COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS
UNIVERSITI TEKNOLOGI MARA
MERBOK, KEDAH

DIPLOMA IN LIBRARY INFORMATICS
(IM144)

PROGRAMMING FOR LIBRARIES
(IML208)

GROUP ASSINGMENT: “SPORT CENTRE”

PREPARED BY:

NAME	MATRICS NUMBER
BALQIS BATRISYIA BINTI AZHAROLKHAIR	2022818714
HAWA NABILA BINTI HANIF	2022625444
NUR ALYA IXORA BINTI ROSLAN	2022661708
NUR ATIFAH SOPHIA BINTI SABRY	2022457012

CLASS:
KCDIM1443F

PREPARED FOR:
SIR AIRUL SHAZWAN BIN NORSHAHIMI

SUBMISSION DATE:
17 JANUARY 2024

GROUP ASSINGMENT: “SPORT CENTRE”

BALQIS BATRISYIA BINTI AZHAROLKHAIR	2022818714
HAWA NABILA BINTI HANIF	2022625444
NUR ALYA IXORA BINTI ROSLAN	2022661708
NUR ATIFAH SOPHIA BINTI SABRY	2022457012

KCDIM1443F

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS
UNIVERSITI TEKNOLOGI MARA
MERBOK, KEDAH

17 JANUARY 2024

STUDENT PLEDGE OF ACADEMIC INTEGRITY

As a student of Universiti Teknologi MARA (UiTM), it is my responsibility to act in accordance with UiTM's academic assessment and evaluation policy. I hereby pledge to act and uphold academic integrity and pursue scholarly activities in UiTM with honesty andresponsible manner. I will not engage or tolerate acts of academic dishonesty, academic misconduct, or academic fraud including but not limited to:

a. **Cheating:** Using or attempt to use any unauthorized device, assistance, sources, practice or materials while completing academic assessments. This include but not limited to copying from another, allowing others to copy, unauthorized collaboration on an assignment or open book tests, or engaging in any act or conduct that can be construed as cheating.

b. **Plagiarism:** Using or attempts to use the work of others (ideas, design, words, art, music, etc.) without acknowledging the source; using or purchasing materials prepared by another person or agency or engaging in other behavior that a reasonable person would consider as plagiarism.

c. **Fabrication:** Falsifying data, information, or citations in any academic assessment and evaluation.

d. **Deception:** Providing false information with intend to deceive an instructor concerning any academic assessment and evaluation.

e. **Furnishing false information:** Providing false information or false representationto any UiTM official, instructor, or office.

With this pledge, I am fully aware that I am obliged to conduct myself with utmost honesty and integrity. I fully understand that a disciplinary action can be taken against me if I, in anymanner, violate this pledge.

Name: Balqis Batrisyia binti Azharolkhair

Matric Number: 2022818714

Course Code: IML208

Programme code: CDIM144

Faculty / Campus: UITM Kampus Sungai Petani

*Students are required to sign one pledge for each course taken.

STUDENT PLEDGE OF ACADEMIC INTEGRITY

As a student of Universiti Teknologi MARA (UiTM), it is my responsibility to act in accordance with UiTM's academic assessment and evaluation policy. I hereby pledge to act and uphold academic integrity and pursue scholarly activities in UiTM with honesty andresponsible manner. I will not engage or tolerate acts of academic dishonesty, academic misconduct, or academic fraud including but not limited to:

a. **Cheating:** Using or attempt to use any unauthorized device, assistance, sources, practice or materials while completing academic assessments. This include but not limited to copying from another, allowing others to copy, unauthorized collaboration on an assignment or open book tests, or engaging in any act or conduct that can be construed as cheating.
b. **Plagiarism:** Using or attempts to use the work of others (ideas, design, words, art, music, etc.) without acknowledging the source; using or purchasing materials prepared by another person or agency or engaging in other behavior that a reasonable person would consider as plagiarism.
c. **Fabrication:** Falsifying data, information, or citations in any academic assessment and evaluation.
d. **Deception:** Providing false information with intend to deceive an instructor concerning any academic assessment and evaluation.
e. **Furnishing false information:** Providing false information or false representationto any UiTM official, instructor, or office.

With this pledge, I am fully aware that I am obliged to conduct myself with utmost honesty and integrity. I fully understand that a disciplinary action can be taken against me if I, in anymanner, violate this pledge.

Name: Hawa Nabila binti Hanif

Matric Number: 2022625444

Course Code: IML208

Programme code: CDIM144

Faculty / Campus: UITM Kampus Sungai Petani

*Students are required to sign one pledge for each course taken.

STUDENT PLEDGE OF ACADEMIC INTEGRITY

As a student of Universiti Teknologi MARA (UiTM), it is my responsibility to act in accordance with UiTM's academic assessment and evaluation policy. I hereby pledge to act and uphold academic integrity and pursue scholarly activities in UiTM with honesty andresponsible manner. I will not engage or tolerate acts of academic dishonesty, academic misconduct, or academic fraud including but not limited to:

a. **Cheating:** Using or attempt to use any unauthorized device, assistance, sources, practice or materials while completing academic assessments. This include but not limited to copying from another, allowing others to copy, unauthorized collaboration on an assignment or open book tests, or engaging in any act or conduct that can be construed as cheating.

b. **Plagiarism:** Using or attempts to use the work of others (ideas, design, words, art, music, etc.) without acknowledging the source; using or purchasing materials prepared by another person or agency or engaging in other behavior that a reasonable person would consider as plagiarism.

c. **Fabrication:** Falsifying data, information, or citations in any academic assessment and evaluation.

d. **Deception:** Providing false information with intend to deceive an instructor concerning any academic assessment and evaluation.

e. **Furnishing false information:** Providing false information or false representationto any UiTM official, instructor, or office.

With this pledge, I am fully aware that I am obliged to conduct myself with utmost honesty and integrity. I fully understand that a disciplinary action can be taken against me if I, in anymanner, violate this pledge.

Name: Nur Alya Ixora binti Roslan

Matric Number: 2022661708

Course Code: IML208

Programme code: CDIM144

Faculty / Campus: UITM Kampus Sungai Petani

*Students are required to sign one pledge for each course taken.

Bahagian Pentaksiran dan Penilaian Akademik 2021

STUDENT PLEDGE OF ACADEMIC INTEGRITY

As a student of Universiti Teknologi MARA (UiTM), it is my responsibility to act in accordance with UiTM's academic assessment and evaluation policy. I hereby pledge to act and uphold academic integrity and pursue scholarly activities in UiTM with honesty andresponsible manner. I will not engage or tolerate acts of academic dishonesty, academic misconduct, or academic fraud including but not limited to:

a. **Cheating:** Using or attempt to use any unauthorized device, assistance, sources, practice or materials while completing academic assessments. This include but not limited to copying from another, allowing others to copy, unauthorized collaboration on an assignment or open book tests, or engaging in any act or conduct that can be construed as cheating.
b. **Plagiarism:** Using or attempts to use the work of others (ideas, design, words, art, music, etc.) without acknowledging the source; using or purchasing materials prepared by another person or agency or engaging in other behavior that a reasonable person would consider as plagiarism.
c. **Fabrication:** Falsifying data, information, or citations in any academic assessment and evaluation.
d. **Deception:** Providing false information with intend to deceive an instructor concerning any academic assessment and evaluation.
e. **Furnishing false information:** Providing false information or false representationto any UiTM official, instructor, or office.

With this pledge, I am fully aware that I am obliged to conduct myself with utmost honesty and integrity. I fully understand that a disciplinary action can be taken against me if I, in anymanner, violate this pledge.

