**Day 04 Task Allocation**

Below is an example of a Python program to manage an asset inventory system using a PostgreSQL database. This program includes functionality for adding, deleting, and listing assets, along with generating reports for daily, weekly, and monthly activities.

**Prerequisites**

Before you start, make sure you have installed the required packages:

**Source Code: -**

**pip install psycopg2-binary**

***Database related: -***

### PostgreSQL Table Setup

First, create a PostgreSQL table to store the asset information. Here's an SQL script to create the table:

**Source Code: -**

CREATE TABLE assets (

id SERIAL PRIMARY KEY,

name VARCHAR(255) NOT NULL,

description TEXT,

category VARCHAR(100),

purchase\_date DATE NOT NULL,

value DECIMAL(12, 2),

added\_on TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**Python Code for Asset Inventory Management System: -**

**import psycopg2**

**from psycopg2 import sql**

**from datetime import datetime, timedelta**

**# Database connection**

**def connect\_db():**

**return psycopg2.connect(**

**dbname="your\_db\_name",**

**user="your\_username",**

**password="your\_password",**

**host="your\_host",**

**port="your\_port"**

**)**

**# Function to add a new asset**

**def add\_asset(name, description, category, purchase\_date, value):**

**try:**

**conn = connect\_db()**

**cur = conn.cursor()**

**insert\_query = """**

**INSERT INTO assets (name, description, category, purchase\_date, value)**

**VALUES (%s, %s, %s, %s, %s);**

**"""**

**cur.execute(insert\_query, (name, description, category, purchase\_date, value))**

**conn.commit()**

**print(f"Asset '{name}' added successfully.")**

**except Exception as e:**

**print(f"Error adding asset: {e}")**

**finally:**

**cur.close()**

**conn.close()**

**# Function to delete an asset by ID**

**def delete\_asset(asset\_id):**

**try:**

**conn = connect\_db()**

**cur = conn.cursor()**

**delete\_query = "DELETE FROM assets WHERE id = %s;"**

**cur.execute(delete\_query, (asset\_id,))**

**conn.commit()**

**print(f"Asset ID '{asset\_id}' deleted successfully.")**

**except Exception as e:**

**print(f"Error deleting asset: {e}")**

**finally:**

**cur.close()**

**conn.close()**

**# Function to list all assets**

**def list\_assets():**

**try:**

**conn = connect\_db()**

**cur = conn.cursor()**

**cur.execute("SELECT \* FROM assets ORDER BY added\_on DESC;")**

**assets = cur.fetchall()**

**print("ID | Name | Description | Category | Purchase Date | Value | Added On")**

**print("-" \* 70)**

**for asset in assets:**

**print(f"{asset[0]} | {asset[1]} | {asset[2]} | {asset[3]} | {asset[4]} | {asset[5]} | {asset[6]}")**

**except Exception as e:**

**print(f"Error listing assets: {e}")**

**finally:**

**cur.close()**

**conn.close()**

**# Function to generate reports**

**def generate\_report(period='daily'):**

**try:**

**conn = connect\_db()**

**cur = conn.cursor()**

**if period == 'daily':**

**date\_from = datetime.now().date()**

**elif period == 'weekly':**

**date\_from = datetime.now().date() - timedelta(days=7)**

**elif period == 'monthly':**

**date\_from = datetime.now().date().replace(day=1)**

**else:**

**print("Invalid period. Choose 'daily', 'weekly', or 'monthly'.")**

**return**

**query = """**

**SELECT \* FROM assets WHERE added\_on >= %s ORDER BY added\_on DESC;**

**"""**

**cur.execute(query, (date\_from,))**

**assets = cur.fetchall()**

**print(f"Report for {period.capitalize()} period from {date\_from}:")**

**print("ID | Name | Description | Category | Purchase Date | Value | Added On")**

**print("-" \* 70)**

**for asset in assets:**

**print(f"{asset[0]} | {asset[1]} | {asset[2]} | {asset[3]} | {asset[4]} | {asset[5]} | {asset[6]}")**

**except Exception as e:**

**print(f"Error generating report: {e}")**

**finally:**

**cur.close()**

**conn.close()**

**# Main function**

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

**# Example usage**

**while True:**

**print("\nAsset Inventory Management System")**

**print("1. Add New Asset")**

**print("2. Delete Asset")**

**print("3. List All Assets")**

**print("4. Generate Report")**

**print("5. Exit")**

**choice = input("Enter your choice: ")**

**if choice == '1':**

**name = input("Enter asset name: ")**

**description = input("Enter asset description: ")**

**category = input("Enter asset category: ")**

**purchase\_date = input("Enter purchase date (YYYY-MM-DD): ")**

**value = float(input("Enter asset value: "))**

**add\_asset(name, description, category, purchase\_date, value)**

**elif choice == '2':**

**asset\_id = int(input("Enter asset ID to delete: "))**

**delete\_asset(asset\_id)**

**elif choice == '3':**

**list\_assets()**

**elif choice == '4':**

**period = input("Enter report period (daily, weekly, monthly): ").lower()**

**generate\_report(period)**

**elif choice == '5':**

**print("Exiting...")**

**break**

**else:**

**print("Invalid choice, please try again.")**

### How It Works

1. **Add New Asset**: You can add a new asset by providing its name, description, category, purchase date, and value.
2. **Delete Asset**: You can delete an asset by providing its ID.
3. **List All Assets**: Lists all the assets in the inventory.
4. **Generate Report**: Generates a report for assets added in the last day, week, or month based on user input.

### Running the Code

Replace the placeholders for dbname, user, password, host, and port in the connect\_db() function with your actual PostgreSQL database credentials.

You can extend and modify this code according to your specific requirements.