**Assignment #2**

**Name: Saquib Ahmed Khan**

**Student ID: 100949697**

**Course Name: Scripting**

**Course Code: COSC1104 - 01**

**Date: 15-11-2024**

**Professor Sohaib Mohiuddin**

**Part 1: Identifying the Problems**

**Problem 1: Cloud Storage Usage Analyzer**

**Description:** Managing cloud storage efficiently is a common challenge, especially with multiple files and folders accumulating over time. A Python-based solution that scans a local folder synced with cloud storage (e.g., Google Drive, Dropbox) can analyze the usage by categorizing files, identifying duplicates, and showing the largest files.

**Value:** This solution would help users identify unnecessary files taking up storage, optimize cloud storage usage, and reduce costs. It can also help prevent running out of storage by providing periodic reports on usage trends.

**Complexity:** Developing this solution involves implementing file operations, analyzing file metadata, and providing summary reports. It requires knowledge of Python's os module and potentially APIs of cloud storage services.

**Expected Libraries:**

* os and shutil for file operations.
* argparse for command-line functionality.
* Optionally, google-api-python-client or dropbox for cloud integration.

**Problem 2: Automated Resource Provisioning Simulator**

**Description:** Cloud computing relies heavily on provisioning resources like virtual machines, storage, or networks. This Python program would simulate a basic cloud resource provisioning system where users can create, delete, and list resources, as well as allocate costs for each.

**Value:** This solution can help students and professionals understand the basics of resource provisioning in the cloud without accessing real cloud platforms. It simulates cost calculation and resource management, providing a simplified learning experience.

**Complexity:** The task involves creating a menu-driven program, using control structures, and maintaining resource information in memory or files. It also introduces the concept of cost calculation for cloud resources.

**Expected Libraries:**

* os or pickle for resource data persistence.
* time for simulating operations.
* Optionally, json for storing configuration data.

**Part 2: Solve the Problem Using Python**

**Code link:** [**cosc-1104-01-f24/assignment2/storage\_analyzer.py at main · Saquib-Ahmed-Khan/cosc-1104-01-f24**](https://github.com/Saquib-Ahmed-Khan/cosc-1104-01-f24/blob/main/assignment2/storage_analyzer.py)

**Part 3: Reflection**

**1. What was the most challenging aspect of solving this problem?**

The hardest part of this assignment was figuring out how to detect duplicate files. I learned that I couldn’t just rely on file names because two files with different names could have the same content. I had to research and understand file hashing (SHA256) to generate unique codes for each file based on its content. Implementing this process was tricky at first, especially for large files, but reading the files in chunks solved the problem.

**2. What was the most valuable thing you learned from this assignment?**

The most valuable thing I learned is how to work with file systems in Python. Before this assignment, I wasn’t familiar with concepts like walking through directories (os.walk), calculating file sizes, or categorizing files by type. I also learned how to use hashing to compare file content, which is a useful skill for detecting duplicates. Overall, this assignment gave me practical knowledge about handling and analyzing files programmatically.

**3. How did you test your completed product? How confident are you that it works reliably?**

I tested the program by creating a sample folder called saquib\_folder with various files of different types and sizes, including duplicates. I ensured the program accurately categorized files, detected duplicates, and listed the largest files. I tested it multiple times with different folder structures, including some edge cases like empty folders and very large files. Based on these tests, I’m confident that the program works reliably under normal conditions.

**4. Is there something you would still like to add to this, or something it makes you want to try next?**

Yes, I’d like to improve this program by adding features like:

* Automatically deleting duplicate files (with user confirmation).
* Generating a report (e.g., in a CSV or HTML file) with all the analysis results.
* Adding a graphical visualization of file types and their usage (e.g., pie charts). This assignment also inspired me to learn more about handling files in cloud storage systems like AWS S3 or Google Drive.