Name: Nur Atifah Sophia binti Sabry

Matric Number: 2022457012

Course Code: IML208

Programme code: CDIM144

Faculty / Campus: UITM Kampus Sungai Petani

*Students are required to sign one pledge for each course taken.

ACKNOWLEDGEMENT

First and foremost, praises and thanks to Allah for his showers of blessings throughout our task to complete the assignment successfully.

To everyone who helped to finish this report, I would like to extend my deepest gratitude. Throughout this process, their assistance, direction, and encouragement have been priceless.

I would like to express my sincere gratitude to Sir Airul Shazwan Bin Norshahimi , our beloved lecturer, for his steadfast support and astute direction. His knowledge and helpful criticism have been very helpful in determining the focus and substance of this study.

Additionally, I want to express my gratitude to my team members for their assistance and contributions. Their commitment and diligence have greatly improved the quality of this report. I am thankful of my friends' and family's patience, support, and understanding during this project's challenging stages. Their unwavering assistance has been as a continual source of inspiration.

In conclusion, we would like to thank all the individuals who helped us in this journey. Your contributions were instrumental in the successful completion of this assignment, and we are deeply grateful for your support. Thank you.

TABLE OF CONTENT

1.0 INTRODCTION

To begin with, the Sports Centre is a facility that provides sports equipment, places, and services to the user. This system of the Sports Centre includes user information, facility management, and reporting damaged tools to ensure that the centre's activities are running smoothly and systematically. In addition, this system aims to be implemented in academic institutions, mainly to help the centre management, which is usually managed by staff and students as helpers. Furthermore, it is user-friendly because this system is easy to handle and adapt to the management.

To be clear, users who want to use the facilities provided by the centre need to fill in their information in the form and submit it so that the data will be recorded in the database. Subsequently, users can choose the court or place they want to use depending on their availability and the equipment they need. All this information is needed to ensure the users book the place and the equipment before use. Thus, there will be less risk of overlapping bookings that eventually lead to other problems of dissatisfaction among the users.

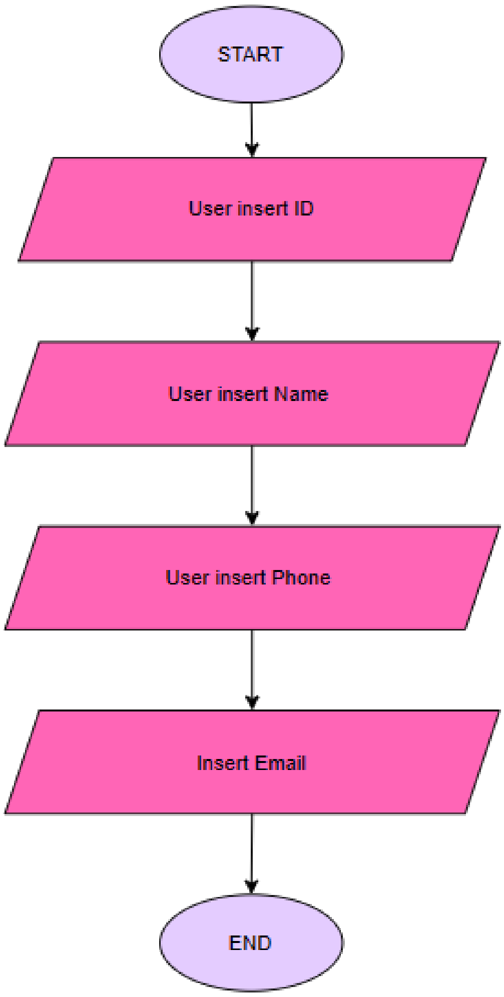
Other than that, there is also a platform to report damaged tools. In this case, the staff or person in responsibility need to fill in the information about the item that has been broken or damaged. The item type and total number of damaged items must be inserted to calculate the total price. The item type, total damages and the total price will be displayed at the end of the form so that the staff can be informed about the report they made as a summary. All the data will be recorded in the database to ensure that the management can keep track of the condition of all their equipment and provide the best services and facilities to the users.

1.1 OBJECTIVES

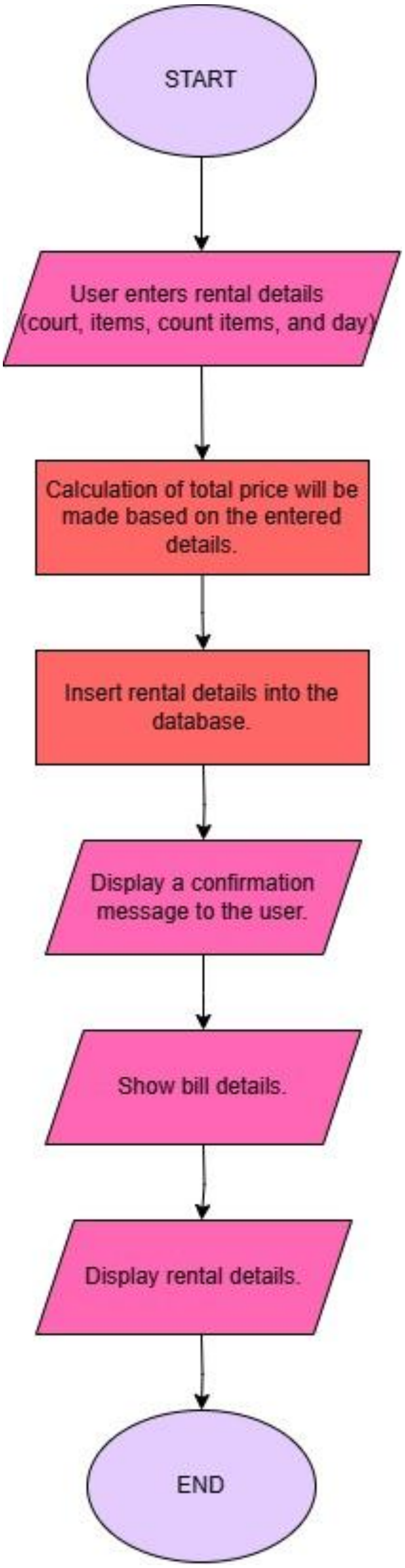
1. To lend items or courts that available.
2. To improve the lending system of the sport centre.
3. To motivate students to exercise more or play sports.
4. Our target user: University students

