMessage("Registering Student..");
			pDialog.setIndeterminate(false);
			pDialog.setCancelable(true);
			pDialog.show();
		}

		protected String doInBackground(String... args) {
			String studentFirstName = firstName.getText().toString();
			String studentMiddleName = middleName.getText().toString();
			String studentLastName = lastName.getText().toString();
			String StudentStandard = standard.getText().toString();
			String studentBranch = branch.getText().toString();
			String studentRollNumber = rollNumber.getText().toString();
			String studentPassword = password.getText().toString();
			String studentConfirmPassword = confirmPassword.getText().toString();
			String studentSecurityQuestion = securityQuestion.getText().toString();
			String studentSecurityQuestionAnswer =
securityQuestionAnswer.getText().toString();

			final String username = (studentFirstName.substring(0, 1)+
studentMiddleName.substring(0, 1)+ studentLastName.substring(0, 1) + StudentStandard
			+ studentBranch + studentRollNumber).toUpperCase();

			List<NameValuePair> params = new ArrayList<NameValuePair>();
			params.add(new BasicNameValuePair("firstName", studentFirstName));
			params.add(new BasicNameValuePair("middleName",
studentMiddleName));
			params.add(new BasicNameValuePair("lastName", studentLastName));
			params.add(new BasicNameValuePair("standard", StudentStandard));
			params.add(new BasicNameValuePair("branch", studentBranch));
			params.add(new BasicNameValuePair("rollNumber",
studentRollNumber));
```

```
                params.add(new BasicNameValuePair("password", studentPassword));
                params.add(new
BasicNameValuePair("confirmPassword",studentConfirmPassword));
                params.add(new
BasicNameValuePair("securityQuestion",studentSecurityQuestion));
                params.add(new BasicNameValuePair("securityQuestionAnswer",
        studentSecurityQuestionAnswer));
                params.add(new BasicNameValuePair("username", username));
                params.add(new BasicNameValuePair("status", "Disabled"));

                JSONObject json =
jsonParser.makeHttpRequest(url_create_student,"POST", params);
                Log.d("Create Response", json.toString());
                try {
                        int success = json.getInt(TAG_SUCCESS);
                        if (success == 1) {

                                runOnUiThread(new Runnable() {
                                        public void run() {
                                                Toast.makeText(getApplicationContext(),
                                                        "Student Registration is done
Successfully", Toast.LENGTH_LONG).show();
                                                Toast.makeText(getApplicationContext(),
                                                        "Your Username
is:"+username.toString(), Toast.LENGTH_LONG).show();
                                                Intent i = new
Intent(getApplicationContext(),StudentLogin.class);
                                                startActivity(i);
                                                finish();
                                        }
                                });
                        } else {
                                runOnUiThread(new Runnable() {
                                        public void run() {
                                                Toast.makeText(getApplicationContext(),
                                                        "Erro to Register Student",
Toast.LENGTH_LONG).show();
                                        }
                                        });
                        }
                } catch (JSONException e) {
                        e.printStackTrace();
                }
                return null;
        }
        protected void onPostExecute(String file_url) {
```

```
                        pDialog.dismiss();
                }
        }
}
```

**Upload Video:**

```
public class UploadVideo extends Activity implements OnClickListener{
        public static final String VIDEO_PREFERENCE="video";
        public static final String VIDEO_PREFERENCE1="video1";
        String fileName=null;
        private ProgressDialog pDialog;
        JSONParser jsonParser = new JSONParser();
        private TextView messageText;
        private Button uploadButton, btnselectvideo;
        private ImageView imageview;
        private int serverResponseCode = 0;
        private ProgressDialog dialog = null;
        private String upLoadServerUri = null;
        private String filepath = null;
        int FLAG = 0;
private static final String TAG_SUCCESS = "success";
        //uploading details
        String serverLocation="http://10.0.2.2/";
        String pathSeperator="/";
        String folder="VirtualClassroom";
        private static String url_insert_videodetails =
"http://10.0.2.2/virtualClassroom/create_video_entry.php";
        EditText txtstandard;
        EditText txtsubject;
        String script="upload_to_server.php";
        String location="VirtualClassroom";
        String branch="CSE";

@Override
        public void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.uploadvideo);
                txtstandard=(EditText)findViewById(R.id.videouploadStandard);
                txtsubject=(EditText)findViewById(R.id.videouploadSubject);
                uploadButton = (Button) findViewById(R.id.uploadButton);
                messageText = (TextView) findViewById(R.id.messageText);
                btnselectvideo = (Button) findViewById(R.id.button_selectvideo);
                imageview = (ImageView) findViewById(R.id.imageView_pic);
                btnselectvideo.setOnClickListener(this);
                uploadButton.setOnClickListener(this);
```

