UNIVERSITI TEKNOLOGI MARA

KEDAH BRANCH

SCHOOL OF INFORMATION
SCIENCE

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

DIPLOMA IN INFORMATION MANAGEMENT (IM144)

IML208: PROGRAMMING FOR LIBRARIES

GROUP PROJECT:

HALL BOOKING SYSTEM

Prepared by:
MUHAMMAD NAZHAN SYAHMI
(2022261582)

MUHAMMAD IZZAT SYAHMI BIN MOHAMMAD NIZAM
(2022661584)

LUZMAN AL-THAQIF BIN NARZARUDDIN
(2022860858)

GROUP KIM1443E.

Prepared for:
SIR AIRUL SHAZWAN BIN
NORSHAHIMI

Submission date:

16th January 2024

TITLE OF ASSIGNMENT:

GROUP PROJECT: HALL BOOKING

PREPARED BY:

MUHAMMAD NAZHAN SYAHMI BIN SAIFUL RIDZUAN
(2022261582)
MUHAMMAD IZZAT SYAHMI BIN MOHAMMAD NIZAM
(2022661584)
LUZMAN AL-THAQIF BIN NARZARUDDIN
(2022860858)

GROUP KIM1443E

IM144 – DIPLOMA IN LIBRARY INFORMATION

SCHOOL OF INFORMATION SCIENCE

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

UNIVESITI TEKNOLOGI MARA (UITM)

KEDAH BRANCH

Acknowledgement

Assalamualaikum W.B.T

We would like to convey our sincere gratitude to our lecturer SIR AIRUL SHAZWAN BIN NORSHAHIMI for his guidance and for helping us in completing our assignment. This assignment is understandable because of his help. As students we are still learning. While completing this assignment, we learnt something new and valuable.

Other than that, we would like to thank all our friends for inviting us to do this assignment together and showing us how it is done. Their help in giving clarity to doubt by answering all of our questions, help us with finishing our assignment. Finally, we also would like to thank our family for their support in accomplishing this assignment. Their support gave us motivation to not give up and charge through the wall.

