

Steps to run the project

Step 1: Save the file as app.py

app.py

```
import sqlite3
from flask import Flask, render_template, request
import tkinter as tk
from tkinter import ttk

app = Flask(__name__)

# Create a SQLite database and connect to it
conn = sqlite3.connect("lost_and_found.db")
cursor = conn.cursor()

# Create a table to store lost and found items
cursor.execute("""
    CREATE TABLE IF NOT EXISTS items (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        name TEXT NOT NULL,
        description TEXT,
        location TEXT,
        status TEXT NOT NULL
    )
""")
conn.commit()

@app.route('/add_item', methods=['POST'])
def add_item():
    name = request.form['name']
    description = request.form['description']
    location = request.form['location']

    cursor.execute("INSERT INTO items (name, description, location, status)
VALUES (?, ?, ?, ?)",
                    (name, description, location, "Lost"))
    conn.commit()
    return "Item added successfully."

@app.route('/find_item', methods=['POST'])
def find_item():
    name = request.form['name']
    cursor.execute("SELECT * FROM items WHERE name=?", (name,))
    item = cursor.fetchone()
    if item:
        return f"Name: {item[1]}\nDescription: {item[2]}\nLocation: {item[3]}\nStatus: {item[4]}"
    else:
```

```

        return "Item not found."

@app.route('/update_status_found', methods=['POST'])
def update_status_found():
    name = request.form['name']
    cursor.execute("UPDATE items SET status=? WHERE name=?", ("Found",
name))
    conn.commit()
    return f"Status of item '{name}' updated to 'Found'."

@app.route('/list_items')
def list_items():
    cursor.execute("SELECT * FROM items")
    items = cursor.fetchall()
    result = ""
    if items:
        for item in items:
            result += f"Name: {item[1]}\nDescription: {item[2]}\nLocation:
{item[3]}\nStatus: {item[4]}\n{'-'*20}\n"
        else:
            result = "No items found."
    return result

def clear_entries():
    name_entry.delete(0, tk.END)
    description_entry.delete(0, tk.END)
    location_entry.delete(0, tk.END)

def add_item_tkinter():
    name = name_entry.get()
    description = description_entry.get()
    location = location_entry.get()

    cursor.execute("INSERT INTO items (name, description, location, status)
VALUES (?, ?, ?, ?)",
                    (name, description, location, "Lost"))
    conn.commit()
    result_label.config(text="Item added successfully.")
    clear_entries()

def find_item_tkinter():
    name = name_entry.get()
    cursor.execute("SELECT * FROM items WHERE name=?", (name,))
    item = cursor.fetchone()
    if item:
        result_label.config(text=f"Name: {item[1]}\nDescription:
{item[2]}\nLocation: {item[3]}\nStatus: {item[4]}")
    else:

```

```

        result_label.config(text="Item not found.")

def update_status_found_tkinter():
    name = name_entry.get()
    cursor.execute("UPDATE items SET status=? WHERE name=?", ("Found",
name))
    conn.commit()
    result_label.config(text=f"Status of item '{name}' updated to 'Found'.")
    list_items_tkinter() # Refresh the list of items to reflect the updated
status

def list_items_tkinter():
    cursor.execute("SELECT * FROM items")
    items = cursor.fetchall()
    if items:
        result_label.config(text="")
        for item in items:
            result_label.config(text=result_label.cget("text") + f"\nName:
{item[1]}\nDescription: {item[2]}\nLocation: {item[3]}\nStatus:
{item[4]}\n{'-'*20}\n")
    else:
        result_label.config(text="No items found.")

# Create the main window
root = tk.Tk()
root.title("Lost and Found System")

# Create and layout widgets
frame = ttk.Frame(root)
frame.grid(column=0, row=0, padx=10, pady=10)

name_label = ttk.Label(frame, text="Name:")
name_label.grid(column=0, row=0, padx=5, pady=5, sticky=tk.W)

name_entry = ttk.Entry(frame)
name_entry.grid(column=1, row=0, padx=5, pady=5)

description_label = ttk.Label(frame, text="Description:")
description_label.grid(column=0, row=1, padx=5, pady=5, sticky=tk.W)

description_entry = ttk.Entry(frame)
description_entry.grid(column=1, row=1, padx=5, pady=5)

location_label = ttk.Label(frame, text="Location:")
location_label.grid(column=0, row=2, padx=5, pady=5, sticky=tk.W)

location_entry = ttk.Entry(frame)
location_entry.grid(column=1, row=2, padx=5, pady=5)