```
}
	@Override
	public void onClick(View arg0) {
		if (arg0 == btnselectvideo) {
			FLAG = 1;
			Intent intent = new Intent();
			intent.setType("video/*");
			intent.setAction(Intent.ACTION_GET_CONTENT);
			startActivityForResult(
					Intent.createChooser(intent, "Complete action using"), 1);
			if (btnselectvideo.isPressed()) {
				Drawable bitmap =
getResources().getDrawable(R.drawable.video);
				imageview.setImageDrawable(bitmap);
			}
		} else if (arg0 == uploadButton) {
			if (FLAG == 0) {
				Toast.makeText(UploadVideo.this,"Please select video
!!!",Toast.LENGTH_LONG).show();
			} else {
				if (filepath != null) {
					dialog = ProgressDialog.show(UploadVideo.this,
"","Uploading file...", true);
					messageText.setText("uploading started.....");
					new Thread(new Runnable() {
						public void run() {
							uploadFile(filepath);
						}
					}).start();
				} else {
					Toast.makeText(UploadVideo.this, "Please try again !!!",
							Toast.LENGTH_LONG).show();
				}
			}
		}
	}
	@Override
	protected void onActivityResult(int requestCode, int resultCode, Intent data) {
		if (requestCode == 1 && resultCode == RESULT_OK) {
			Uri selectedImageUri = data.getData();
			filepath = getPath(selectedImageUri);
			Bitmap bitmap = BitmapFactory.decodeFile(filepath);
			String videoName=filepath.substring(12);
			 SharedPreferences settings=
getSharedPreferences(VIDEO_PREFERENCE1, 0);
	 SharedPreferences.Editor editor= settings.edit();
```

```
        editor.putString("videoName", videoName);
        editor.commit();
                        imageview.setImageBitmap(bitmap);
                        messageText.setText("Uploading file Name:" + videoName);
                }
        }
        class uploadFile extends AsyncTask<String, String, String>
        {
                @Override
              protected void onPreExecute() {
                super.onPreExecute();
                pDialog = new ProgressDialog(UploadVideo.this);
                pDialog.setMessage("Creating ..");
                pDialog.setIndeterminate(false);
                pDialog.setCancelable(true);
                pDialog.show();
              }
                 protected String doInBackground(String... args) {
                String videouploadstandard = txtstandard.getText().toString();
                String videouploadsubject = txtsubject.getText().toString();
                SharedPreferences setting2 = getSharedPreferences(VIDEO_PREFERENCE1, 0);
                        String video= setting2.getString("videoName", "Wrong");
        // Building Parameters
                List<NameValuePair> params = new ArrayList<NameValuePair>();
                params.add(new BasicNameValuePair("standard", videouploadstandard));
                params.add(new BasicNameValuePair("subject", videouploadsubject));
                params.add(new BasicNameValuePair("videoName", video));
                // getting JSON Object
                JSONObject json = jsonParser.makeHttpRequest(url_insert_videodetails,
"POST", params);
                // check log cat fro response
                Log.d("Create Response", json.toString());
                // check for success tag
                try {
                   int success = json.getInt(TAG_SUCCESS);
                   if (success == 1) {
                        runOnUiThread(new Runnable() {
                                                public void run() {
                        Toast.makeText(getApplicationContext(),"video uploading is done",
Toast.LENGTH_LONG).show();
                                                }
                                        })
                   } else {
                        runOnUiThread(new Runnable() {
                                                public void run() {
```

```
                        Toast.makeText(getApplicationContext(),     "Error to upload video",
Toast.LENGTH_LONG).show();
                                                    }
                                                });
                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }
                return null;
            }
protected void onPostExecute(String file_url) {
                // dismiss the dialog once done
                pDialog.dismiss();
            }
        }
        @SuppressWarnings("deprecation")
        public String getPath(Uri uri) {
                String[] projection = { MediaStore.Images.Media.DATA };
                Cursor cursor = managedQuery(uri, projection, null, null, null);
                int column_index = cursor
                                .getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
                cursor.moveToFirst();
                return cursor.getString(column_index);
        }
        public int uploadFile(final String sourceFileUri) {
                //String fileName = sourceFileUri;
                fileName=sourceFileUri;//this is added new
                HttpURLConnection conn = null;
                DataOutputStream dos = null;
                String lineEnd = "\r\n";
                String twoHyphens = "--";
                String boundary = "*****";
                int bytesRead, bytesAvailable, bufferSize;
                byte[] buffer;
                int maxBufferSize = 1 * 1024;
                File sourceFile = new File(sourceFileUri);
                if (!sourceFile.isFile()) {
                        dialog.dismiss();
                        Log.e("uploadFile", "Source File not exist :" + filepath);
                        runOnUiThread(new Runnable() {
                                public void run() {
                                        messageText.setText("Source File not exist :" + filepath);
                                }
                        });
                        return 0;
                } else {
```