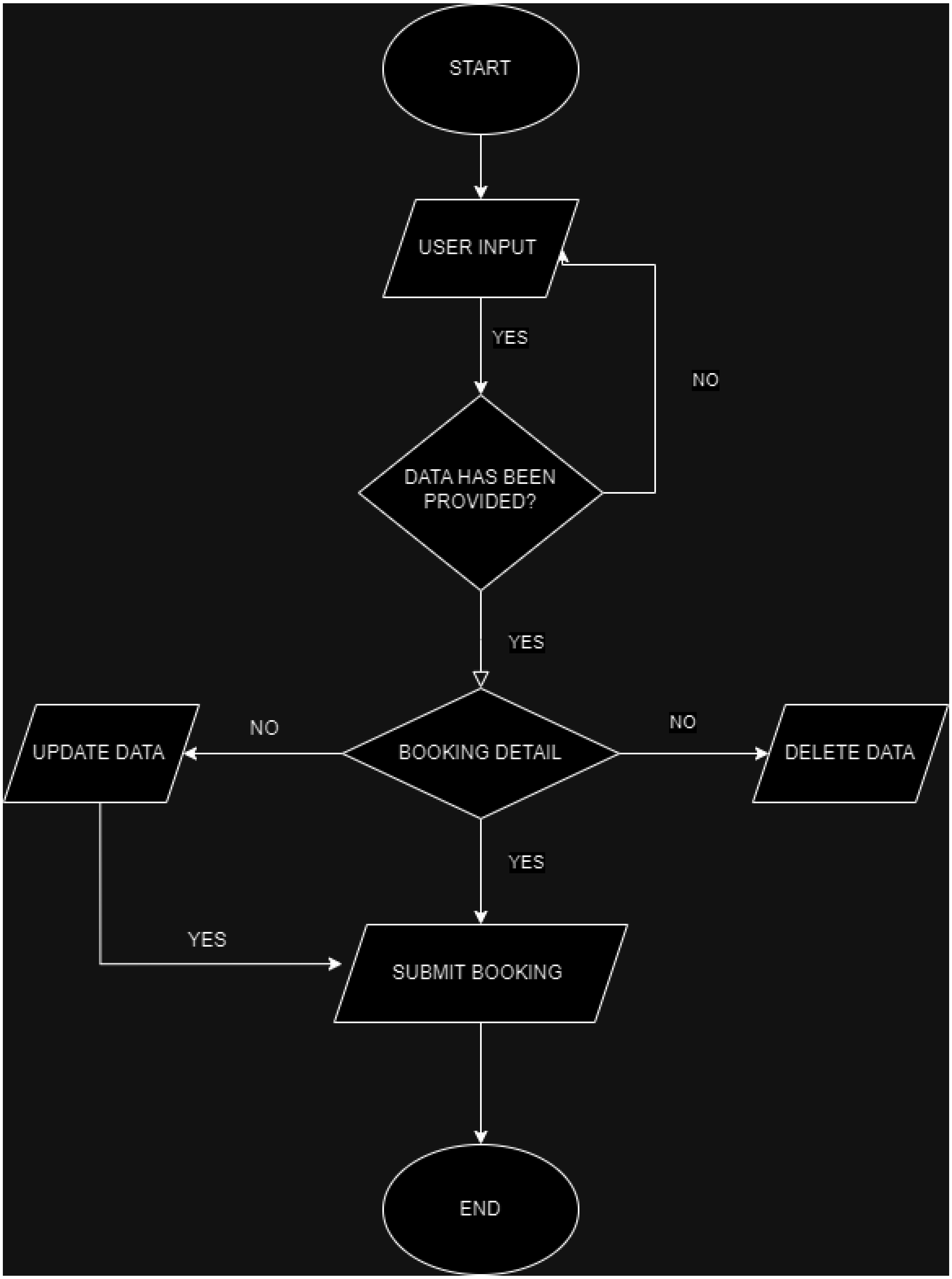
TABLE OF CONTENT

1.0 Introduction

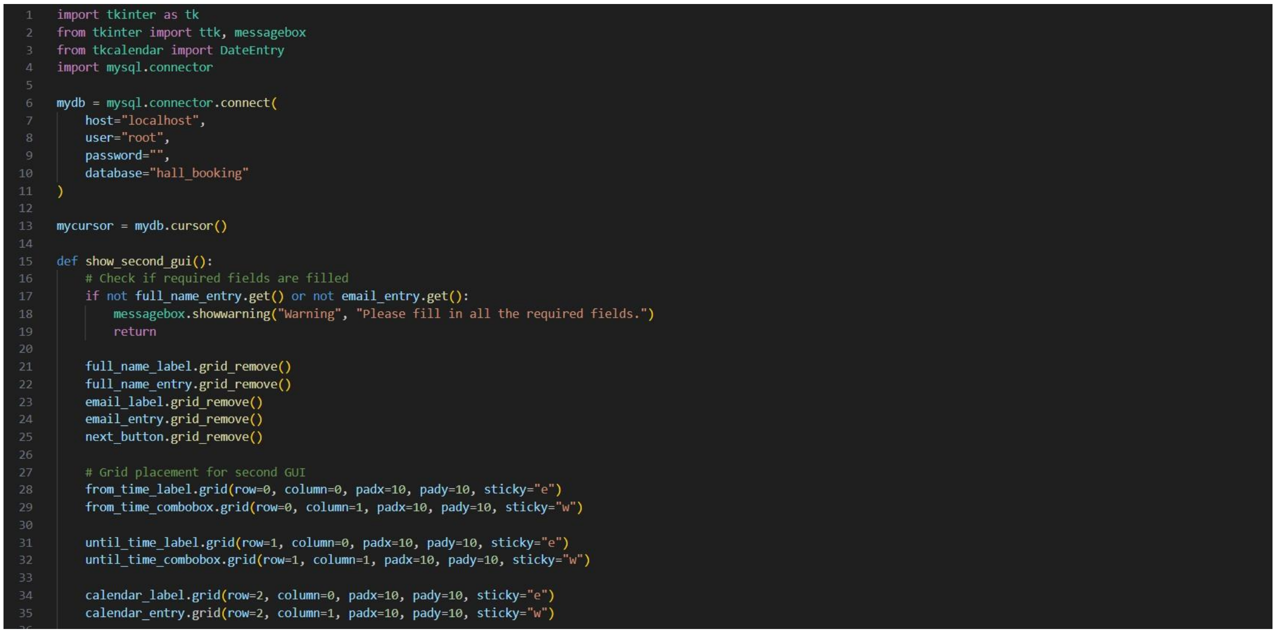
In this final project, our focus is directed towards the complex landscape of event management, where we discuss the details of creating and implementing a strong hall booking system using Tkinter in Python. The goal is to create a graphical user interface (GUI) that is easy to use and simple so that people may schedule event venues with ease. Important information is recorded by the system, such as the user's complete name, email address, wanted hall, date, and event time. Our code ensures a responsive and well-rounded user experience by smoothly integrating features for submission, changing current bookings, and efficiently deleting reservations, all without compromising on surface aesthetics.

This assignment presents a chance to improve problem-solving abilities, user interface design expertise, and general programming knowledge as we set out on our adventure. Our hall booking system not only meets but beyond the expectations of users involved in event planning by striking a balance between aesthetic appeal and practical effectiveness. This shows the potential of programming in creating focused on users solutions for modern difficulties in event management. We hope that with this assignment, it will help students to get a better knowledge of software development in the context of event management by offering a hands-on experience that bridges the gap between theoretical programming principles and their actual implementation.

