SCHOOL OF INFORMATION SCIENCE,

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

UNIVERSITI TEKNOLOGI MARA,

MERBOK, KEDAH

DIPLOMA IN INFORMATIC LIBRARY

(IM144)

PROGRAMMING FOR LIBRARIES

(IML208)

INDIVIDUAL ASSIGNMENT

“RESTAURANT TABLE RESERVATION (CUSTOMER)”

PREPARED BY:

WAN ALIAH FARISYA BINTI WAN NORUL AZAN

CLASS: KCDIM1443E

PREPARED FOR:

SIR AIRUL SHAZWAN BIN NORSHAHIMI

SUBMISSION DATE:

4 JANUARY 2024

INDIVIDUAL ASSIGNMENT

“RESTAURANT TABLE RESERVATION (CUSTOMER)”

BY:

WAN ALIAH FARISYA BINTI WAN NORUL AZAN

(2022865804)

SCHOOL OF INFORMATION SCIENCE,

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

UNIVERSITI TEKNOLOGI MARA,

MERBOK, KEDAH

4 JANUARY 2024

ACKNOWLEDGEMENT

Assalamualaikum w.b.t

First of all, I would like to praise ALLAH S.W.T. for giving me this opportunity to do this assignment going smoothly. Without His blessings, I could not succeed in solving this assignment.

Secondly, I would love to give appreciation to the lecturer, Sir Airul Shazwan Bin Norshahimi for all the guidance and knowledge that he shared with me throughout the process of finishing this assignment. Without his guidance, I could not manage to complete this task easily. I also really appreciate your hard work in teaching me this subject.

Finally, I would like to say thank you to my family and friends for supporting me and never giving up on giving me encouragement and prayers that have kept me going till now. I hope with this assignment, I can use the information for something beneficial to us in the future.

