



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

I 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

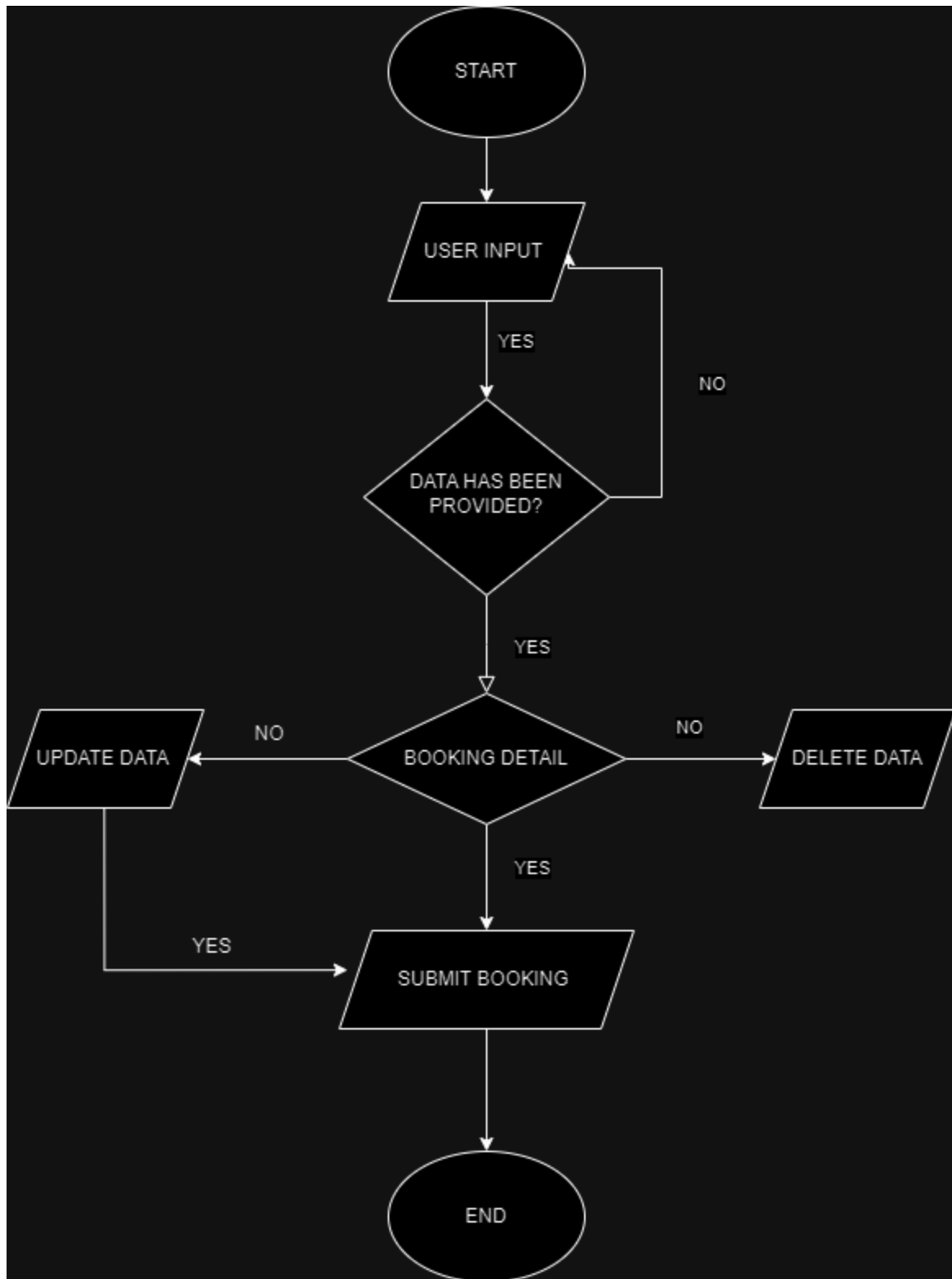
ACKNOWLEDGMENT		
1.0	INTRODUCTION	1
2.0	FLOWCHART	2
3.0	SNAPSHOT OF CODE	3
4.0	SNAPSHOT OF GUI	6
5.0	SNAPSHOT OF DATABSE	7