```
                        try {
                                FileInputStream fileInputStream = new FileInputStream(
                                        sourceFile);
                                String standard=txtstandard.getText().toString();
                                String subject=txtsubject.getText().toString();
                                 SharedPreferences settings=
getSharedPreferences(VIDEO_PREFERENCE, 0);
            SharedPreferences.Editor editor= settings.edit();
            editor.putString("standard", standard);
            editor.putString("subject", subject);
            editor.commit();
          upLoadServerUri=serverLocation+location+pathSeperator+standard+pathSeperator+bran
ch+pathSeperator+subject+pathSeperator+script;
                                URL url = new URL(upLoadServerUri);
                                conn = (HttpURLConnection) url.openConnection();
                                conn.setDoInput(true); // Allow Inputs
                                conn.setDoOutput(true); // Allow Outputs
                                conn.setUseCaches(false); // Don't use a Cached Copy
                                conn.setChunkedStreamingMode(1024);//i have added extra
                                conn.setRequestMethod("POST");
                                conn.setRequestProperty("Connection", "Keep-Alive");
                                conn.setRequestProperty("ENCTYPE", "multipart/form-data");
                                conn.setRequestProperty("Content-Type",
                                        "multipart/form-data;boundary=" + boundary);
                                conn.setRequestProperty("uploaded_file", fileName);
                                dos = new DataOutputStream(conn.getOutputStream());
                                dos.writeBytes(twoHyphens + boundary + lineEnd);
                                dos.writeBytes("Content-Disposition: form-data;
name=\"uploaded_file\";filename=\""+ fileName + "\"" + lineEnd);
                                dos.writeBytes(lineEnd);
                                bytesAvailable = fileInputStream.available();
                                bufferSize = Math.min(bytesAvailable, maxBufferSize);
                                buffer = new byte[bufferSize];
                                bytesRead = fileInputStream.read(buffer, 0, bufferSize);
                                while (bytesRead > 0) {
                                        dos.write(buffer, 0, bufferSize);
                                        bytesAvailable = fileInputStream.available();
                                        bufferSize = Math.min(bytesAvailable, maxBufferSize);
                                        bytesRead = fileInputStream.read(buffer, 0, bufferSize);
                                }
                                dos.writeBytes(lineEnd);
                                dos.writeBytes(twoHyphens + boundary + twoHyphens +
lineEnd);
                                serverResponseCode = conn.getResponseCode();
                                String serverResponseMessage = conn.getResponseMessage();
```

```
                                Log.i("uploadFile", "HTTP Response is : "+
serverResponseMessage + ": " + serverResponseCode);
                                if (serverResponseCode == 200) {
                                        runOnUiThread(new Runnable() {
                                                public void run() {
                                                        new uploadFile().execute();
        Toast.makeText(UploadVideo.this,"File Upload Complete.",
Toast.LENGTH_SHORT).show();
                                                        finish();
                                                }
                                        });
                                }
                                fileInputStream.close();
                                dos.flush();
                                dos.close();
                        } catch (MalformedURLException ex) {
                                dialog.dismiss();
                                ex.printStackTrace();
                                runOnUiThread(new Runnable() {
                                        public void run() {
        messageText.setText("MalformedURLException Exception : check script url.");
        Toast.makeText(UploadVideo.this,"MalformedURLException",
Toast.LENGTH_SHORT).show();
                                        }
                                });
                                Log.e("Upload file to server", "error: " + ex.getMessage(), ex);
                        } catch (Exception e) {
                                dialog.dismiss();
                                e.printStackTrace();
                                runOnUiThread(new Runnable() {
                                        public void run() {
                                                messageText.setText("Got Exception : see logcat ");
                                                Toast.makeText(UploadVideo.this,
                                                                "Got Exception : see logcat ",
                                                                Toast.LENGTH_SHORT).show();
                                        }
                                });
                                Log.e("Upload file to server Exception",
                                                "Exception : " + e.getMessage(), e);
                        }
                        dialog.dismiss();
                        return serverResponseCode;
                }
        }
}
```

**Record Video:**

