UNIVERSITY TEKNOLOGI MARA

KEDAH BRANCH

SCHOOL OF INFORMATION SCIENCE

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

DIPLOMA IN LIBRARY INFORMATICS

(CDIM144)

PROGRAMMING FOR LIBRARIES

(IML208)

INDIVIDUAL 1: “LIRBARY NOTICE PAYMENT”

PREPARED BY:

SITI NUR AMIRAH BINTI SHAHRIL BADRUL AMIN AULIA (2022618958)

CLASS: KCDIM1443F

PREPARED FOR:

SIR AIRUL SHAZWAN BIN NORSHAHIMI

SUBMISSION DATE:

WEEK 12

ASSESSMENT 1:
“LIRBARY NOTICE PAYMENT”

PREPARED BY:

SITI NUR AMIRAH BINTI SHAHRIL BADRUL AMIN AULIA (2022618958)

CDIM144 (DIPLOMA IN LIBRARY INFORMATICS)

SCHOOL OF INFORMATION SCIENCE

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

UNIVERSITY TEKNOLOGI MARA

KEDAH BRANCH

TABLE OF CONTENTS

1.0 INTRODUCTION

In this assignment, I was assigned to do simple basic python in any field of work. So, I had chosen library field as my scope of work as my guidance in completing my individual assignment.

In short, there is many activities happened in library. For example, acquisition and book loan. But what if student suddenly return the book exceed the day limit? What if there is no such a system to record the data of the book that got returned late? It is believing the library and librarians will have a hard time in handling the management. That's why I'm creating a basic python program called "Notice Payment". It is a coding that can record user data and total charge they will be needed to pay to the librarian in charge.

For the coding, I used python help. As for the database, I use XAMPP Control and phpMyAdmin to help me sync the python and the database.

