UNIVERSITI TEKNOLOGI MARA (UiTM)

CAWANGAN KEDAH

KAMPUS SUNGAI PETANI

SCHOOL OF INFORMATION SCIENCES STUDIES

COLLEGE OF COMPUTING, INFORMATICS, AND MEDIA

DIPLOMA IN LIBRARY INFORMATICS

(CDIM144)

PROGRAMMING FOR LIBRARIES

(IML208)

ASSIGNMENT 1 : INDIVIDUAL PROJECT (ONLINE TICKET CONCERT)

PREPARED BY:

NUR ANISA NAZIHAH BINTI KELANA (2022850266)

KCDIM1443B

PREPARED FOR:

SIR AIRUL SHAZWAN BIN NORSHAHIMI

SUBMISSION DATE: WEEK 12

ASSIGNMENT 1 : INDIVIDUAL PROJECT (ONLINE TICKET CONCERT)

NUR ANISA NAZIHAH BINTI KELANA

2022850266

KCDIM1443B

DIPLOMA IN LIBRARY INFORMATICS

SCHOOL OF INFORMATION STUDIES

COLLEGE OF COMPUTING, INFORMATICS AND MEDIA

UNIVERSITI TEKNOLOGI MARA (UiTM) CAWANGAN KEDAH

4TH JANUARY 2024

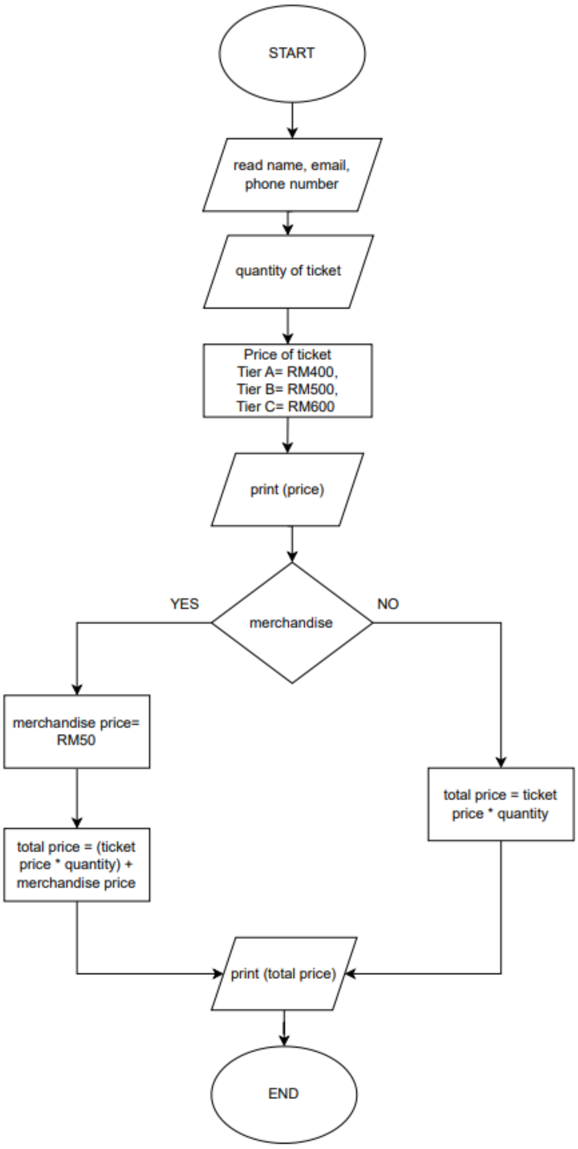
TABLE OF CONTENT

1.0 INTRODUCTION

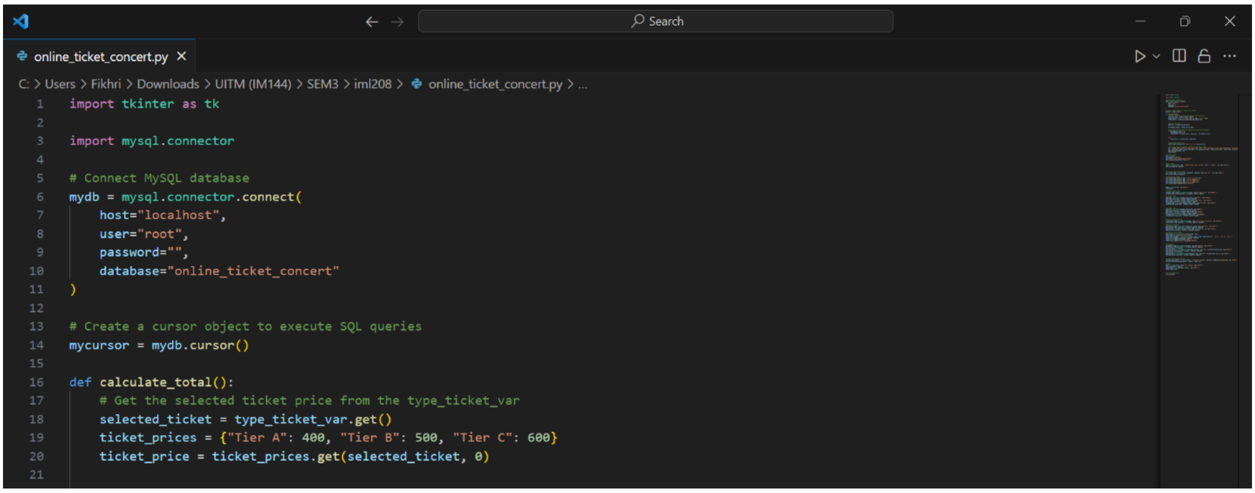
I decided to do this academic project on an online ticket concert because it's a popular topic, especially for fans. Furthermore, it can have a positive impact because we can find alternatives for buying online tickets to concerts. Firstly, the customer needs to enter their name, email address, and phone number for the coding. Then, the consumer can select the quantity of tickets and the sort of price. Next, customers have a choice of three ticket types: Tier A (RM400), Tier B (RM500), and Tier C (RM600).

Moreover, the customer has the option of purchasing stuff or not. The merchandise costs RM50. The overall cost of the ticket, including the number of tickets and merch, is the outcome. For this assignment, I can explore more details about GUI Tkinter such as making a label, frame, and checkbutton. Furthermore, this assignment also gave me more details and guided me to write coding more correctly.