```
public class RecordVideo extends Activity {
        // Activity request codes
   private static final int CAMERA_CAPTURE_IMAGE_REQUEST_CODE = 100;
   private static final int CAMERA_CAPTURE_VIDEO_REQUEST_CODE = 200;
   public static final int MEDIA_TYPE_IMAGE = 1;
   public static final int MEDIA_TYPE_VIDEO = 2;
   // directory name to store captured images and videos
   private static final String IMAGE_DIRECTORY_NAME = "";
   private Uri fileUri; // file url to store image/video
   //private ImageView imgPreview;
   private VideoView videoPreview;
   private Button btnRecordVideo,btnUploadRecordVideo;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.record_video);
      videoPreview = (VideoView) findViewById(R.id.videoPreview);
      btnRecordVideo = (Button) findViewById(R.id.btnRecordVideo);
      btnUploadRecordVideo =  (Button) findViewById(R.id.btnUploadRecordVideo);
      btnUploadRecordVideo.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            // capture picture
                  Intent i = new Intent(getApplicationContext(),UploadVideo.class);
                              startActivity(i);
         }
      });
btnRecordVideo.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            // record video
            recordVideo();
         }
      });
      // Checking camera availability
      if (!isDeviceSupportCamera()) {
         Toast.makeText(getApplicationContext(),
               "Sorry! Your device doesn't support camera",
               Toast.LENGTH_LONG).show();
         // will close the app if the device does't have camera
         finish();
      }
   }
```

```
private boolean isDeviceSupportCamera() {
   if (getApplicationContext().getPackageManager().hasSystemFeature(
         PackageManager.FEATURE_CAMERA)) {
      // this device has a camera
      return true;
   } else {
      // no camera on this device
      return false;
   }
}
private void recordVideo() {
   Intent intent = new Intent(MediaStore.ACTION_VIDEO_CAPTURE);
   fileUri = getOutputMediaFileUri(MEDIA_TYPE_VIDEO);
   // set video quality
   intent.putExtra(MediaStore.EXTRA_VIDEO_QUALITY, 1);
   intent.putExtra(MediaStore.EXTRA_OUTPUT, fileUri);
   startActivityForResult(intent, CAMERA_CAPTURE_VIDEO_REQUEST_CODE);
}
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
   if (requestCode == CAMERA_CAPTURE_IMAGE_REQUEST_CODE) {
      if (resultCode == RESULT_OK) {
      } else if (resultCode == RESULT_CANCELED) {
         Toast.makeText(getApplicationContext(),
               "User cancelled image capture", Toast.LENGTH_SHORT)
               .show();
      } else {
        Toast.makeText(getApplicationContext(),
               "Sorry! Failed to capture image", Toast.LENGTH_SHORT)
               .show();
      }
   } else
      if (requestCode == CAMERA_CAPTURE_VIDEO_REQUEST_CODE) {
      if (resultCode == RESULT_OK) {
      } else if (resultCode == RESULT_CANCELED) {
        Toast.makeText(getApplicationContext(),
               "User cancelled video recording", Toast.LENGTH_SHORT)
               .show();
      } else {
         Toast.makeText(getApplicationContext(),
               "Sorry! Failed to record video", Toast.LENGTH_SHORT)
               .show();
      }
   }
}
public Uri getOutputMediaFileUri(int type) {
```

```
        return Uri.fromFile(getOutputMediaFile(type));
    }
    private static File getOutputMediaFile(int type) {
        File mediaStorageDir = new
File(Environment.getExternalStorageDirectory(),IMAGE_DIRECTORY_NAME);
        if (!mediaStorageDir.exists()) {
            if (!mediaStorageDir.mkdirs()) {
                Log.d(IMAGE_DIRECTORY_NAME, "Oops! Failed create "
                        + IMAGE_DIRECTORY_NAME + " directory");
                return null;
            }
        }
String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss",
                Locale.getDefault()).format(new Date());
        File mediaFile;
        if (type == MEDIA_TYPE_IMAGE) {
            mediaFile = new File(mediaStorageDir.getPath() + File.separator
                    + "IMG_" + timeStamp + ".jpg");
        } else if (type == MEDIA_TYPE_VIDEO) {
            mediaFile = new File(mediaStorageDir.getPath() + File.separator
                    + "VID_" + timeStamp + ".mp4");
        } else {
            return null;
        }
        return mediaFile;
    }
}
```