PROJECT

Module Title:	Administrative Scripting	**Module Code:**	SN53002FP
Assignment:	Develop script to automate system job and to harden system.	**Duration**	

Objectives:

To test the student's skill to:

1. Write script for retrieving and sorting system information.
2. Apply exception technique to detect and handle script error.
3. Apply pattern matching algorithm to detect virus signature in log files.
4. Write script to protect data in information store.
5. Write script to synchronize execution of multiple system jobs.
6. Create task scheduler for scheduling system jobs.
7. Configure task scheduler for scheduling system jobs.

Major Tools & Equipment:

1. Personal Computer.
2. Windows 10 or later.
3. Python 3.9 and above.
4. PyCharm Community Editor 2019 or equivalent.

Scenario:

A cloud hosting company providing file storage services for user to upload and download files had contacted you to develop a script to automate maintenance tasks. The maintenance tasks include the following:

1: Download all files from the ftp or equivalent server to local system.

2: Separate images and text related files into two different folders, one for image and one for text.

3: Identify files containing virus pattern and move the files into a separate folder.

4: Archive the files in the text and image folder according to type and creation date.

5: Ping the health of FTP server and upload the zip files into the ftp server if FTP is healthy.

6: Create a log file to record completion of task 1 to task 6.

You will be given the following

- Folder containing text files and image files with some files infected by virus.
- Virus signature file containing the virus pattern.
- Configuration file containing all settings required for the project.

Instructions:

PART A: Data from information store manipulated correctly according to script requirement.

1. Create a project script with the following name "**StorageManagement**".

2. Create python script, **StorageManager**, to perform required tasks.

3. Write python code to download all files from FTP or equivalent server to local system.

   Before creating script to automate the task, test to make sure uploading and downloading of file via server is working. If this is not working, proceed to do the next task first by using the resource folder from local machine.

Once this is working, explore and write the script to access server and download all files from selected folder into client machine. The IP address, source folder from server and the destination folder in local machine will be read from a configuration file as follows:

Ipaddress= xx.xx.xx.xx
server_folder= <location1>
client_folder= <location2>

Configuration file will be given out separately along with the resource files.
Log successful outcome or any error encountered into a summary text file.

4. Write python code to separate images and text related files into two different folders, one for image and one for text.
   Sort the file and **move** files into the image and text folder location specified in the configuration file as follows:

   client_image_folder =<location3>
   client_text_folder =<location4>

   Log successful outcome or any error encountered into a summary text file.

PART B: Anomalies detected in log files using appropriate script.

1. Write python code to identify files containing virus pattern and move the files into a separate folder.

   Scan text file in text folder and **move** all text files with virus to the folder location specified in the configuration file as follows.

   client_virus_folder=<location5>

   Use regular expression to detect the pattern and log the infected file name into summary text file.

PART C: Data in information store protected correctly according to requirement.

1. Write python code to archive the files in the images and text folder according to type and creation date.

   Archive the file according to their type and date of creation as follows:

   /text/sample1.txt created on Dec 2022 archive to 202212-txt.zip
   /text/sample2.txt created on Dec 2022 archive to 202212-txt.zip
   /text/sample3.txt created on Jan 2023 archive to 202301-txt.zip
   /text/sample4.txt created on Jan 2023 archive to 202301-txt.zip

   /image/sample1.jpg created on Dec 2022 archive to 202212-img.zip
   /image/sample2.pngcreated on Dec 2022 archive to 202212-img.zip
   /image/sample3.jpg created on Jan 2023 archive to 202301-img.zip
   /image/sample4.pngcreated on Jan 2023 archive to 202301-img.zip

   Protect all the zip files with encryption. Use your name as the password and store the zip file in the location specified in the configuration file as follows:

   client_zip_folder=<location6>

   Log successful outcome or any error encountered into a summary text file.

PART D: Script error handled with exception handling according to required process.

1. Write python code to ping the health of FTP or equivalent server and upload the zip files into the server if server is healthy.

   Use python library to ping the connectivity of server. If connection is established, upload the image and text zip files into the location specified in the configuration file as follows:

   server_zip_folder=<location7>

   If connection is not established, wait for 5 sec and try again. Raise an error message "Server unavailable for file upload".

   If connection is not available after the second retry.

   Log successful outcome or any error encountered into a summary text file.

PART E: Script executed correctly without error according to requirement.

1. Write python code to record the activities in a summary file.

   Use python logging library to log the task activity into a file with the following name **yyyymmdd_hhmmss_yourname.log**. The log file should follow the following format:

   yyyy-mm-dd hh:mm:ss, ownername - INFO – <activities>

   An example of content in the log file as follows:

   2023-02-21 08:08:08, root –INFO – Downloading file from ftp done.
   2023-02-21 08:08:08, root –INFO – File sorting done.
   2023-02-21 08:08:08, root –INFO – Scan file for virus done.
   2023-02-21 08:08:08, root –INFO – Archive file done.
   2023-02-21 08:08:18, root –INFO – Ping ftp connectivity done.
   2023-02-21 8:08:18, root –INFO – Upload zip file done.

PART F: System jobs scheduled according to requirement.

1. Create a python script to automate the following:

   a. Create task scheduler with the name "**StorageManagement**".
   b. Configure task scheduler for **StorageManagement** to start on the date, time and interval as given in the specification.

   The scheduler can be a library from python or Window Task Scheduler created within python script.

---End---