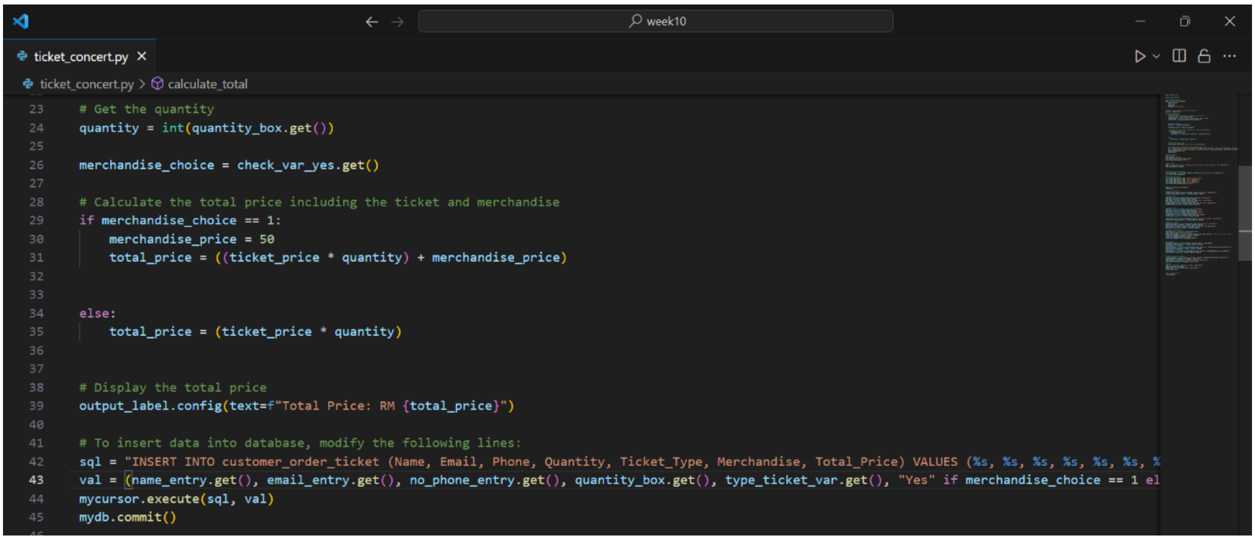
2.0 FLOW CHART


START
read name, email, phone number
quantity of ticket
Price of ticket Tier A= RM400, Tier B= RM500, Tier C= RM600
print (price)
merchandise
YES
NO
merchandise price= RM50
total price = (ticket price * quantity) + merchandise price
total price = ticket price * quantity
print (total price)
END

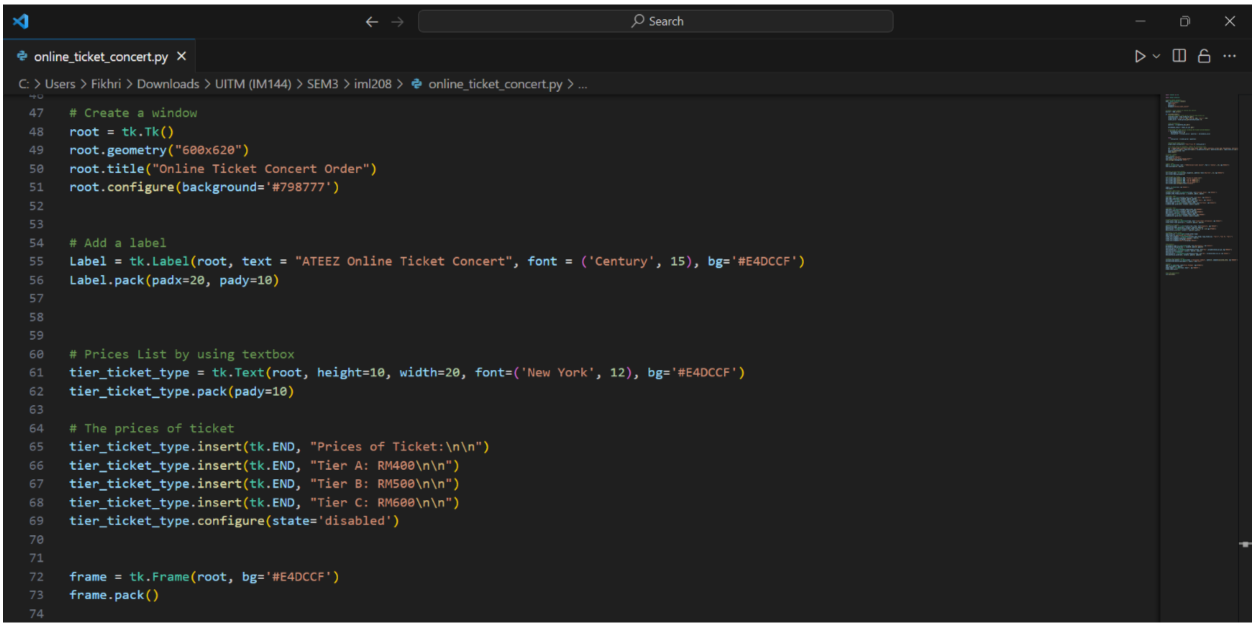
3.0 CODING (PYTHON)


Search
online_ticket_concert.py
C: > Users > Fikhri > Downloads > UITM (IM144) > SEM3 > iml208 > online_ticket_concert.py > ...