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

```
1 import tkinter as tk
2 from tkinter import ttk, messagebox
3 from tkcalendar import DateEntry
4 import mysql.connector
5
6 mydb = mysql.connector.connect(
7     host="localhost",
8     user="root",
9     password="",
10    database="hall_booking"
11 )
12
13 mycursor = mydb.cursor()
14
15 def show_second_gui():
16     # Check if required fields are filled
17     if not full_name_entry.get() or not email_entry.get():
18         messagebox.showwarning("Warning", "Please fill in all the required fields.")
19     return
20
21 full_name_label.grid_remove()
22 full_name_entry.grid_remove()
23 email_label.grid_remove()
24 email_entry.grid_remove()
25 next_button.grid_remove()
26
27 # Grid placement for second GUI
28 from_time_label.grid(row=0, column=0, padx=10, pady=10, sticky="e")
29 from_time_combobox.grid(row=0, column=1, padx=10, pady=10, sticky="w")
30
31 until_time_label.grid(row=1, column=0, padx=10, pady=10, sticky="e")
32 until_time_combobox.grid(row=1, column=1, padx=10, pady=10, sticky="w")
33
34 calendar_label.grid(row=2, column=0, padx=10, pady=10, sticky="e")
35 calendar_entry.grid(row=2, column=1, padx=10, pady=10, sticky="w")
36
```

```
37 event_name_label.grid(row=3, column=0, padx=10, pady=10, sticky="e")
38 event_name_entry.grid(row=3, column=1, padx=10, pady=10, sticky="w")
39
40 hall_name_label.grid(row=4, column=0, padx=10, pady=10, sticky="e")
41 hall_name_combobox.grid(row=4, column=1, padx=10, pady=10, sticky="w")
42
43 #button placement
44 submit_button.grid(row=5, column=0, padx=10, pady=10)
45 update_button.grid(row=5, column=1, padx=10, pady=10)
46 delete_button.grid(row=5, column=2, padx=10, pady=10)
47
48 def submit_booking():
49
50     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():
51         messagebox.showwarning("Warning", "Please fill in all the required fields.")
52     return
53
54     # Gather input data by user
55     full_name = full_name_entry.get()
56     email = email_entry.get()
57     from_time = from_time_combobox.get()
58     until_time = until_time_combobox.get()
59     calendar_date = calendar_entry.get()
60     event_name = event_name_entry.get()
61     hall_name = hall_name_combobox.get()
62
63     # Insert hall data
64     insert_hall_sql = "INSERT INTO `halls` (hall_name) VALUES (%s)"
65     insert_hall_val = (hall_name,)
66
67     try:
68         mycursor.execute(insert_hall_sql, insert_hall_val)
69         mydb.commit()
70         messagebox.showinfo("Success", "Hall information has been successfully inserted.")
71     except Exception as e:
72         messagebox.showerror("Error", f"Error inserting hall data: {e}")
```

```

74 # Insert booking data
75 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)"
76 insert_booking_val = (full_name, email, hall_name, from_time, until_time, calendar_date, event_name)
77
78 try:
79     mycursor.execute(insert_booking_sql, insert_booking_val)
80     mydb.commit()
81     messagebox.showinfo("Success", "Booking information has been successfully inserted.")
82 except Exception as e:
83     messagebox.showerror("Error", f"Error inserting booking data: {e}")
84
85 #update function
86 def update_booking():
87     # Check if required fields are filled
88     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():
89         messagebox.showwarning("Warning", "Please fill in all the required fields.")
90         return
91
92     # Gather input data
93     full_name = full_name_entry.get()
94     email = email_entry.get()
95     from_time = from_time_combobox.get()
96     until_time = until_time_combobox.get()
97     calendar_date = calendar_entry.get()
98     event_name = event_name_entry.get()
99     hall_name = hall_name_combobox.get()
100
101     # Update booking information
102     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"
103     update_booking_val = (full_name, email, hall_name, from_time, until_time, calendar_date, event_name)
104
105     try:
106         mycursor.execute(update_booking_sql, update_booking_val)
107         mydb.commit()
108         messagebox.showinfo("Success", "Booking information has been successfully updated.")
109     except Exception as e:
110         messagebox.showerror("Error", f"Error updating booking data: {e}")

