

```

1      _____Host Monitoring_____
2
3      from urllib.request import Request, urlopen
4      from urllib.error import URLError, HTTPError
5      import smtplib
6      from email.mime.multipart import MIMEMultipart
7      from email.mime.text import MIMEText
8      import subprocess
9      import ipaddress
10
11     email_list = ['to mail id']
12     space = ', '
13
14     def send_mail(receivers, subject, message):
15         sender = 'from mail id'
16         mail_host = 'smtp.gmail.com'
17         port_number = 587
18         username = 'from mail id'
19         password = 'password'
20
21         msg = MIMEText(message)
22         msg['From'] = "Host-Monitoring<sender>"
23         msg['To'] = space.join(receivers)
24         msg['Subject'] = subject
25
26         smtpObj = smtplib.SMTP(mail_host, port_number)
27         smtpObj.starttls()
28         smtpObj.login(username, password)
29         smtpObj.sendmail(sender, receivers, msg.as_string())
30         smtpObj.quit()
31
32     send_mail(email_list, ('Task Done(Daily) | Host-Monitoring'), ('Host-Monitoring task
completed successfully'))
33
34     def main():
35         all_hosts =
36         ['192.168.150.104', '192.168.150.132', '192.168.150.136', '172.20.1.11', '202.88.232.58',
37         '192.168.240.104']
38
39     # Configure subprocess to hide the console window
40     info = subprocess.STARTUPINFO()
41     info.dwFlags |= subprocess.STARTF_USESHOWWINDOW
42     info.wShowWindow = subprocess.SW_HIDE
43
44     # For each IP address in the subnet,
45     # run the ping command with subprocess.Popen interface
46     for i in range(len(all_hosts)):
47         output = subprocess.Popen(['ping', '-n', '1', '-w', '500', str(all_hosts[i])],
48         stdout=subprocess.PIPE, startupinfo=info).communicate()[0]
49
50         if "Destination host unreachable" in output.decode('utf-8'):
51             print(str(all_hosts[i]), "is Offline")
52             send_mail(email_list, ('Host Down -' + str(all_hosts[i])),
53             (str(all_hosts[i]) + ' is not reachable'))
54         elif "Request timed out" in output.decode('utf-8'):
55             print(str(all_hosts[i]), "is Offline")
56             send_mail(email_list, ('Host Down -' + str(all_hosts[i])),
57             (str(all_hosts[i]) + ' is not reachable'))
58         else:
59             print(str(all_hosts[i]), "is Online")
60
61     if __name__ == '__main__':
62         main()
63
64     _____Site Availability_____
65
66     from urllib.request import Request, urlopen
67     from urllib.error import URLError, HTTPError
68     import smtplib

```

```

64 from email.mime.multipart import MIMEMultipart
65 from email.mime.text import MIMEText
66
67 email_list = ['ashir.v@accelfrontline.com']
68 space = ', '
69
70 def send_mail(receivers, subject, message):
71     sender = 'from mail id'
72     mail_host = 'smtp.gmail.com'
73     port_number = 587
74     username = 'from mail id'
75     password = 'password'
76
77     msg = MIMEText(message)
78     msg['From'] = "Site-Monitoring <sender>"
79     msg['To'] = space.join(receivers)
80     msg['Subject'] = subject
81
82     smtpObj = smtplib.SMTP(mail_host, port_number)
83     smtpObj.starttls()
84     smtpObj.login(username, password)
85     smtpObj.sendmail(sender, receivers, msg.as_string())
86     smtpObj.quit()
87
88 #send_mail(email_list, ('Task Done | Site Monitoring'), ('Site availability monitoring
task completed successfully'))
89 #, 'http://192.168.240.112/'
90 def main():
91     sites = ['http://helpdesk/pro_users/login', 'http://it-stores.ushustech.com']
92
93     for url in sites:
94         try:
95             response = urlopen(url)
96
97         except HTTPError as e:
98             print('Site Down - ', url, ' Error Code: ', e.code)
99             send_mail(email_list, ('Site Down -' + url), (url + ' site is not
reachable'))
100         except URLError as e:
101             print('Site Down - ', url)
102             send_mail(email_list, ('Site Down -' + url), (url + ' site is not
reachable'))
103         else:
104             print("Site Up - ", url)
105             #send_mail(email_list, ('Site Up -'+url), (url+' site is available'))
106
107 if __name__ == '__main__':
108     main()
109
110 _____Login Helpdesk_____
111
112 from selenium import webdriver
113 from selenium.webdriver.chrome.options import Options
114 from time import sleep
115
116 #username = input('Enter Username:')
117 #password = input('Enter Password:')
118 input = input('Open helpdesk portal?')
119 username = 'Ashir.V@Accelfrontline.com'
120 password = 'Accell23'
121 Chrome_Driver_Path = 'D:/Ashir_AFL/Doc_AFL/Scripts/chromedriver.exe'
122 URL = 'http://helpdesk/pro_users/login'
123
124 # Right click on the "Username" field of login page and select "inspect element".
125 # We will use the value of the "name" attribute for this input which is "Username".
126
127 Username_attribute = 'pro_user_email'
128
129 #Right click on the "Password" field of login page and select "inspect element".

