**Day 06 Task Allocation: -**

Below is a basic example of a Python script for an Asset Inventory Management System using PostgreSQL as the database. This script is a simplified version to get you started. It includes the use of psycopg2 to connect to the PostgreSQL database and tkinter for a simple graphical interface that includes your logo.

**Prerequisites:**

* Install psycopg2 for PostgreSQL connection: pip install psycopg2
* Install tkinter (comes pre-installed with Python on most systems)

**Python Script:**

**Source Code:-**

**import psycopg2**

**from tkinter import \***

**from tkinter import messagebox**

**from PIL import Image, ImageTk**

**# Database Connection**

**def connect\_db():**

**try:**

**conn = psycopg2.connect(**

**dbname="your\_dbname",**

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

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

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

**port="your\_port"**

**)**

**return conn**

**except Exception as e:**

**messagebox.showerror("Database Error", f"Error connecting to the database: {e}")**

**return None**

**# Create Asset Table if not exists**

**def create\_table():**

**conn = connect\_db()**

**if conn:**

**try:**

**cur = conn.cursor()**

**cur.execute("""**

**CREATE TABLE IF NOT EXISTS assets (**

**id SERIAL PRIMARY KEY,**

**asset\_name VARCHAR(100),**

**asset\_type VARCHAR(50),**

**purchase\_date DATE,**

**status VARCHAR(20)**

**)**

**""")**

**conn.commit()**

**cur.close()**

**conn.close()**

**except Exception as e:**

**messagebox.showerror("Table Error", f"Error creating table: {e}")**

**# Insert Asset into Database**

**def insert\_asset():**

**conn = connect\_db()**

**if conn:**

**try:**

**cur = conn.cursor()**

**cur.execute("""**

**INSERT INTO assets (asset\_name, asset\_type, purchase\_date, status)**

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

**""", (asset\_name.get(), asset\_type.get(), purchase\_date.get(), status.get()))**

**conn.commit()**

**messagebox.showinfo("Success", "Asset added successfully")**

**cur.close()**

**conn.close()**

**except Exception as e:**

**messagebox.showerror("Insert Error", f"Error inserting asset: {e}")**

**# Main Application Window**

**root = Tk()**

**root.title("Asset Inventory Management - SSAP SOLUTIONS LLP")**

**root.geometry("500x500")**

**# Load and Display Logo**

**logo = Image.open("path\_to\_your\_logo.png") # Replace with your logo path**

**logo = logo.resize((200, 100), Image.ANTIALIAS)**

**logo = ImageTk.PhotoImage(logo)**

**logo\_label = Label(image=logo)**

**logo\_label.pack()**

**# Asset Entry Form**

**Label(root, text="Asset Name").pack()**

**asset\_name = Entry(root)**

**asset\_name.pack()**

**Label(root, text="Asset Type").pack()**

**asset\_type = Entry(root)**

**asset\_type.pack()**

**Label(root, text="Purchase Date (YYYY-MM-DD)").pack()**

**purchase\_date = Entry(root)**

**purchase\_date.pack()**

**Label(root, text="Status").pack()**

**status = Entry(root)**

**status.pack()**

**# Submit Button**

**submit\_button = Button(root, text="Add Asset", command=insert\_asset)**

**submit\_button.pack()**

**# Initialize Database**

**create\_table()**

**# Run Application**

**root.mainloop()**

**Explanation:**

1. **Database Connection**: The connect\_db function connects to your PostgreSQL database using psycopg2. You need to replace the placeholder values (your\_dbname, your\_username, your\_password, your\_host, your\_port) with your actual database credentials.
2. **Create Table**: The create\_table function creates an assets table in the database if it doesn't already exist.
3. **Insert Asset**: The insert\_asset function inserts a new asset into the database when the user fills in the form and clicks the "Add Asset" button.
4. **Tkinter GUI**: The script uses tkinter to create a simple graphical user interface (GUI) where users can input asset details. The logo is displayed at the top of the window.
5. **Logo**: Replace "path\_to\_your\_logo.png" with the actual path to your SSAP SOLUTIONS LLP logo file.

**Notes:**

* This script is just a starting point. You can expand it by adding more features such as viewing, updating, and deleting assets.
* Ensure that the PostgreSQL server is running, and the credentials provided are correct.
* The PIL library is used to handle the logo image, and tkinter is used for the GUI.

Let me know if you need further enhancements or have any specific requirements!