



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 ASSIGNMENT: SPORT CENTRE BOOKING SYSTEM

PREPARED BY:

NAME	STUDENT ID
SITI NUR AMELIA BINTI MOHD KHALID RASIDI	2023855608

CLASS: KCDIM144 3B

PREPARED FOR:

EN. MOHD FIRDAUS BIN MOHD HELMI

SUBMISSION DATE:
18th December 2024

INDIVIDUAL ASSIGNMENT: SPORT CENTRE BOOKING SYSTEM

SITI NUR AMELIA BINTI MOHD KHALID RASIDI

2023855608

KCDIM144 3B

UNIVERSITI TEKNOLOGI MARA (UiTM) KEDAH, KAMPUS SUNGAI PETANI

SCHOOL OF INFORMATION SCIENCE

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

DIPLOMA IN LIBRARY INFORMATICS

18th December 2024

ACKNOWLEDGEMENT

Assalamualaikum w.b.t,

Alhamdulillah, I would like to thank God for giving me the strength to complete the assignment smoothly. And because of His grace, I was encouraged to continue the assignment until I succeed.

I, as a part of the students from CDIM144 Library Informatics, want to convey my sincere gratitude to En. Mohd Firdaus Bin Mohd Helmi, my lecturer, for his entire attention and infinite help in completing this assignment. Without his guidance, I could not even finish this assignment completely all by myself.

Apart from that, I would like to thank all my beloved classmates for helping me without any hesitation throughout this assignment. Their help in answering the questions in the group or private chat truly gives clarity to all my doubts and issues.

Finally, I would like to thank my family for their continuous support until I accomplish this assignment. With their support and motivation, I was determined to not give up halfway and finish this assignment easily. Also, all that is involved directly or indirectly in this assignment.

X 

NAME: SITI NUR AMELIA BINTI MOHD KHALID RASIDI

STUDENT ID: 2023855608

CLASS: KCDIM144 3B

STUDENT PLEDGE



As a student of Universiti Teknologi MARA (UiTM), it is my responsibility to act in accordance with UiTM's academic assessment and evaluation policy. I hereby pledge to act and uphold academic integrity and pursue scholarly activities in UiTM with honesty and responsible manner. I will not engage or tolerate acts of academic dishonesty, academic misconduct, or academic fraud including but not limited to:

- a. **Cheating:** Using or attempt to use any unauthorized device, assistance, sources, practice, or materials while completing academic assessments. This include but not limited to copying from another, allowing others to copy, unauthorized collaboration on an assignment or open book tests, or engaging in any act or conduct that can be construed as cheating.
- b. **Plagiarism:** Using or attempts to use the work of others (ideas, design, words, art, music, etc.) without acknowledging the source; using or purchasing materials prepared by another person or agency or engaging in other behaviour that a reasonable person would consider as plagiarism.
- c. **Fabrication:** Falsifying data, information, or citations in any academic assessment and evaluation.
- d. **Deception:** Providing false information with intend to deceive an instructor concerning any academic assessment and evaluation.
- e. **Furnishing false information:** Providing false information or false representation to any UiTM official, instructor, or office.

With this pledge, I am fully aware that I am obliged to conduct myself with utmost honesty and integrity. I fully understand that a disciplinary action can be taken against me if I, in any manner, violate this pledge.

Name: SITI NUR AMELIA BINTI MOHD KHALID RASIDI

Matric Number: 2023855608

Course Code: IML208

Programme code: KCDIM144.

Faculty / Campus: SCHOOL OF INFORMATION SCIENCE

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS, KAMPUS SUNGAI PETANI.

*Students are required to sign one pledge for each course taken.

Bahagian Pentaksiran dan Penilaian Akademik 2021

TABLE OF CONTENT

CONTENT	PAGE
ACKNOWLEDGEMENT	iii
STUDENT PLEDGE	iv
TABLE OF CONTENT	v
PROMPT DATA AND FUNCTION	1 - 2
CONDITIONAL STATEMENT	3
RESULT	4 - 6
STRENGTH AND KAIZEN (ROOM FOR IMPROVEMENT)	6
REFERENCE	

Project Name: Sport Centre Booking System

File Name: Sport.py

Prompt Data:

- i. Name
- ii. Phone Number
- iii. Date

Function:

- i. Create Data

```
◆ Sport_GUI.py > ...
1  import tkinter as tk
2  from tkinter import messagebox
3  from datetime import datetime
4
5  # Global variable for keeping track of booking
6  bookings = [] # List to store bookings
7
8  # Function to create a new booking
9  def create_booking():
10     name = name_var.get()
11     phone = phone_var.get()
12     booking_date = booking_date_var.get()
13     sport = sport_var.get()
14     court = court_var.get()
15     time_slot = time_slot_var.get()
16
17     # Validate that all fields are filled
18     if not name or not phone or not booking_date or not sport or not court or not time_slot:
19         messagebox.showerror("Error", "All fields are required!")
20         return
21
22     # Add the booking to the list
23     booking = {
24         "name": name,
25         "phone": phone,
26         "booking_date": booking_date,
27         "sport": sport,
28         "court": f"Court {court}",
29         "time": time_slot
30     }
31     bookings.append(booking)
32     messagebox.showinfo("Success", f"Your {sport} session at {court} has been booked at {time_slot}.")
33
```