2.0 FLOWCHART

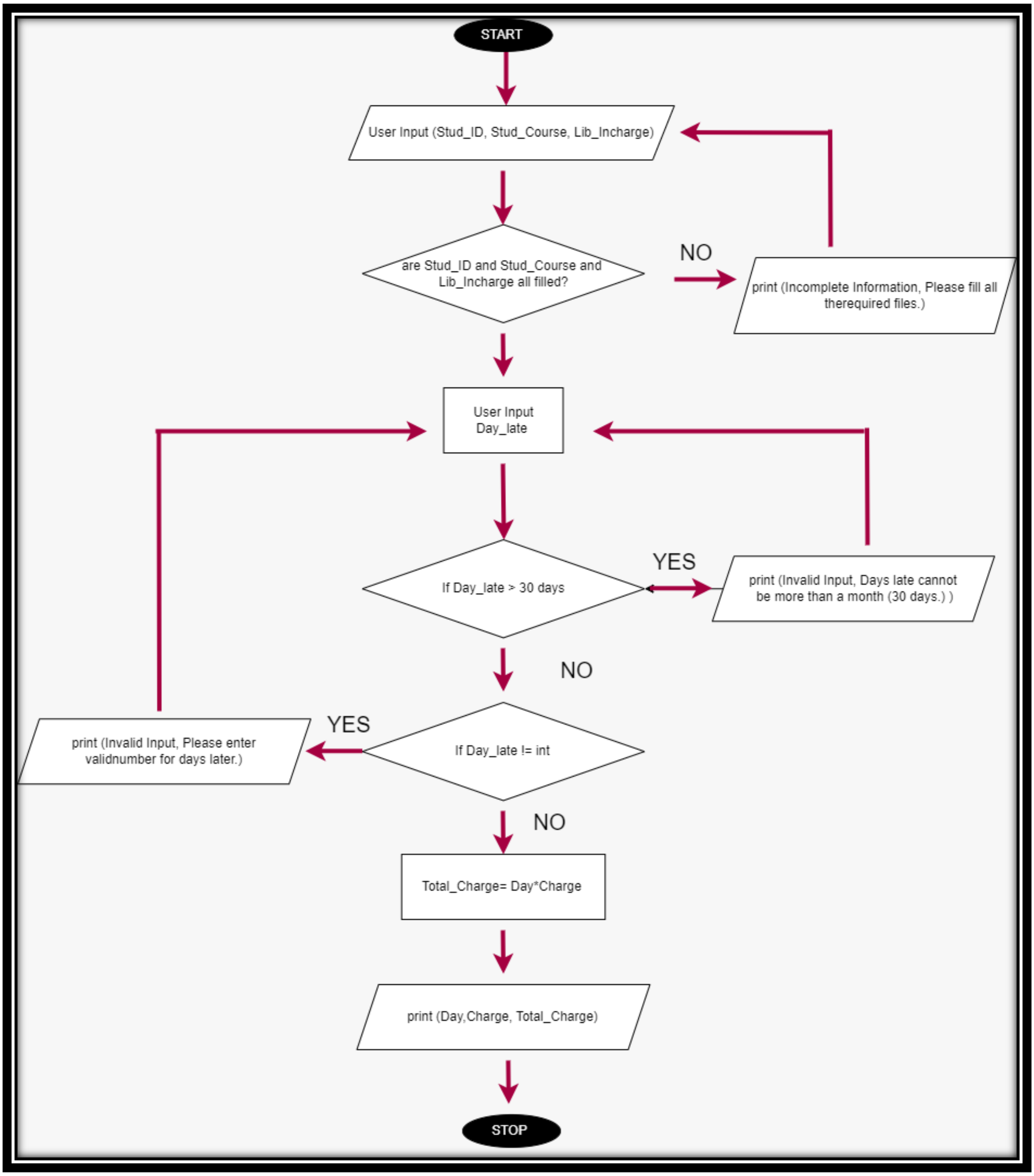
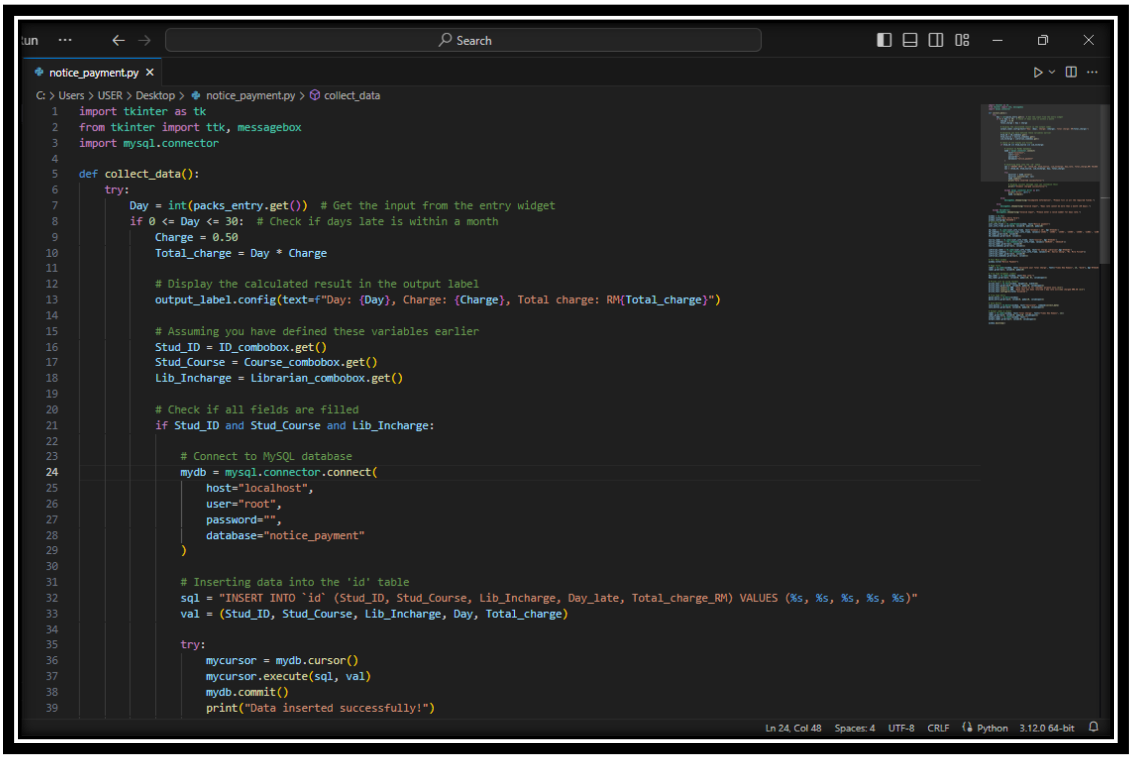

