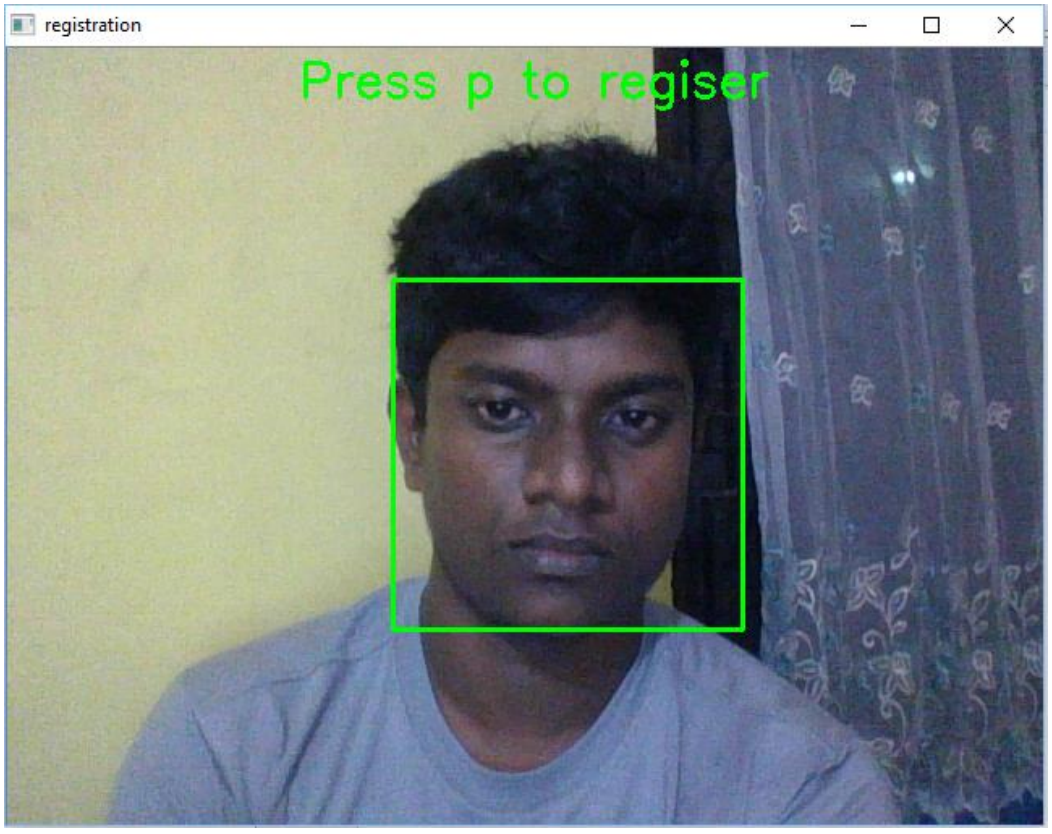
Source Code

```
import cv2
import os
import numpy as np
import time

def draw_rectangle(img, rect):#function for drawing rectangles
    (x, y, w, h) = rect
    cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)

def faceDetection(img):
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)#convert to grayscale
    face_cascade = cv2.CascadeClassifier('lbpcascade_frontalface.xml')#detect faces using lbpcascade
    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.2, minNeighbors=5);
    if (len(faces) == 0):
        return None, None
    (x, y, w, h) = faces[0]#get coordinates of the faces
    return gray[y:y + w, x:x + h], faces[0]

def registration():
    subject_images_names = os.listdir("Registered-images")
    faces = []
    labels = []
    label=1
    for image_name in subject_images_names:

            if image_name.startswith("."):
                continue;
            image_path = "Registered-images" + "/" + image_name

            image = cv2.imread(image_path)

            cv2.imshow("Registering new User...", cv2.resize(image, (400, 500)))
            cv2.waitKey(100)

            face, rect = faceDetection(image)#get faces to a array

            if face is not None:
                faces.append(face)
                labels.append(label)

    cv2.destroyAllWindows()
    cv2.waitKey(1)
    cv2.destroyAllWindows()

    return faces, labels


print("Registering new user...")

camera = cv2.VideoCapture(0)
reg_button=False
i=0
while True:
    ret,rframe=camera.read()
    cv2.putText(rframe, "Press p to regiser", (180, 30),
                cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
    face, rect = faceDetection(rframe)
    if rect is not None:#draw rectangles around faces in registering feed
        draw_rectangle(rframe,[rect[0],rect[1],rect[2],rect[3]])

    cv2.imshow("registration",rframe)

