```
import RPi.GPIO as GPIO
import time
import subprocess
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.mime.base import MIMEBase
from email import encoders
import os

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BOARD)
def sendmail():
    email_user = 'smita.apr25@gmail.com'
    email_password = 'smitaapr25'
    email_send = ['sourabhk104@gmail.com','
alishakumari1597@gmail.com','kanupam2016@gmail.com']
    subject = 'intruder detected'
    msg = MIMEMultipart()
    msg['From'] = email_user
    msg['To'] = email_send[0]
    msg['Subject'] = subject
    body = 'Here is the attached link of the video. 192.168.43.80:8081'
    msg.attach(MIMEText(body,'plain'))
    text = msg.as_string()
    server = smtplib.SMTP('smtp.gmail.com',587)
    server.starttls()
    server.login(email_user,email_password)
    server.sendmail(email_user,email_send,text)
    server.quit()
def sendmail1():
    email_user = 'smita.apr25@gmail.com'
    email_password = 'smitaapr25'
    email_send = ['sourabhk104@gmail.com','
alishakumari1597@gmail.com','kanupam2016@gmail.com']
    subject = 'wrong password'
    msg = MIMEMultipart()
    msg['From'] = email_user
    msg['To'] = email_send[0]
    msg['Subject'] = subject
    body = 'WRONG PASSWORD DETECTED.Here is the attached picture of the person'
    msg.attach(MIMEText(body,'plain'))
    filename='pic.jpg'
    attachment  =open(filename,'rb')
    part = MIMEBase('application','octet-stream')
    part.set_payload((attachment).read())
    encoders.encode_base64(part)
    part.add_header('Content-Disposition',"attachment; filename= "+filename)
    msg.attach(part)
    text = msg.as_string()
    server = smtplib.SMTP('smtp.gmail.com',587)
    server.starttls()
    server.login(email_user,email_password)
    server.sendmail(email_user,email_send,text)
    server.quit()
```

```
PIRpin= 3

GPIO.setup(PIRpin,GPIO.IN)

LCD_RS = 37
LCD_E  = 35
LCD_D4 = 33
LCD_D5 = 31
LCD_D6 = 29
LCD_D7 = 23

LCD_WIDTH = 16    # Maximum characters per line
LCD_CHR = True
LCD_CMD = False

LCD_LINE_1 = 0x80 # LCD RAM address for the 1st line
LCD_LINE_2 = 0xC0 # LCD RAM address for the 2nd line


# Timing constants
E_PULSE = 0.0005
E_DELAY = 0.0005


GPIO.setup(LCD_E, GPIO.OUT)  # E
GPIO.setup(LCD_RS, GPIO.OUT) # RS
GPIO.setup(LCD_D4, GPIO.OUT) # DB4
GPIO.setup(LCD_D5, GPIO.OUT) # DB5
GPIO.setup(LCD_D6, GPIO.OUT) # DB6
GPIO.setup(LCD_D7, GPIO.OUT) # DB7



def lcd_init():
  # Initialise display
  lcd_byte(0x33,LCD_CMD) # 110011 Initialise
  lcd_byte(0x32,LCD_CMD) # 110010 Initialise
  lcd_byte(0x06,LCD_CMD) # 000110 Cursor move direction
  lcd_byte(0x0C,LCD_CMD) # 001100 Display On,Cursor Off, Blink Off
  lcd_byte(0x28,LCD_CMD) # 101000 Data length, number of lines, font size
  lcd_byte(0x01,LCD_CMD) # 000001 Clear display
  time.sleep(E_DELAY)

def lcd_byte(bits, mode):
  # Send byte to data pins
  # bits = data
  # mode = True  for character
  #        False for command

  GPIO.output(LCD_RS, mode) # RS

  # High bits
  GPIO.output(LCD_D4, False)
```

```
  GPIO.output(LCD_D5, False)
  GPIO.output(LCD_D6, False)
  GPIO.output(LCD_D7, False)
  if bits&0x10==0x10:
    GPIO.output(LCD_D4, True)
  if bits&0x20==0x20:
    GPIO.output(LCD_D5, True)
  if bits&0x40==0x40:
    GPIO.output(LCD_D6, True)
  if bits&0x80==0x80:
    GPIO.output(LCD_D7, True)

  # Toggle 'Enable' pin
  lcd_toggle_enable()

  # Low bits
  GPIO.output(LCD_D4, False)
  GPIO.output(LCD_D5, False)