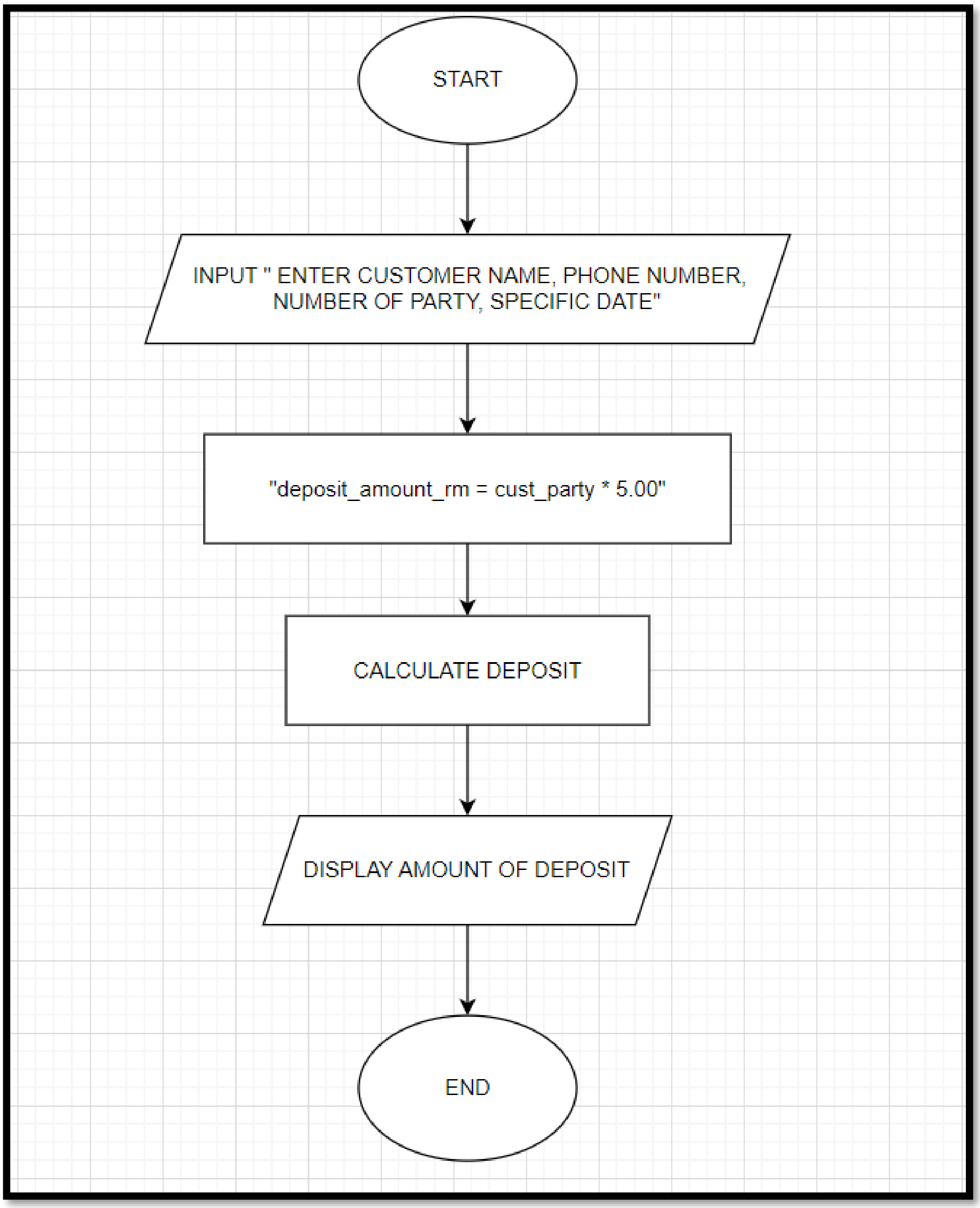
TABLE OF CONTENT

1.0 INTRODUCTION

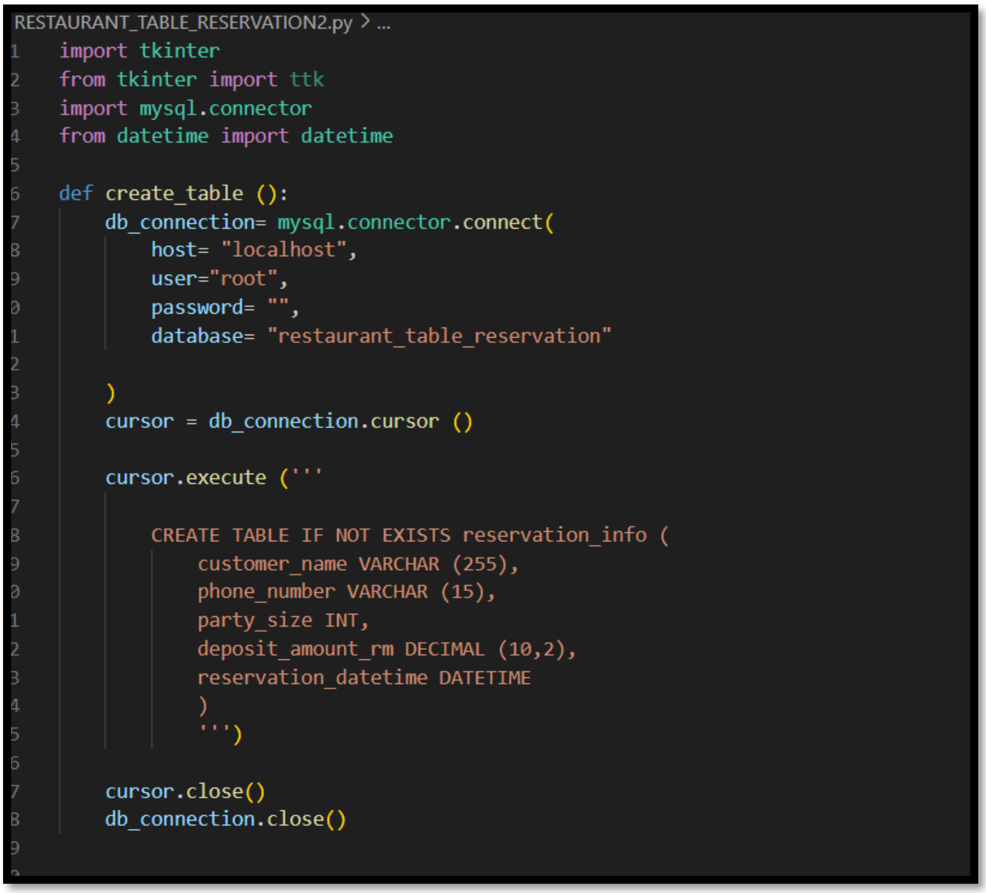
This assignment is exactly about restaurant table reservations that consist of coding, GUI, and database interface. Also, this assignment shows me the importance of all the interfaces in our daily life. It is because it can make our routine or work easier and faster. As for database integration, it utilizes MySQL to store and manage reservation information. As for Tkinter GUI, it employs the Tkinter library to create an intuitive graphical user interface for data entry. Other than that, deposit calculation dynamically calculates the deposit amount based on the number of guests in the party. For date time, it captures reservation date and time, enhancing precision and organization.