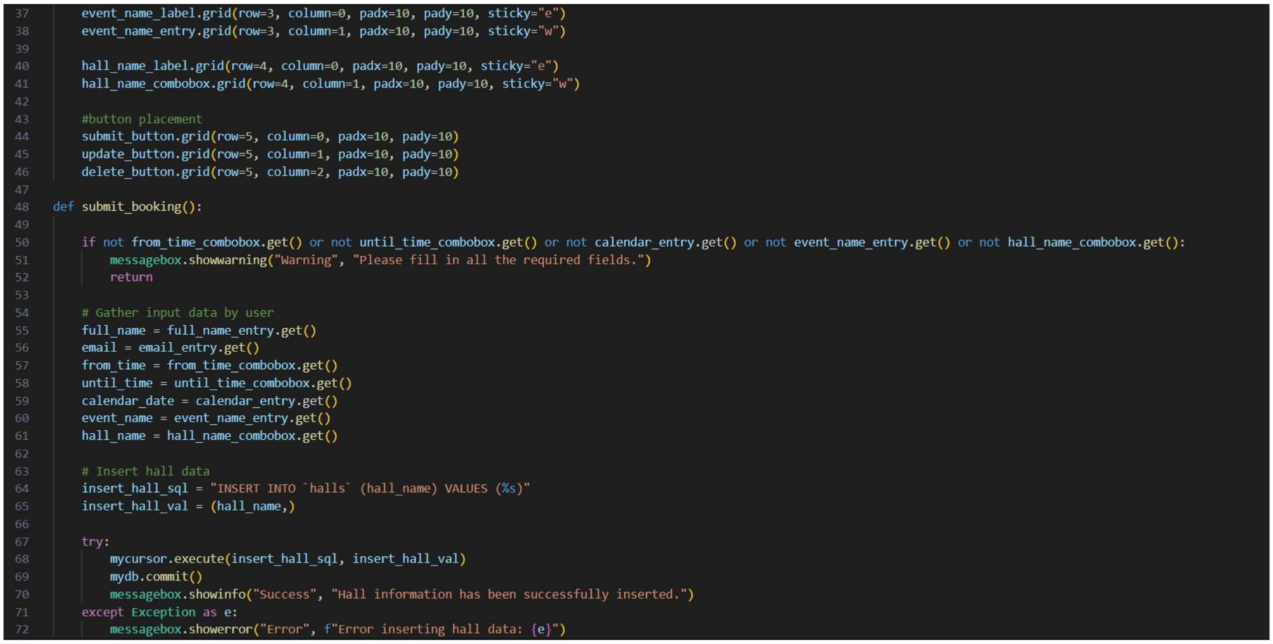
2.0 Flowchart



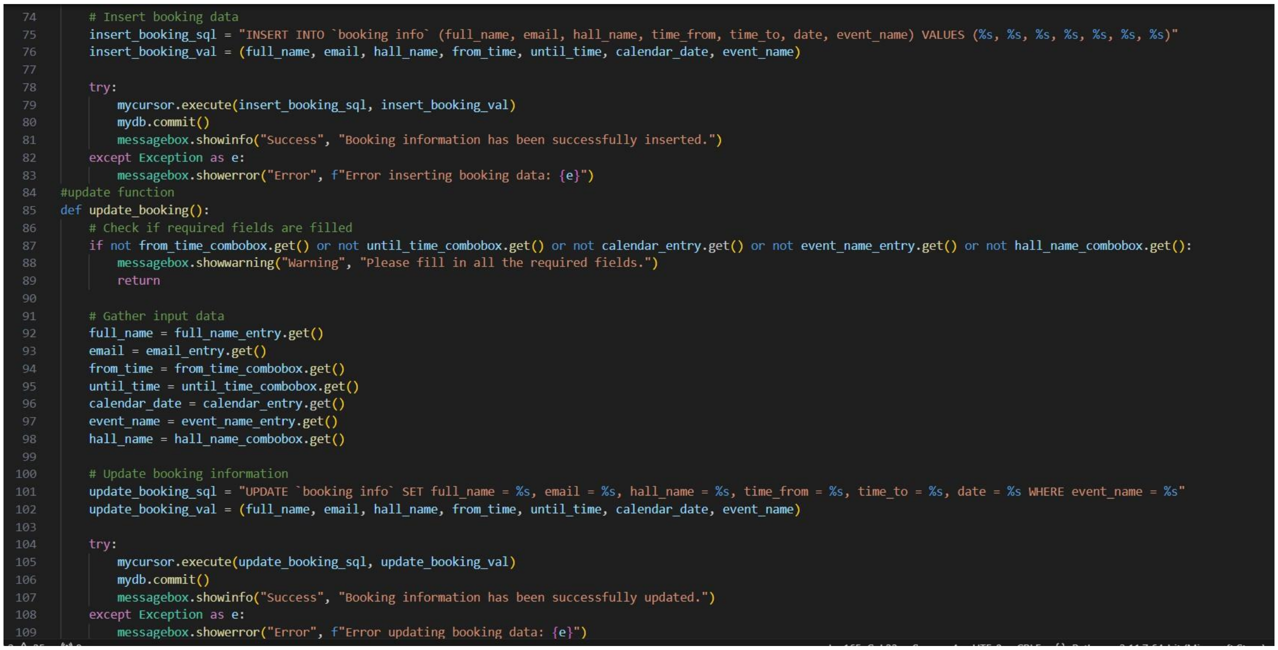
3.0 Snapshot of code



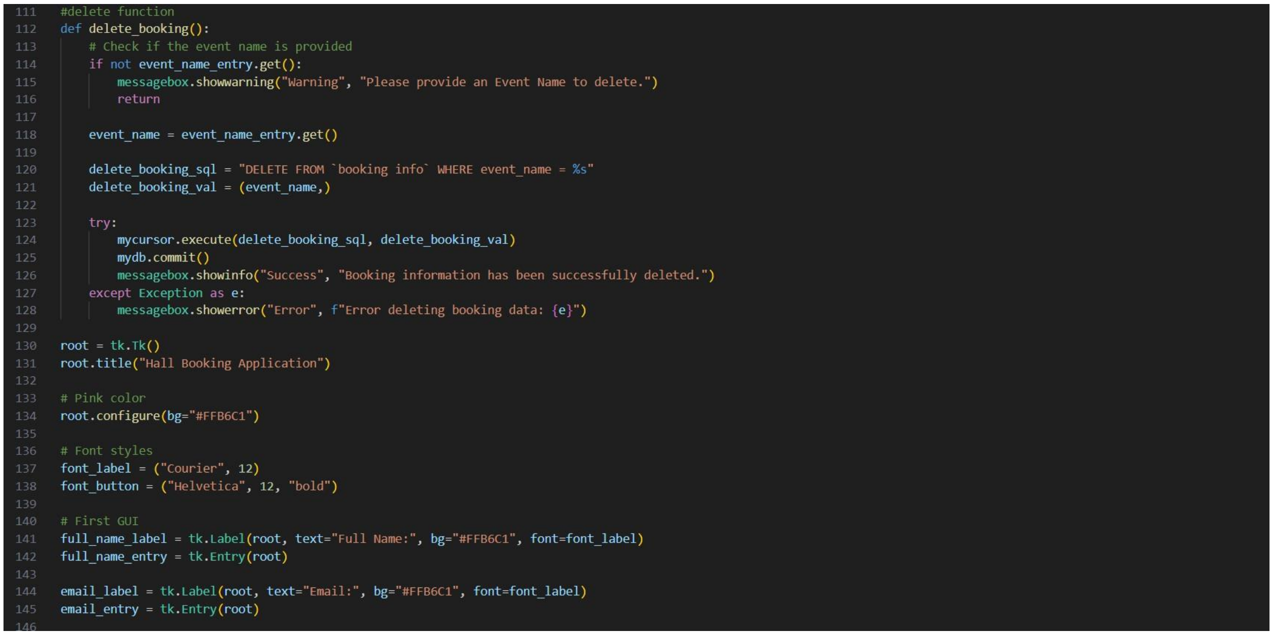
```python
import tkinter as tk
from tkinter import ttk, messagebox
from tkcalendar import DateEntry
import mysql.connector

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="hall_booking"
)

mycursor = mydb.cursor()

def show_second_gui():
    # Check if required fields are filled
    if not full_name_entry.get() or not email_entry.get():
        messagebox.showwarning("Warning", "Please fill in all the required fields.")
        return

    full_name_label.grid_remove()
    full_name_entry.grid_remove()
    email_label.grid_remove()
    email_entry.grid_remove()
    next_button.grid_remove()

    # Grid placement for second GUI
    from_time_label.grid(row=0, column=0, padx=10, pady=10, sticky="e")
    from_time_combobox.grid(row=0, column=1, padx=10, pady=10, sticky="w")

    until_time_label.grid(row=1, column=0, padx=10, pady=10, sticky="e")
    until_time_combobox.grid(row=1, column=1, padx=10, pady=10, sticky="w")

    calendar_label.grid(row=2, column=0, padx=10, pady=10, sticky="e")
    calendar_entry.grid(row=2, column=1, padx=10, pady=10, sticky="w")
```