  GPIO.output(LCD_D6, False)
  GPIO.output(LCD_D7, False)
  if bits&0x01==0x01:
    GPIO.output(LCD_D4, True)
  if bits&0x02==0x02:
    GPIO.output(LCD_D5, True)
  if bits&0x04==0x04:
    GPIO.output(LCD_D6, True)
  if bits&0x08==0x08:
    GPIO.output(LCD_D7, True)

  # Toggle 'Enable' pin
  lcd_toggle_enable()

def lcd_toggle_enable():
  # Toggle enable
  time.sleep(E_DELAY)
  GPIO.output(LCD_E, True)
  time.sleep(E_PULSE)
  GPIO.output(LCD_E, False)
  time.sleep(E_DELAY)

def lcd_string(message,line):
  # Send string to display

  message = message.ljust(LCD_WIDTH," ")

  lcd_byte(line, LCD_CMD)

  for i in range(LCD_WIDTH):
    lcd_byte(ord(message[i]),LCD_CHR)




MATRIX = [
        ["1","2","3","A"],
```

```
        ["4","5","6","B"],
        ["7","8","9","C"],
        ["*","0","#","D"]
]
ROW = [7,11,13,15] # BCM numbering; Board numbering is: 7,8,10,11 (see pinout.xyz/)
COL = [12,16,22,18] # BCM numbering; Board numbering is: 12,13,15,16 (see pinout.xyz/)
p=""
for j in range(4):
    GPIO.setup(COL[j],GPIO.OUT)
    GPIO.output(COL[j],1)

for i in range(4):
    GPIO.setup(ROW[i], GPIO.IN, pull_up_down = GPIO.PUD_UP)
c="1234"
k=0
j=0
s=""
lcd_init()

try:
    while True:
        os.system("sudo service motion stop")
        lcd_string("                ",LCD_LINE_1)
        lcd_string("                ",LCD_LINE_2)
        i=GPIO.input(PIRpin)
        if(i==1):
            print ("intruder")
            os.system("sudo motion service restart")
            os.system("sudo motion")
            sendmail()
            p=""
            s=""
            k=0
            lcd_string("                ",LCD_LINE_1)
            lcd_string("                ",LCD_LINE_2)
            lcd_string("ENTER PASSWORD",LCD_LINE_1)
            while (k==0):
                for j in range(4):
                    GPIO.output(COL[j],0)
                    for i in range(4):
                        if GPIO.input(ROW[i])==0:
                            while(GPIO.input(ROW[i])==0):
                                pass
                            s+="*"
                            lcd_string(s,LCD_LINE_2)
                            p=p+MATRIX[i][j]
                            if len(p)==len(c):
                              if p==c:
                                lcd_string("            ",LCD_LINE_1)
                                lcd_string("            ",LCD_LINE_2)
                                lcd_string("WELCOME",LCD_LINE_1)
                              else:
                                lcd_string("            ",LCD_LINE_1)
                                lcd_string("            ",LCD_LINE_2)
                                lcd_string("WRONG PASSWORD",LCD_LINE_1)
                                os.system("sudo service motion stop")
                                subprocess.call("fswebcam -d /dev/video0 -r 1024x768 -S0
```

```
"+"pic.jpg",shell=True)
                        os.system("sudo service motion restart")
                        os.system("sudo motion")
                        sendmail1()
                       k=1
                GPIO.output(COL[j],1)
            time.sleep(5)
        else:
            os.system("sudo service motion stop")
            print ("No intruder")
            lcd_string("             ",LCD_LINE_1)
            lcd_string("             ",LCD_LINE_2)
            time.sleep(1)



except KeyboardInterrupt:
    GPIO.cleanup()
```