CodeAlpha Internship

Cloud Computing Tasks

Prepared for: Mariam Khaled Taha Elgmmal

Task 1: Data Redundancy Removal System

■ Overview:
This system prevents duplicate data entries (emails or names) in a simple user database.

■ Code (Python):
```
----------
users = []

def add_user(name):
if name in users:
return "■■ ■■■■■ ■■■■■■"
users.append(name)
return "■ ■■■ ■■■■■■■"

print(add_user("Mariam"))
print(add_user("Mariam"))
print(add_user("Ali"))
```

■■ Expected Output:
```
----------
■ ■■■ ■■■■■■■
■■ ■■■■■ ■■■■■■
■ ■■■ ■■■■■■■
```

■ Explanation:
- Creates a list of users.
- Adds a new name only if it is not already in the list.
- Prevents redundancy (■■■■■■■).

Task 4: Cloud Storage Simulation

■ Overview:
This task simulates a cloud storage system where users can upload, list, and delete files.

■ Code (Python):

```
cloud_storage = []

def upload_file(file_name):
cloud_storage.append(file_name)
return f"■ {file_name} uploaded."

def list_files():
return cloud_storage if cloud_storage else "■ No files."

def delete_file(file_name):
if file_name in cloud_storage:
cloud_storage.remove(file_name)
return f"■■ {file_name} deleted."
return "■■ File not found."

# Example Usage
print(upload_file("file1.txt"))
print(upload_file("photo.png"))
print(list_files())
print(delete_file("file1.txt"))
print(list_files())
```

■■ Expected Output:

```
■ file1.txt uploaded.
■ photo.png uploaded.
['file1.txt', 'photo.png']
■■ file1.txt deleted.
['photo.png']
```

■ Explanation:
- Mimics cloud storage behavior.
- Users can upload, list, and delete files.
- Helps understand basic cloud storage logic.