



COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

UNIVERSITI TEKNOLOGI MARA (UiTM),

CAWANGAN KEDAH, KAMPUS SUNGAI PETANI

DIPLOMA IN INFORMATICS LIBRARY

(CDIM144)

PROGRAMMING FOR LIBRARIES

(IML208)

INDIVIDUAL ASSIGNMENT: DENTAL CLINIC APPPOINTMENT SYSTEM

PREPARED BY:

SITI HADIRAH BINTI NASIB

(2023664186)

CLASS: KCDIM1443B

PREPARED FOR:

MOHD FIRDAUS BIN MOHD HELMI

SUBMISSION DATE:

WEEK 10

INDIVIDUAL ASSIGNMENT: DENTAL CLINIC APPPOINTMENT SYSTEM

PREPARED BY:

SITI HADIRAH BINTI NASIB

(2023664186)

CLASS: KCDIM1443B

CDIM144 – DIPLOMA IN INFORMATICS LIBRARY

ACKNOWLEDGEMENT

Assalamualaikum w.b.t,

first and foremost I would like to say thanks for Allah because of His Grace the journey of finishing this assignment run smoothly.

Next, I would say thank you to miss Zuraidah binti Arif for gave the tips on finding the information, gave us the guideline on how to organize this assignment in perfect format and the advice to improve the assignment. I don't know how to start this assignment without her guidance. I also won't forget to be thankful for having a supporting family. My parents always pray for my success and won't ever give up to ask about my health. They always take care of me even though I am far away from home. My love from them never be less day by day.

Other than that, I would say thanks for my classmates for kindly helping me qlarify any doubts or misunderstood throught the chat group or in the conversation. Last but not least, I would say thank you for myself because in short period of time I can finished this assignment. The improvement or mistakes that I made hopefully would be the lessons for the next time.

Project Name: Dental Clinic Appointment

File name: dentalappointment.py

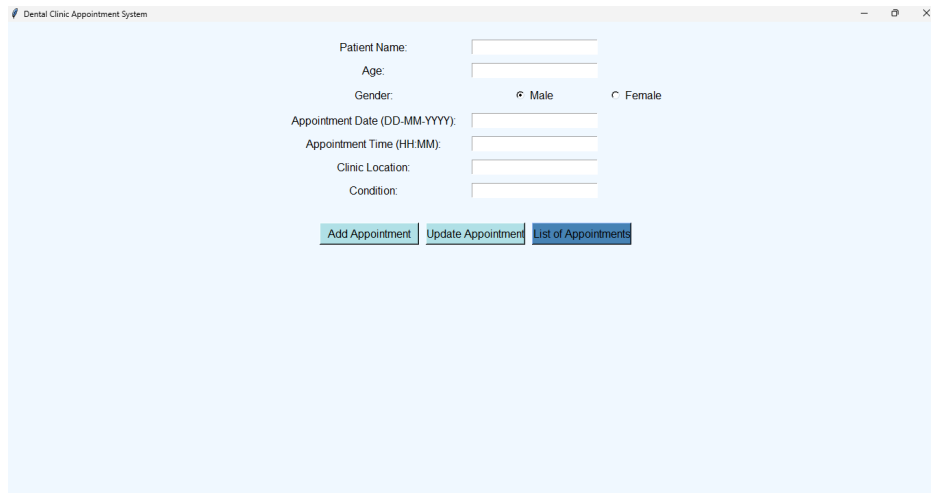
Prompt Data:

- i. Name
- ii. Age
- iii. Gender (Male/Female)
- iv. Phone number
- v. Appointment date
- vi. Appointment time
- vii. Clinic location
- viii. Condition

Function :

i. Create data

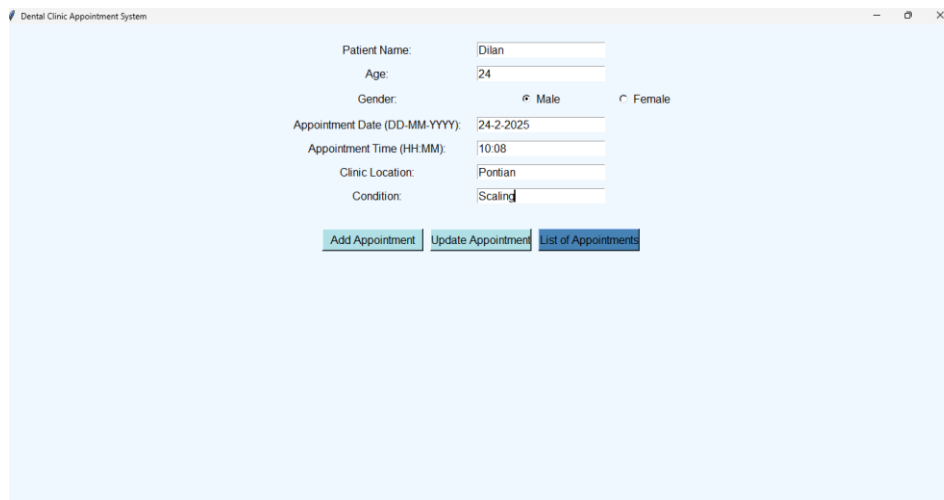
The first step is let's start with open the GUI.