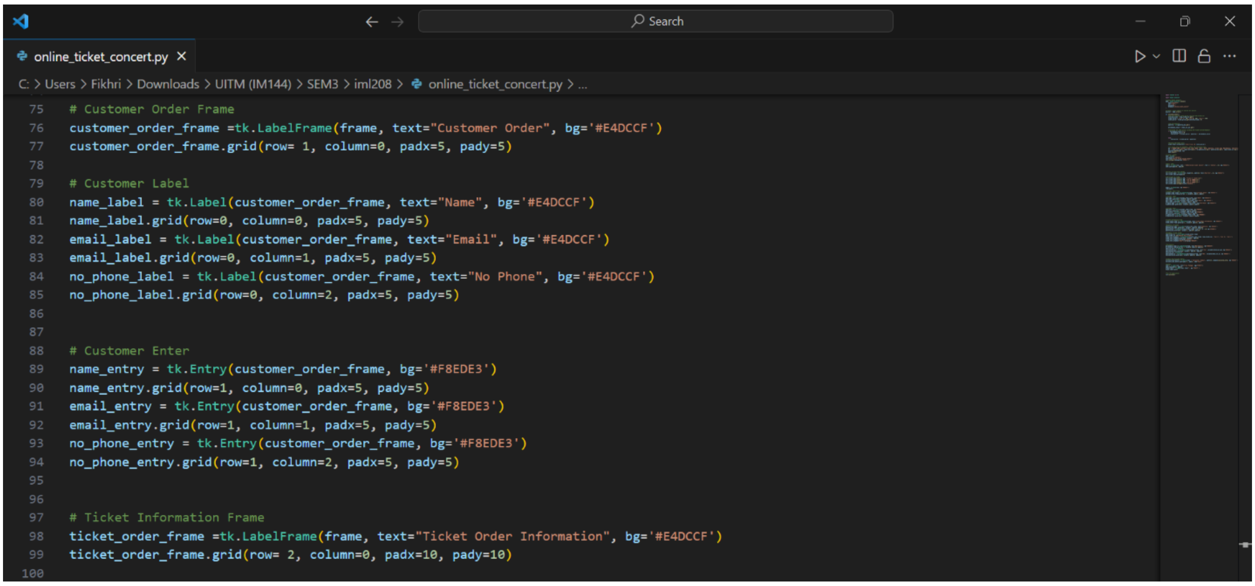
```python
import tkinter as tk

import mysql.connector

# Connect MySQL database
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="online_ticket_concert"
)

# Create a cursor object to execute SQL queries
mycursor = mydb.cursor()

def calculate_total():
    # Get the selected ticket price from the type_ticket_var
    selected_ticket = type_ticket_var.get()
    ticket_prices = {"Tier A": 400, "Tier B": 500, "Tier C": 600}
    ticket_price = ticket_prices.get(selected_ticket, 0)

```



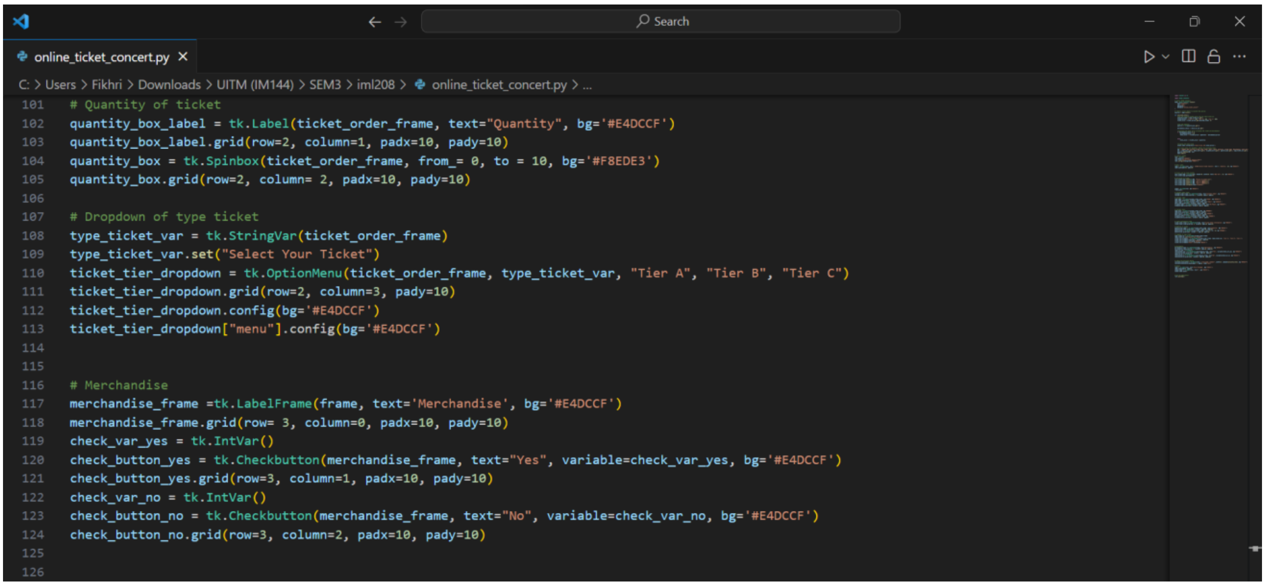
```python
    # Get the quantity
    quantity = int(quantity_box.get())

    merchandise_choice = check_var_yes.get()

    # Calculate the total price including the ticket and merchandise
    if merchandise_choice == 1:
        merchandise_price = 50
        total_price = ((ticket_price * quantity) + merchandise_price)


    else:
        total_price = (ticket_price * quantity)


    # Display the total price
    output_label.config(text=f"Total Price: RM {total_price}")

    # To insert data into database, modify the following lines:
    sql = "INSERT INTO customer_order_ticket (Name, Email, Phone, Quantity, Ticket_Type, Merchandise, Total_Price) VALUES (%s, %s, %s, %s, %s, %s, %
    val = (name_entry.get(), email_entry.get(), no_phone_entry.get(), quantity_box.get(), type_ticket_var.get(), "Yes" if merchandise_choice == 1 el
    mycursor.execute(sql, val)
    mydb.commit()
```


Search
online_ticket_concert.py
C: > Users > Fikhri > Downloads > UITM (IM144) > SEM3 > iml208 > online_ticket_concert.py > ...

