Assessment Submission

(Problem No - II)

**Code:**

from tkinter import \*

import smtplib

f=Tk()

f.title("Gmail Gui")

send\_email=StringVar()

send\_pass=StringVar()

recv\_email=StringVar()

msg\_body=None

def layout():

menuBar=Menu(f)

menuBar.add\_command(label="Instructions",command=instruct)

menuBar.add\_command(label="About",command=about)

f.config(menu=menuBar)

sender\_email=Label(f,text="Sender's Gmail ID: ")

sender\_entry=Entry(f,textvariable=send\_email,bd=3)

sender\_pass=Label(f,text="Sender's Gmail Pass: ")

sender\_passentry=Entry(f,show='\*',textvariable=send\_pass,bd=3)

receiver\_email=Label(f,text="Receiver's Email: ")

receiver\_entry=Entry(f,textvariable=recv\_email,bd=3)

msg\_label=Label(f,text='Message')

global msg\_body

msg\_body=Text(f,height=5,width=15,bd=3)

send=Button(f,text='Send',width=15,command=mail,bd=3)

cancel=Button(f,text='Cancel',width=15,command=destroy,bd=3)

sender\_email.grid(row=0,column=0,padx=5,pady=3)

sender\_entry.grid(row=0,column=1,padx=5,pady=3)

sender\_pass.grid(row=1,column=0,padx=5,pady=3)

sender\_passentry.grid(row=1,column=1,padx=5,pady=3)

receiver\_email.grid(row=2,column=0,padx=5,pady=3)

receiver\_entry.grid(row=2,column=1,padx=5,pady=3)

msg\_label.grid(row=3,column=0,padx=5,pady=3)

msg\_body.grid(row=3,column=1,padx=5,pady=3)

send.grid(row=4,column=0,padx=5,pady=3)

cancel.grid(row=4,column=1,padx=5,pady=3)

f.mainloop()

def destroy():

f.destroy()

def msg\_box():

messagebox.showinfo("Email Info","Mail Sent")

def instruct():

messagebox.showinfo("Instruction","switch 'allow less secure apps' to ON from\nhttps://myaccount.google.com/u/0/security?hl=en&pli=1#connectedapps\nbefore using the app!!")

def about():

messagebox.showinfo("About","This app is for only educational purpose\nCreated by Yagnesh Vakharia\nNon-Copywrite")

def mail():

try:

if send\_email.get()=="" or send\_pass.get()=="" or recv\_email.get()=="":

messagebox.showerror("Error","Please enter the complete details.")

else:

server=smtplib.SMTP('smtp.gmail.com',587)

server.starttls()

a=send\_email.get()

b=send\_pass.get()

c=msg\_body.get('1.0',END)

d=recv\_email.get()

server.login(a,b)

server.sendmail(a,d,c)

server.close()

msg\_box()

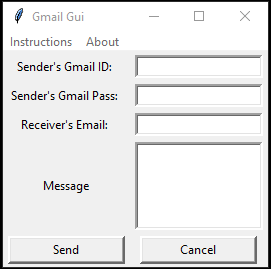
except Exception as e:

print(e)

a=messagebox.askokcancel("Error","Read instructions")

layout()

**Screenshot:**

****

**Creating messages with attachments**

def create\_message\_with\_attachment(  
    sender, to, subject, message\_text, file):  
  """Create a message for an email.  
  
  Args:  
    sender: Email address of the sender.  
    to: Email address of the receiver.  
    subject: The subject of the email message.  
    message\_text: The text of the email message.  
    file: The path to the file to be attached.  
  
  Returns:  
    An object containing a base64url encoded email object.  
  """  
  message = MIMEMultipart()  
  message['to'] = to  
  message['from'] = sender  
  message['subject'] = subject  
  
  msg = MIMEText(message\_text)  
  message.attach(msg)  
  
  content\_type, encoding = mimetypes.guess\_type(file)  
  
  if content\_type is None or encoding is not None:  
    content\_type = 'application/octet-stream'  
  main\_type, sub\_type = content\_type.split('/', 1)  
  if main\_type == 'text':  
    fp = open(file, 'rb')  
    msg = MIMEText(fp.read(), \_subtype=sub\_type)  
    fp.close()  
  elif main\_type == 'image':  
    fp = open(file, 'rb')  
    msg = MIMEImage(fp.read(), \_subtype=sub\_type)  
    fp.close()  
  elif main\_type == 'audio':  
    fp = open(file, 'rb')  
    msg = MIMEAudio(fp.read(), \_subtype=sub\_type)  
    fp.close()  
  else:  
    fp = open(file, 'rb')  
    msg = MIMEBase(main\_type, sub\_type)  
    msg.set\_payload(fp.read())  
    fp.close()  
  filename = os.path.basename(file)  
  msg.add\_header('Content-Disposition', 'attachment', filename=filename)  
  message.attach(msg)  
  
  return {'raw': base64.urlsafe\_b64encode(message.as\_string())}