The screenshot shows the 'Dental Clinic Appointment System' window. It contains a form with the following fields and controls:

- Patient Name:
- Age:
- Gender: ☒ Male ☐ Female
- Appointment Date (DD-MM-YYYY):
- Appointment Time (HH:MM):
- Clinic Location:
- Condition:
- Buttons:

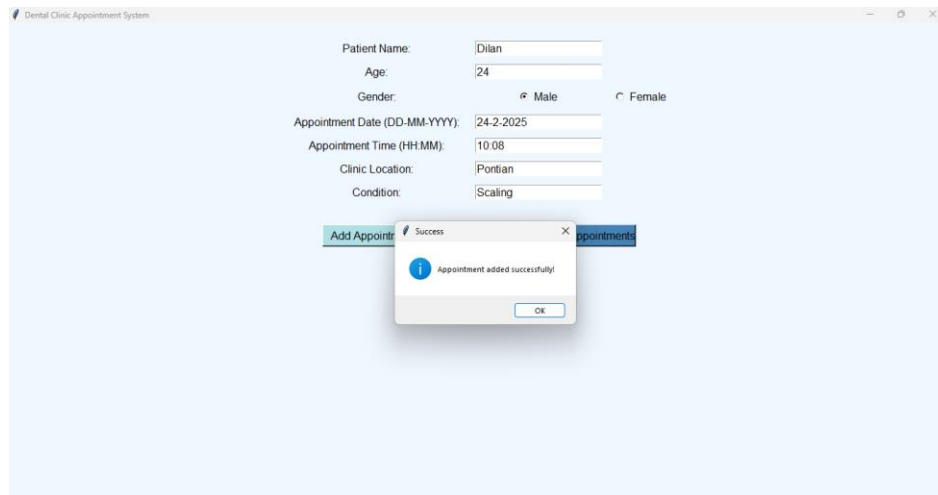
Then, insert your data on the system and click “Add Appointment” button.



The screenshot shows the same 'Dental Clinic Appointment System' window, but now with data entered into the form fields:

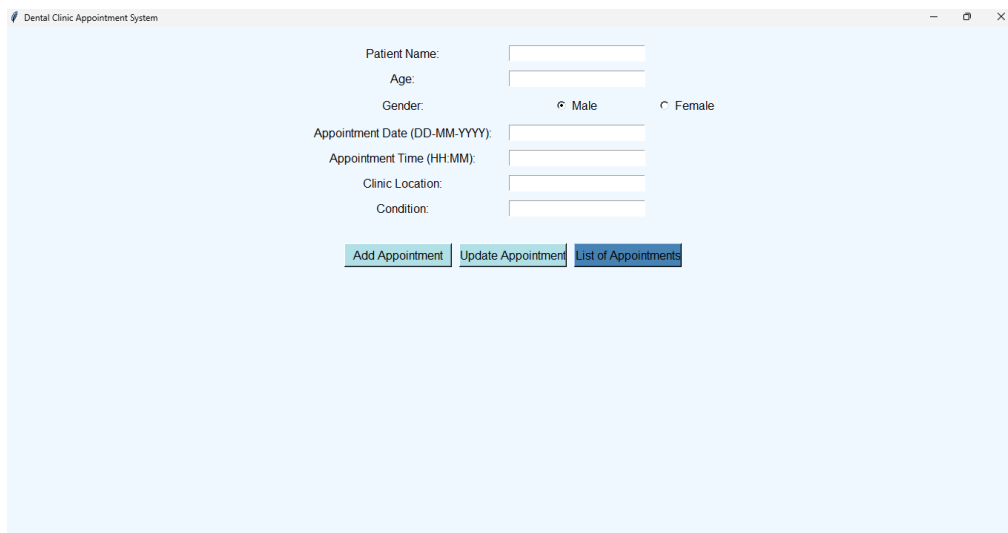
- Patient Name:
- Age:
- Gender: ☒ Male ☐ Female
- Appointment Date (DD-MM-YYYY):
- Appointment Time (HH:MM):
- Clinic Location:
- Condition:
- Buttons:

The result will be like this. Click “Ok” and the system will automatically restart into normal form.



ii. Read data

Go and click “List of Appointments” button.