START
User Input (Stud_ID, Stud_Course, Lib_Incharge)
are Stud_ID and Stud_Course and Lib_Incharge all filled?
NO
print (Incomplete Information, Please fill all therequired files.)
User Input Day_late
If Day_late > 30 days
YES
print (Invalid Input, Days late cannot be more than a month (30 days.))
NO
If Day_late != int
YES
print (Invalid Input, Please enter validnumber for days later.)
NO
Total_Charge= Day*Charge
print (Day,Charge, Total_Charge)
STOP

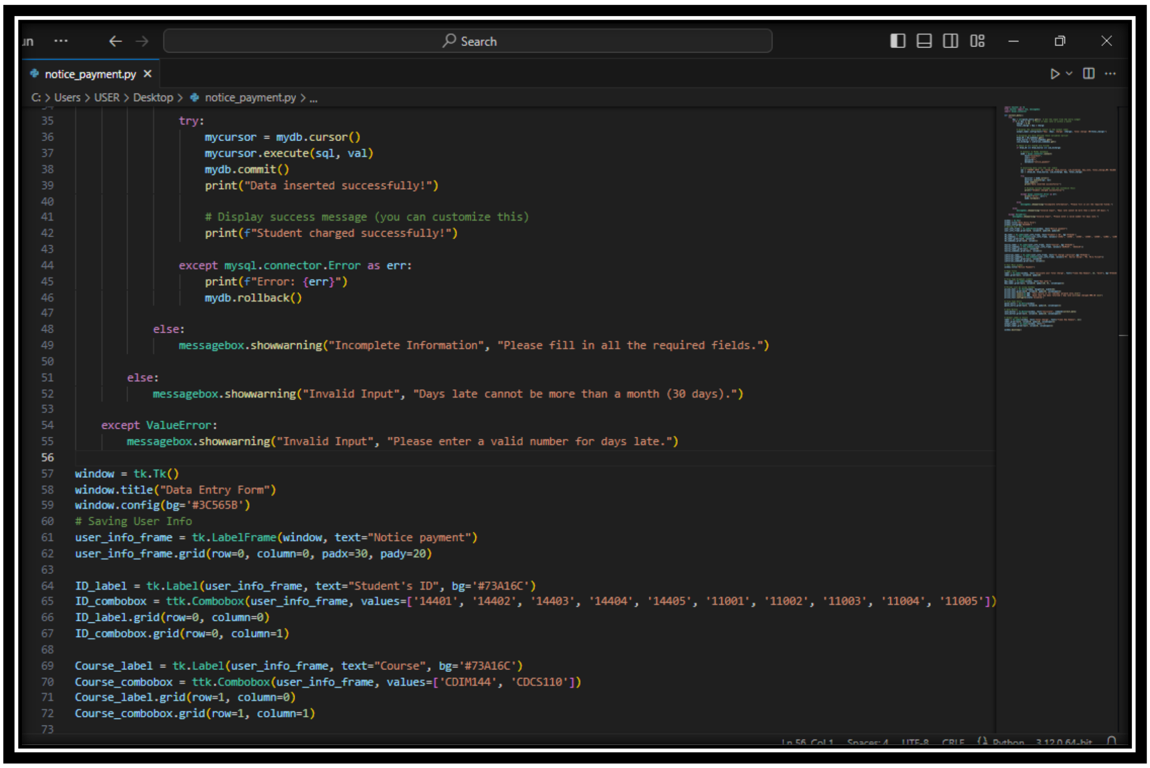
Figure 1: My flowchart

3.0 SNAPSHOT OF THE CODING


Search
notice_payment.py
C: > Users > USER > Desktop > notice_payment.py > collect_data
Ln 24, Col 48   Spaces: 4   UTF-8   CRLF   Python   3.12.0 64-bit