```

```

130 # In the script we will need to use the value of the "name" attribute for this input
    which is "Password".
131
132 Password_attribute = 'pro_user_password'
133
134 chrome_options = Options()
135 chrome_options.add_argument("--disable-infobars")
136
137 driver = webdriver.Chrome(executable_path =
    Chrome_Driver_Path,chrome_options=chrome_options)
138 driver.maximize_window()
139 driver.implicitly_wait(20)
140
141 driver.get(URL)
142 print("Opened WebPage")
143 sleep(1)
144
145 username_box = driver.find_element_by_id(Username_attribute)
146 username_box.send_keys(username)
147 print("Username entered")
148 sleep(1)
149
150 password_box = driver.find_element_by_id>Password_attribute)
151 password_box.send_keys(password)
152 print("Password entered")
153
154 login_box = driver.find_element_by_xpath("//button[contains(@class, 'sui-bttn-primary
    sui-bttn')]")
155 login_box.click()
156 print("logged-in")
157 driver.get('http://helpdesk/tickets/v2#my_tickets')
158
159 #driver.find_element_by_class_name('sui-dropdown').click()
160
161 _____Memory Usage Alert_____
162
163 from urllib.error import URLError, HTTPError
164 import smtplib
165 from email.mime.multipart import MIMEMultipart
166 from email.mime.text import MIMEText
167 import psutil
168 import platform
169 import datetime
170 from requests import get
171 import socket
172
173 host_name = socket.gethostname()
174 host_ip = socket.gethostbyname(host_name)
175 #print("Hostname : ", host_name)
176 #print("IP    : ", host_ip)
177
178 memory_percent = psutil.virtual_memory()[2]
179 cpu_percent = psutil.cpu_percent(interval=1)
180 disk_percent = psutil.disk_usage('/') [3]
181 boot_time = datetime.datetime.fromtimestamp(psutil.boot_time())
182 running_since = boot_time.strftime("%A %d. %B %Y")
183 os, name, version, _, _ = platform.uname()
184 version = version.split('-')[0]
185 cores = psutil.cpu_count()
186 ip = get('https://api.ipify.org').text
187
188 email_list = ['ashir.v@accelfrontline.com']
189 space = ', '
190 def send_mail(receivers, subject, message):
191     sender = 'from mail id'
192     mail_host = 'smtp.gmail.com'
193     port_number = 587
194     username = 'from mail id'
195     password = 'password'

```

```

196
197     msg = MIMEText(message)
198     msg['From'] = "Memory Usage Alert <sender>"
199     msg['To'] = space.join(receivers)
200     msg['Subject'] = subject
201
202     smtpObj = smtplib.SMTP(mail_host, port_number)
203     smtpObj.starttls()
204     smtpObj.login(username, password)
205     smtpObj.sendmail(sender, receivers, msg.as_string())
206     smtpObj.quit()
207
208 def main():
209     if memory_percent >= 90:
210         print('Memory Usage      -', memory_percent, '%')
211         send_mail(email_list, ('Memory usage is high - ' + host_name), ('Memory usage
            is ' + str(memory_percent) + ' %'))
212     else:
213         print('Memory usage is normal')
214         #send_mail(email_list, ('Memory usage is normal - ' + host_name), ('Memory
            usage is ' + str(memory_percent) + ' %'))
215
216 if __name__ == '__main__':
217     main()
218
219 #print('External IP      -', ip)
220 #print('Memory Usage     -', memory_percent, '%')
221 #print('CPU Usage        -', cpu_percent, '%')
222 #print('Disk Usage       -', disk_percent, '%')
223 #print('Machine Boot Time -', boot_time)
224 #print('Running Since    -', running_since)
225 #print('OS Version       -', os, version)
226 #print('CPU Cores        -', cores)
227
228 Memory_Usage_Alert.vbs-----
229 Set WshShell = CreateObject("WScript.Shell")
230 WshShell.Run chr(34) & "D:\Ashir_AFL\Doc_AFL\Scripts\Memory_Usage_Alert.py" & Chr(34), 0
231 Set WshShell = Nothing
232 -----
233 _____Open WebPage_____
234
235 import webbrowser
236
237 input = input('Open sharepoint?')
238 sites = 'https://accelfrontline.sharepoint.com/teams/IT-Dept/'
239 chrome_path = 'C:/Program Files (x86)/Google/Chrome/Application/chrome.exe %s'
240 webbrowser.get(chrome_path).open(sites)
241 print('Sharepoint opened')
242
243 import webbrowser
244 from datetime import datetime
245
246 input = input('Open daily meeting report?')
247 now = datetime.now()
248 date = now.strftime("report-daily_meeting_%Y_%m_%d.pdf")
249 sites = '
https://accelfrontline.sharepoint.com/teams/IT-Dept/Shared%20Documents/Daily%20Meetings/
'+date
250 chrome_path = 'C:/Program Files (x86)/Google/Chrome/Application/chrome.exe %s'
251 webbrowser.get(chrome_path).open(sites)
252 print('Meeting report opened')
253
254 _____Send Mail_____
255
256 import smtplib
257 from email.mime.multipart import MIMEMultipart
258 from email.mime.text import MIMEText
259
260 def main():

```

```
261     sender = 'from mail id'
262     receivers = 'to mail id'
263     mail_host = 'smtp.gmail.com'
264     port_number = 587
265     username = 'from mail id'
266     password = 'password'
267     subject = input('Enter Subject :')
268     content = input('Please enter mail content :')
269
270     msg = MIMEMultipart()
271     msg['From'] = "Sticky Notes <sender>"
272     msg['To'] = receivers
273     msg['Subject'] = subject
274
275     body = content
276
277     msg.attach(MIMEText(body, 'plain'))
278
279     smtpObj = smtplib.SMTP(mail_host, port_number)
280     smtpObj.starttls()
281     smtpObj.login(username, password)
282     text = msg.as_string()
283     smtpObj.sendmail(sender, receivers, text)
284     print("Successfully sent email")
285     smtpObj.quit()
286
287 if __name__ == '__main__':
288     main()
289
290
```