Moreover, the integration of a database into a restaurant table reservation system guarantees that reservation data is kept in an easily retrievable and manageable format. In addition to improving user experience, the Tkinter GUI makes entering and visualizing reservation data easier. The ability of automation to perform precise and timely calculations without the need for human intervention is demonstrated by the dynamic deposit amount calculation based on the number of guests. Better scheduling and organization are further benefits of standardizing the format in which reservation dates and times are recorded. I will be able to comprehend how technology can drastically improve our lives in many ways by streamlining and streamlining tasks.

2.0 FLOWCHART

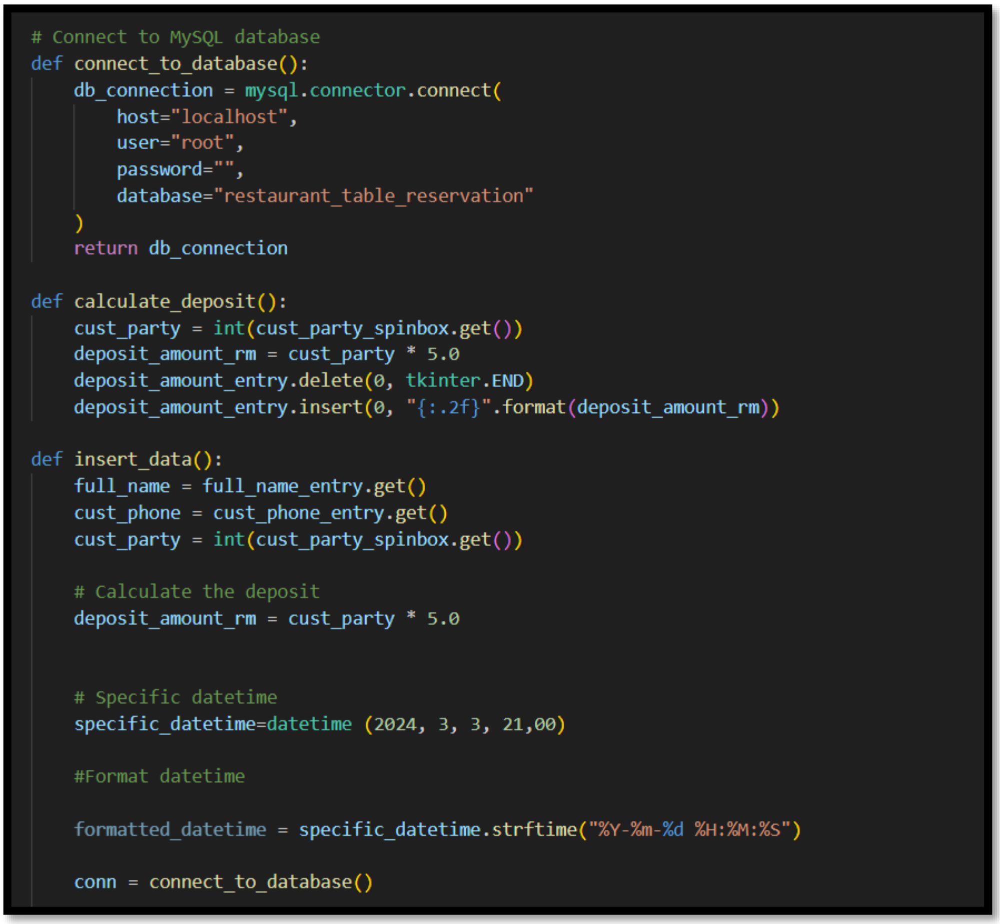

START
INPUT " ENTER CUSTOMER NAME, PHONE NUMBER, NUMBER OF PARTY, SPECIFIC DATE"
"deposit_amount_rm = cust_party * 5.00"
CALCULATE DEPOSIT
DISPLAY AMOUNT OF DEPOSIT
END