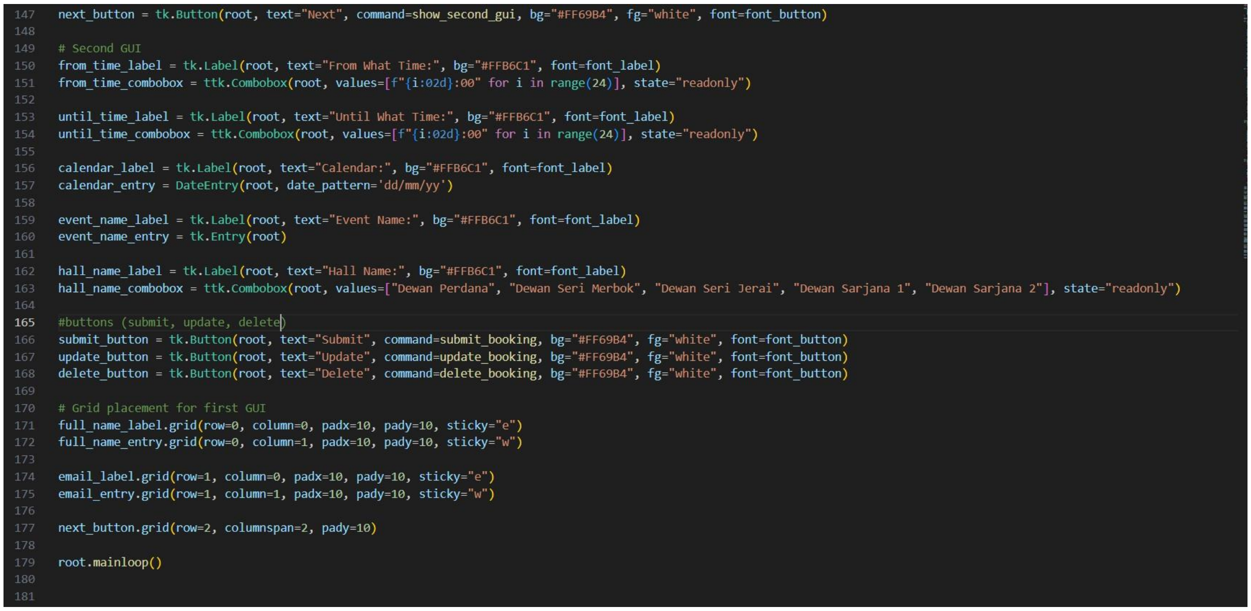
```python
    event_name_label.grid(row=3, column=0, padx=10, pady=10, sticky="e")
    event_name_entry.grid(row=3, column=1, padx=10, pady=10, sticky="w")

    hall_name_label.grid(row=4, column=0, padx=10, pady=10, sticky="e")
    hall_name_combobox.grid(row=4, column=1, padx=10, pady=10, sticky="w")

    #button placement
    submit_button.grid(row=5, column=0, padx=10, pady=10)
    update_button.grid(row=5, column=1, padx=10, pady=10)
    delete_button.grid(row=5, column=2, padx=10, pady=10)

def submit_booking():

    if not from_time_combobox.get() or not until_time_combobox.get() or not calendar_entry.get() or not event_name_entry.get() or not hall_name_combobox.get():
        messagebox.showwarning("Warning", "Please fill in all the required fields.")
        return

    # Gather input data by user
    full_name = full_name_entry.get()
    email = email_entry.get()
    from_time = from_time_combobox.get()
    until_time = until_time_combobox.get()
    calendar_date = calendar_entry.get()
    event_name = event_name_entry.get()
    hall_name = hall_name_combobox.get()

    # Insert hall data
    insert_hall_sql = "INSERT INTO `halls` (hall_name) VALUES (%s)"
    insert_hall_val = (hall_name,)

    try:
        mycursor.execute(insert_hall_sql, insert_hall_val)
        mydb.commit()
        messagebox.showinfo("Success", "Hall information has been successfully inserted.")
    except Exception as e:
        messagebox.showerror("Error", f"Error inserting hall data: {e}")
```



```python
    # Insert booking data
    insert_booking_sql = "INSERT INTO `booking info` (full_name, email, hall_name, time_from, time_to, date, event_name) VALUES (%s, %s, %s, %s, %s, %s, %s)"
    insert_booking_val = (full_name, email, hall_name, from_time, until_time, calendar_date, event_name)

    try:
        mycursor.execute(insert_booking_sql, insert_booking_val)
        mydb.commit()
        messagebox.showinfo("Success", "Booking information has been successfully inserted.")
    except Exception as e:
        messagebox.showerror("Error", f"Error inserting booking data: {e}")
#update function
def update_booking():
    # Check if required fields are filled
    if not from_time_combobox.get() or not until_time_combobox.get() or not calendar_entry.get() or not event_name_entry.get() or not hall_name_combobox.get():
        messagebox.showwarning("Warning", "Please fill in all the required fields.")
        return

    # Gather input data
    full_name = full_name_entry.get()
    email = email_entry.get()
    from_time = from_time_combobox.get()
    until_time = until_time_combobox.get()
    calendar_date = calendar_entry.get()
    event_name = event_name_entry.get()
    hall_name = hall_name_combobox.get()

    # Update booking information
    update_booking_sql = "UPDATE `booking info` SET full_name = %s, email = %s, hall_name = %s, time_from = %s, time_to = %s, date = %s WHERE event_name = %s"
    update_booking_val = (full_name, email, hall_name, from_time, until_time, calendar_date, event_name)

    try:
        mycursor.execute(update_booking_sql, update_booking_val)
        mydb.commit()
        messagebox.showinfo("Success", "Booking information has been successfully updated.")
    except Exception as e:
        messagebox.showerror("Error", f"Error updating booking data: {e}")
```