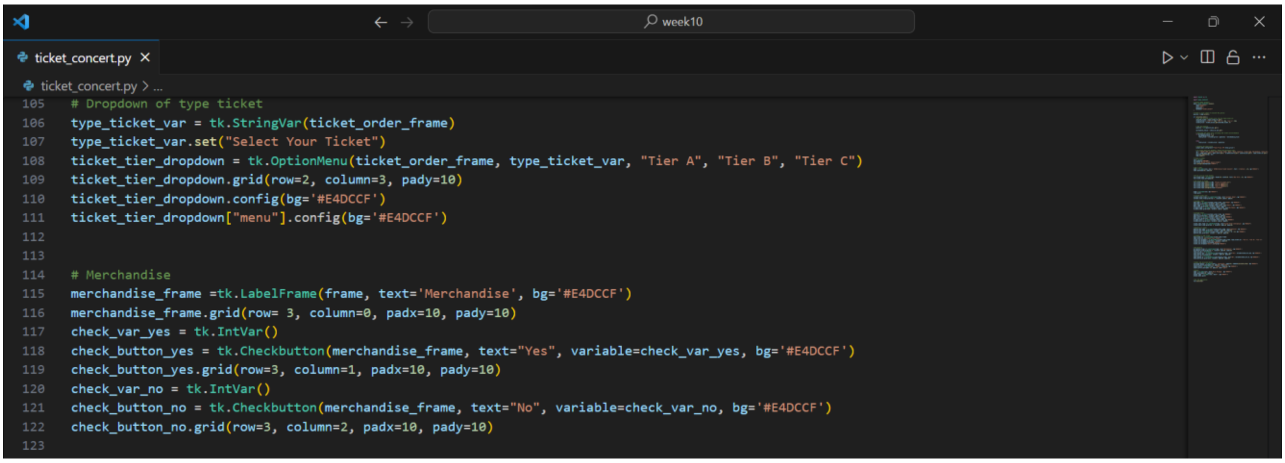
```python
# Create a window
root = tk.Tk()
root.geometry("600x620")
root.title("Online Ticket Concert Order")
root.configure(background='#798777')


# Add a label
Label = tk.Label(root, text = "ATEEZ Online Ticket Concert", font = ('Century', 15), bg='#E4DCCF')
Label.pack(padx=20, pady=10)



# Prices List by using textbox
tier_ticket_type = tk.Text(root, height=10, width=20, font=('New York', 12), bg='#E4DCCF')
tier_ticket_type.pack(pady=10)

# The prices of ticket
tier_ticket_type.insert(tk.END, "Prices of Ticket:\n\n")
tier_ticket_type.insert(tk.END, "Tier A: RM400\n\n")
tier_ticket_type.insert(tk.END, "Tier B: RM500\n\n")
tier_ticket_type.insert(tk.END, "Tier C: RM600\n\n")
tier_ticket_type.configure(state='disabled')


frame = tk.Frame(root, bg='#E4DCCF')
frame.pack()

```


Search
online_ticket_concert.py
C: > Users > Fikhri > Downloads > UITM (IM144) > SEM3 > iml208 > online_ticket_concert.py > ...

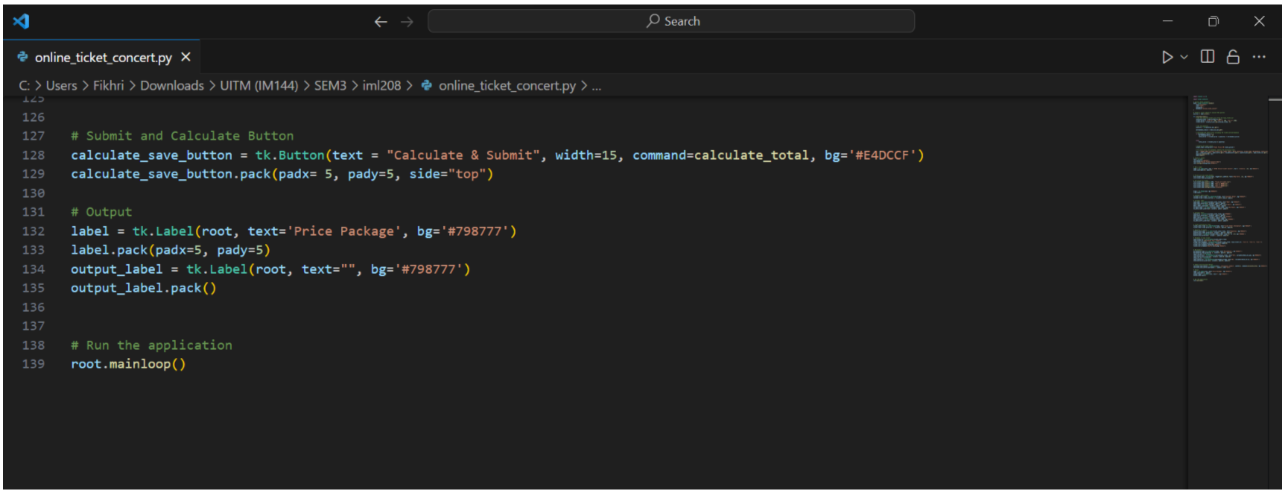
```python
# Customer Order Frame
customer_order_frame =tk.LabelFrame(frame, text="Customer Order", bg='#E4DCCF')
customer_order_frame.grid(row= 1, column=0, padx=5, pady=5)

# Customer Label
name_label = tk.Label(customer_order_frame, text="Name", bg='#E4DCCF')
name_label.grid(row=0, column=0, padx=5, pady=5)
email_label = tk.Label(customer_order_frame, text="Email", bg='#E4DCCF')
email_label.grid(row=0, column=1, padx=5, pady=5)
no_phone_label = tk.Label(customer_order_frame, text="No Phone", bg='#E4DCCF')
no_phone_label.grid(row=0, column=2, padx=5, pady=5)


# Customer Enter
name_entry = tk.Entry(customer_order_frame, bg='#F8EDE3')
name_entry.grid(row=1, column=0, padx=5, pady=5)
email_entry = tk.Entry(customer_order_frame, bg='#F8EDE3')
email_entry.grid(row=1, column=1, padx=5, pady=5)
no_phone_entry = tk.Entry(customer_order_frame, bg='#F8EDE3')
no_phone_entry.grid(row=1, column=2, padx=5, pady=5)


# Ticket Information Frame
ticket_order_frame =tk.LabelFrame(frame, text="Ticket Order Information", bg='#E4DCCF')
ticket_order_frame.grid(row= 2, column=0, padx=10, pady=10)