3.0 SNAPSHOT OF CODE

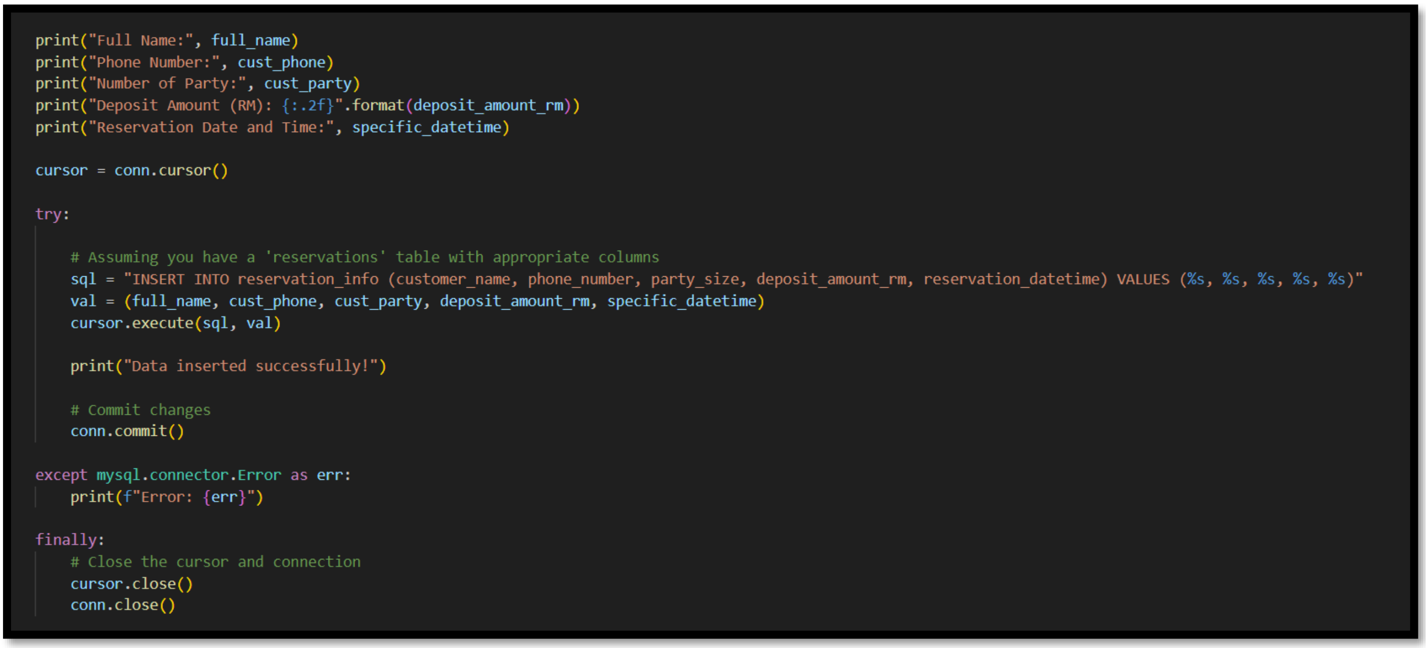


RESTAURANT_TABLE_RESERVATION2.py > ...