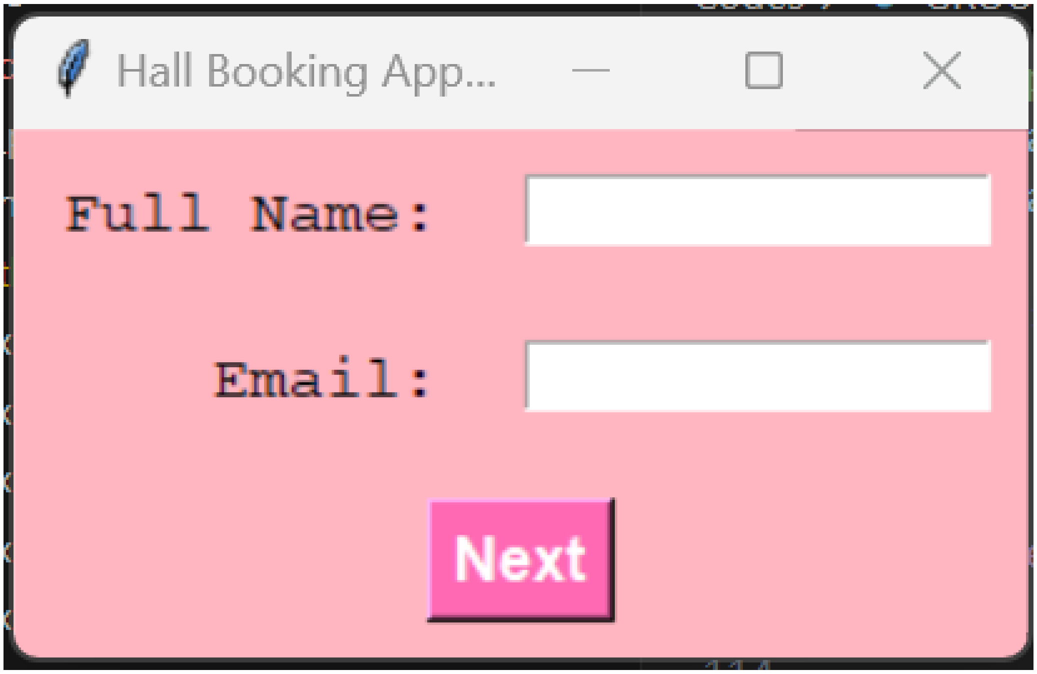


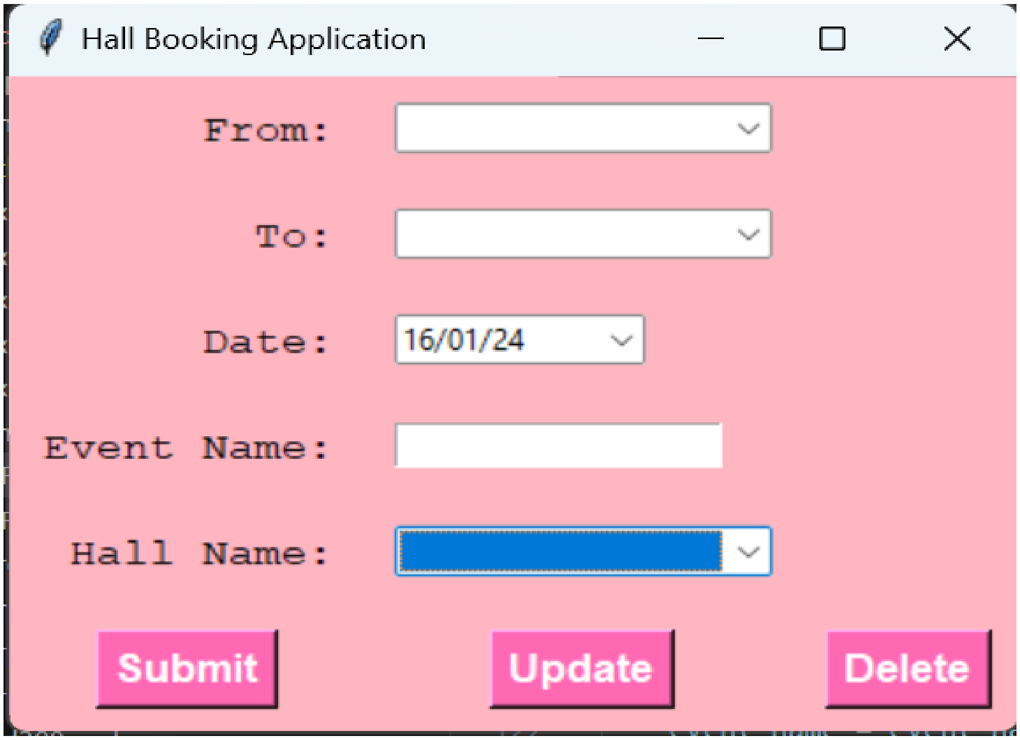
```python
#delete function
def delete_booking():
    # Check if the event name is provided
    if not event_name_entry.get():
        messagebox.showwarning("Warning", "Please provide an Event Name to delete.")
        return

    event_name = event_name_entry.get()

    delete_booking_sql = "DELETE FROM `booking info` WHERE event_name = %s"
    delete_booking_val = (event_name,)

    try:
        mycursor.execute(delete_booking_sql, delete_booking_val)
        mydb.commit()
        messagebox.showinfo("Success", "Booking information has been successfully deleted.")
    except Exception as e:
        messagebox.showerror("Error", f"Error deleting booking data: {e}")

root = tk.Tk()
root.title("Hall Booking Application")

# Pink color
root.configure(bg="#FFB6C1")

# Font styles
font_label = ("Courier", 12)
font_button = ("Helvetica", 12, "bold")

# First GUI
full_name_label = tk.Label(root, text="Full Name:", bg="#FFB6C1", font=font_label)
full_name_entry = tk.Entry(root)

email_label = tk.Label(root, text="Email:", bg="#FFB6C1", font=font_label)
email_entry = tk.Entry(root)
```



```python
next_button = tk.Button(root, text="Next", command=show_second_gui, bg="#FF69B4", fg="white", font=font_button)

# Second GUI
from_time_label = tk.Label(root, text="From What Time:", bg="#FFB6C1", font=font_label)
from_time_combobox = ttk.Combobox(root, values=[f"{i:02d}:00" for i in range(24)], state="readonly")

until_time_label = tk.Label(root, text="Until What Time:", bg="#FFB6C1", font=font_label)
