```
___________________Host Monitoring___________________

from urllib.request import Request, urlopen
from urllib.error import URLError, HTTPError
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
import subprocess
import ipaddress

email_list = ['to mail id']
space = ', '

def send_mail(receivers, subject, message):
    sender = 'from mail id'
    mail_host = 'smtp.gmail.com'
    port_number = 587
    username = 'from mail id'
    password = 'password'

    msg = MIMEText(message)
    msg['From'] = "Host-Monitoring<sender>"
    msg['To'] = space.join(receivers)
    msg['Subject'] = subject

    smtpObj = smtplib.SMTP(mail_host, port_number)
    smtpObj.starttls()
    smtpObj.login(username, password)
    smtpObj.sendmail(sender, receivers, msg.as_string())
    smtpObj.quit()

send_mail(email_list, ('Task Done(Daily) | Host-Monitoring'), ('Host-Monitoring task completed successfully'))

def main():
    all_hosts = ['192.168.150.104','192.168.150.132','192.168.150.136','172.20.1.11','202.88.232.58', '192.168.240.104']

# Configure subprocess to hide the console window
    info = subprocess.STARTUPINFO()
    info.dwFlags |= subprocess.STARTF_USESHOWWINDOW
    info.wShowWindow = subprocess.SW_HIDE

# For each IP address in the subnet,
# run the ping command with subprocess.popen interface
    for i in range(len(all_hosts)):
        output = subprocess.Popen(['ping', '-n', '1', '-w', '500', str(all_hosts[i])], stdout=subprocess.PIPE, startupinfo=info).communicate()[0]

        if "Destination host unreachable" in output.decode('utf-8'):
            print(str(all_hosts[i]), "is Offline")
            send_mail(email_list, ('Host Down -' + str(all_hosts[i])), (str(all_hosts[i]) + ' is not reachable'))
        elif "Request timed out" in output.decode('utf-8'):
            print(str(all_hosts[i]), "is Offline")
            send_mail(email_list, ('Host Down -' + str(all_hosts[i])), (str(all_hosts[i]) + ' is not reachable'))
        else:
            print(str(all_hosts[i]), "is Online")

if __name__ == '__main__':
    main()

___________________Site_Availability_________________

from urllib.request import Request, urlopen
from urllib.error import URLError, HTTPError
import smtplib
```

```
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText

email_list = ['ashir.v@accelfrontline.com']
space = ', '

def send_mail(receivers, subject, message):
    sender = 'from mail id'
    mail_host = 'smtp.gmail.com'
    port_number = 587
    username = 'from mail id'
    password = 'password'

    msg = MIMEText(message)
    msg['From'] = "Site-Monitoring <sender>"
    msg['To'] = space.join(receivers)
    msg['Subject'] = subject

    smtpObj = smtplib.SMTP(mail_host, port_number)
    smtpObj.starttls()
    smtpObj.login(username, password)
    smtpObj.sendmail(sender, receivers, msg.as_string())
    smtpObj.quit()

#send_mail(email_list, ('Task Done | Site Monitoring'), ('Site availability monitoring task completed successfully'))
#, 'http://192.168.240.112/'
def main():
    sites = ['http://helpdesk/pro_users/login', 'http://it-stores.ushustech.com']

    for url in sites:
        try:
            response = urlopen(url)

        except HTTPError as e:
            print('Site Down - ', url, '   Error Code: ', e.code)
            send_mail(email_list, ('Site Down -' + url), (url + ' site is not reachable'))
        except URLError as e:
            print('Site Down - ', url)
            send_mail(email_list, ('Site Down -' + url), (url + ' site is not reachable'))
        else:
            print("Site Up   - ", url)
            #send_mail(email_list,('Site Up -'+url),(url+' site is available'))

if __name__ == '__main__':
    main()
```

___________________Login Helpdesk___________________

```
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from time import sleep

#username = input('Enter Username:')
#password = input('Enter Password:')
input = input('Open helpdesk portal?')
username = 'Ashir.V@Accelfrontline.com'
password = 'Accel123'
Chrome_Driver_Path = 'D:/Ashir_AFL/Doc_AFL/Scripts/chromedriver.exe'
URL = 'http://helpdesk/pro_users/login'

# Right click on the “Username” field of login page and select “inspect element”.
# We will use the value of the “name” attribue for this input which is “Username”.

Username_attribue = 'pro_user_email'

#Right click on the “Password” field of login page and select “inspect element”.