This is how user can read their appointment schedule.

ID	Patient Name	Age	Gender	Appointment Date	Appointment Time	Clinic Location	Condition
1	Kai	22	Male	2-1-2025	9:00	Kajang	Scaling
2	Manja	13	Female	31-12-2024	10:40	Shah Alam	Braces
3	Aira	34	Female	3-2-2025	10:12	Merbok	wisdom tooth
4	Yeonjun	26	Female	3-3-2025	9:25	Muar	Tooth extraction
5	Dilan	24	Male	24-2-2025	10:08	Pontian	Scaling

[Load for Update](#)
[Cancel Appointment](#)

iii. Update data

First, click on “Update Appointment” button.

Dental Clinic Appointment System

Patient Name:

Age:

Gender:
☒ Male
☐ Female

Appointment Date (DD-MM-YYYY):

Appointment Time (HH:MM):

Clinic Location:

Condition:

[Add Appointment](#)
[Update Appointment](#)
[List of Appointments](#)

Second, choose the data you want to change and click “Load for Update” button.

List of Appointments

ID	Patient Name	Age	Gender	Appointment Date	Appointment Time	Clinic Location	Condition
1	Kai	22	Male	2-1-2025	9:00	Kajang	Scaling
2	Manja	13	Female	31-12-2024	10:40	Shah Alam	Braces
3	Aira	34	Female	3-2-2025	10:12	Merbok	wisdom tooth
4	Yeonjun	26	Female	3-3-2025	9:25	Muar	Tooth extraction
5	Dilan	24	Male	24-2-2025	10:08	Pontian	Scaling

Load for Update

Cancel Appointment

It will appear like this. Now, we need to change the gender from Female to Male.

Before:

Dental Clinic Appointment System

Patient Name:

Yeonjun

Age:

26

Gender:

Male

Female

Appointment Date (DD-MM-YYYY):

3-3-2025

Appointment Time (HH:MM):

9:25

Clinic Location:

Muar

Condition:

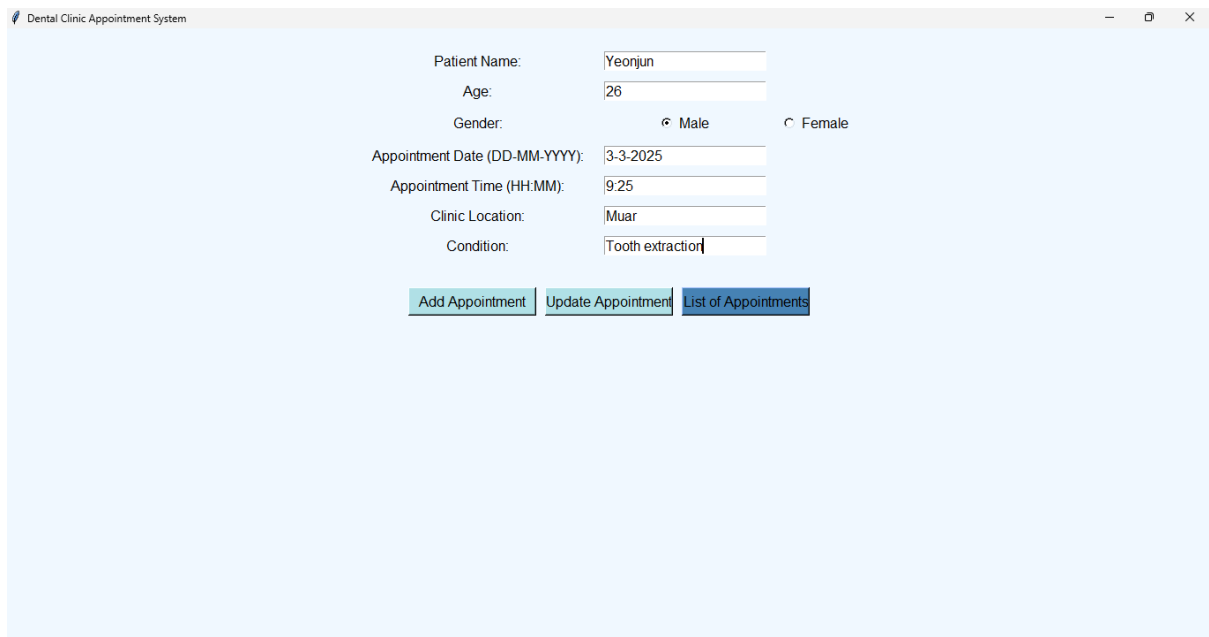
Tooth extraction

Add Appointment

Update Appointment

List of Appointments

After:



Dental Clinic Appointment System

Patient Name: Yeonjun

Age: 26

Gender: ☒ Male ☐ Female

Appointment Date (DD-MM-YYYY): 3-3-2025

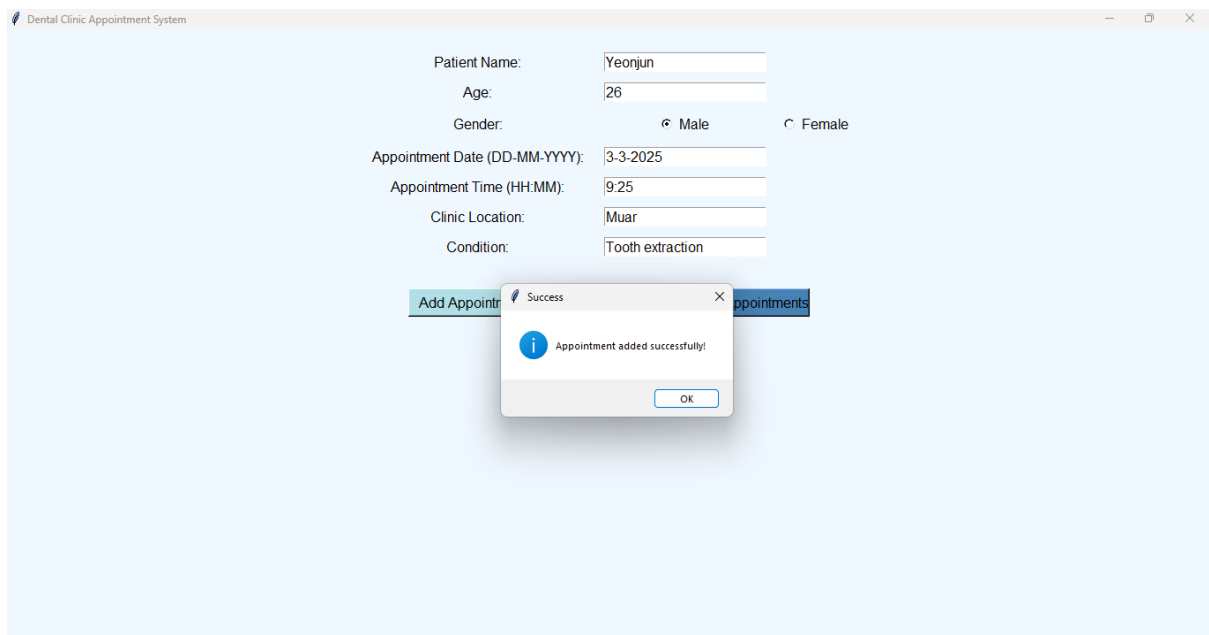
Appointment Time (HH:MM): 9:25

Clinic Location: Muar

Condition: Tooth extraction

[Add Appointment](#) [Update Appointment](#) [List of Appointments](#)

After change the data, click “Add Appointment” and it will appear like this:



Dental Clinic Appointment System

Patient Name: Yeonjun

Age: 26

Gender: ☒ Male ☐ Female

Appointment Date (DD-MM-YYYY): 3-3-2025

Appointment Time (HH:MM): 9:25

Clinic Location: Muar

Condition: Tooth extraction

[Add Appointment](#) [Update Appointment](#) [List of Appointments](#)