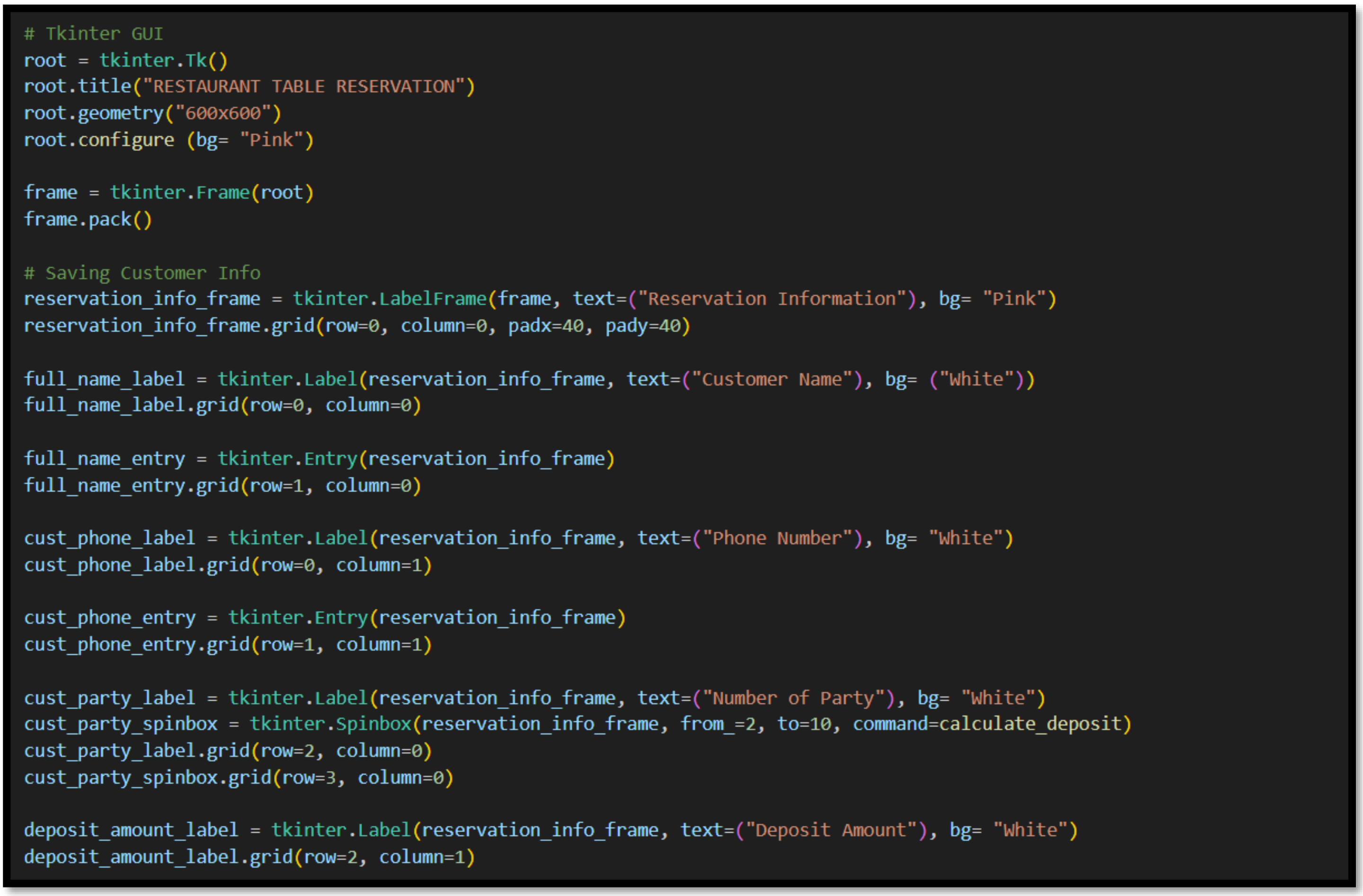
```
import tkinter
from tkinter import ttk
import mysql.connector
from datetime import datetime

def create_table ():
    db_connection= mysql.connector.connect(
        host= "localhost",
        user="root",
        password= "",
        database= "restaurant_table_reservation"

    )
    cursor = db_connection.cursor ()

    cursor.execute ('''

        CREATE TABLE IF NOT EXISTS reservation_info (
            customer_name VARCHAR (255),
            phone_number VARCHAR (15),
            party_size INT,
            deposit_amount_rm DECIMAL (10,2),
            reservation_datetime DATETIME
            )
            ''')

    cursor.close()
    db_connection.close()

```



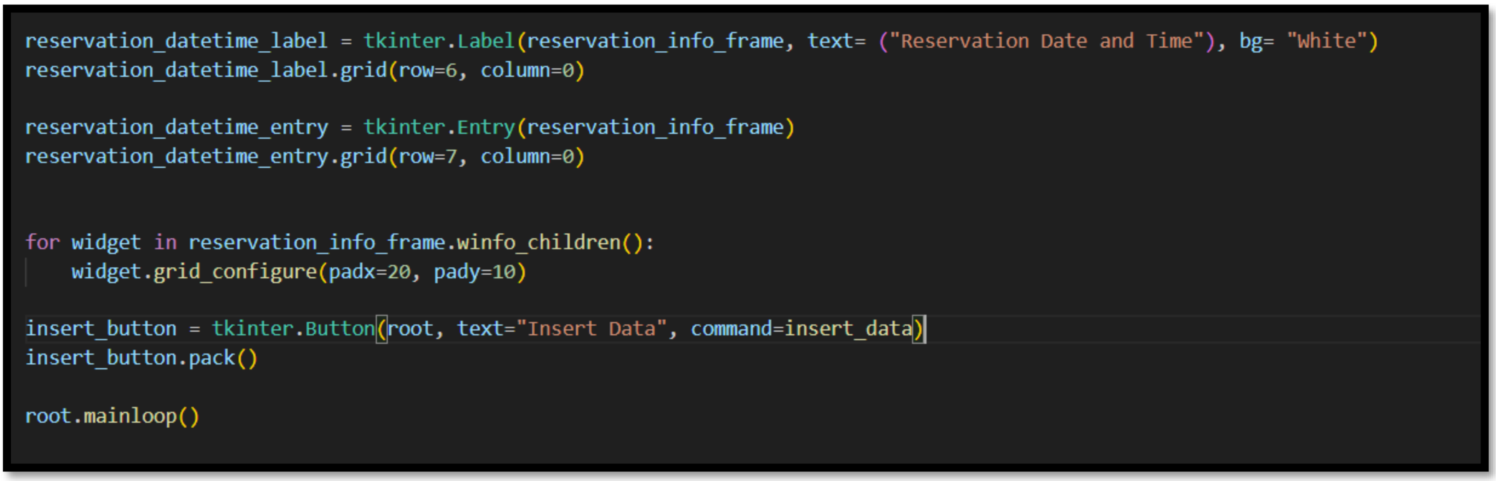
```
# Connect to MySQL database
def connect_to_database():
    db_connection = mysql.connector.connect(
        host="localhost",
        user="root",
        password="",
        database="restaurant_table_reservation"
    )
    return db_connection

def calculate_deposit():
    cust_party = int(cust_party_spinbox.get())
    deposit_amount_rm = cust_party * 5.0
    deposit_amount_entry.delete(0, tkinter.END)
    deposit_amount_entry.insert(0, "{:.2f}".format(deposit_amount_rm))

def insert_data():
    full_name = full_name_entry.get()
    cust_phone = cust_phone_entry.get()
    cust_party = int(cust_party_spinbox.get())

    # Calculate the deposit
    deposit_amount_rm = cust_party * 5.0


    # Specific datetime
    specific_datetime=datetime (2024, 3, 3, 21,00)

    #Format datetime

    formatted_datetime = specific_datetime.strftime("%Y-%m-%d %H:%M:%S")

    conn = connect_to_database()
```



```
print("Full Name:", full_name)
print("Phone Number:", cust_phone)
print("Number of Party:", cust_party)
print("Deposit Amount (RM): {:.2f}".format(deposit_amount_rm))
print("Reservation Date and Time:", specific_datetime)

cursor = conn.cursor()

try:

    # Assuming you have a 'reservations' table with appropriate columns
    sql = "INSERT INTO reservation_info (customer_name, phone_number, party_size, deposit_amount_rm, reservation_datetime) VALUES (%s, %s, %s, %s, %s)"
    val = (full_name, cust_phone, cust_party, deposit_amount_rm, specific_datetime)
    cursor.execute(sql, val)

    print("Data inserted successfully!")

    # Commit changes
    conn.commit()

except mysql.connector.Error as err:
    print(f"Error: {err}")

finally:
    # Close the cursor and connection
    cursor.close()
    conn.close()
```