- ii. Read Data

```
33
34 # Function to view all bookings (Read)
35 def view_bookings():
36     if not bookings:
37         messagebox.showinfo("No Bookings", "No bookings found.")
38         return
39
40     bookings_text = "Current Bookings:\n"
41     for i, booking in enumerate(bookings, 1):
42         bookings_text += f"{i}. {booking['sport']} - {booking['court']} at {booking['time']}\n"
43         bookings_text += f"    Name: {booking['name']}, Phone: {booking['phone']}, Date: {booking['booking_date']}\n"
44
45     messagebox.showinfo("Bookings", bookings_text)
46
```

iii. Update Data

```
46
47 # Function to update a booking
48 def update_booking():
49     view_bookings() # View bookings before updating
50
51     booking_number = int(booking_number_var.get()) - 1
52     if 0 <= booking_number < len(bookings):
53         # Get the selected booking
54         booking = bookings[booking_number]
55
56         # Update details
57         new_sport = sport_var.get()
58         new_court = court_var.get()
59         new_time_slot = time_slot_var.get()
60
61         if new_sport:
62             booking["sport"] = new_sport
63         if new_court:
64             booking["court"] = f"Court {new_court}"
65         if new_time_slot:
66             booking["time"] = new_time_slot
67
68         messagebox.showinfo("Success", "Booking updated successfully!")
69     else:
70         messagebox.showerror("Error", "Invalid booking number!")
71
```

iv. Delete Existing Data

```
71
72 # Function to delete a booking
73 def delete_booking():
74     view_bookings() # View bookings before deleting
75
76     booking_number = int(booking_number_var.get()) - 1
77     if 0 <= booking_number < len(bookings):
78         del bookings[booking_number]
79         messagebox.showinfo("Success", "Booking deleted successfully!")
80     else:
81         messagebox.showerror("Error", "Invalid booking number!")
82
```

Conditional Statement: Yes

```
16
17     # Validate that all fields are filled
18     if not name or not phone or not booking_date or not sport or not court or not time_slot:
19         messagebox.showerror("Error", "All fields are required!")
20         return
21
22     # Add the booking to the list
23     booking = {}
24     "name": name,
25     "phone": phone,
26     "booking_date": booking_date,
27     "sport": sport,
28     "court": f"Court {court}",
29     "time": time_slot
30
31     bookings.append(booking)
32     messagebox.showinfo("Success", f"Your {sport} session at {court} has been booked at {time_slot}.")
33
34     # Function to view all bookings (Read)
35     def view_bookings():
36         if not bookings:
37             messagebox.showinfo("No Bookings", "No bookings found.")
38             return
39
40         bookings_text = "Current Bookings:\n"
41         for i, booking in enumerate(bookings, 1):
42             bookings_text += f"{i}. {booking['sport']} - {booking['court']} at {booking['time']}\n"
43             bookings_text += f"    Name: {booking['name']}, Phone: {booking['phone']}, Date: {booking['booking_date']}\n"
44
```

```
46
47     # Function to update a booking
48     def update_booking():
49         view_bookings() # View bookings before updating
50
51         booking_number = int(booking_number_var.get()) - 1
52         if 0 <= booking_number < len(bookings):
53             # Get the selected booking
54             booking = bookings[booking_number]
55
56             # Update details
57             new_sport = sport_var.get()
58             new_court = court_var.get()
59             new_time_slot = time_slot_var.get()
60
61             if new_sport:
62                 booking["sport"] = new_sport
63             if new_court:
64                 booking["court"] = f"Court {new_court}"
65             if new_time_slot:
66                 booking["time"] = new_time_slot
67
68             messagebox.showinfo("Success", "Booking updated successfully!")
69         else:
70             messagebox.showerror("Error", "Invalid booking number!")
71
72     # Function to delete a booking
73     def delete_booking():
74         view_bookings() # View bookings before deleting
75
76         booking_number = int(booking_number_var.get()) - 1
77         if 0 <= booking_number < len(bookings):
78             del bookings[booking_number]
79             messagebox.showinfo("Success", "Booking deleted successfully!")
80         else:
81             messagebox.showerror("Error", "Invalid booking number!")
82
```


GUI: Yes

A screenshot of a GUI form for creating a booking. The form is titled "Enter Your Name" and contains several input fields and buttons. The fields are: "Enter Phone Number", "Enter Date of Booking (YYYY-MM-DD)", "Select Sport", "Select Court", "Select Time Slot", "Create Booking", "View Bookings", "Enter Booking Number to Update/Delete", "Update Booking", and "Delete Booking".

