UNIVERSITI TEKNOLOGI MARA

(UiTM) KEDAH, KAMPUS SUNGAI PETANI

SCHOOL OF INFORMATION SCIENCE

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

DIPLOMA IN INFORMATICS LIBRARY

[CDIM144]

PROGRAMMING FOR LIBRARIES [IML208]

INDIVIDUAL PROJECT:

PREPARED BY:

NAME	STUDENT ID
NURUL AMNA ZAFIRAH BINTI MOHD ASHID	2023620732

GROUP: KCDIM144 3E

PREPARED FOR:

MOHD FIRDAUS BIN MOHD HELMI

SUBMISSION DATE:

18 DECEMBER 2024

1. Project Name: Hospital Appointment System

2. File name: hosp.appointment.py

3. Prompt Data:

i) appointment id: Uniquely identifies each appointment.

ii) patient name: Identifies the person who needs the appointment.

iii) doctor name: Determines which doctor will attend the patient.

iv) appointment date: Schedules the exact date of the appointment.

v) reason: Provides context or purpose for the appointment.

4. Function:

i) Create Appointment: Collect user input and adds a new appointment to the list.

ii) View Appointment: Displays all current appointments in a numbered list.

iii) Update Appointment: Modify an appointment by choosing its number.

iv) Delete Appointment: Allows users to remove an appointment by selecting its number.

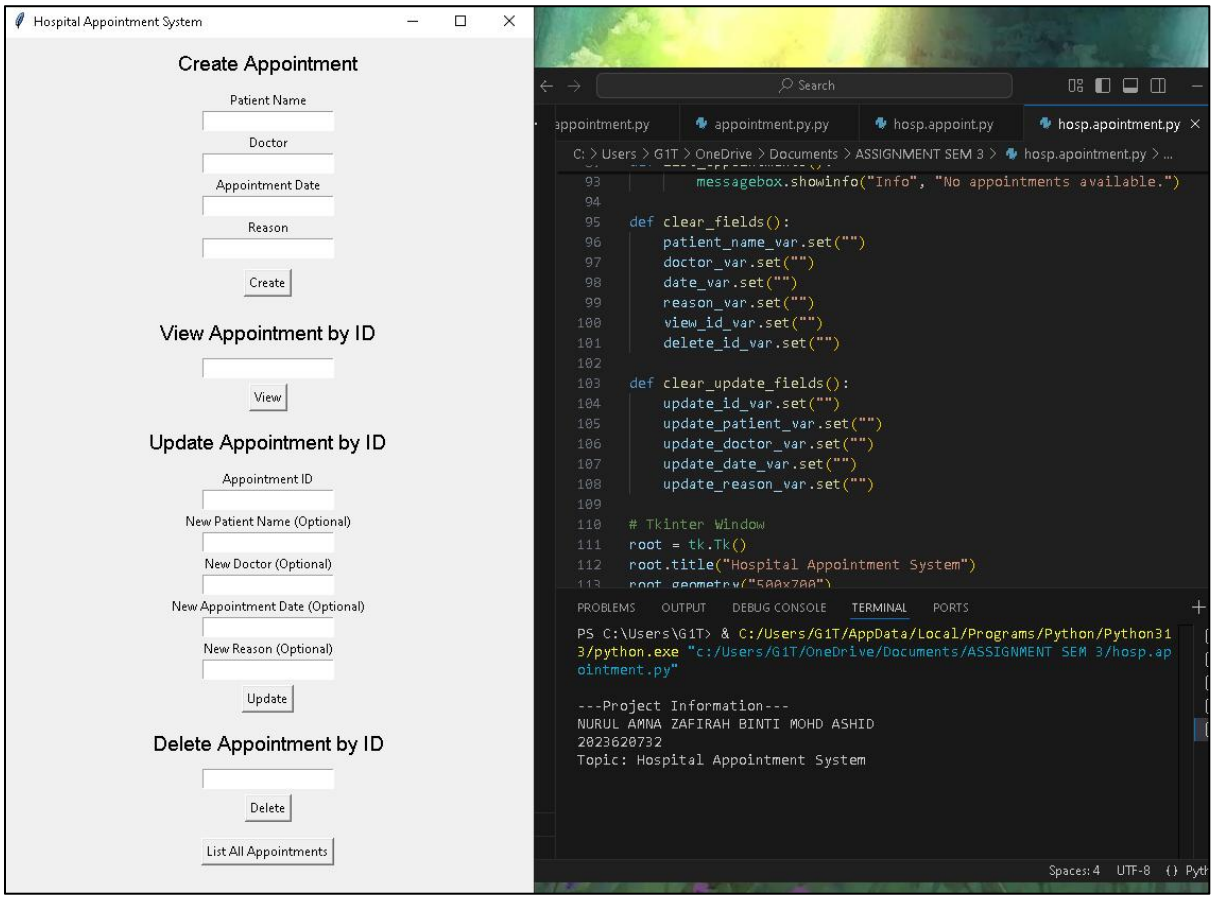
v) List All Appointment: Shows all existing appointments.

vi) Exit: Stops the program and exits the main loop.