```

```
add_button = ttk.Button(frame, text="Add Item", command=add_item_tkinter)
add_button.grid(column=0, row=3, columnspan=2, padx=5, pady=5)

find_button = ttk.Button(frame, text="Find Item", command=find_item_tkinter)
find_button.grid(column=0, row=4, columnspan=2, padx=5, pady=5)

update_found_button = ttk.Button(frame, text="Update Status to Found",
command=update_status_found_tkinter)
update_found_button.grid(column=0, row=5, columnspan=2, padx=5, pady=5)

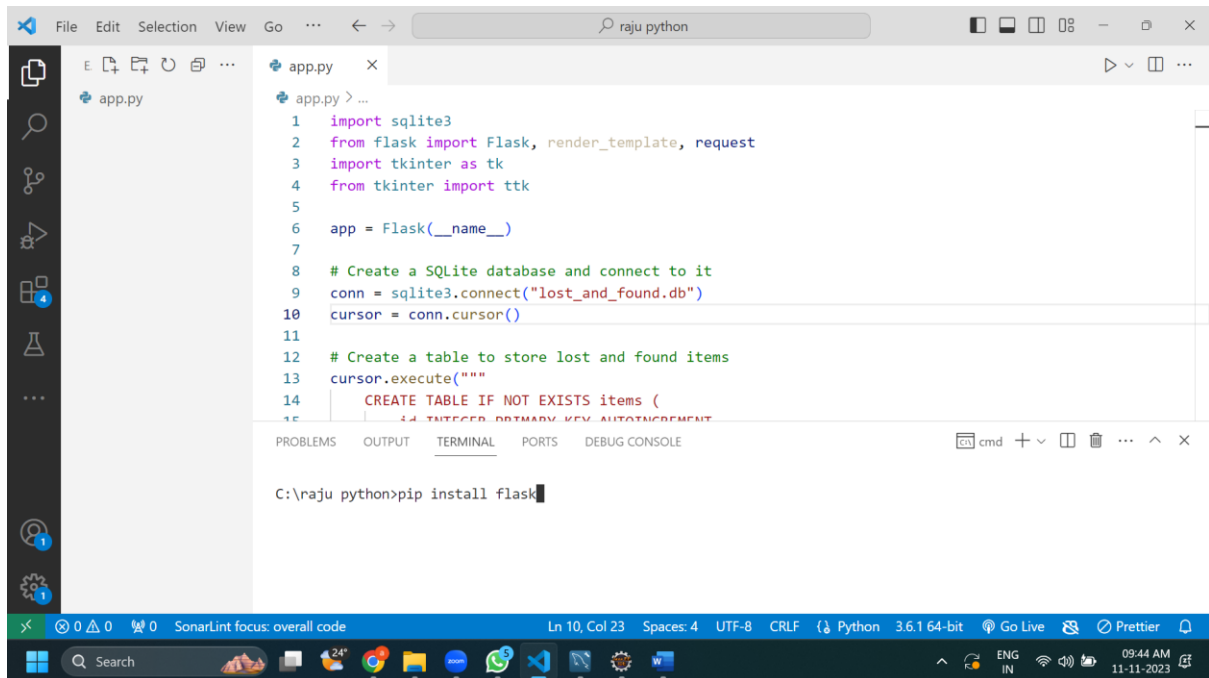
list_button = ttk.Button(frame, text="List All Items",
command=list_items_tkinter)
list_button.grid(column=0, row=6, columnspan=2, padx=5, pady=5)

result_label = ttk.Label(frame, text="", wraplength=300)
result_label.grid(column=0, row=7, columnspan=2, padx=5, pady=10)

root.mainloop()

if __name__ == '__main__':
    app.run(debug=True)
```