2.0 FLOWCHART

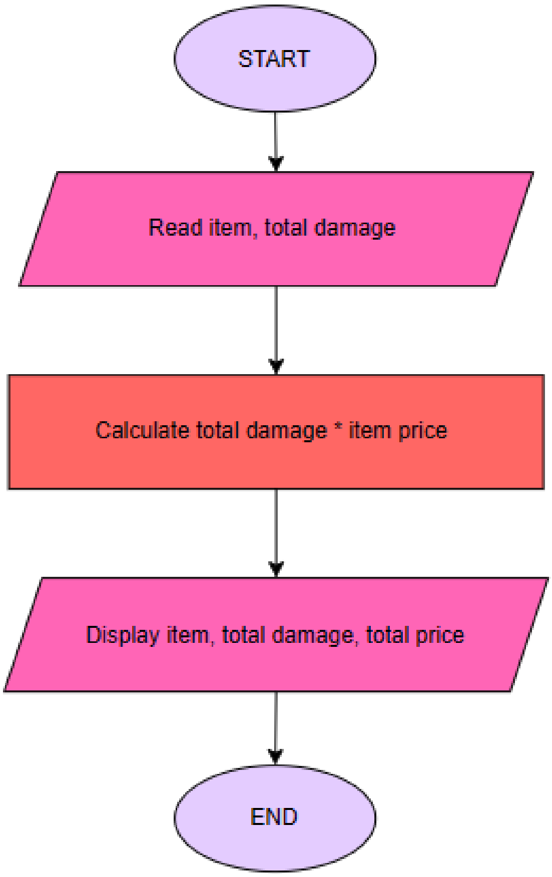
2.1 USER ACCOUNT



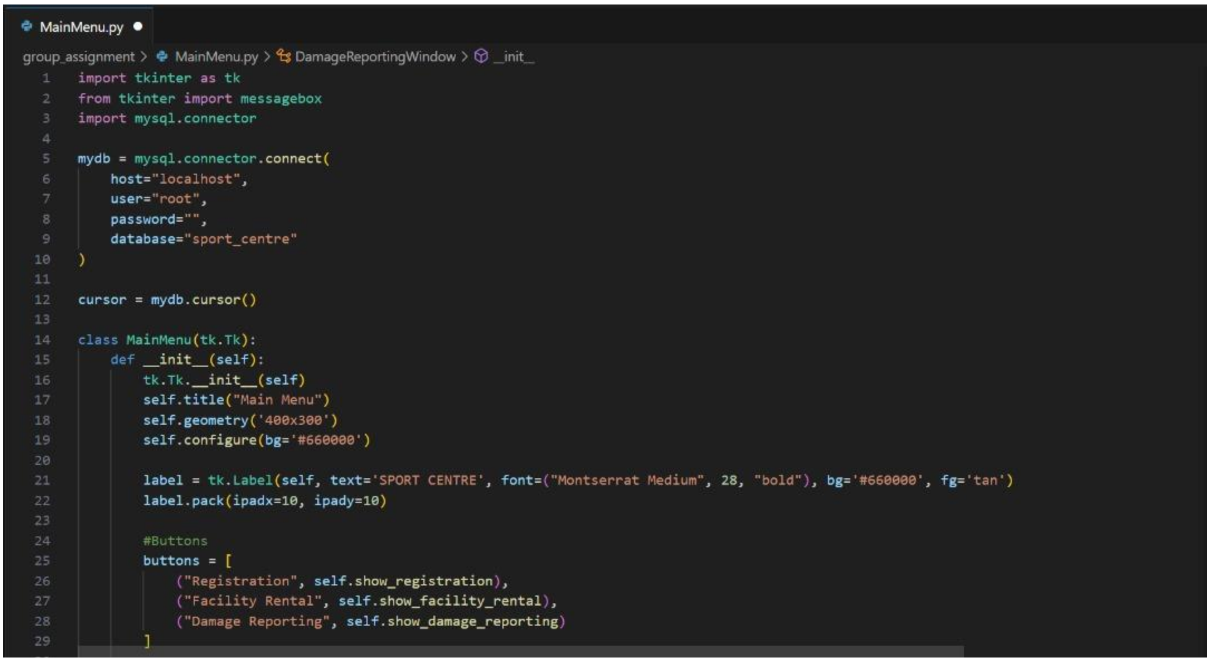
2.2 FACILITY MANAGEMENT


START
User enters rental details (court, items, count items, and day)
Calculation of total price will be made based on the entered details.
Insert rental details into the database.
Display a confirmation message to the user.
Show bill details.
Display rental details.
END

2.3 REPORTING TOOLS

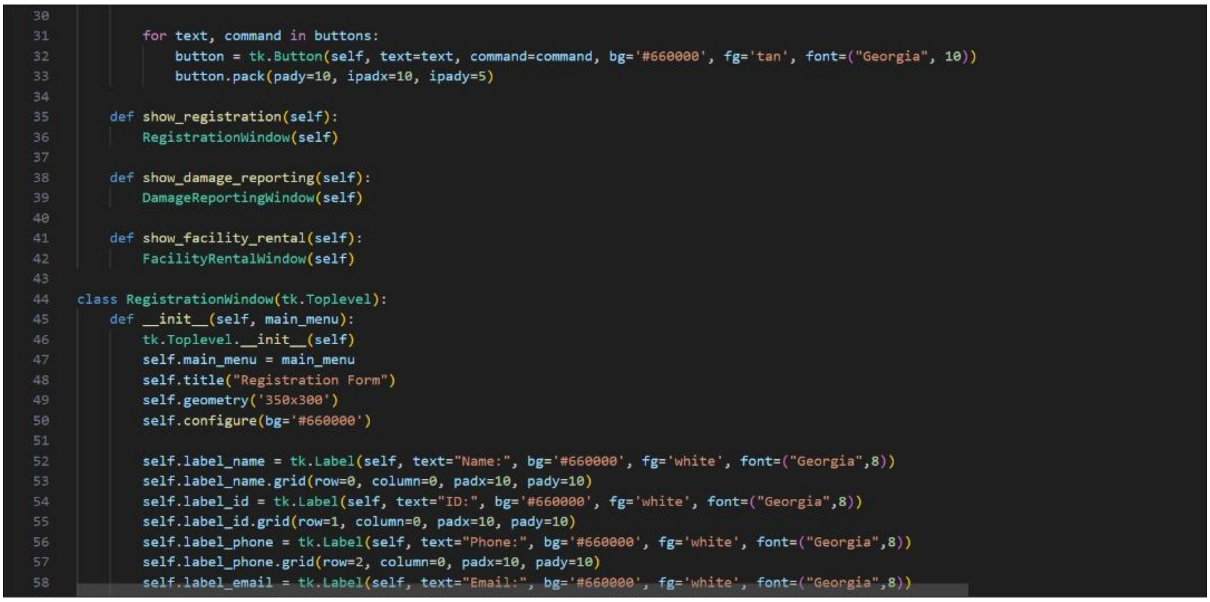

START
Read item, total damage
Calculate total damage * item price
Display item, total damage, total price
END