```

```

111 #delete function
112 def delete_booking():
113     # Check if the event name is provided
114     if not event_name_entry.get():
115         messagebox.showwarning("Warning", "Please provide an Event Name to delete.")
116         return
117
118     event_name = event_name_entry.get()
119
120     delete_booking_sql = "DELETE FROM `booking info` WHERE event_name = %s"
121     delete_booking_val = (event_name,)
122
123     try:
124         mycursor.execute(delete_booking_sql, delete_booking_val)
125         mydb.commit()
126         messagebox.showinfo("Success", "Booking information has been successfully deleted.")
127     except Exception as e:
128         messagebox.showerror("Error", f"Error deleting booking data: {e}")
129
130 root = tk.Tk()
131 root.title("Hall Booking Application")
132
133 # Pink color
134 root.configure(bg="#FFB6C1")
135
136 # Font styles
137 font_label = ("Courier", 12)
138 font_button = ("Helvetica", 12, "bold")
139
140 # First GUI
141 full_name_label = tk.Label(root, text="Full Name:", bg="#FFB6C1", font=font_label)
142 full_name_entry = tk.Entry(root)
143
144 email_label = tk.Label(root, text="Email:", bg="#FFB6C1", font=font_label)
145 email_entry = tk.Entry(root)
146

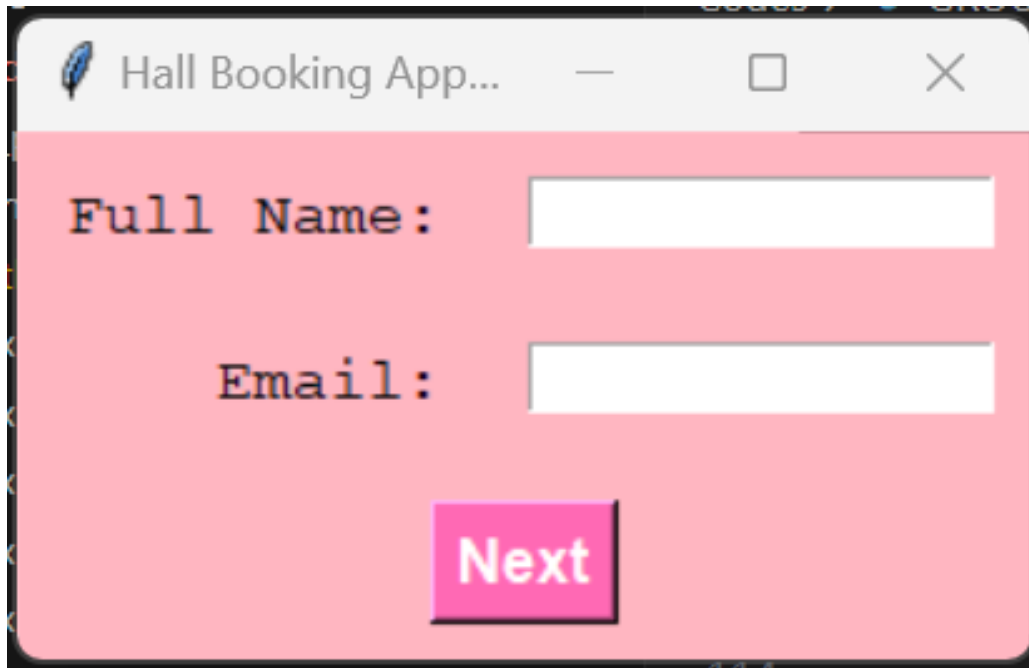
```

```

147 next_button = tk.Button(root, text="Next", command=show_second_gui, bg="#FF69B4", fg="white", font=font_button)
148
149 # Second GUI
150 from_time_label = tk.Label(root, text="From What Time:", bg="#FFB6C1", font=font_label)
151 from_time_combobox = ttk.Combobox(root, values=[f"{i:02d}:00" for i in range(24)], state="readonly")
152
153 until_time_label = tk.Label(root, text="Until What Time:", bg="#FFB6C1", font=font_label)
154 until_time_combobox = ttk.Combobox(root, values=[f"{i:02d}:00" for i in range(24)], state="readonly")
155
156 calendar_label = tk.Label(root, text="Calendar:", bg="#FFB6C1", font=font_label)
157 calendar_entry = DateEntry(root, date_pattern='dd/mm/yy')
158
159 event_name_label = tk.Label(root, text="Event Name:", bg="#FFB6C1", font=font_label)
160 event_name_entry = tk.Entry(root)
161
162 hall_name_label = tk.Label(root, text="Hall Name:", bg="#FFB6C1", font=font_label)
163 hall_name_combobox = ttk.Combobox(root, values=["Dewan Perdana", "Dewan Seri Merbok", "Dewan Seri Jerai", "Dewan Sanjana 1", "Dewan Sanjana 2"], state="readonly")
164
165 #buttons (submit, update, delete)
166 submit_button = tk.Button(root, text="Submit", command=submit_booking, bg="#FF69B4", fg="white", font=font_button)
167 update_button = tk.Button(root, text="Update", command=update_booking, bg="#FF69B4", fg="white", font=font_button)
168 delete_button = tk.Button(root, text="Delete", command=delete_booking, bg="#FF69B4", fg="white", font=font_button)
169
170 # Grid placement for first GUI
171 full_name_label.grid(row=0, column=0, padx=10, pady=10, sticky="e")
172 full_name_entry.grid(row=0, column=1, padx=10, pady=10, sticky="w")
173
174 email_label.grid(row=1, column=0, padx=10, pady=10, sticky="e")
175 email_entry.grid(row=1, column=1, padx=10, pady=10, sticky="w")
176
177 next_button.grid(row=2, columnspan=2, pady=10)
178
179 root.mainloop()
180
181