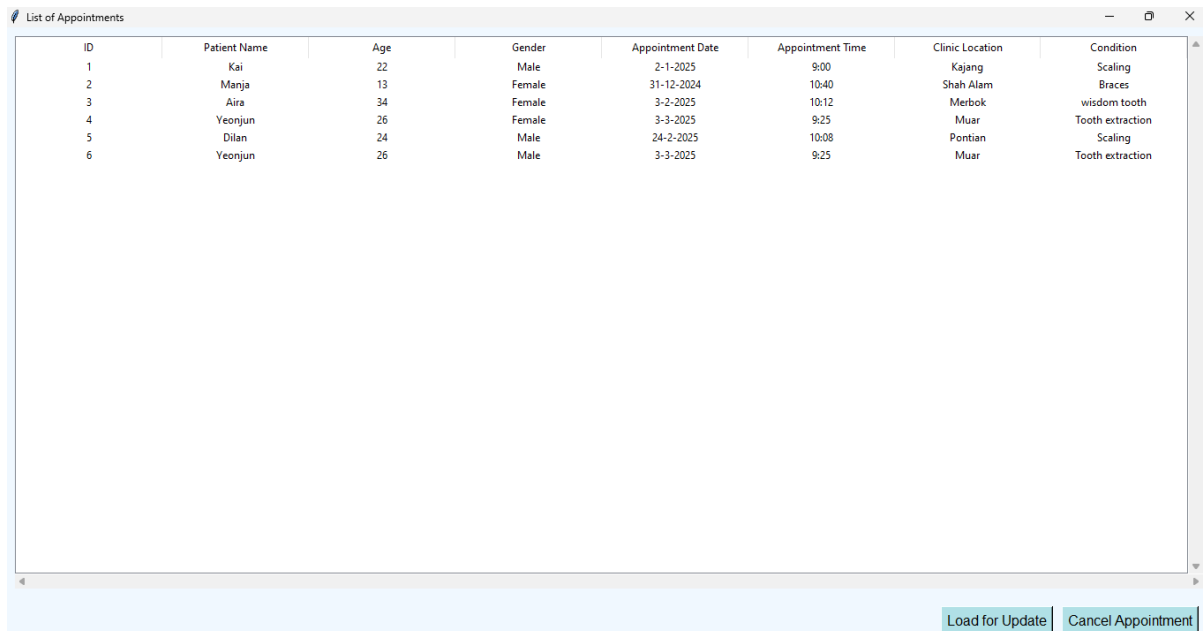
Success

Appointment added successfully!

OK

iv. Delete existing data

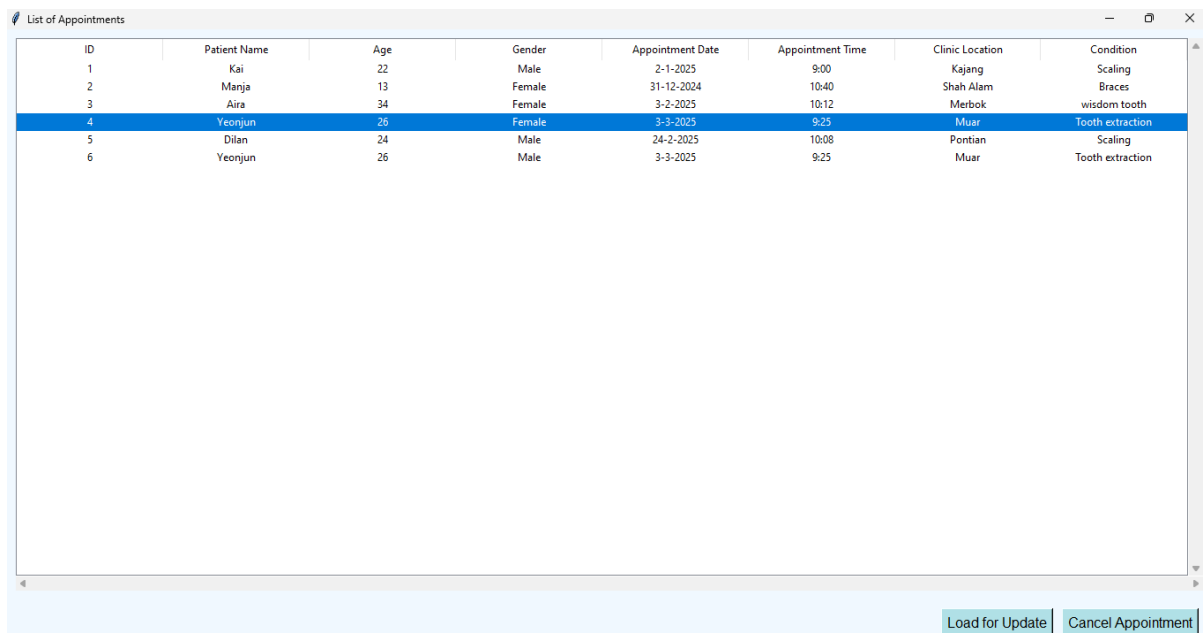
After updating the list, it turned like this:



ID	Patient Name	Age	Gender	Appointment Date	Appointment Time	Clinic Location	Condition
1	Kai	22	Male	2-1-2025	9:00	Kajang	Scaling
2	Manja	13	Female	31-12-2024	10:40	Shah Alam	Braces
3	Aira	34	Female	3-2-2025	10:12	Merbok	wisdom tooth
4	Yeonjun	26	Female	3-3-2025	9:25	Muar	Tooth extraction
5	Dilan	24	Male	24-2-2025	10:08	Pontian	Scaling
6	Yeonjun	26	Male	3-3-2025	9:25	Muar	Tooth extraction

Load for Update Cancel Appointment

All you have to do is choose the data you need to remove and click on the “Cancel Appointment”.