3.0 CODE

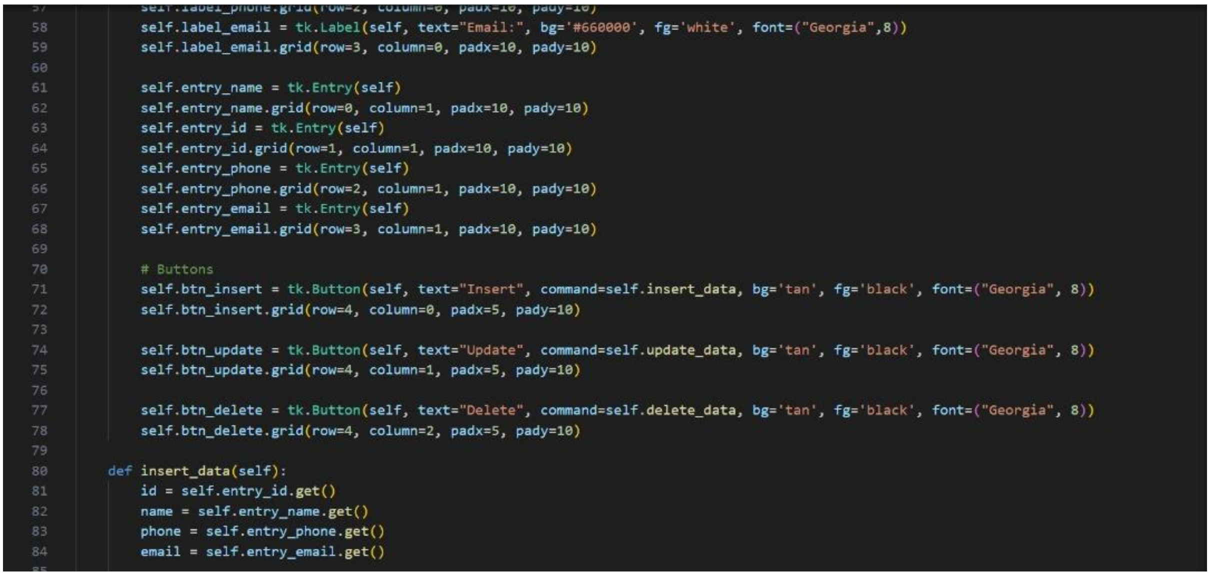


MainMenu.py ●
group_assignment > MainMenu.py > DamageReportingWindow > __init__

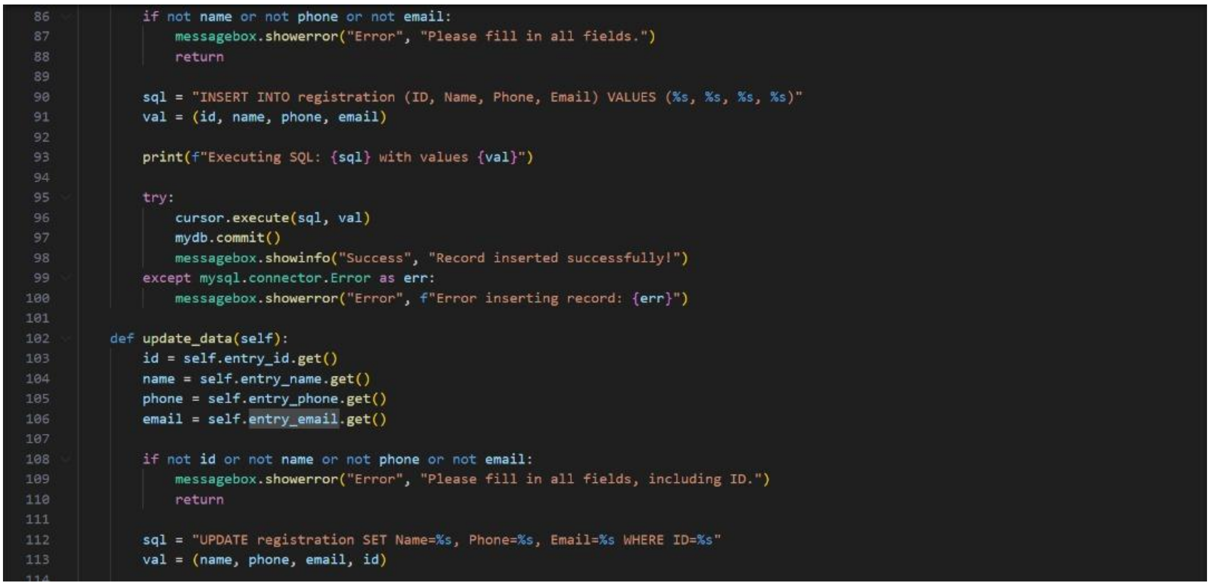
```
import tkinter as tk
from tkinter import messagebox
import mysql.connector

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

cursor = mydb.cursor()

class MainMenu(tk.Tk):
    def __init__(self):
        tk.Tk.__init__(self)
        self.title("Main Menu")
        self.geometry('400x300')
        self.configure(bg='#660000')

        label = tk.Label(self, text='SPORT CENTRE', font=("Montserrat Medium", 28, "bold"), bg='#660000', fg='tan')
        label.pack(ipadx=10, ipady=10)

        #Buttons
        buttons = [
            ("Registration", self.show_registration),
            ("Facility Rental", self.show_facility_rental),
            ("Damage Reporting", self.show_damage_reporting)
        ]
```



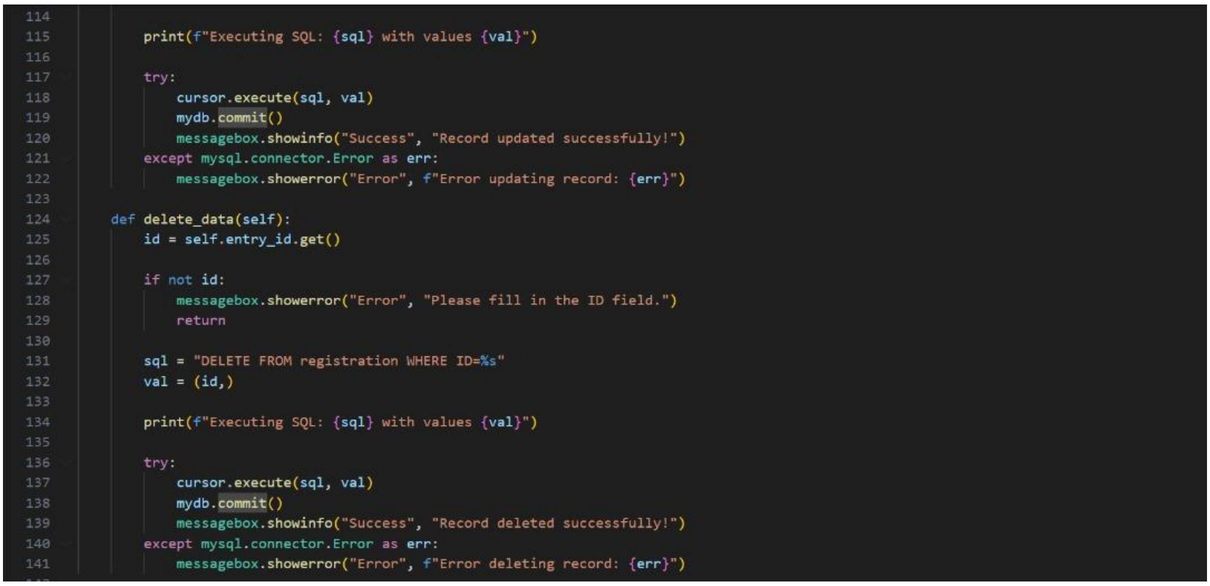
```

        for text, command in buttons:
            button = tk.Button(self, text=text, command=command, bg='#660000', fg='tan', font=("Georgia", 10))
            button.pack(pady=10, ipadx=10, ipady=5)

    def show_registration(self):
        RegistrationWindow(self)

    def show_damage_reporting(self):
        DamageReportingWindow(self)

    def show_facility_rental(self):
        FacilityRentalWindow(self)

class RegistrationWindow(tk.Toplevel):
    def __init__(self, main_menu):
        tk.Toplevel.__init__(self)
        self.main_menu = main_menu
        self.title("Registration Form")
        self.geometry('350x300')
        self.configure(bg='#660000')

        self.label_name = tk.Label(self, text="Name:", bg='#660000', fg='white', font=("Georgia",8))
        self.label_name.grid(row=0, column=0, padx=10, pady=10)
        self.label_id = tk.Label(self, text="ID:", bg='#660000', fg='white', font=("Georgia",8))
        self.label_id.grid(row=1, column=0, padx=10, pady=10)
        self.label_phone = tk.Label(self, text="Phone:", bg='#660000', fg='white', font=("Georgia",8))
        self.label_phone.grid(row=2, column=0, padx=10, pady=10)
        self.label_email = tk.Label(self, text="Email:", bg='#660000', fg='white', font=("Georgia",8))
```