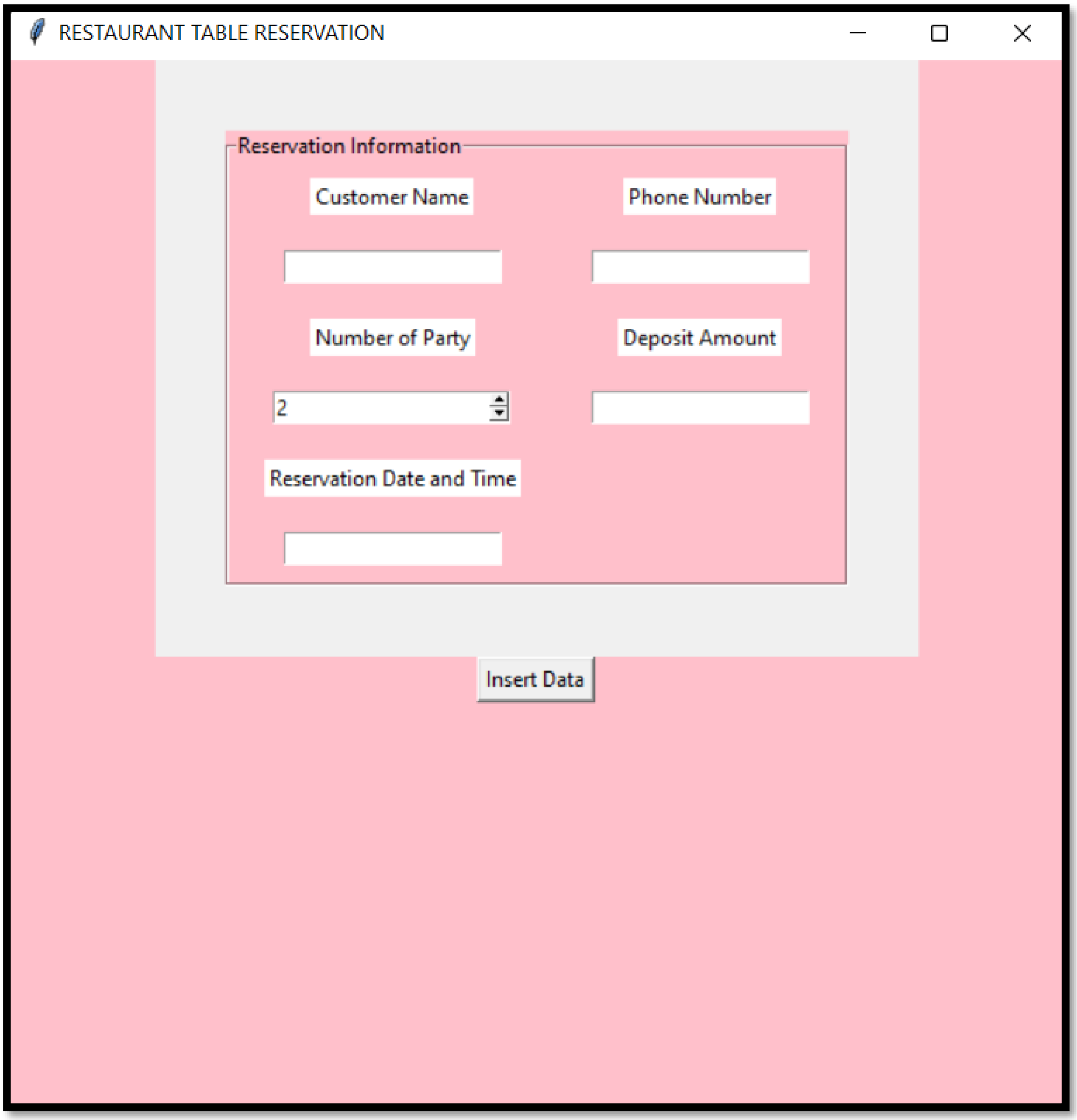


```
# Tkinter GUI
root = tkinter.Tk()
root.title("RESTAURANT TABLE RESERVATION")
root.geometry("600x600")
root.configure (bg= "Pink")

frame = tkinter.Frame(root)
frame.pack()

# Saving Customer Info
reservation_info_frame = tkinter.LabelFrame(frame, text=("Reservation Information"), bg= "Pink")
reservation_info_frame.grid(row=0, column=0, padx=40, pady=40)

full_name_label = tkinter.Label(reservation_info_frame, text=("Customer Name"), bg= ("White"))
full_name_label.grid(row=0, column=0)

full_name_entry = tkinter.Entry(reservation_info_frame)
full_name_entry.grid(row=1, column=0)

cust_phone_label = tkinter.Label(reservation_info_frame, text=("Phone Number"), bg= "White")
cust_phone_label.grid(row=0, column=1)

cust_phone_entry = tkinter.Entry(reservation_info_frame)
cust_phone_entry.grid(row=1, column=1)

cust_party_label = tkinter.Label(reservation_info_frame, text=("Number of Party"), bg= "White")
cust_party_spinbox = tkinter.Spinbox(reservation_info_frame, from_=2, to=10, command=calculate_deposit)
cust_party_label.grid(row=2, column=0)
cust_party_spinbox.grid(row=3, column=0)

deposit_amount_label = tkinter.Label(reservation_info_frame, text=("Deposit Amount"), bg= "White")
deposit_amount_label.grid(row=2, column=1)
```