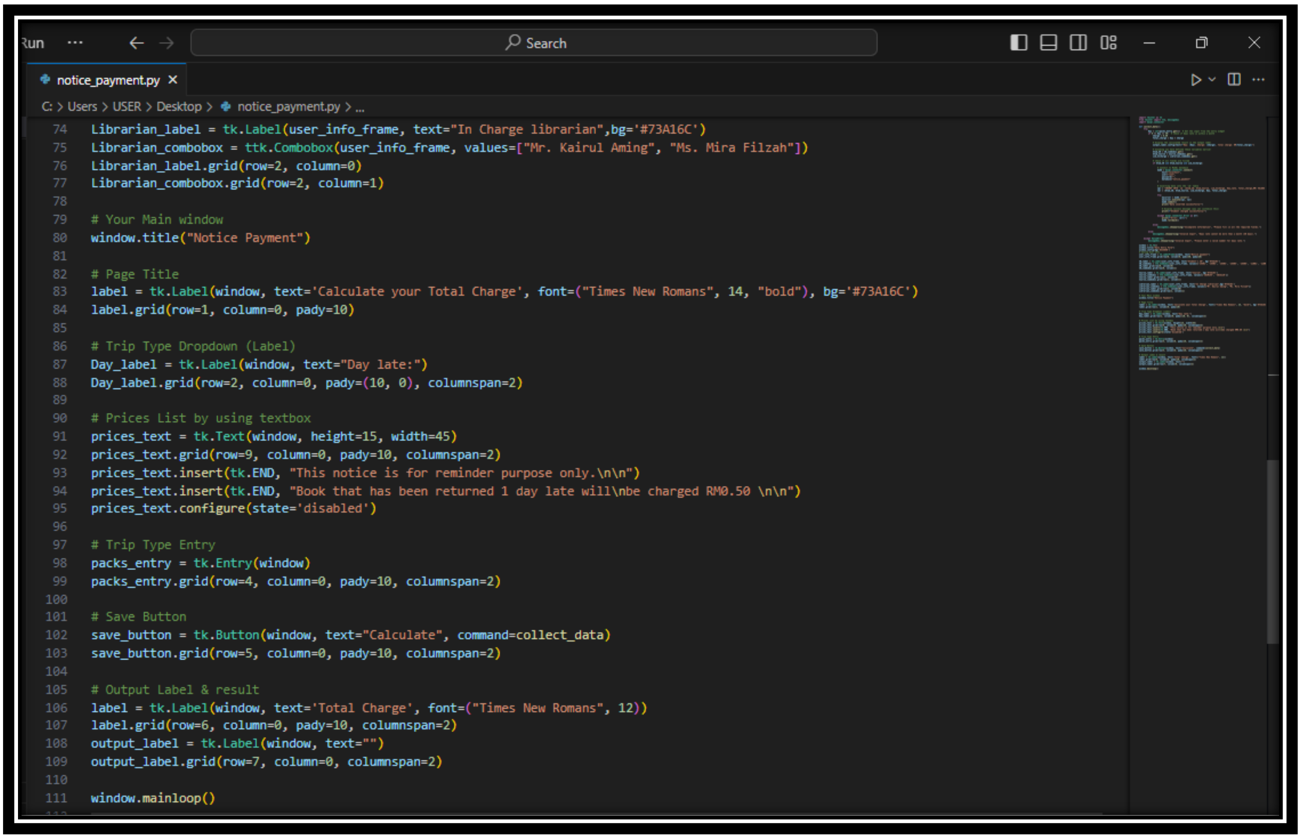
```python
import tkinter as tk
from tkinter import ttk, messagebox
import mysql.connector

def collect_data():
    try:
        Day = int(packs_entry.get())  # Get the input from the entry widget
        if 0 <= Day <= 30:  # Check if days late is within a month
            Charge = 0.50
            Total_charge = Day * Charge

            # Display the calculated result in the output label
            output_label.config(text=f"Day: {Day}, Charge: {Charge}, Total charge: RM{Total_charge}")

            # Assuming you have defined these variables earlier
            Stud_ID = ID_combobox.get()
            Stud_Course = Course_combobox.get()
            Lib_Incharge = Librarian_combobox.get()

            # Check if all fields are filled
            if Stud_ID and Stud_Course and Lib_Incharge:

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

                # Inserting data into the 'id' table
                sql = "INSERT INTO `id` (Stud_ID, Stud_Course, Lib_Incharge, Day_late, Total_charge_RM) VALUES (%s, %s, %s, %s, %s)"
                val = (Stud_ID, Stud_Course, Lib_Incharge, Day, Total_charge)

                try:
                    mycursor = mydb.cursor()
                    mycursor.execute(sql, val)
                    mydb.commit()
                    print("Data inserted successfully!")
```