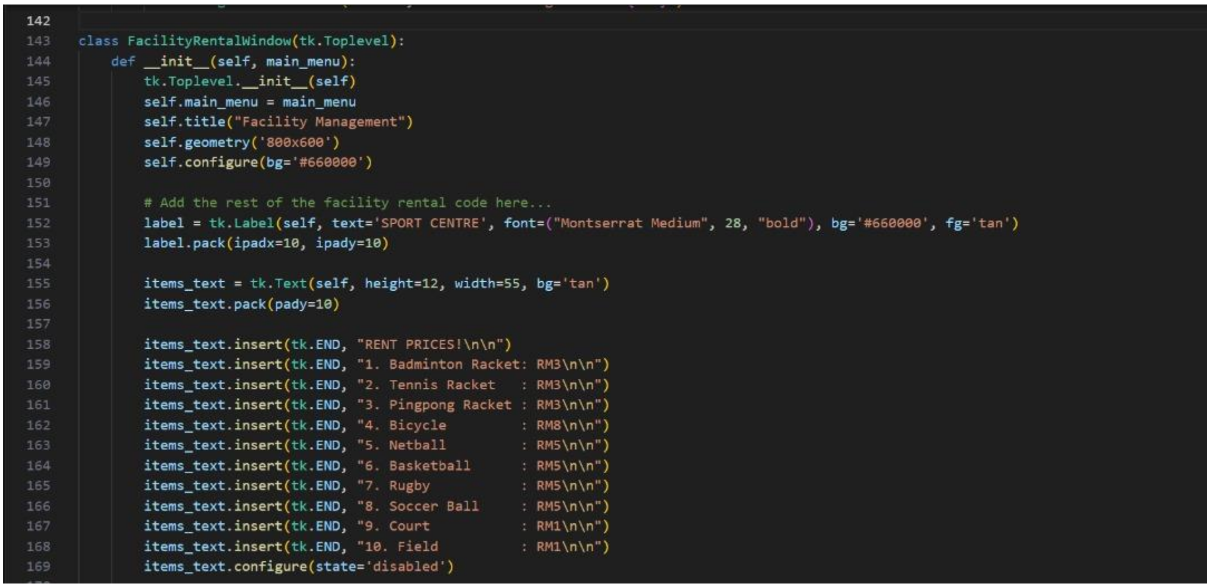
```
        self.label_phone.grid(row=2, column=0, padx=10, pady=10)
        self.label_email = tk.Label(self, text="Email:", bg='#660000', fg='white', font=("Georgia",8))
        self.label_email.grid(row=3, column=0, padx=10, pady=10)

        self.entry_name = tk.Entry(self)
        self.entry_name.grid(row=0, column=1, padx=10, pady=10)
        self.entry_id = tk.Entry(self)
        self.entry_id.grid(row=1, column=1, padx=10, pady=10)
        self.entry_phone = tk.Entry(self)
        self.entry_phone.grid(row=2, column=1, padx=10, pady=10)
        self.entry_email = tk.Entry(self)
        self.entry_email.grid(row=3, column=1, padx=10, pady=10)

        # Buttons
        self.btn_insert = tk.Button(self, text="Insert", command=self.insert_data, bg='tan', fg='black', font=("Georgia", 8))
        self.btn_insert.grid(row=4, column=0, padx=5, pady=10)

        self.btn_update = tk.Button(self, text="Update", command=self.update_data, bg='tan', fg='black', font=("Georgia", 8))
        self.btn_update.grid(row=4, column=1, padx=5, pady=10)

        self.btn_delete = tk.Button(self, text="Delete", command=self.delete_data, bg='tan', fg='black', font=("Georgia", 8))
        self.btn_delete.grid(row=4, column=2, padx=5, pady=10)

    def insert_data(self):
        id = self.entry_id.get()
        name = self.entry_name.get()
        phone = self.entry_phone.get()
        email = self.entry_email.get()
```



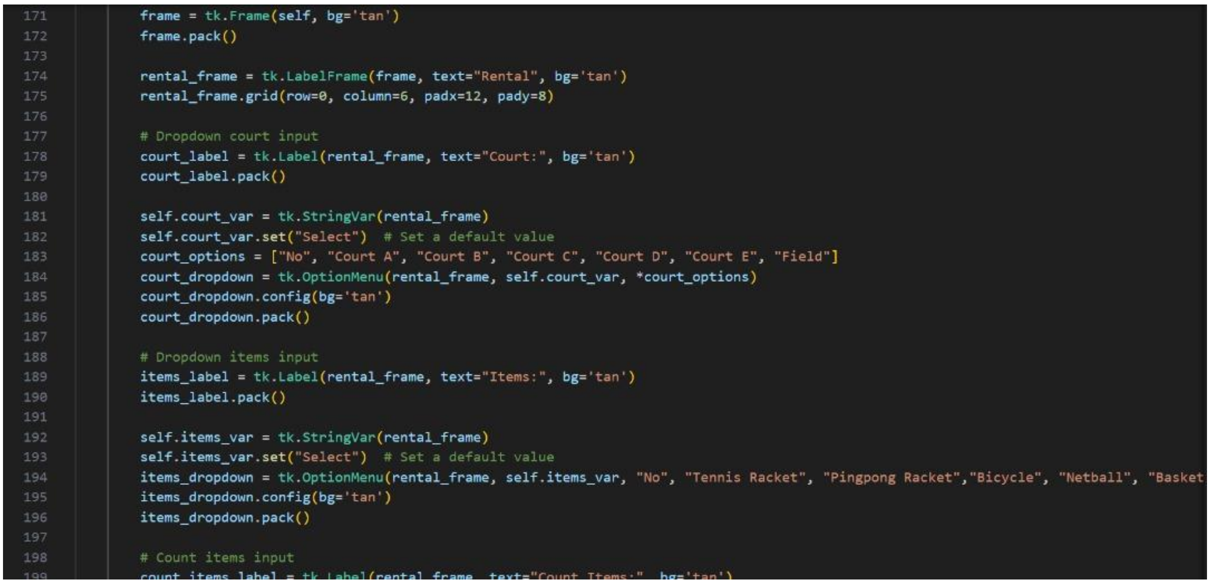
```
        if not name or not phone or not email:
            messagebox.showerror("Error", "Please fill in all fields.")
            return

        sql = "INSERT INTO registration (ID, Name, Phone, Email) VALUES (%s, %s, %s, %s)"
        val = (id, name, phone, email)

        print(f"Executing SQL: {sql} with values {val}")

        try:
            cursor.execute(sql, val)
            mydb.commit()
            messagebox.showinfo("Success", "Record inserted successfully!")
        except mysql.connector.Error as err:
            messagebox.showerror("Error", f"Error inserting record: {err}")

    def update_data(self):
        id = self.entry_id.get()
        name = self.entry_name.get()
        phone = self.entry_phone.get()
        email = self.entry_email.get()

        if not id or not name or not phone or not email:
            messagebox.showerror("Error", "Please fill in all fields, including ID.")
            return

        sql = "UPDATE registration SET Name=%s, Phone=%s, Email=%s WHERE ID=%s"
        val = (name, phone, email, id)
```