```


Search
online_ticket_concert.py
C: > Users > Fikhri > Downloads > UITM (IM144) > SEM3 > iml208 > online_ticket_concert.py > ...

```python
# Quantity of ticket
quantity_box_label = tk.Label(ticket_order_frame, text="Quantity", bg='#E4DCCF')
quantity_box_label.grid(row=2, column=1, padx=10, pady=10)
quantity_box = tk.Spinbox(ticket_order_frame, from_= 0, to = 10, bg='#F8EDE3')
quantity_box.grid(row=2, column= 2, padx=10, pady=10)

# Dropdown of type ticket
type_ticket_var = tk.StringVar(ticket_order_frame)
type_ticket_var.set("Select Your Ticket")
ticket_tier_dropdown = tk.OptionMenu(ticket_order_frame, type_ticket_var, "Tier A", "Tier B", "Tier C")
ticket_tier_dropdown.grid(row=2, column=3, pady=10)
ticket_tier_dropdown.config(bg='#E4DCCF')
ticket_tier_dropdown["menu"].config(bg='#E4DCCF')


# Merchandise
merchandise_frame =tk.LabelFrame(frame, text='Merchandise', bg='#E4DCCF')
merchandise_frame.grid(row= 3, column=0, padx=10, pady=10)
check_var_yes = tk.IntVar()
check_button_yes = tk.Checkbutton(merchandise_frame, text="Yes", variable=check_var_yes, bg='#E4DCCF')
check_button_yes.grid(row=3, column=1, padx=10, pady=10)
check_var_no = tk.IntVar()
check_button_no = tk.Checkbutton(merchandise_frame, text="No", variable=check_var_no, bg='#E4DCCF')
check_button_no.grid(row=3, column=2, padx=10, pady=10)
```

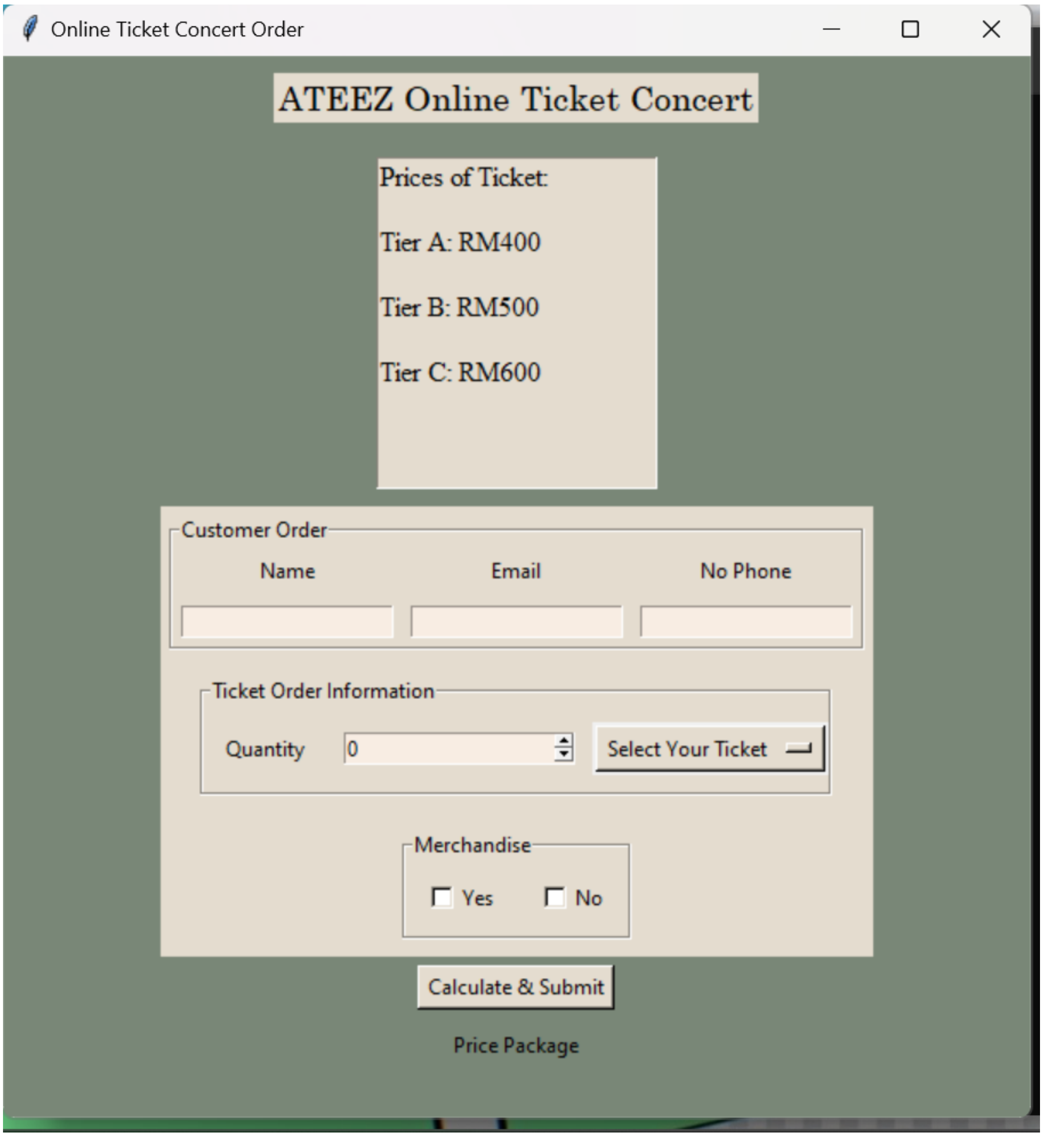


```python
# Dropdown of type ticket
type_ticket_var = tk.StringVar(ticket_order_frame)
type_ticket_var.set("Select Your Ticket")
ticket_tier_dropdown = tk.OptionMenu(ticket_order_frame, type_ticket_var, "Tier A", "Tier B", "Tier C")
ticket_tier_dropdown.grid(row=2, column=3, pady=10)
ticket_tier_dropdown.config(bg='#E4DCCF')
ticket_tier_dropdown["menu"].config(bg='#E4DCCF')


# Merchandise
merchandise_frame =tk.LabelFrame(frame, text='Merchandise', bg='#E4DCCF')
merchandise_frame.grid(row= 3, column=0, padx=10, pady=10)
check_var_yes = tk.IntVar()
check_button_yes = tk.Checkbutton(merchandise_frame, text="Yes", variable=check_var_yes, bg='#E4DCCF')
check_button_yes.grid(row=3, column=1, padx=10, pady=10)
check_var_no = tk.IntVar()
check_button_no = tk.Checkbutton(merchandise_frame, text="No", variable=check_var_no, bg='#E4DCCF')
check_button_no.grid(row=3, column=2, padx=10, pady=10)
```