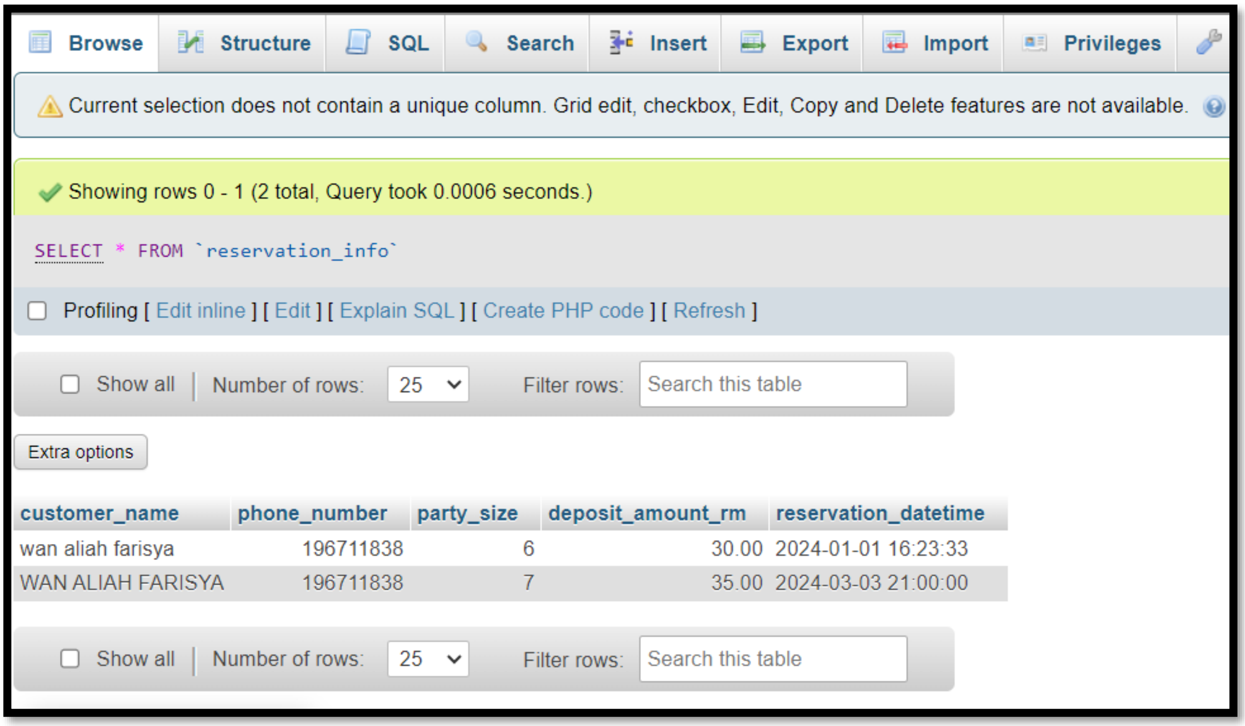


```python
reservation_datetime_label = tkinter.Label(reservation_info_frame, text= ("Reservation Date and Time"), bg= "white")
reservation_datetime_label.grid(row=6, column=0)

reservation_datetime_entry = tkinter.Entry(reservation_info_frame)
reservation_datetime_entry.grid(row=7, column=0)


for widget in reservation_info_frame.winfo_children():
    widget.grid_configure(padx=20, pady=10)

insert_button = tkinter.Button(root, text="Insert Data", command=insert_data)
insert_button.pack()

root.mainloop()
```