Figure 2: python code


Search
notice_payment.py
C: > Users > USER > Desktop > notice_payment.py > ...
Ln 56, Col 1   Spaces: 4   UTF-8   CRLF   Python   3.12.0 64-bit

```python
                try:
                    mycursor = mydb.cursor()
                    mycursor.execute(sql, val)
                    mydb.commit()
                    print("Data inserted successfully!")

                    # Display success message (you can customize this)
                    print(f"Student charged successfully!")

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

            else:
                messagebox.showwarning("Incomplete Information", "Please fill in all the required fields.")

        else:
            messagebox.showwarning("Invalid Input", "Days late cannot be more than a month (30 days).")

    except ValueError:
        messagebox.showwarning("Invalid Input", "Please enter a valid number for days late.")

window = tk.Tk()
window.title("Data Entry Form")
window.config(bg='#3C565B')
# Saving User Info
user_info_frame = tk.LabelFrame(window, text="Notice payment")
user_info_frame.grid(row=0, column=0, padx=30, pady=20)

ID_label = tk.Label(user_info_frame, text="Student's ID", bg='#73A16C')
ID_combobox = ttk.Combobox(user_info_frame, values=['14401', '14402', '14403', '14404', '14405', '11001', '11002', '11003', '11004', '11005'])
ID_label.grid(row=0, column=0)
ID_combobox.grid(row=0, column=1)

Course_label = tk.Label(user_info_frame, text="Course", bg='#73A16C')
Course_combobox = ttk.Combobox(user_info_frame, values=['CDIM144', 'CDCS110'])
Course_label.grid(row=1, column=0)
Course_combobox.grid(row=1, column=1)
```