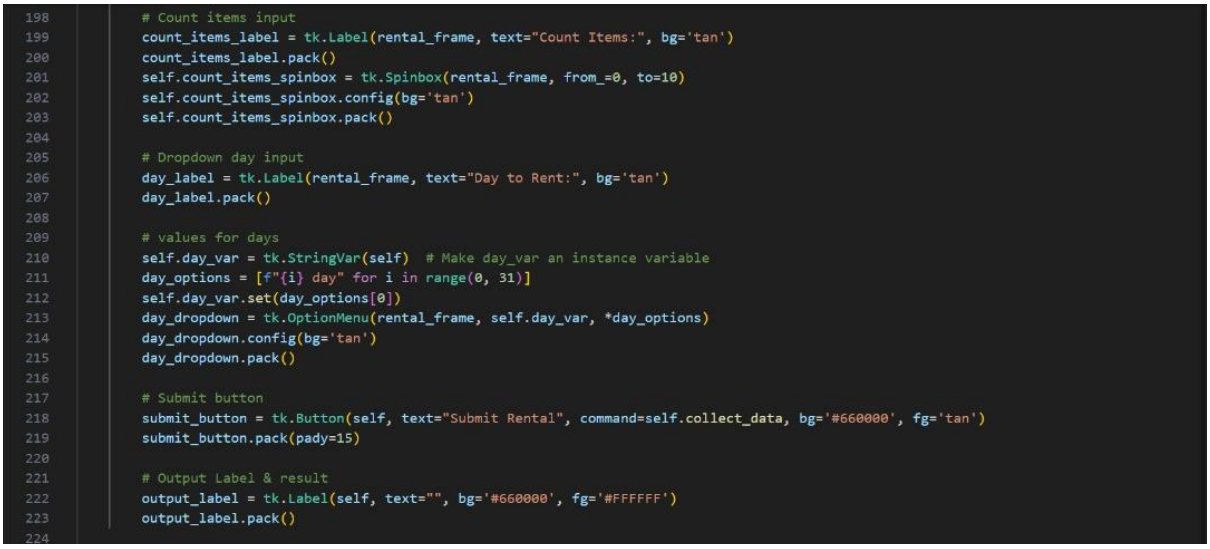
```
        print(f"Executing SQL: {sql} with values {val}")

        try:
            cursor.execute(sql, val)
            mydb.commit()
            messagebox.showinfo("Success", "Record updated successfully!")
        except mysql.connector.Error as err:
            messagebox.showerror("Error", f"Error updating record: {err}")

    def delete_data(self):
        id = self.entry_id.get()

        if not id:
            messagebox.showerror("Error", "Please fill in the ID field.")
            return

        sql = "DELETE FROM registration WHERE ID=%s"
        val = (id,)

        print(f"Executing SQL: {sql} with values {val}")

        try:
            cursor.execute(sql, val)
            mydb.commit()
            messagebox.showinfo("Success", "Record deleted successfully!")
        except mysql.connector.Error as err:
            messagebox.showerror("Error", f"Error deleting record: {err}")
```



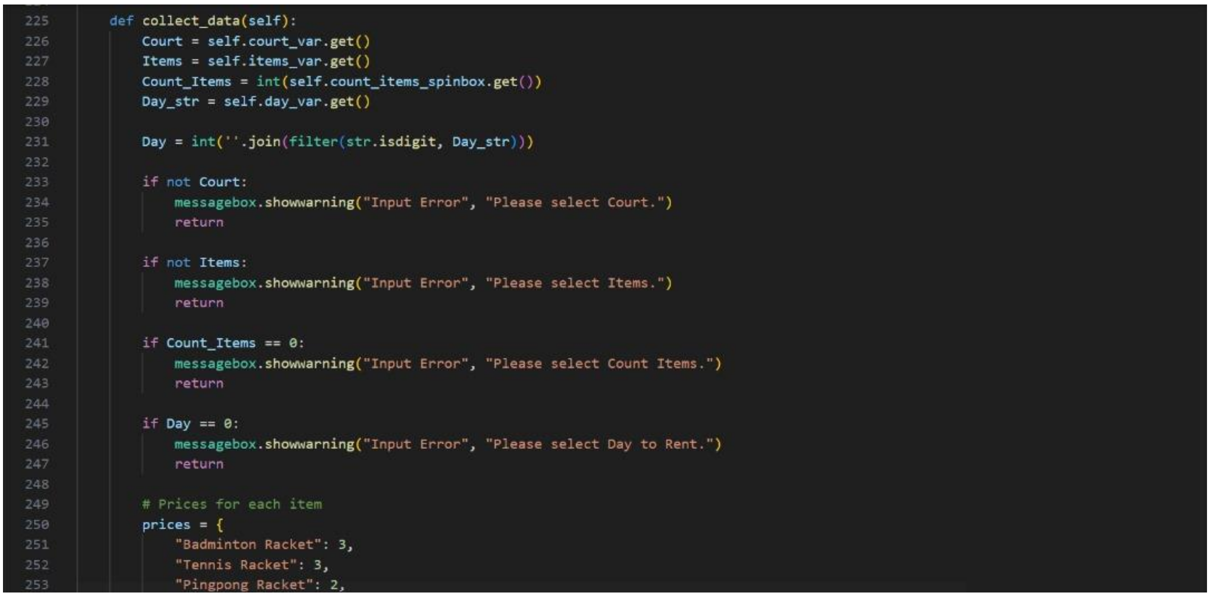
```

class FacilityRentalWindow(tk.Toplevel):
    def __init__(self, main_menu):
        tk.Toplevel.__init__(self)
        self.main_menu = main_menu
        self.title("Facility Management")
        self.geometry('800x600')
        self.configure(bg='#660000')

        # Add the rest of the facility rental code here...
        label = tk.Label(self, text='SPORT CENTRE', font=("Montserrat Medium", 28, "bold"), bg='#660000', fg='tan')
        label.pack(ipadx=10, ipady=10)

        items_text = tk.Text(self, height=12, width=55, bg='tan')
        items_text.pack(pady=10)

        items_text.insert(tk.END, "RENT PRICES!\n\n")
        items_text.insert(tk.END, "1. Badminton Racket: RM3\n\n")
        items_text.insert(tk.END, "2. Tennis Racket   : RM3\n\n")
        items_text.insert(tk.END, "3. Pingpong Racket : RM3\n\n")
        items_text.insert(tk.END, "4. Bicycle         : RM8\n\n")
        items_text.insert(tk.END, "5. Netball         : RM5\n\n")
        items_text.insert(tk.END, "6. Basketball      : RM5\n\n")
        items_text.insert(tk.END, "7. Rugby           : RM5\n\n")
        items_text.insert(tk.END, "8. Soccer Ball     : RM5\n\n")
        items_text.insert(tk.END, "9. Court           : RM1\n\n")
        items_text.insert(tk.END, "10. Field          : RM1\n\n")
        items_text.configure(state='disabled')
```



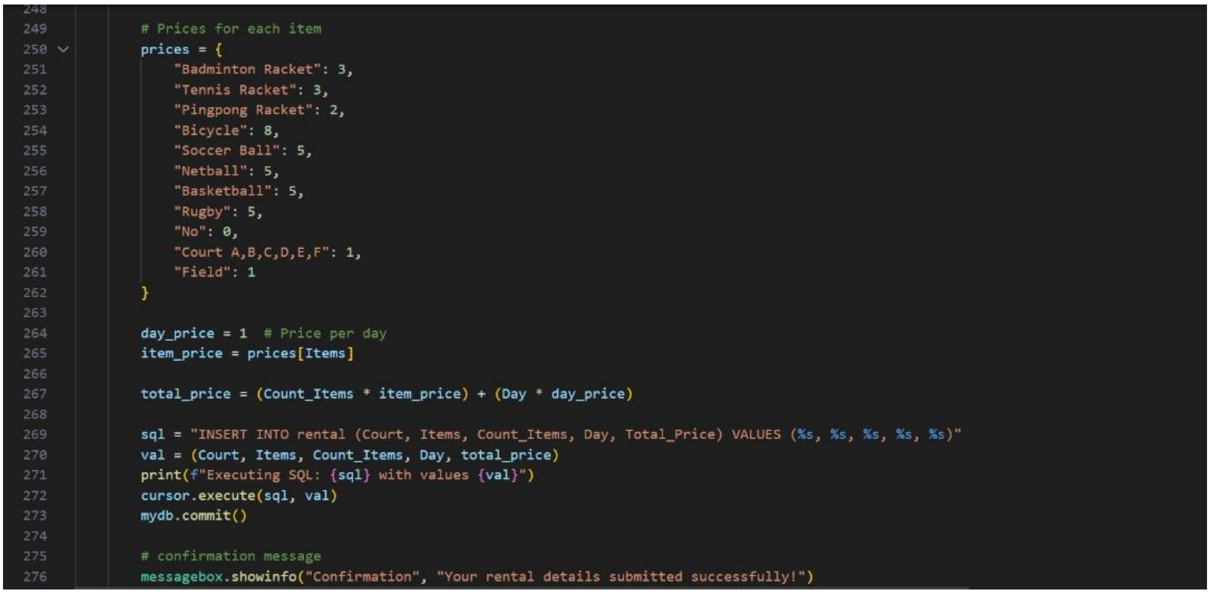
```
        frame = tk.Frame(self, bg='tan')
        frame.pack()

        rental_frame = tk.LabelFrame(frame, text="Rental", bg='tan')
        rental_frame.grid(row=0, column=6, padx=12, pady=8)

        # Dropdown court input
        court_label = tk.Label(rental_frame, text="Court:", bg='tan')
        court_label.pack()

        self.court_var = tk.StringVar(rental_frame)
        self.court_var.set("Select")  # Set a default value
        court_options = ["No", "Court A", "Court B", "Court C", "Court D", "Court E", "Field"]
        court_dropdown = tk.OptionMenu(rental_frame, self.court_var, *court_options)
        court_dropdown.config(bg='tan')
        court_dropdown.pack()

        # Dropdown items input
        items_label = tk.Label(rental_frame, text="Items:", bg='tan')
        items_label.pack()

        self.items_var = tk.StringVar(rental_frame)
        self.items_var.set("Select")  # Set a default value
        items_dropdown = tk.OptionMenu(rental_frame, self.items_var, "No", "Tennis Racket", "Pingpong Racket","Bicycle", "Netball", "Basket
        items_dropdown.config(bg='tan')
        items_dropdown.pack()

        # Count items input
        count_items_label = tk.Label(rental_frame, text="Count Items:", bg='tan')
```