Step 2: Go to project location and run the following commands



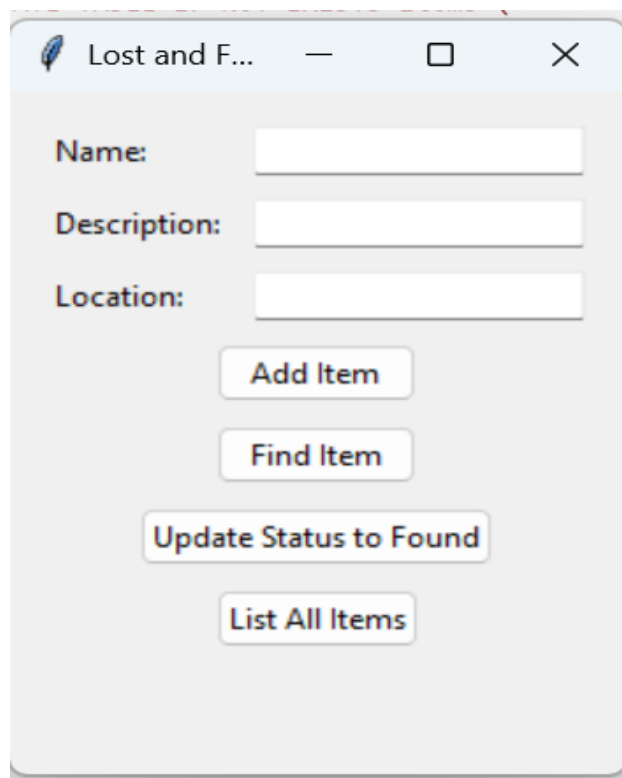
The screenshot shows a Visual Studio Code editor window with a file named `app.py` open. The code in the editor is as follows:

```
1 import sqlite3
2 from flask import Flask, render_template, request
3 import tkinter as tk
4 from tkinter import ttk
5
6 app = Flask(__name__)
7
8 # Create a SQLite database and connect to it
9 conn = sqlite3.connect("lost_and_found.db")
10 cursor = conn.cursor()
11
12 # Create a table to store lost and found items
13 cursor.execute("""
14     CREATE TABLE IF NOT EXISTS items (
15         id INTEGER PRIMARY KEY,
16         name TEXT NOT NULL,
17         description TEXT NOT NULL,
18         location TEXT NOT NULL,
19         status TEXT NOT NULL
20     )
21 """)
22
23 # Home page
24 @app.route("/")
25 def home():
26     return render_template("index.html")
27
28 # Add item route
29 @app.route("/add", methods=["POST"])
30 def add_item():
31     name = request.form["name"]
32     description = request.form["description"]
33     location = request.form["location"]
34     status = "Lost"
35     cursor.execute("INSERT INTO items (name, description, location, status) VALUES (?, ?, ?, ?)", (name, description, location, status))
36     conn.commit()
37     return render_template("index.html")
38
39 # Find item route
40 @app.route("/find", methods=["POST"])
41 def find_item():
42     name = request.form["name"]
43     description = request.form["description"]
44     location = request.form["location"]
45     status = "Found"
46     cursor.execute("UPDATE items SET status = ? WHERE name = ? AND description = ? AND location = ?", (status, name, description, location))
47     conn.commit()
48     return render_template("index.html")
49
50 # Update status to Found route
51 @app.route("/update", methods=["POST"])
52 def update_status():
53     name = request.form["name"]
54     description = request.form["description"]
55     location = request.form["location"]
56     status = "Found"
57     cursor.execute("UPDATE items SET status = ? WHERE name = ? AND description = ? AND location = ?", (status, name, description, location))
58     conn.commit()
59     return render_template("index.html")
60
61 # List all items route
62 @app.route("/list", methods=["GET"])
63 def list_items():
64     cursor.execute("SELECT * FROM items")
65     items = cursor.fetchall()
66     return render_template("list_items.html", items=items)
67
68 if __name__ == "__main__":
69     app.run(debug=True)
```

Below the editor, the TERMINAL panel shows the command `C:\raju python>pip install flask` being executed.

C:\raju python>pip install flask

CC:\raju python>python app.py



The screenshot shows a web application window titled "Lost and F...". It contains three input fields labeled "Name:", "Description:", and "Location:". Below these fields are four buttons: "Add Item", "Find Item", "Update Status to Found", and "List All Items".