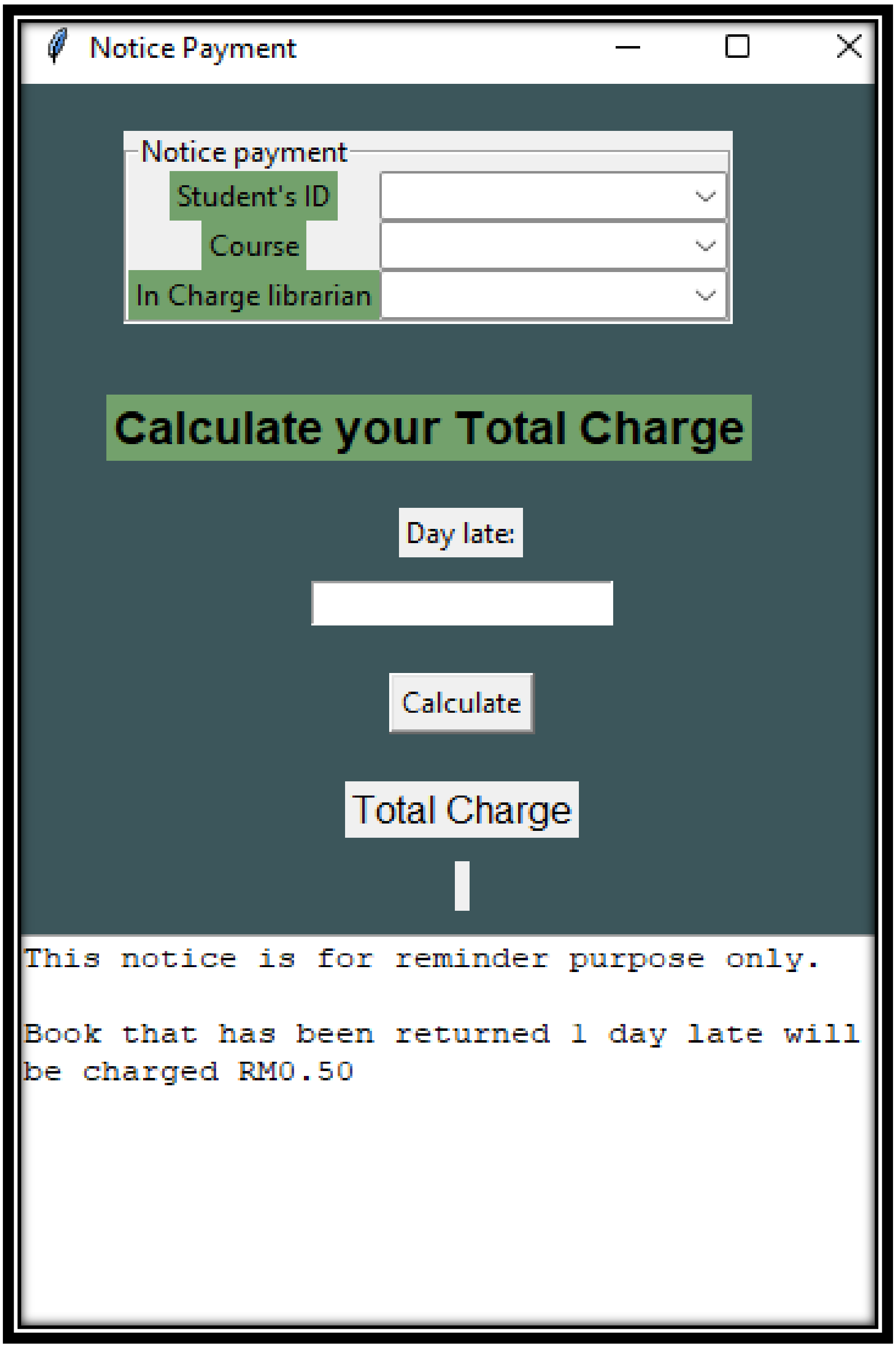
Figure 3: python code



```python
Librarian_label = tk.Label(user_info_frame, text="In Charge librarian",bg='#73A16C')
Librarian_combobox = ttk.Combobox(user_info_frame, values=["Mr. Kairul Aming", "Ms. Mira Filzah"])
Librarian_label.grid(row=2, column=0)
Librarian_combobox.grid(row=2, column=1)

# Your Main window
window.title("Notice Payment")

# Page Title
label = tk.Label(window, text='Calculate your Total Charge', font=("Times New Romans", 14, "bold"), bg='#73A16C')
label.grid(row=1, column=0, pady=10)

# Trip Type Dropdown (Label)
Day_label = tk.Label(window, text="Day late:")
Day_label.grid(row=2, column=0, pady=(10, 0), columnspan=2)

# Prices List by using textbox
prices_text = tk.Text(window, height=15, width=45)
prices_text.grid(row=9, column=0, pady=10, columnspan=2)
prices_text.insert(tk.END, "This notice is for reminder purpose only.\n\n")
prices_text.insert(tk.END, "Book that has been returned 1 day late will\nbe charged RM0.50 \n\n")
prices_text.configure(state='disabled')

# Trip Type Entry
packs_entry = tk.Entry(window)
packs_entry.grid(row=4, column=0, pady=10, columnspan=2)

# Save Button
save_button = tk.Button(window, text="Calculate", command=collect_data)
save_button.grid(row=5, column=0, pady=10, columnspan=2)

# Output Label & result
label = tk.Label(window, text='Total Charge', font=("Times New Romans", 12))
label.grid(row=6, column=0, pady=10, columnspan=2)
output_label = tk.Label(window, text="")
output_label.grid(row=7, column=0, columnspan=2)

window.mainloop()
```