5. Conditional Statement:

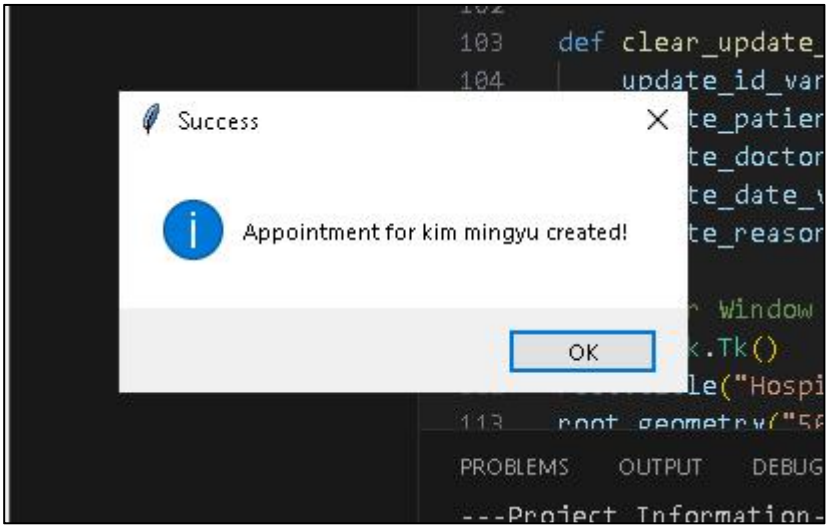
If, Elif, Else: Yes

6. GUI: Yes


Hospital Appointment System
Create Appointment
Patient Name
Doctor
Appointment Date
Reason
Create
View Appointment by ID
View
Update Appointment by ID
Appointment ID
New Patient Name (Optional)
New Doctor (Optional)
New Appointment Date (Optional)
New Reason (Optional)
Update
Delete Appointment by ID
Delete
List All Appointments
messagebox.showinfo("Info", "No appointments available.")
def clear_fields():
patient_name_var.set("")
doctor_var.set("")
date_var.set("")
reason_var.set("")
view_id_var.set("")
delete_id_var.set("")
def clear_update_fields():
update_id_var.set("")
update_patient_var.set("")
update_doctor_var.set("")
update_date_var.set("")
update_reason_var.set("")
# Tkinter Window
root = tk.Tk()
root.title("Hospital Appointment System")
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\G1T> & C:/Users/G1T/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/G1T/OneDrive/Documents/ASSIGNMENT SEM 3/hosp.appointment.py"
---Project Information---
NURUL AMNA ZAFIRAH BINTI MOHD ASHID
2023620732
Topic: Hospital Appointment System

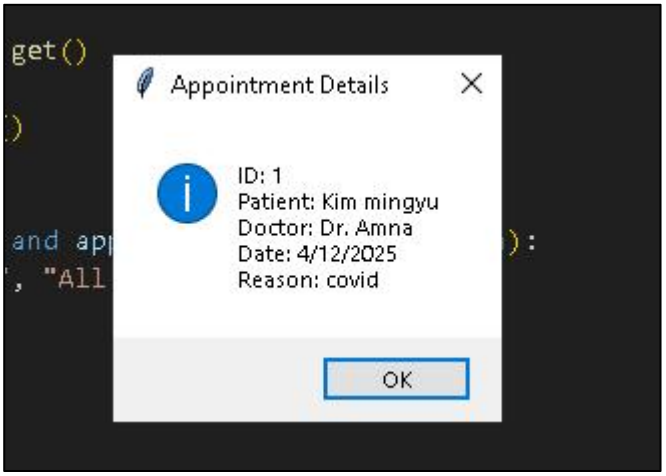
7. Result:

7.1 Create Appointment:

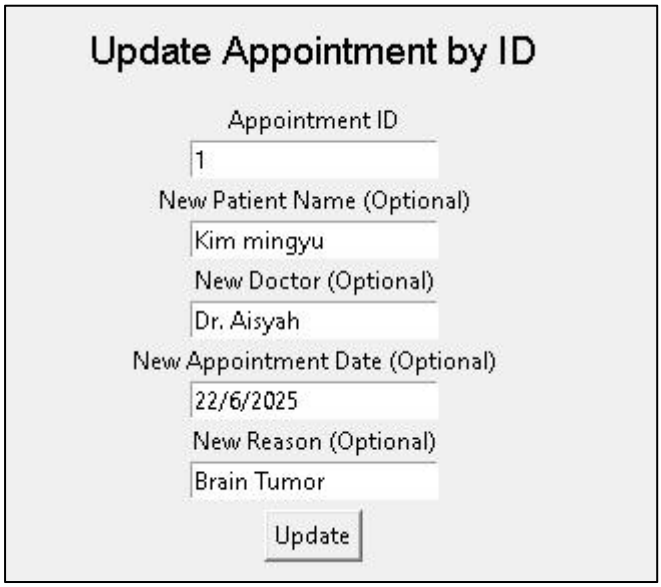


7.2 View Appointment

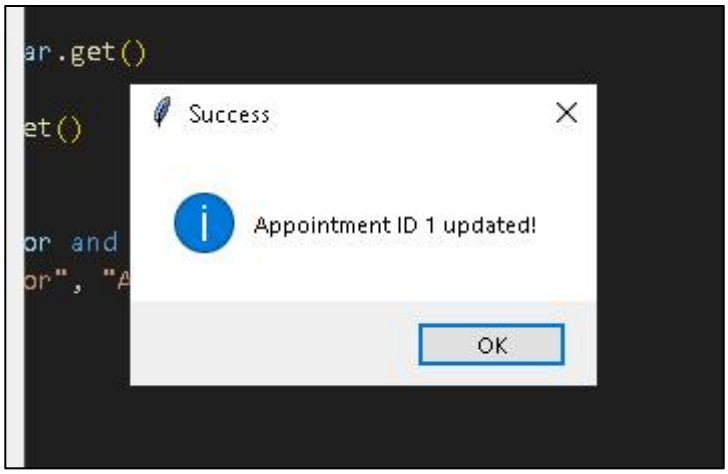
An appointment ID is automatically assigned upon the creation of an appointment, starting sequentially from the number 1.



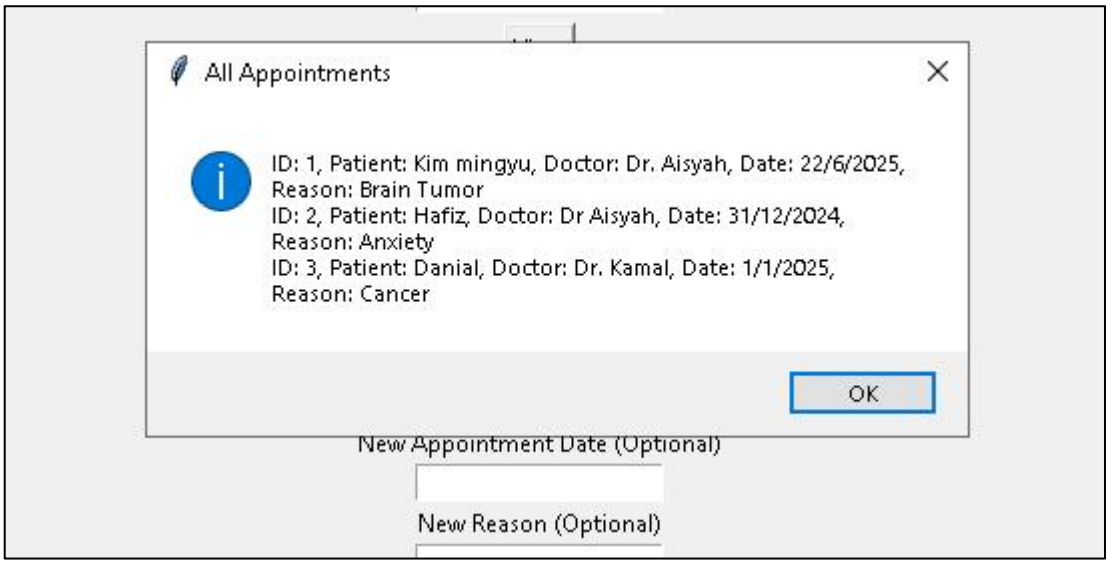
7.3 Update Appointment



You can update all details related to your appointment, including assigning a new doctor, by providing your appointment ID.



7.4 List All Appointment



You can view a list of all appointments, including their scheduled dates. Appointments will remain in the system unless you choose to delete them.

7.5 Delete Appointment

You can delete an appointment by providing the corresponding appointment ID.

Strengths:

1. Clear Functionality:

Each function has a clear purpose: create, view, update, delete, and list appointments. This makes the code easy to understand and maintain.

2. User-Friendly GUI:

The use of Tkinter provides a simple graphical interface that is intuitive for users to navigate. Labels, buttons, and input fields are arranged logically.

3. Well-Organized Structure:

Functions are modular and separated for specific tasks, adhering to good coding practices.

4. Update Functionality:

Adding the update function shows completeness and ensures users can modify appointments as needed.

5. Dynamic IDs:

Appointment IDs are automatically assigned using the length of the appointments list, simplifying record identification.

Kaizen:

1. Improved UI Layout:

The UI layout can become cluttered with more appointments.

2. Update Logic:

When updating an appointment, it should display the existing details in the entry fields for editing.

3. Scalability:

Currently, appointments are stored in a simple list, which works fine for small data. For larger systems, consider integrating a database.

4. Implement Data Persistence

The current system does not save appointments when the program is closed.