Result:

i. Create data

A screenshot of the same GUI form as above, but with a success message overlay. The message is: "Success Your Badminton session at 2 has been booked at 9:00 AM - 11:00 AM." The message is displayed in a small window with an "OK" button. The form fields are filled with: "Amelia" for Name, "011-23456789" for Phone Number, "2024-12-30" for Date of Booking, "Badminton" for Sport, and "2" for Court. The "Create Booking" button is highlighted.

ii. Read data

Enter Your Name

Amelia

Enter Phone Number

011-23456789

Enter Date of Booking (YYYY-MM-DD)

2024-12-30

Select Sport

Badminton

Bookings

Current Bookings:
1. Badminton - Court 2 at 9:00 AM - 11:00 AM
Name: Amelia, Phone: 011-23456789, Date: 2024-12-30

OK

View Bookings

Enter Booking Number to Update/Delete

Update Booking

Delete Booking

iii. Update data

Enter Your Name

Amelia

Enter Phone Number

011-23456789

Enter Date of Booking (YYYY-MM-DD)

2024-12-30

Select Sport

Badminton

Bookings

Current Bookings:
1. Badminton - Court 2 at 9:00 AM - 11:00 AM
Name: Amelia, Phone: 011-23456789, Date: 2024-12-30

OK

View Bookings

Enter Booking Number to Update/Delete

Update Booking

Delete Booking

Enter Your Name

Amelia

Enter Phone Number

011-23456789

Enter Date of Booking (YYYY-MM-DD)

2024-12-30

Select Sport

Badminton

Bookings

Current Bookings:
1. Badminton - Court 3 at 11:30 AM - 2:30 PM
Name: Amelia, Phone: 011-23456789, Date: 2024-12-30

OK

View Bookings

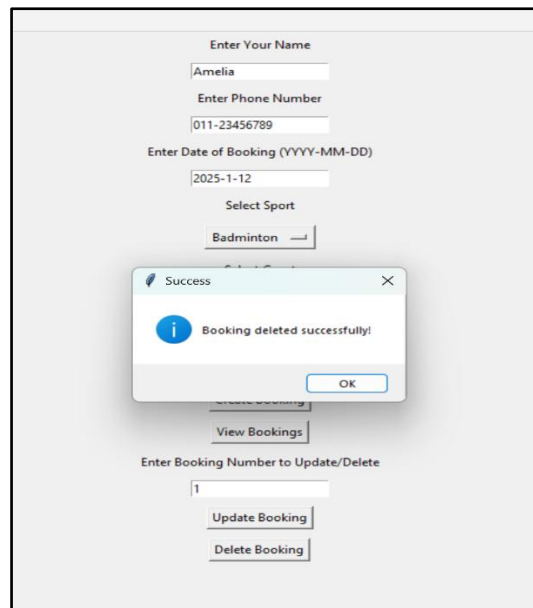
Enter Booking Number to Update/Delete

1

Update Booking

Delete Booking

iv. Delete data



The screenshot displays a web application interface for managing bookings. At the top, there are input fields for 'Enter Your Name' (containing 'Amelia'), 'Enter Phone Number' (containing '011-23456789'), and 'Enter Date of Booking (YYYY-MM-DD)' (containing '2025-1-12'). Below these is a 'Select Sport' dropdown menu currently set to 'Badminton'. A modal window titled 'Success' is centered on the screen, displaying a blue information icon and the text 'Booking deleted successfully!'. An 'OK' button is located at the bottom right of the modal. In the background, partially obscured by the modal, are buttons for 'View Bookings', 'Update Booking', and 'Delete Booking'. At the bottom of the form, there is an input field labeled 'Enter Booking Number to Update/Delete' with the number '1' entered, and buttons for 'Update Booking' and 'Delete Booking'.

Strength:

1. Increase staff productivity.
2. Flexibility in modification:
 - i. Booking updates: The update functionality allows workers to quickly adjust booking details.
 - ii. Delete booking: The delete function enable workers to remove outdated or erroneous booking, keeping the system clean and up to date.
3. Real-time information:
 - i. Instant feedback: whenever a user makes a booking, the system immediately provides a confirmation with details.
 - ii. Booking overview: The ability to view all bookings at once gives workers a snapshot of current activity.

Kaizen (Room for Improvement):

1. Improving booking duration handling:
 - i. Variable durations: Allow users to select variable booking durations instead of just entering number manually.
 - ii. Multiple session: Allow users to book multiple sessions at once or recurring sessions.
2. Customizable time slot:
 - i. Custom time slots: Consider allowing sports centre to define their own time slots, rather than hardcoding them into the system. This would make the system more flexible and adaptable to different scheduling requirements.
3. User feedback and ratings:
 - i. Booking suggestions: Based on past bookings, the system could recommend sports, courts, or times for the user to make future bookings easier.

REFERENCE

- GeeksforGeeks. (2021, November 12). *Movie tickets Booking management system in Python*. GeeksforGeeks. <https://www.geeksforgeeks.org/movie-tickets-booking-management-system-in-python/>
- GeeksforGeeks. (2022, December 9). *CRUD Operation in Python using MySQL*. GeeksforGeeks. <https://www.geeksforgeeks.org/crud-operation-in-python-using-mysql/>
- GeeksforGeeks. (2024a, June 19). *Python Tkinter*. GeeksforGeeks. <https://www.geeksforgeeks.org/python-gui-tkinter/>