Search
online_ticket_concert.py
C: > Users > Fikhri > Downloads > UITM (IM144) > SEM3 > iml208 > online_ticket_concert.py > ...

```python
# Submit and Calculate Button
calculate_save_button = tk.Button(text = "Calculate & Submit", width=15, command=calculate_total, bg='#E4DCCF')
calculate_save_button.pack(padx= 5, pady=5, side="top")

# Output
label = tk.Label(root, text='Price Package', bg='#798777')
label.pack(padx=5, pady=5)
output_label = tk.Label(root, text="", bg='#798777')
output_label.pack()


# Run the application
root.mainloop()
```

4.0 GUI



Online Ticket Concert Order

ATEEZ Online Ticket Concert

Prices of Ticket:

Tier A: RM400

Tier B: RM500

Tier C: RM600

Customer Order

Name Email No Phone

Ticket Order Information

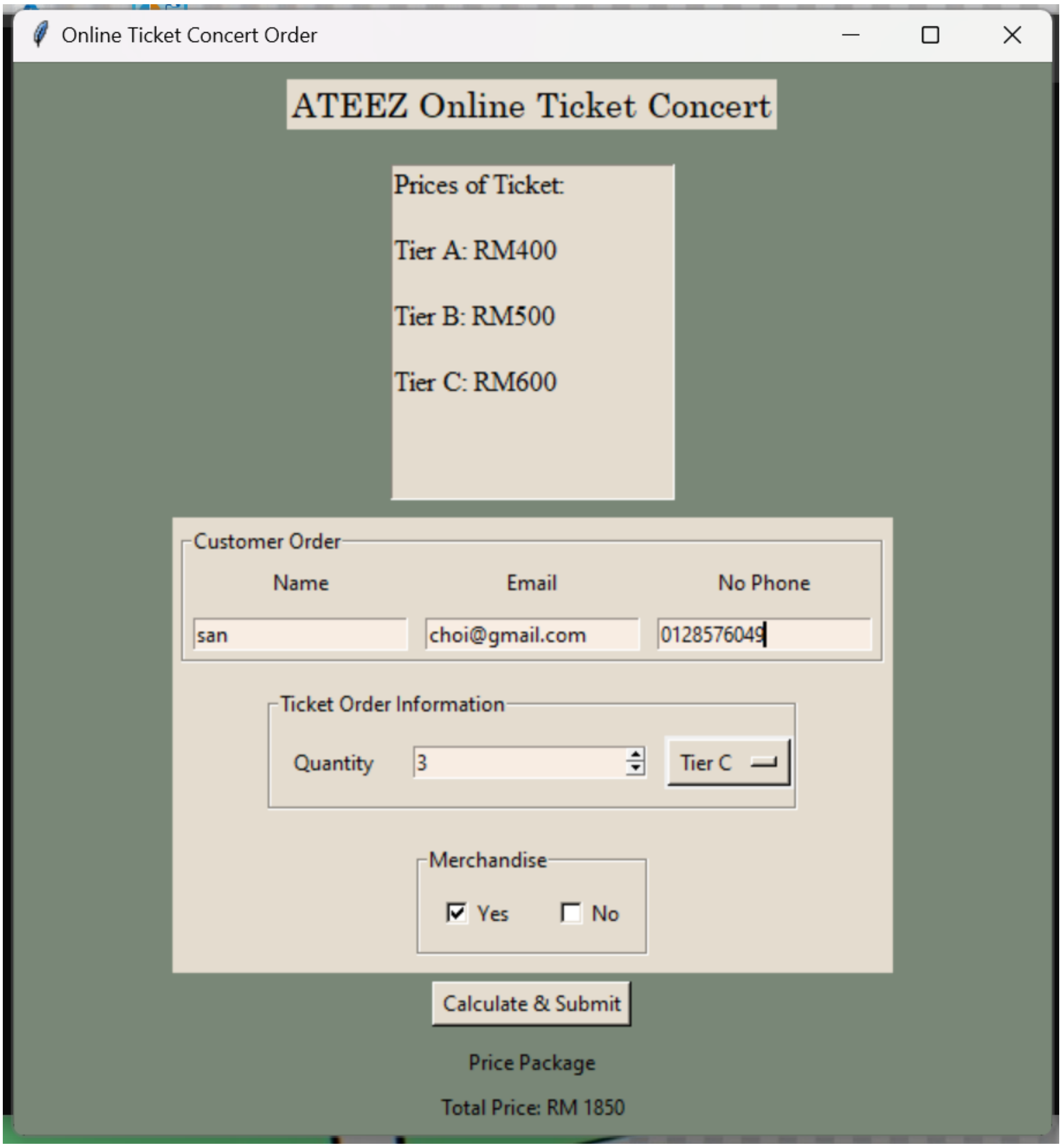
Quantity 0 Select Your Ticket

Merchandise

□ Yes □ No

Calculate & Submit

Price Package



Online Ticket Concert Order

ATEEZ Online Ticket Concert

Prices of Ticket:

Tier A: RM400

Tier B: RM500

Tier C: RM600

Customer Order

Name	Email	No Phone
san	choi@gmail.com	0128576049

Ticket Order Information

Quantity 3 Tier C

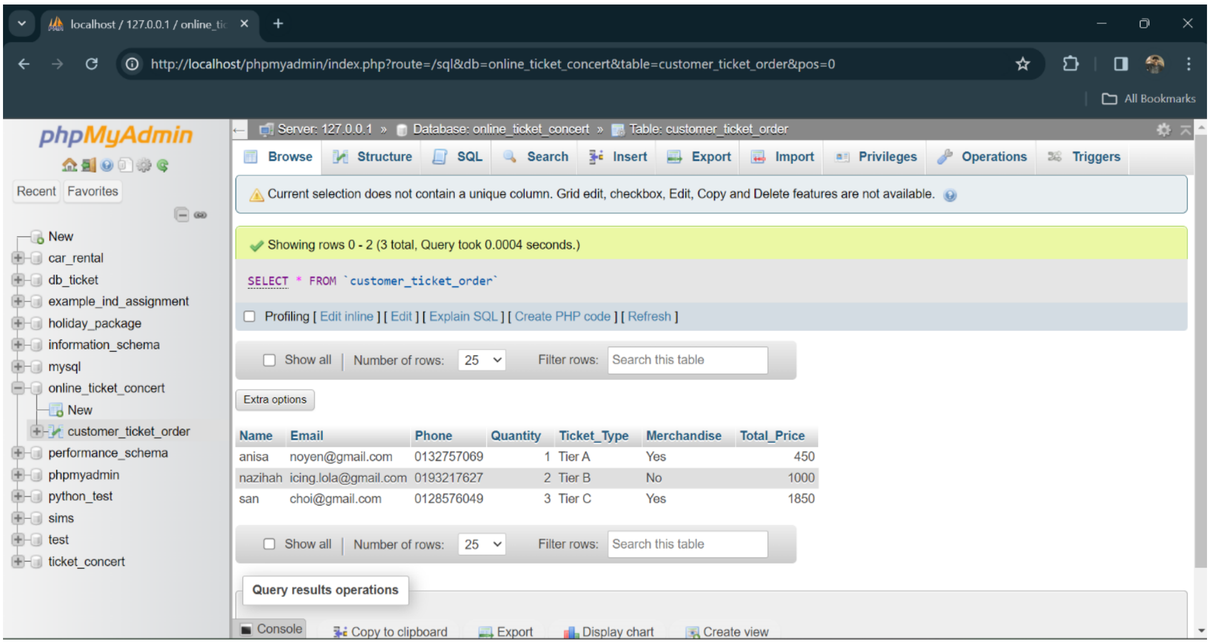
Merchandise

☑ Yes □ No

Price Package

Total Price: RM 1850

5.0 DATABASE

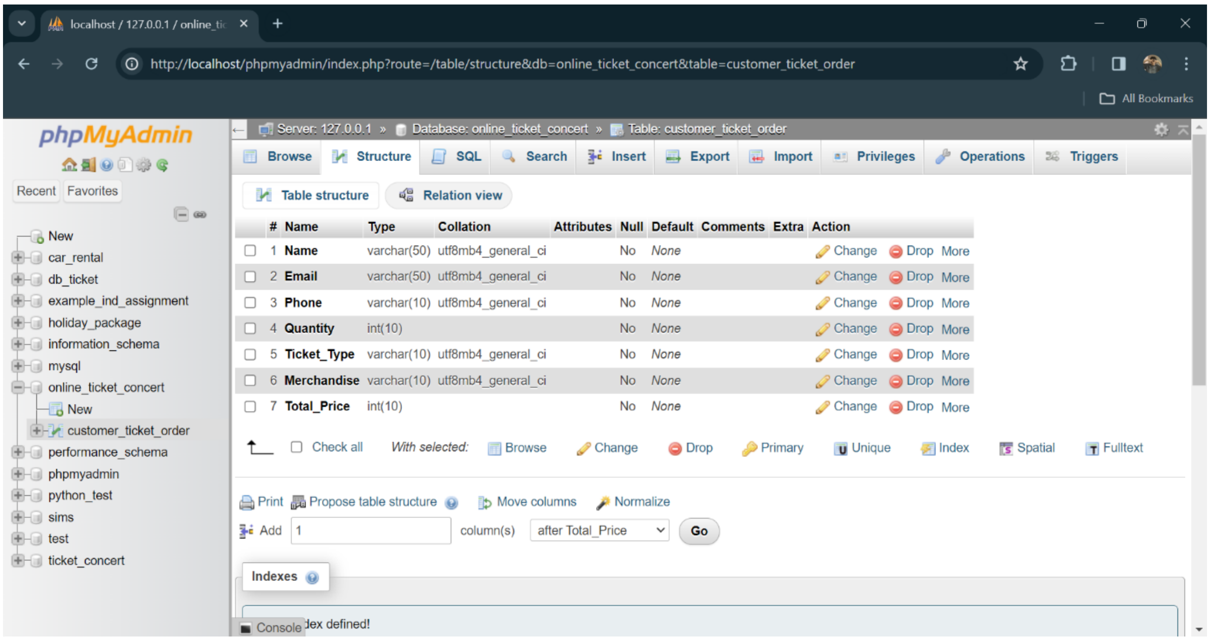


Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - 2 (3 total, Query took 0.0004 seconds.)

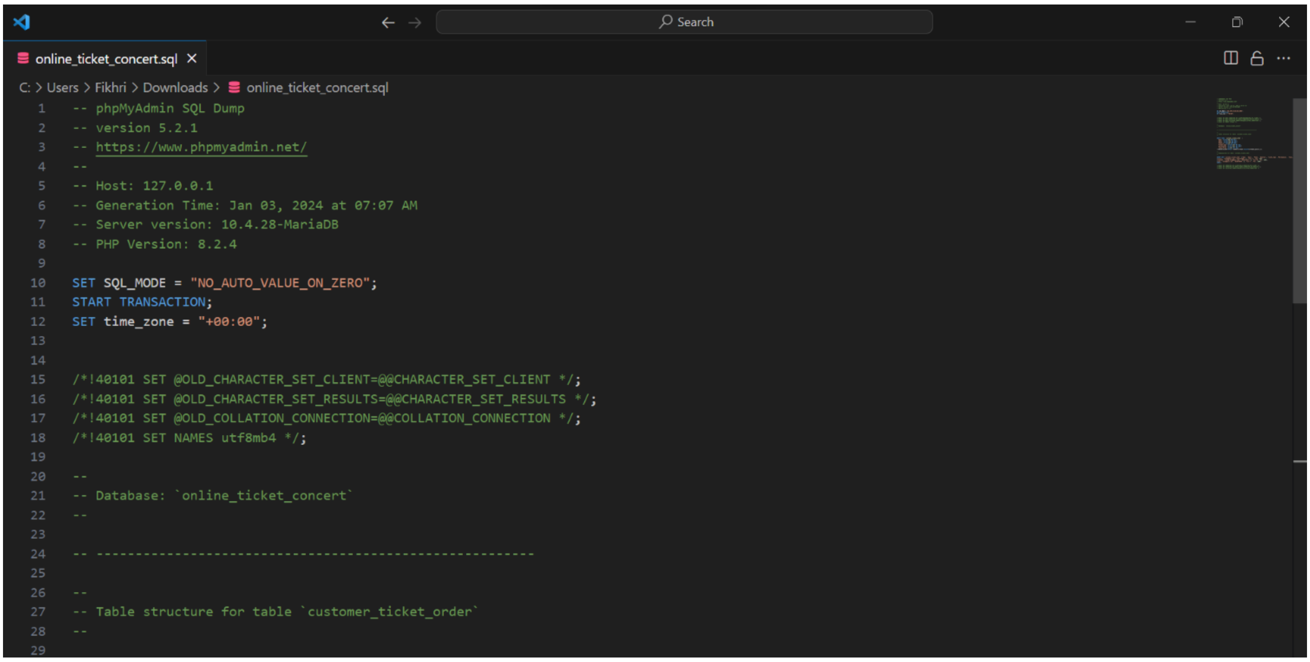
```
SELECT * FROM `customer_ticket_order`
```

