**Day 03 Task Allocation**

**Here is a basic Python script for an Asset Inventory Management System that allows you to add, delete, and manage assets. It also includes reporting functions for daily, weekly, and monthly reports. The system stores data in a simple list and uses CSV files to persist the data.**

**Source Code: -**

import csv

from datetime import datetime, timedelta

# Define file name for storing assets

FILE\_NAME = 'asset\_inventory.csv'

# Function to load assets from CSV file

def load\_assets():

assets = []

try:

with open(FILE\_NAME, mode='r') as file:

reader = csv.DictReader(file)

for row in reader:

assets.append(row)

except FileNotFoundError:

pass

return assets

# Function to save assets to CSV file

def save\_assets(assets):

with open(FILE\_NAME, mode='w', newline='') as file:

writer = csv.DictWriter(file, fieldnames=['Asset ID', 'Name', 'Category', 'Purchase Date', 'Status'])

writer.writeheader()

writer.writerows(assets)

# Function to add a new asset

def add\_asset():

assets = load\_assets()

asset\_id = input('Enter Asset ID: ')

name = input('Enter Asset Name: ')

category = input('Enter Asset Category: ')

purchase\_date = input('Enter Purchase Date (YYYY-MM-DD): ')

status = input('Enter Status (Active/Inactive): ')

assets.append({

'Asset ID': asset\_id,

'Name': name,

'Category': category,

'Purchase Date': purchase\_date,

'Status': status

})

save\_assets(assets)

print('Asset added successfully!')

# Function to delete an asset by ID

def delete\_asset():

assets = load\_assets()

asset\_id = input('Enter Asset ID to delete: ')

assets = [asset for asset in assets if asset['Asset ID'] != asset\_id]

save\_assets(assets)

print('Asset deleted successfully!')

# Function to view all assets

def view\_assets():

assets = load\_assets()

for asset in assets:

print(asset)

# Function to generate reports

def generate\_report(time\_frame):

assets = load\_assets()

report\_date = datetime.now()

if time\_frame == 'daily':

start\_date = report\_date - timedelta(days=1)

elif time\_frame == 'weekly':

start\_date = report\_date - timedelta(weeks=1)

elif time\_frame == 'monthly':

start\_date = report\_date - timedelta(days=30)

else:

print('Invalid time frame!')

return

report\_assets = [asset for asset in assets if datetime.strptime(asset['Purchase Date'], '%Y-%m-%d') >= start\_date]

print(f"{time\_frame.capitalize()} Report ({start\_date.strftime('%Y-%m-%d')} to {report\_date.strftime('%Y-%m-%d')})")

for asset in report\_assets:

print(asset)

# Main menu

def main():

while True:

print("\nAsset Inventory Management System")

print("1. Add Asset")

print("2. Delete Asset")

print("3. View All Assets")

print("4. Generate Daily Report")

print("5. Generate Weekly Report")

print("6. Generate Monthly Report")

print("7. Exit")

choice = input("Enter your choice: ")

if choice == '1':

add\_asset()

elif choice == '2':

delete\_asset()

elif choice == '3':

view\_assets()

elif choice == '4':

generate\_report('daily')

elif choice == '5':

generate\_report('weekly')

elif choice == '6':

generate\_report('monthly')

elif choice == '7':

break

else:

print("Invalid choice! Please try again.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**How it Works:**

1. **Load/Save Assets:** The system loads the asset data from a CSV file on startup and saves any changes to the file.
2. **Add Asset:** Allows the user to add a new asset by entering the required details.
3. **Delete Asset:** Deletes an asset by its ID.
4. **View Assets:** Displays all assets currently in the inventory.
5. **Generate Reports:** Generates reports for daily, weekly, and monthly time frames based on the purchase date of assets.

**Running the Code:**

1. Copy the script into a .py file (e.g., asset\_inventory.py).
2. Run the script in your Python environment: python asset\_inventory.py.

The data will be stored in a CSV file named asset\_inventory.csv, which will be created in the same directory as the script.