until_time_combobox = ttk.Combobox(root, values=[f"{i:02d}:00" for i in range(24)], state="readonly")

calendar_label = tk.Label(root, text="Calendar:", bg="#FFB6C1", font=font_label)
calendar_entry = DateEntry(root, date_pattern='dd/mm/yy')

event_name_label = tk.Label(root, text="Event Name:", bg="#FFB6C1", font=font_label)
event_name_entry = tk.Entry(root)

hall_name_label = tk.Label(root, text="Hall Name:", bg="#FFB6C1", font=font_label)
hall_name_combobox = ttk.Combobox(root, values=["Dewan Perdana", "Dewan Seri Merbok", "Dewan Seri Jerai", "Dewan Sarjana 1", "Dewan Sarjana 2"], state="readonly")

#buttons (submit, update, delete)
submit_button = tk.Button(root, text="Submit", command=submit_booking, bg="#FF69B4", fg="white", font=font_button)
update_button = tk.Button(root, text="Update", command=update_booking, bg="#FF69B4", fg="white", font=font_button)
delete_button = tk.Button(root, text="Delete", command=delete_booking, bg="#FF69B4", fg="white", font=font_button)

# Grid placement for first GUI
full_name_label.grid(row=0, column=0, padx=10, pady=10, sticky="e")
full_name_entry.grid(row=0, column=1, padx=10, pady=10, sticky="w")

email_label.grid(row=1, column=0, padx=10, pady=10, sticky="e")
email_entry.grid(row=1, column=1, padx=10, pady=10, sticky="w")

next_button.grid(row=2, columnspan=2, pady=10)

root.mainloop()
```

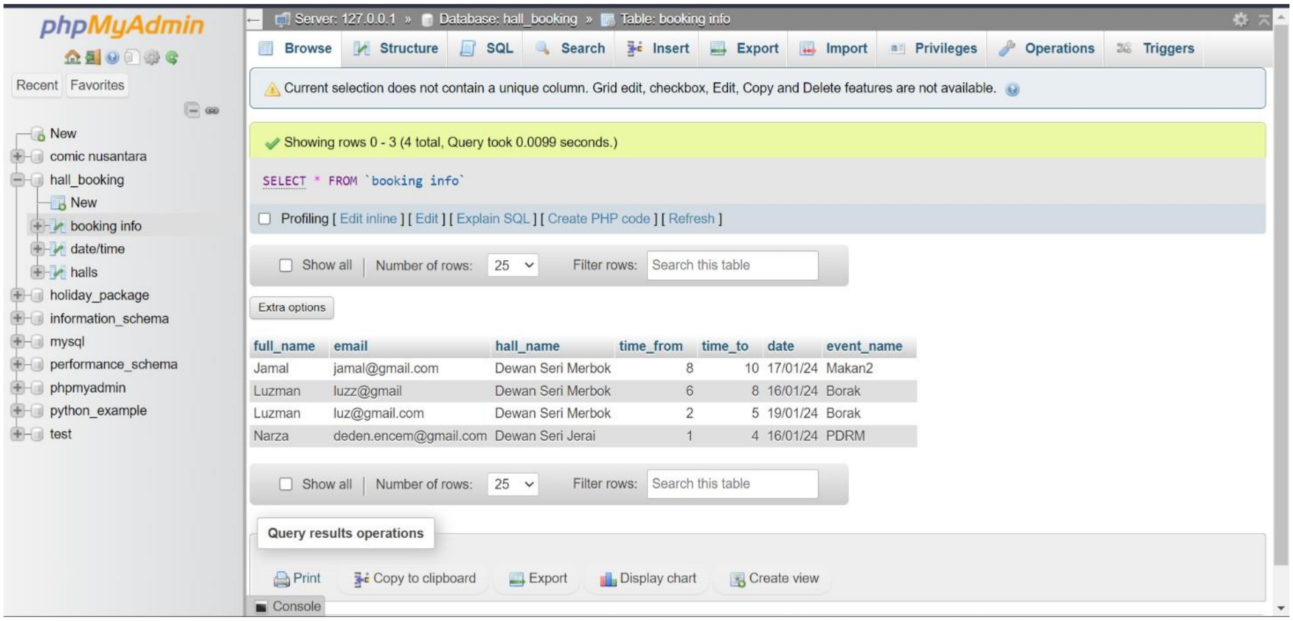
4.0 Snapshot of GUI





Date: 16/01/24

5.0 Snapshot of Database



Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - 3 (4 total, Query took 0.0099 seconds.)