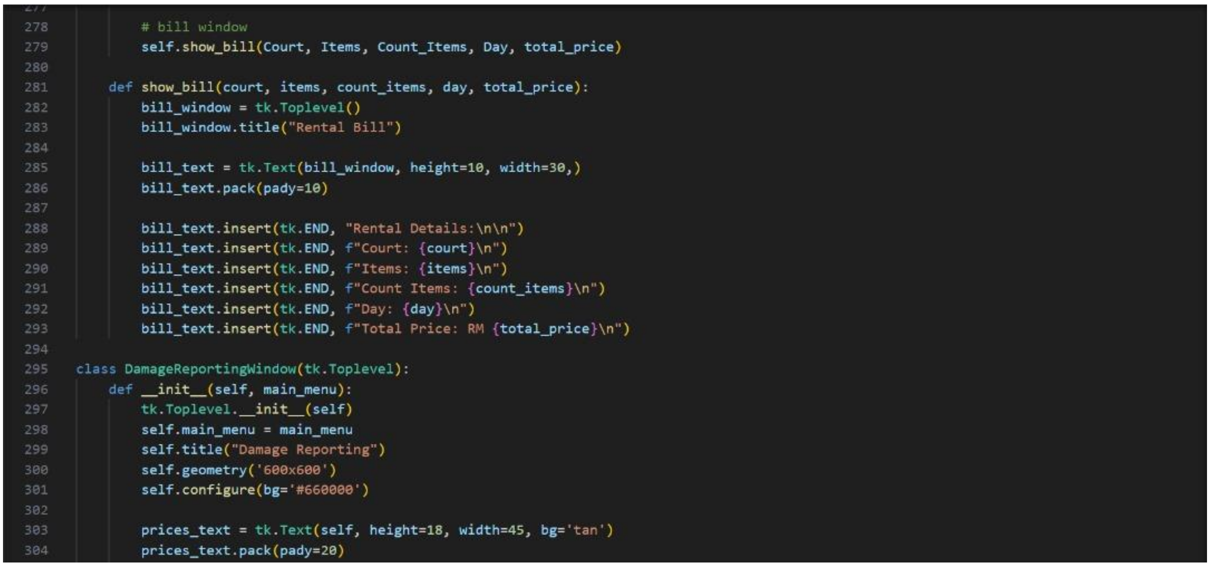
```
        # Count items input
        count_items_label = tk.Label(rental_frame, text="Count Items:", bg='tan')
        count_items_label.pack()
        self.count_items_spinbox = tk.Spinbox(rental_frame, from_=0, to=10)
        self.count_items_spinbox.config(bg='tan')
        self.count_items_spinbox.pack()

        # Dropdown day input
        day_label = tk.Label(rental_frame, text="Day to Rent:", bg='tan')
        day_label.pack()

        # values for days
        self.day_var = tk.StringVar(self)  # Make day_var an instance variable
        day_options = [f"{i} day" for i in range(0, 31)]
        self.day_var.set(day_options[0])
        day_dropdown = tk.OptionMenu(rental_frame, self.day_var, *day_options)
        day_dropdown.config(bg='tan')
        day_dropdown.pack()

        # Submit button
        submit_button = tk.Button(self, text="Submit Rental", command=self.collect_data, bg='#660000', fg='tan')
        submit_button.pack(pady=15)

        # Output Label & result
        output_label = tk.Label(self, text="", bg='#660000', fg='#FFFFFF')
        output_label.pack()
```



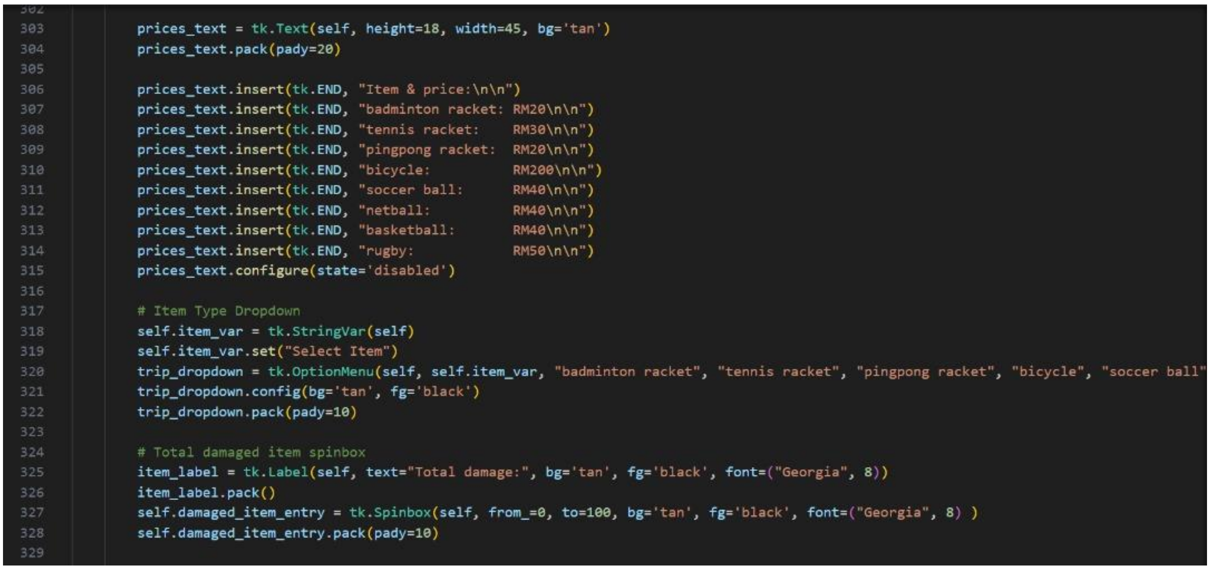
```python
    def collect_data(self):
        Court = self.court_var.get()
        Items = self.items_var.get()
        Count_Items = int(self.count_items_spinbox.get())
        Day_str = self.day_var.get()

        Day = int(''.join(filter(str.isdigit, Day_str)))

        if not Court:
            messagebox.showwarning("Input Error", "Please select Court.")
            return

        if not Items:
            messagebox.showwarning("Input Error", "Please select Items.")
            return

        if Count_Items == 0:
            messagebox.showwarning("Input Error", "Please select Count Items.")
            return

        if Day == 0:
            messagebox.showwarning("Input Error", "Please select Day to Rent.")
            return

        # Prices for each item
        prices = {
            "Badminton Racket": 3,
            "Tennis Racket": 3,
            "Pingpong Racket": 2,
```