3.0 SNAPSHOT OF GUI



4.0 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 - 1 (2 total, Query took 0.0006 seconds.)

```
SELECT * FROM `reservation_info`
```

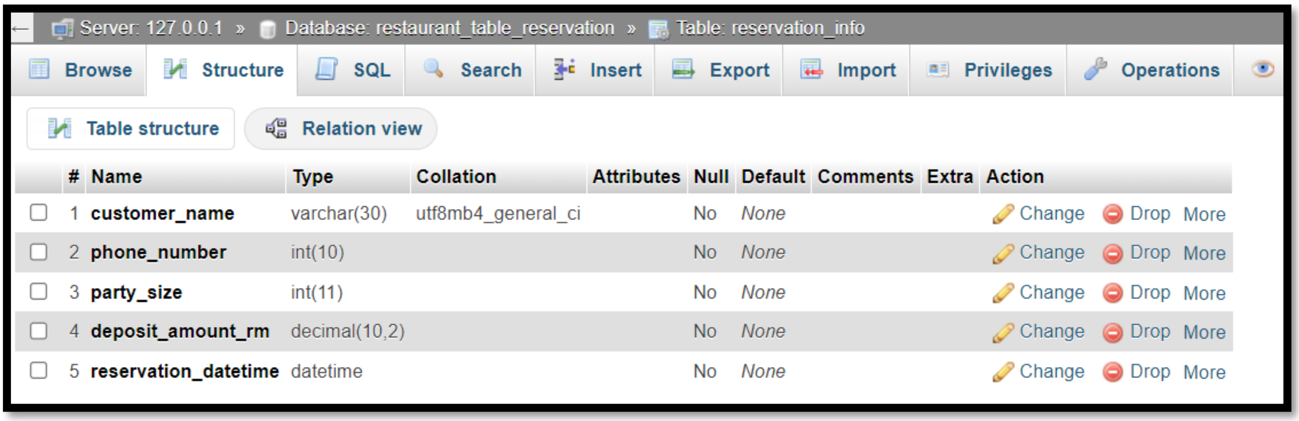
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

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

Extra options

customer_name	phone_number	party_size	deposit_amount_rm	reservation_datetime
wan aliah farisya	196711838	6	30.00	2024-01-01 16:23:33
WAN ALIAH FARISYA	196711838	7	35.00	2024-03-03 21:00:00

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



Server: 127.0.0.1 » Database: restaurant_table_reservation » Table: reservation_info

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**customer_name**	varchar(30)	utf8mb4_general_ci		No	*None*			Change Drop More
□	2	**phone_number**	int(10)			No	*None*			Change Drop More
□	3	**party_size**	int(11)			No	*None*			Change Drop More
□	4	**deposit_amount_rm**	decimal(10,2)			No	*None*			Change Drop More
□	5	**reservation_datetime**	datetime			No	*None*			Change Drop More