```

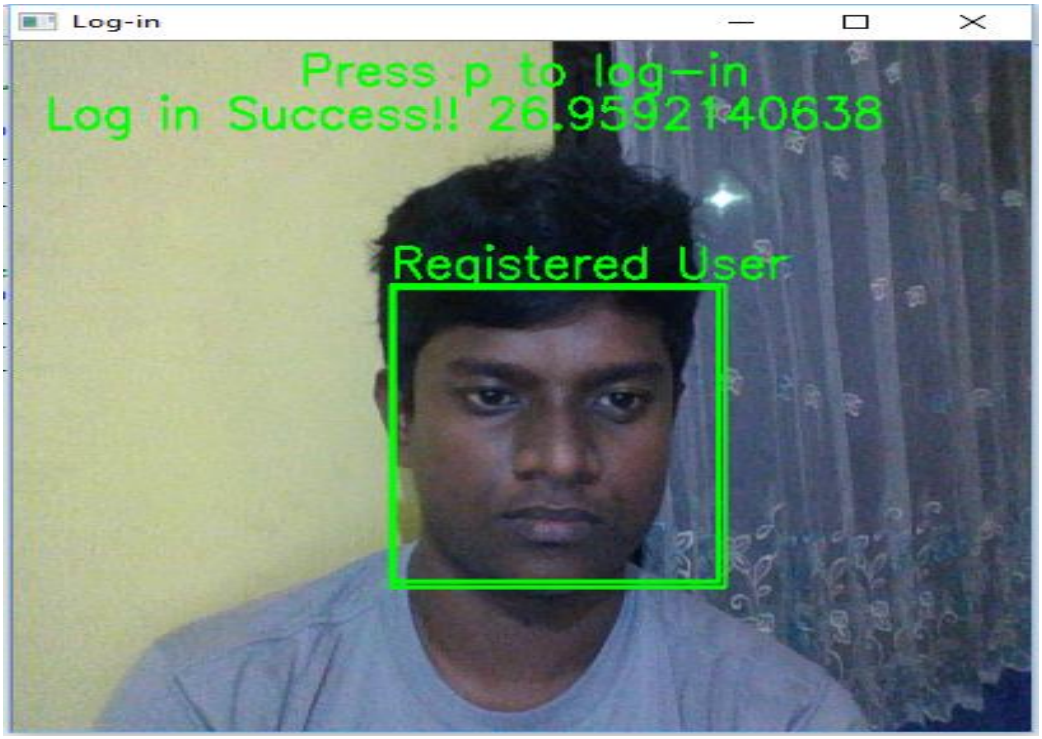
```
    if cv2.waitKey(1) == ord('p'):
        reg_button=True
    if reg_button:
        cv2.imwrite('Registered-images\\%s.png' % i, rframe)
        time.sleep(0.2)
        if i==12:#take 12 snaps when registering a person
            camera.release()
            break
        i=i+1

faces, labels = registration()
print("Registration Successful")

face_recognizer = cv2.face.LBPHFaceRecognizer_create()#initialize LBP face recognizer


try:
    face_recognizer.train(faces, np.array(labels))#trainingg LBP face recognizer


    def faceRecognition(login_image):#function for face recognnition
        img = login_image.copy()
        face, rect = faceDetection(img)

        label, confidence = face_recognizer.predict(face)

        print confidence

        if (confidence < 50):#accept only confidence is below 50
            draw_rectangle(img, rect)

            cv2.putText(img, "Registered User", ( rect[0], rect[1] - 5),
                        cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
            cv2.putText(img, "Log in Success!! " + str(confidence), (20, 60),
                        cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
        else:
            cv2.putText(img, "Log in Failed!! " + str(confidence), (20, 60),
                        cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)

        return img


    print("Recognizing on registered users...")
    camera2 = cv2.VideoCapture(0)
    while True:#get video feed when log in to the system

        ret, frame = camera2.read()

        cv2.putText(frame, "Press p to log-in", (180, 30),
                    cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
        face, rect = faceDetection(frame)
        if rect is not None:
            draw_rectangle(frame, [rect[0], rect[1], rect[2], rect[3]])
        cv2.imshow("Log-in", frame)
        if cv2.waitKey(1) == ord('p'):
            cv2.imwrite('login-image\\test1.png', frame)#write image
            camera2.release()
            break

    login_image = cv2.imread("login-image/test1.png")

    predicted_img1 = faceRecognition(login_image)#recogize the face using trained model
```

```
125
126         cv2.imshow("Log-in", cv2.resize(predicted_img1, (400, 500)))
127
128         cv2.waitKey(0)
129         cv2.destroyAllWindows()
130         cv2.waitKey(1)
131         cv2.destroyAllWindows()
132
133     except cv2.error:
134         print "No faces detected"
135         cv2.waitKey(0)
136         cv2.destroyAllWindows()
137         cv2.waitKey(1)
138         cv2.destroyAllWindows()
139
140
141
142
```

Registration Window



d Log-in Window(Success)



Log-in Window(Success)