```

```
# In the script we will need to use the value of the “name” attribue for this input which is “Password”.

Password_attribue = 'pro_user_password'

chrome_options = Options()
chrome_options.add_argument("--disable-infobars")

driver = webdriver.Chrome(executable_path = Chrome_Driver_Path,chrome_options=chrome_options)
driver.maximize_window()
driver.implicitly_wait(20)

driver.get(URL)
print("Opened WebPage")
sleep(1)

username_box = driver.find_element_by_id(Username_attribue)
username_box.send_keys(username)
print("Username entered")
sleep(1)

password_box = driver.find_element_by_id(Password_attribue)
password_box.send_keys(password)
print("Password entered")

login_box = driver.find_element_by_xpath("//button[contains(@class, 'sui-bttn-primary sui-bttn')]")
login_box.click()
print("loged-in")
driver.get('http://helpdesk/tickets/v2#my_tickets')

#driver.find_element_by_class_name('sui-dropdown').click()

___________________Memory Usage Alert_________________

from urllib.error import URLError, HTTPError
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
import psutil
import platform
import datetime
from requests import get
import socket

host_name = socket.gethostname()
host_ip = socket.gethostbyname(host_name)
#print("Hostname : ", host_name)
#print("IP   : ", host_ip)

memory_percent = psutil.virtual_memory()[2]
cpu_percent = psutil.cpu_percent(interval=1)
disk_percent = psutil.disk_usage('/')[3]
boot_time = datetime.datetime.fromtimestamp(psutil.boot_time())
running_since = boot_time.strftime("%A %d. %B %Y")
os, name, version, _, _, _ = platform.uname()
version = version.split('-')[0]
cores = psutil.cpu_count()
ip = get('https://api.ipify.org').text

email_list = ['ashir.v@accelfrontline.com']
space = ', '
def send_mail(receivers, subject, message):
    sender = 'from mail id'
    mail_host = 'smtp.gmail.com'
    port_number = 587
    username = 'from mail id'
    password = 'password'

    msg = MIMEText(message)
    msg['From'] = "Memory Usage Alert <sender>"
    msg['To'] = space.join(receivers)
    msg['Subject'] = subject

    smtpObj = smtplib.SMTP(mail_host, port_number)
    smtpObj.starttls()
    smtpObj.login(username, password)
    smtpObj.sendmail(sender, receivers, msg.as_string())
    smtpObj.quit()

def main():
    if memory_percent >= 90:
        print('Memory Usage      -', memory_percent, '%')
        send_mail(email_list, ('Memory usage is high - ' + host_name), ('Memory usage is ' + str(memory_percent) + ' %'))
    else:
        print('Memory usage is normal')
        #send_mail(email_list, ('Memory usage is normal - ' + host_name), ('Memory usage is ' + str(memory_percent) + ' %'))

if __name__ == '__main__':
    main()

#print('External IP       -', ip)
#print('Memory Usage      -', memory_percent, '%')
#print('CPU Usage         -', cpu_percent, '%')
#print('Disk Usage        -', disk_percent, '%')
#print('Machine Boot Time -', boot_time)
#print('Running Since     -', running_since)
#print('OS Version        -', os, version)
#print('CPU Cores         -', cores)

Memory_Usage_Alert.vbs-------------------------------------------------------------------
Set WshShell = CreateObject("WScript.Shell")
WshShell.Run chr(34) & "D:\Ashir_AFL\Doc_AFL\Scripts\Memory_Usage_Alert.py" & Chr(34), 0
Set WshShell = Nothing
-----------------------------------------------------------------------------------------
___________________Open WebPage_________________

import webbrowser

input = input('Open sharepoint?')
sites = 'https://accelfrontline.sharepoint.com/teams/IT-Dept/'
chrome_path = 'C:/Program Files (x86)/Google/Chrome/Application/chrome.exe %s'
webbrowser.get(chrome_path).open(sites)
print('Sharepoint opened')

import webbrowser
from datetime import datetime

input = input('Open daily meeting report?')
now = datetime.now()
date = now.strftime("report-daily_meeting_%Y_%m_%d.pdf")
sites = 'https://accelfrontline.sharepoint.com/teams/IT-Dept/Shared%20Documents/Daily%20Meetings/'+date
chrome_path = 'C:/Program Files (x86)/Google/Chrome/Application/chrome.exe %s'
webbrowser.get(chrome_path).open(sites)
print('Meeting report opened')

___________________Send Mail_________________

import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText

def main():
    sender = 'from mail id'
    receivers = 'to mail id'
    mail_host = 'smtp.gmail.com'
    port_number = 587
    username = 'from mail id'
    password = 'password'
    subject = input('Enter Subject :')
    content = input('Please enter mail content :')

    msg = MIMEMultipart()
    msg['From'] = "Sticky Notes <sender>"
    msg['To'] = receivers
    msg['Subject'] = subject

    body = content

    msg.attach(MIMEText(body, 'plain'))

    smtpObj = smtplib.SMTP(mail_host, port_number)
    smtpObj.starttls()
    smtpObj.login(username, password)
    text = msg.as_string()
    smtpObj.sendmail(sender, receivers, text)
    print("Successfully sent email")
    smtpObj.quit()

if __name__ == '__main__':
    main()
```