Figure 4: python code

4.0 SNAPSHOT OF THE GUI



```
This notice is for reminder purpose only.

Book that has been returned 1 day late will
be charged RM0.50
```

Figure 5: GUI of my coding

5.0 SNAPSHOT OF THE DATABASE

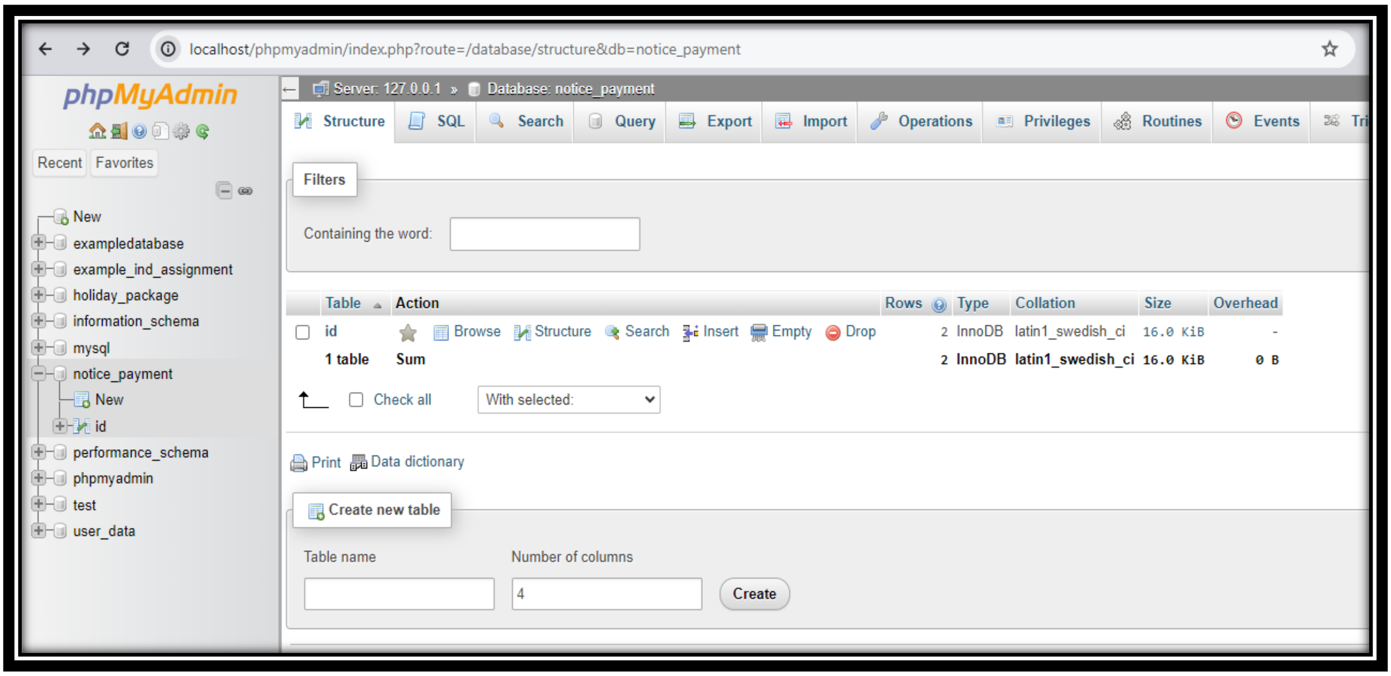
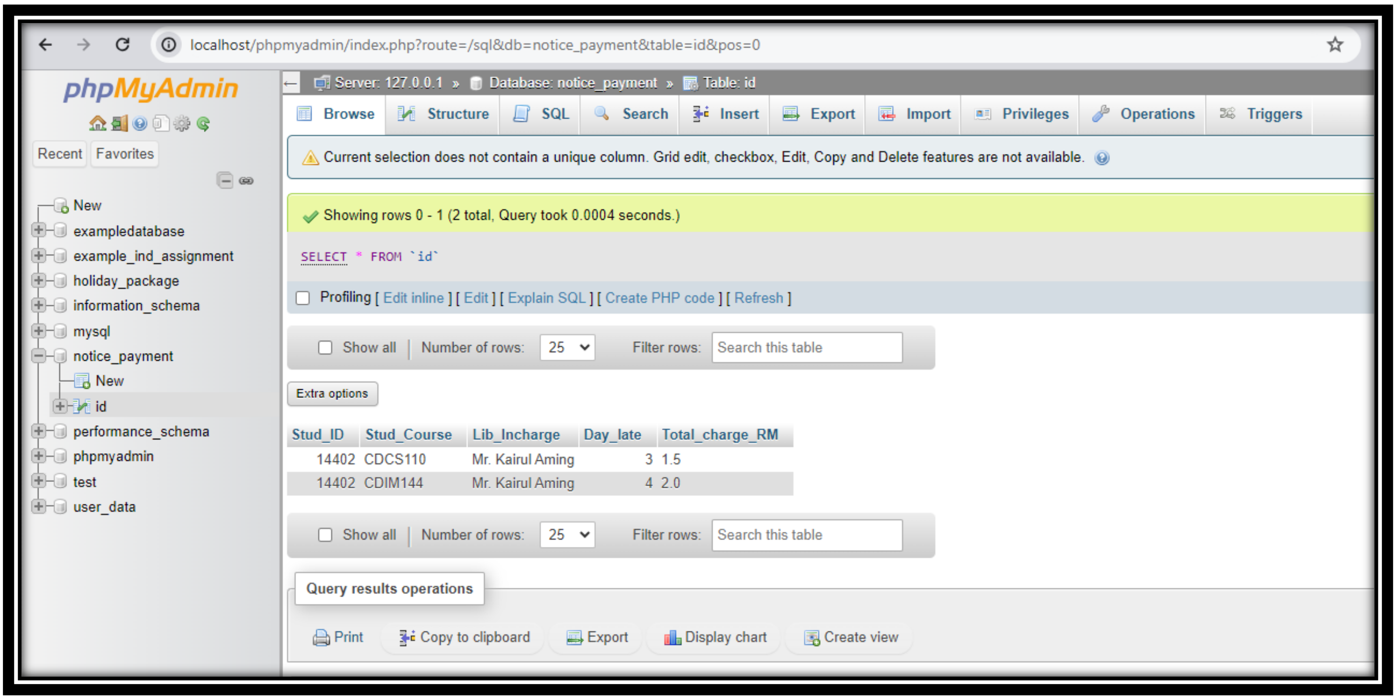