```

4.0 Snapshot of GUI

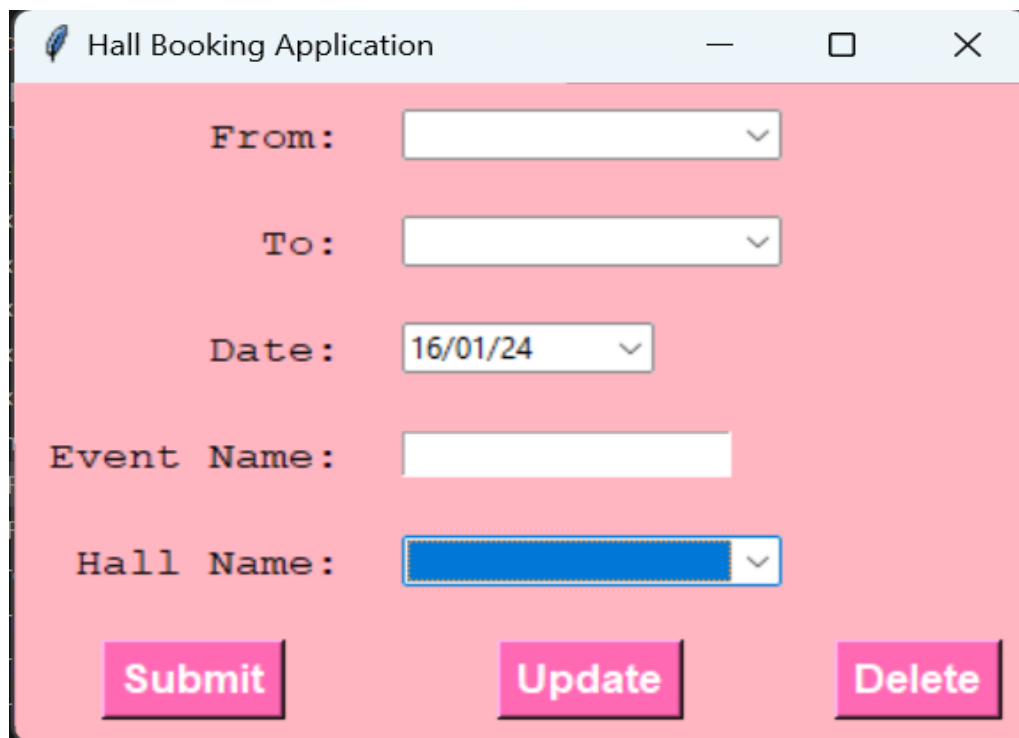


A screenshot of a web browser window titled "Hall Booking App...". The page has a pink background and contains a registration form with two input fields: "Full Name:" and "Email:". Below these fields is a pink button labeled "Next".

Full Name:

Email:

Next



A screenshot of a web browser window titled "Hall Booking Application". The page has a pink background and contains a booking details form with five input fields: "From:", "To:", "Date:", "Event Name:", and "Hall Name:". The "Date:" field is pre-filled with "16/01/24". Below these fields are three pink buttons: "Submit", "Update", and "Delete".

From:

To:

Date:

Event Name:

Hall Name:

Submit **Update** **Delete**

5.0 Snapshot of Database

The screenshot shows the phpMyAdmin interface for the 'hall_booking' database. The 'booking info' table is selected, and its structure is displayed. The table has 6 columns: full_name, email, hall_name, time_from, time_to, and date. The data is as follows:

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

The screenshot shows the phpMyAdmin interface for the 'hall_booking' database. The 'Structure' view is selected, and the table structure is displayed. The table has 6 columns: full_name, email, hall_name, time_from, time_to, and date. The data is as follows:

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

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

⚠ 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`
```

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

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

Extra options

time_from	time_to	calendar_date
6	8 0	
6	8 0	

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

Query results operations

Print Copy to clipboard Export Display chart Create view

Console

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

⚠ 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`
```

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

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

Extra options

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

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

Query results operations

Console Copy to clipboard Export Display chart Create view

THANK YOU