ID	Patient Name	Age	Gender	Appointment Date	Appointment Time	Clinic Location	Condition
1	Kai	22	Male	2-1-2025	9:00	Kajang	Scaling
2	Manja	13	Female	31-12-2024	10:40	Shah Alam	Braces
3	Aira	34	Female	3-2-2025	10:12	Merbok	wisdom tooth
4	Yeonjun	26	Female	3-3-2025	9:25	Muar	Tooth extraction
5	Dilan	24	Male	24-2-2025	10:08	Pontian	Scaling
6	Yeonjun	26	Male	3-3-2025	9:25	Muar	Tooth extraction

Load for Update Cancel Appointment

The result:

Dental Clinic Appointment System

Patient Name:

Age:

Gender: ☒ Male ☐ Female

Appointment Date (DD-MM-YYYY):

Appointment Time (HH:MM):

Clinic Location:

Condition:

[Add Appointment](#) [Cancel Appointment](#)

Success

i Appointment canceled successfully!

[OK](#)

Conditional Statement : Yes

1) if

```
16
17     if name and age and gender and date and time and location and condition:
18         appointment = {
19             "id": len(appointments) + 1,
20             "patient_name": name,
21             "age": age,
22             "gender": gender,
23             "appointment_date": date,
24             "appointment_time": time,
25             "clinic_location": location,
26             "condition": condition
27         }
```

```
28     if not selected_item:
29         messagebox.showerror('Error', 'Please select an appointment to update.')
30     return
```

```
34     if appointment["id"] == selected_id:
35         entry_name.delete(0, tk.END)
36         entry_name.insert(0, appointment["patient_name"])
37         entry_age.delete(0, tk.END)
38         entry_age.insert(0, appointment["age"])
39         gender_var.set(appointment["gender"])
40         entry_date.delete(0, tk.END)
41         entry_date.insert(0, appointment["appointment_date"])
42         entry_time.delete(0, tk.END)
43         entry_time.insert(0, appointment["appointment_time"])
44         entry_location.delete(0, tk.END)
45         entry_location.insert(0, appointment["clinic_location"])
46         entry_condition.delete(0, tk.END)
47         entry_condition.insert(0, appointment["condition"])
48         appointments_window.destroy()
49     return
```

```
53     if not selected_item:
54         messagebox.showerror('Error', 'Please select an appointment to delete.')
55     return
```

2) else

```
31     else:
32         messagebox.showerror('Error', 'Please fill all fields.')
33
```

GUI : Yes (add screenshot)

localhost:50498/Homepage/appointment.aspx

RUSH BRUSH DENTAL CLINIC

Clinic Timings

Morning : 9am-2pm Evening : 5pm-7pm

Appointment Form

Date and Time not available

OK

Name: patient1

Contact: 8978627221

Address: 283/2 New Prem Nagar Karnal

Date: 2021/05/14

Time: 10:00

SUBMIT

SCO 2196 , MUGHAL CANAL , ROHINI , NEW DELHI
9034258XXX 7973806XXX

Activate Windows
Go to Settings to activate Windows.

Result:

Python code

```
1 import tkinter as tk
2 from tkinter import ttk
3 from tkinter import messagebox
4
5 appointments = []
6
7 # Add new appointment (Create)
8 def add_appointment():
9     name = entry_name.get()
10    age = entry_age.get()
11    gender = gender_var.get()
12    date = entry_date.get()
13    time = entry_time.get()
14    location = entry_location.get()
15    condition = entry_condition.get()
16
17    if name and age and gender and date and time and location and condition:
18        appointment = {
19            "id": len(appointments) + 1,
20            "patient_name": name,
21            "age": age,
22            "gender": gender,
23            "appointment_date": date,
24            "appointment_time": time,
25            "clinic_location": location,
26            "condition": condition
27        }
28        appointments.append(appointment)
29        messagebox.showinfo('Success', 'Appointment added successfully!')
30        clear_entries()
31    else:
32        messagebox.showerror('Error', 'Please fill all fields..')
33
34 # Show appointments in a separate GUI
35 def show_appointments_window():
36     def load_to_update():
37         selected_item = tree.selection()
38         if not selected_item:
39             messagebox.showerror('Error', 'Please select an appointment to update.')
40             return
41         selected_id = int(tree.item(selected_item, 'values')[0])
42         for appointment in appointments:
43             if appointment['id'] == selected_id:
44                 entry_name.delete(0, tk.END)
45                 entry_name.insert(0, appointment['patient_name'])
46                 entry_age.delete(0, tk.END)
```

```

47         entry_age.delete(0, tk.END)
48         entry_age.insert(0, appointment["age"])
49         gender_var.set(appointment["gender"])
50         entry_date.delete(0, tk.END)
51         entry_date.insert(0, appointment["appointment_date"])
52         entry_time.delete(0, tk.END)
53         entry_time.insert(0, appointment["appointment_time"])
54         entry_location.delete(0, tk.END)
55         entry_location.insert(0, appointment["clinic_location"])
56         entry_condition.delete(0, tk.END)
57         entry_condition.insert(0, appointment["condition"])
58         appointments_window.destroy()
59         return
60
61 def delete_appointment():
62     selected_item = tree.selection()
63     if not selected_item:
64         messagebox.showerror('Error', 'Please select an appointment to delete.')
65         return
66
67     selected_id = int(tree.item(selected_item, 'values')[0])
68     global appointments
69     appointments = [appointment for appointment in appointments if appointment["id"] != selected_id]
70     messagebox.showinfo('Success', 'Appointment canceled successfully!')
71     refresh_table()
72
73 def refresh_table():
74     for row in tree.get_children():
75         tree.delete(row)
76     for appointment in appointments:
77         tree.insert('', tk.END, values=(
78             appointment["id"],
79             appointment["patient_name"],
80             appointment["age"],
81             appointment["gender"],
82             appointment["appointment_date"],
83             appointment["appointment_time"],
84             appointment["clinic_location"],
85             appointment["condition"]
86         ))
87
88 appointments_window = tk.Toplevel(root)
89 appointments_window.title("List of Appointments")
90 appointments_window.geometry("880x500")
91 appointments_window.configure(bg="#f0f8ff")

```

```

93 # Table Frame
94 table_frame = tk.Frame(appointments_window, bg="#f0f8ff")
95 table_frame.pack(fill=tk.BOTH, expand=True, padx=10, pady=10)
96
97 columns = ('ID', 'Patient Name', 'Age', 'Gender', 'Appointment Date', 'Appointment Time', 'Clinic Location', 'Condition')
98 tree = ttk.Treeview(table_frame, columns=columns, show='headings', height=15)
99
100 for col in columns:
101     tree.heading(col, text=col)
102     tree.column(col, anchor=tk.CENTER, width=100)
103
104 scrollbar_x = ttk.Scrollbar(table_frame, orient=tk.HORIZONTAL, command=tree.xview)
105 scrollbar_y = ttk.Scrollbar(table_frame, orient=tk.VERTICAL, command=tree.yview)
106 tree.config(scrollcommand=scrollbar_x.set, yscrollcommand=scrollbar_y.set)
107 scrollbar_x.pack(side=tk.BOTTOM, fill=tk.X)
108 scrollbar_y.pack(side=tk.RIGHT, fill=tk.Y)
109 tree.pack(fill=tk.BOTH, expand=True)
110
111 # Populate the Treeview
112 refresh_table()
113
114 # Buttons
115 button_frame = tk.Frame(appointments_window, bg="#f0f8ff")
116 button_frame.pack(fill=tk.X, padx=10, pady=10)
117
118 btn_delete = tk.Button(button_frame, text="Cancel Appointment", command=delete_appointment, bg="#b0e0e6", font=("Arial", 12))
119 btn_delete.pack(side=tk.RIGHT, padx=5)
120
121 btn_load = tk.Button(button_frame, text="Load for Update", command=load_to_update, bg="#b0e0e6", font=("Arial", 12))
122 btn_load.pack(side=tk.RIGHT, padx=5)
123
124 # Clear input fields
125 def clear_entries():
126     entry_name.delete(0, tk.END)
127     entry_age.delete(0, tk.END)
128     entry_date.delete(0, tk.END)
129     entry_time.delete(0, tk.END)
130     entry_location.delete(0, tk.END)
131     entry_condition.delete(0, tk.END)
132
133 # Main application window
134 root = tk.Tk()
135 root.title("Dental Clinic Appointment System")
136 root.geometry("600x500")
137 root.configure(bg="#f0f8ff")
138