Name	Email	Phone	Quantity	Ticket_Type	Merchandise	Total_Price
anisa	noyen@gmail.com	0132757069	1	Tier A	Yes	450
nazihah	icing.lola@gmail.com	0193217627	2	Tier B	No	1000
san	choi@gmail.com	0128576049	3	Tier C	Yes	1850

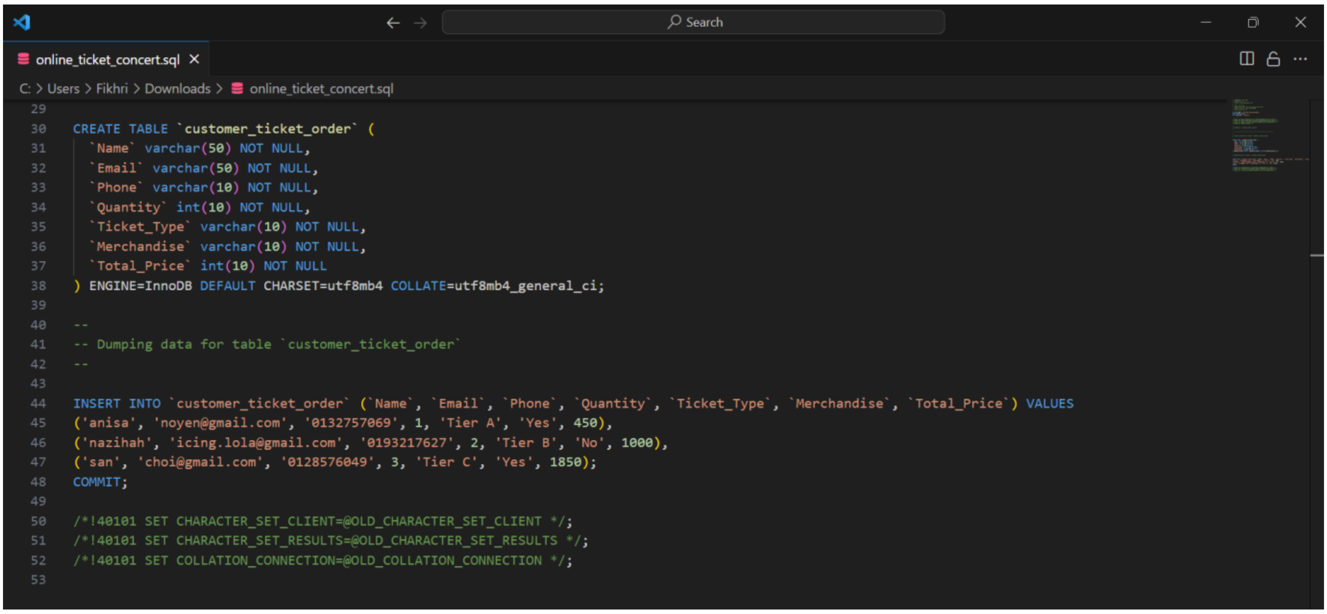


#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Name	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
2	Email	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
3	Phone	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
4	Quantity	int(10)			No	None			Change Drop More
5	Ticket_Type	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
6	Merchandise	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
7	Total_Price	int(10)			No	None			Change Drop More

dex defined!


Search
online_ticket_concert.sql
C: > Users > Fikhri > Downloads > online_ticket_concert.sql

```sql
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Jan 03, 2024 at 07:07 AM
-- Server version: 10.4.28-MariaDB
-- PHP Version: 8.2.4

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `online_ticket_concert`
--

-- --------------------------------------------------------

--
-- Table structure for table `customer_ticket_order`
--

```


Search
online_ticket_concert.sql
C: > Users > Fikhri > Downloads > online_ticket_concert.sql

```sql

CREATE TABLE `customer_ticket_order` (
  `Name` varchar(50) NOT NULL,
  `Email` varchar(50) NOT NULL,
  `Phone` varchar(10) NOT NULL,
  `Quantity` int(10) NOT NULL,
  `Ticket_Type` varchar(10) NOT NULL,
  `Merchandise` varchar(10) NOT NULL,
  `Total_Price` int(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Dumping data for table `customer_ticket_order`
--

INSERT INTO `customer_ticket_order` (`Name`, `Email`, `Phone`, `Quantity`, `Ticket_Type`, `Merchandise`, `Total_Price`) VALUES
('anisa', 'noyen@gmail.com', '0132757069', 1, 'Tier A', 'Yes', 450),
('nazihah', 'icing.lola@gmail.com', '0193217627', 2, 'Tier B', 'No', 1000),
('san', 'choi@gmail.com', '0128576049', 3, 'Tier C', 'Yes', 1850);
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

6.0 CONCLUSION

In conclusion, this assignment has given me more knowledge for coding GUI Python and MySQL Database. Furthermore, I also can calculate this code and get the correct answer. Finally, I should keep studying how to make correctly of coding with more detail and keep it in my memory.