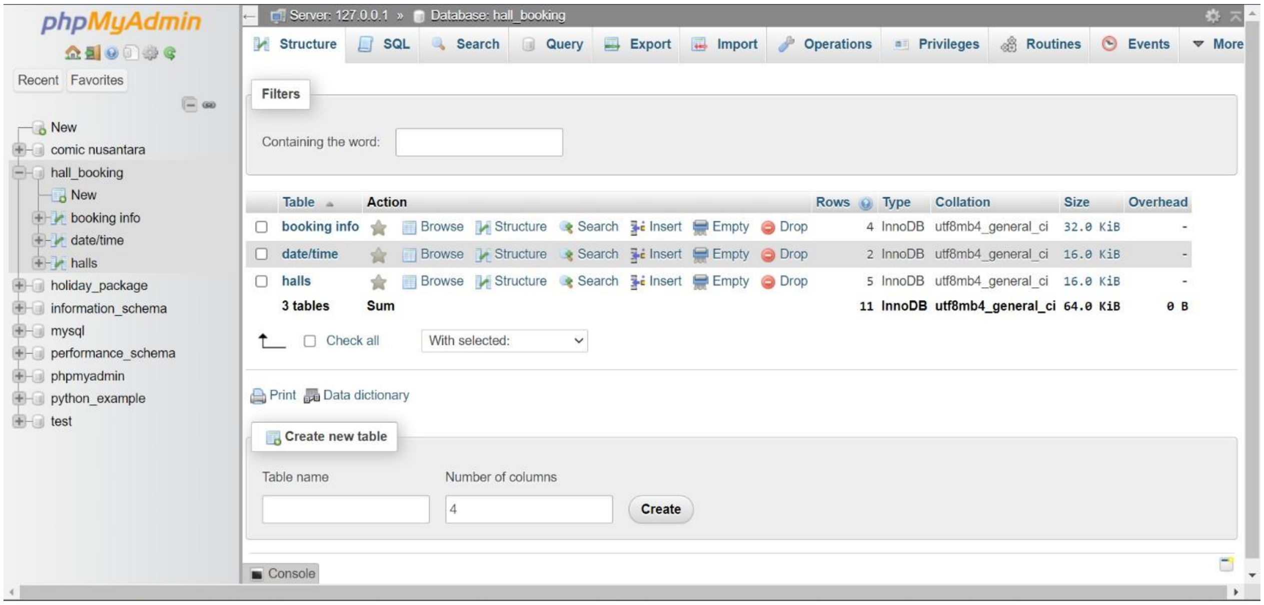
```
SELECT * FROM `booking info`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table

full_name	email	hall_name	time_from	time_to	date	event_name
Jamal	jamal@gmail.com	Dewan Seri Merbok	8	10	17/01/24	Makan2
Luzman	luzz@gmail	Dewan Seri Merbok	6	8	16/01/24	Borak
Luzman	luz@gmail.com	Dewan Seri Merbok	2	5	19/01/24	Borak
Narza	deden.encem@gmail.com	Dewan Seri Jerai	1	4	16/01/24	PDRM

□ Show all | Number of rows: 25 Filter rows: Search this table



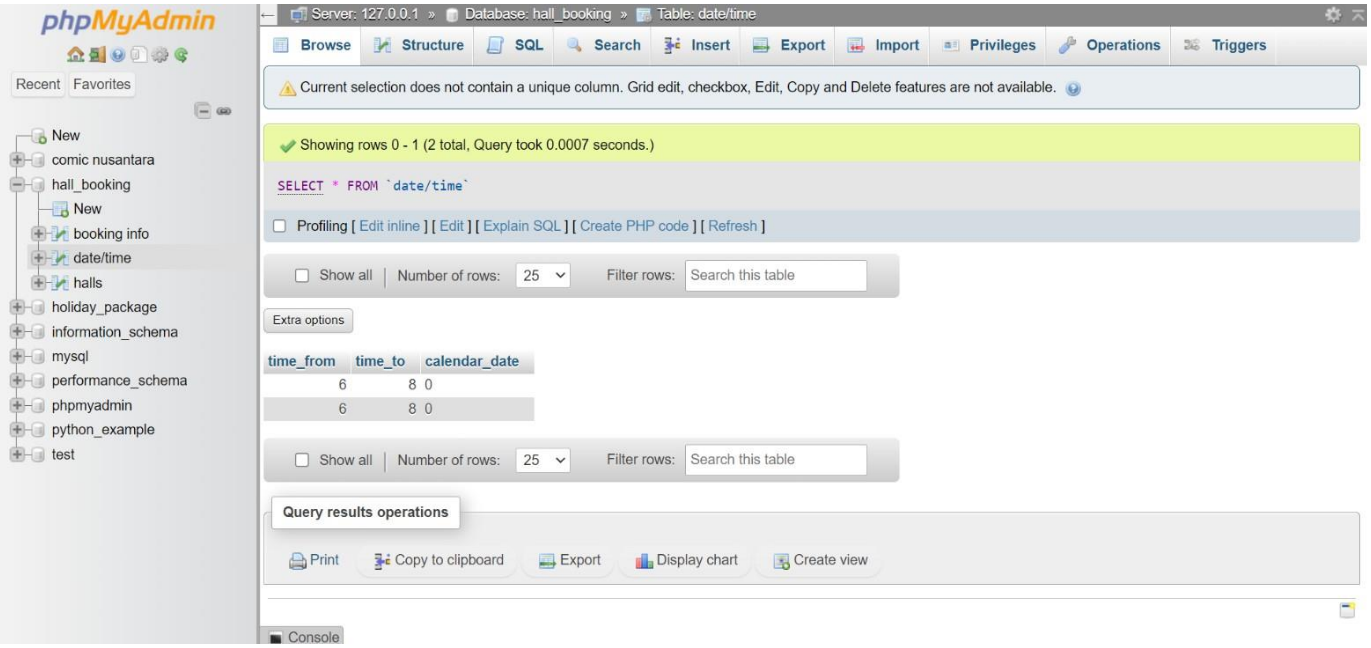
Containing the word:

	Table	Action							Rows	Type	Collation	Size	Overhead
□	booking info	☆	Browse	Structure	Search	Insert	Empty	Drop	4	InnoDB	utf8mb4_general_ci	32.0 KiB	-
□	date/time	☆	Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	halls	☆	Browse	Structure	Search	Insert	Empty	Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
	3 tables	Sum							11	InnoDB	utf8mb4_general_ci	64.0 KiB	0 B

□ Check all With selected:

Table name

Number of columns 4

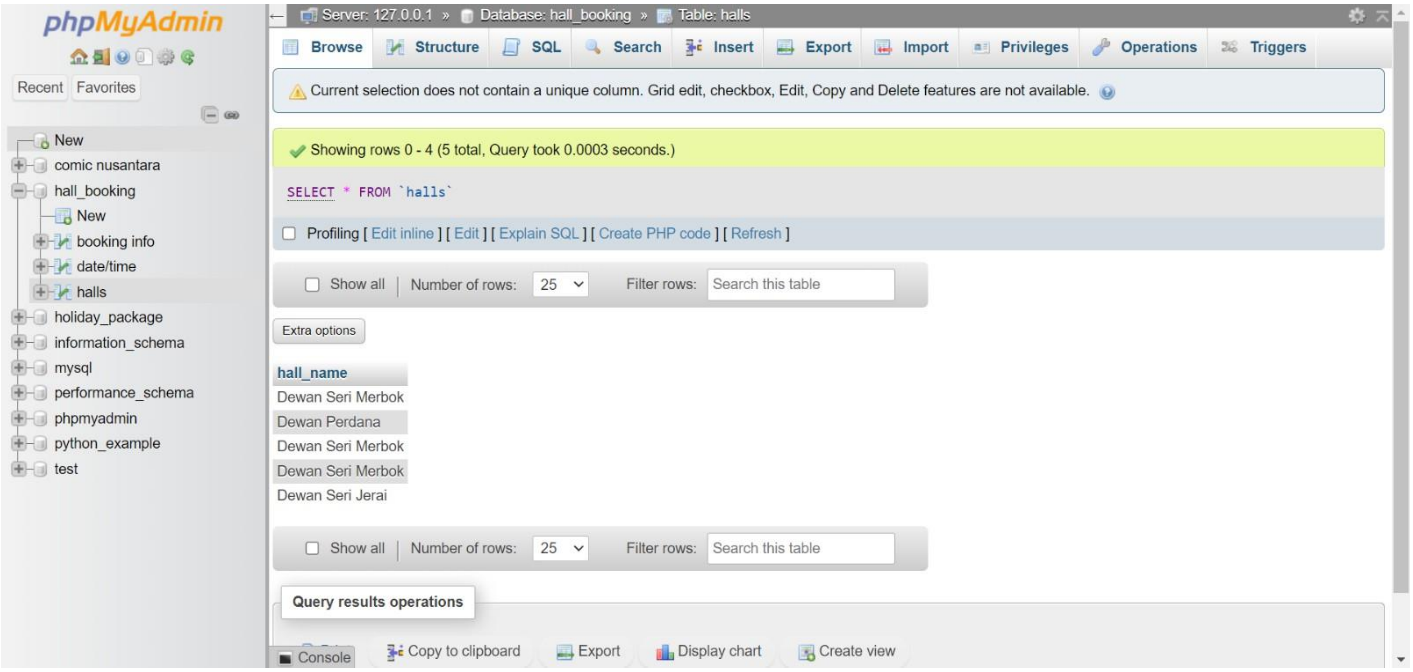

phpMyAdmin
Recent Favorites
New
comic nusantara
hall_booking
New
booking info
date/time
halls
holiday_package
information_schema
mysql
performance_schema
phpmyadmin
python_example
test
Server: 127.0.0.1 » Database: hall_booking » Table: date/time
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table
Extra options
Query results operations
Print Copy to clipboard Export Display chart Create view
Console

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 1 (2 total, Query took 0.0007 seconds.)

```
SELECT * FROM `date/time`
```

time_from	time_to	calendar_date
6	8	0
6	8	0


phpMyAdmin
Recent Favorites
New
comic nusantara
hall_booking
New
booking info
date/time
halls
holiday_package
information_schema
mysql
performance_schema
phpmyadmin
python_example
test
Server: 127.0.0.1 » Database: hall_booking » Table: halls
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table
Extra options
Query results operations
Copy to clipboard Export Display chart Create view
Console

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 4 (5 total, Query took 0.0003 seconds.)

```
SELECT * FROM `halls`
```

hall_name
Dewan Seri Merbok
Dewan Perdana
Dewan Seri Merbok
Dewan Seri Merbok
Dewan Seri Jerai

THANK YOU