Table	Action	Rows	Type	Collation	Size	Overhead
id	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16.0 KiB	-
1 table	Sum	2	InnoDB	latin1_swedish_ci	16.0 KiB	0 B

Figure 6: database snapshot



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 0.0004 seconds.)

```
SELECT * FROM `id`
```

Stud_ID	Stud_Course	Lib_Incharge	Day_late	Total_charge_RM
14402	CDCS110	Mr. Kairul Aming	3	1.5
14402	CDIM144	Mr. Kairul Aming	4	2.0

Figure 7: database snapshot

6.0 CONCLUSION

To summarize everything from this assignment, I learned how to build a simple basic coding for my system called "Notice Payment" as well as what is Python and SQL (database). Not only that, but during the assignment, my knowledge about Python seemed to emerge, which made me know how to detect an error while using Python and even why I can't connect to the database (it's either your Apache or MySQL stop function while you're on).

While doing this assignment too, I encountered several difficulties when working on this project especially in the coding part. It takes me longer to finish the code than it does to finish the database. But I wouldn't say it was simple for the database either. There is some time where my database couldn't work. However, after some consultation with my lecturer, Sir Airul Shazwan Bin Norshahimi and my classmates, the problem has been solved without no time. I would want to thank my lecturer and my friends for their assistance.