```

```

139 # Input Fields Frame
140 input_frame = tk.Frame(root, bg="#f0f8ff")
141 input_frame.pack(pady=20)
142
143 label_name = tk.Label(input_frame, text="Patient Name:", bg="#f0f8ff", font=("Arial", 12))
144 label_name.grid(row=0, column=0, padx=10, pady=5)
145 entry_name = tk.Entry(input_frame, font=("Arial", 12))
146 entry_name.grid(row=0, column=1, padx=10, pady=5)
147
148 label_age = tk.Label(input_frame, text="Age:", bg="#f0f8ff", font=("Arial", 12))
149 label_age.grid(row=1, column=0, padx=10, pady=5)
150 entry_age = tk.Entry(input_frame, font=("Arial", 12))
151 entry_age.grid(row=1, column=1, padx=10, pady=5)
152
153 label_gender = tk.Label(input_frame, text="Gender:", bg="#f0f8ff", font=("Arial", 12))
154 label_gender.grid(row=2, column=0, padx=10, pady=5)
155 gender_var = tk.StringVar()
156 gender_male = tk.Radiobutton(input_frame, text="Male", variable=gender_var, value="Male", bg="#f0f8ff", font=("Arial", 12))
157 gender_male.grid(row=2, column=1, padx=5, pady=5)
158 gender_female = tk.Radiobutton(input_frame, text="Female", variable=gender_var, value="Female", bg="#f0f8ff", font=("Arial", 12))
159 gender_female.grid(row=2, column=2, padx=5, pady=5)
160
161 label_date = tk.Label(input_frame, text="Appointment Date (DD-MM-YYYY):", bg="#f0f8ff", font=("Arial", 12))
162 label_date.grid(row=3, column=0, padx=10, pady=5)
163 entry_date = tk.Entry(input_frame, font=("Arial", 12))
164 entry_date.grid(row=3, column=1, padx=10, pady=5)
165
166 label_time = tk.Label(input_frame, text="Appointment Time (HH:MM):", bg="#f0f8ff", font=("Arial", 12))
167 label_time.grid(row=4, column=0, padx=10, pady=5)
168 entry_time = tk.Entry(input_frame, font=("Arial", 12))
169 entry_time.grid(row=4, column=1, padx=10, pady=5)
170
171 label_location = tk.Label(input_frame, text="Clinic Location:", bg="#f0f8ff", font=("Arial", 12))
172 label_location.grid(row=5, column=0, padx=10, pady=5)
173 entry_location = tk.Entry(input_frame, font=("Arial", 12))
174 entry_location.grid(row=5, column=1, padx=10, pady=5)
175
176 label_condition = tk.Label(input_frame, text="Condition:", bg="#f0f8ff", font=("Arial", 12))
177 label_condition.grid(row=6, column=0, padx=10, pady=5)
178 entry_condition = tk.Entry(input_frame, font=("Arial", 12))
179 entry_condition.grid(row=6, column=1, padx=10, pady=5)
180

```

```

181 # Buttons
182 btn_frame = tk.Frame(root, bg="#f0f8ff")
183 btn_frame.pack(pady=10)
184
185 btn_add = tk.Button(btn_frame, text="Add Appointment", command=add_appointment, bg="#b0e0e6", font=("Arial", 12), width=15)
186 btn_add.pack(side=tk.LEFT, padx=5)
187
188 btn_update = tk.Button(btn_frame, text="Update Appointment", command=lambda: show_appointments_window(), bg="#b0e0e6", font=("Arial", 12), width=15)
189 btn_update.pack(side=tk.LEFT, padx=5)
190
191 btn_list = tk.Button(btn_frame, text="List of Appointments", command=show_appointments_window, bg="#4682b4", font=("Arial", 12), width=15)
192 btn_list.pack(side=tk.LEFT, padx=5)
193
194 # Run the application
195 root.mainloop()

```

GUI for Dental Clinic Appointment System

Dental Clinic Appointment System

Patient Name:

Age:

Gender: ☒ Male ☐ Female

Appointment Date (DD-MM-YYYY):

Appointment Time (HH:MM):

Clinic Location:

Condition:

Final result:

List of Appointments

ID	Patient Name	Age	Gender	Appointment Date	Appointment Time	Clinic Location	Condition
1	Kai	22	Male	2-1-2025	9:00	Kajang	Scaling
2	Manja	13	Female	31-12-2024	10:40	Shah Alam	Braces
3	Aira	34	Female	3-2-2025	10:12	Merbok	wisdom tooth
5	Dilan	24	Male	24-2-2025	10:08	Pontian	Scaling
6	Yeonjun	26	Male	3-3-2025	9:25	Muar	Tooth extraction

Strength:

1. Patients would be alert to the schedule
2. It is user-friendly for anyone who needs to make an appointment
3. Save patients' times from making an appointment directly to the clinic
4. Patient can choose to change or update their data

Kaizen (Room for improvement)

1. This GUI is too basic and did not have any character or emoji that make it interesting.
2. After updating the data into new, it should be add instead of automatically replace the old data.
3. Need insert detail information such as patient's address to sent letter or any important reason. It also need to add patient's phone number and email-address to be extra careful for their health.