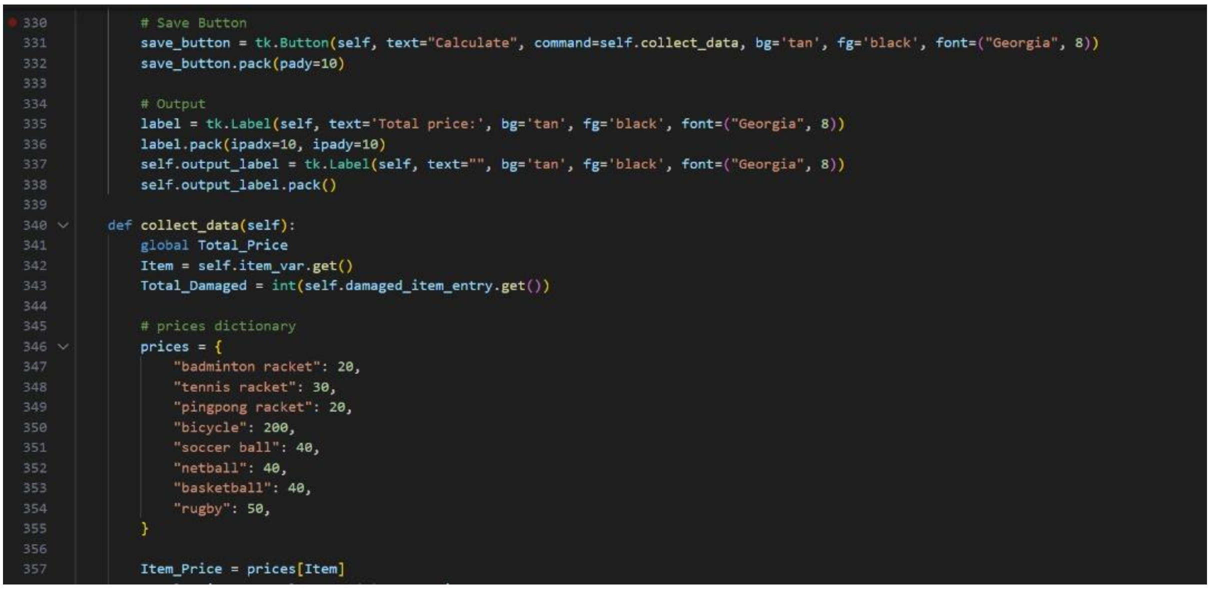
```python
        # Prices for each item
        prices = {
            "Badminton Racket": 3,
            "Tennis Racket": 3,
            "Pingpong Racket": 2,
            "Bicycle": 8,
            "Soccer Ball": 5,
            "Netball": 5,
            "Basketball": 5,
            "Rugby": 5,
            "No": 0,
            "Court A,B,C,D,E,F": 1,
            "Field": 1
        }

        day_price = 1  # Price per day
        item_price = prices[Items]

        total_price = (Count_Items * item_price) + (Day * day_price)

        sql = "INSERT INTO rental (Court, Items, Count_Items, Day, Total_Price) VALUES (%s, %s, %s, %s, %s)"
        val = (Court, Items, Count_Items, Day, total_price)
        print(f"Executing SQL: {sql} with values {val}")
        cursor.execute(sql, val)
        mydb.commit()

        # confirmation message
        messagebox.showinfo("Confirmation", "Your rental details submitted successfully!")
```



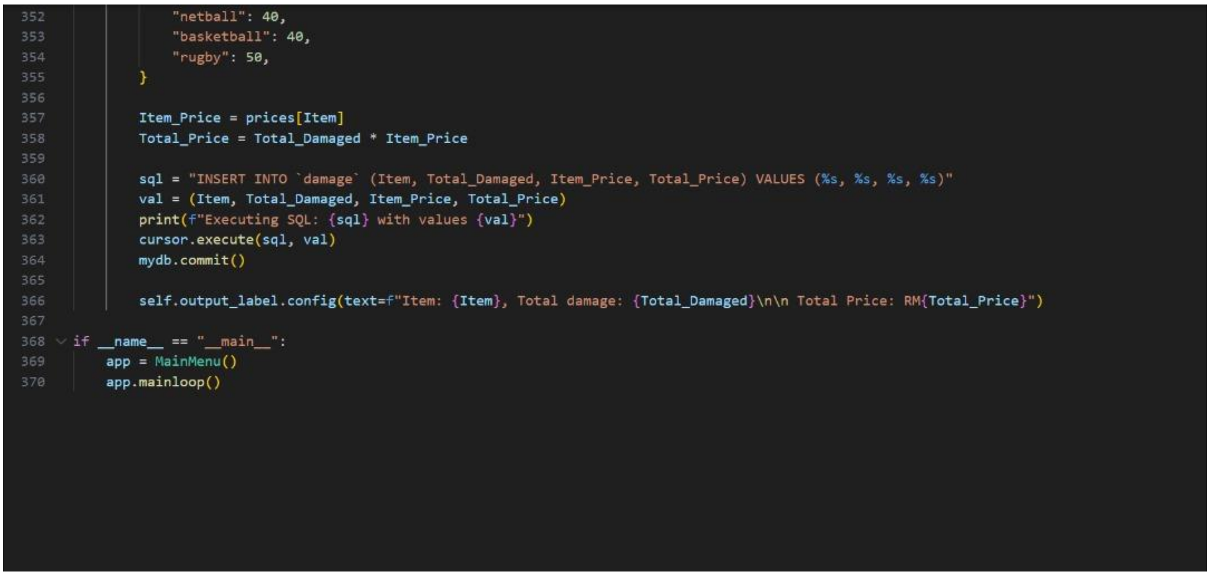
```
        # bill window
        self.show_bill(Court, Items, Count_Items, Day, total_price)

    def show_bill(court, items, count_items, day, total_price):
        bill_window = tk.Toplevel()
        bill_window.title("Rental Bill")

        bill_text = tk.Text(bill_window, height=10, width=30,)
        bill_text.pack(pady=10)

        bill_text.insert(tk.END, "Rental Details:\n\n")
        bill_text.insert(tk.END, f"Court: {court}\n")
        bill_text.insert(tk.END, f"Items: {items}\n")
        bill_text.insert(tk.END, f"Count Items: {count_items}\n")
        bill_text.insert(tk.END, f"Day: {day}\n")
        bill_text.insert(tk.END, f"Total Price: RM {total_price}\n")

class DamageReportingWindow(tk.Toplevel):
    def __init__(self, main_menu):
        tk.Toplevel.__init__(self)
        self.main_menu = main_menu
        self.title("Damage Reporting")
        self.geometry('600x600')
        self.configure(bg='#660000')

        prices_text = tk.Text(self, height=18, width=45, bg='tan')
        prices_text.pack(pady=20)
```



```
        prices_text = tk.Text(self, height=18, width=45, bg='tan')
        prices_text.pack(pady=20)

        prices_text.insert(tk.END, "Item & price:\n\n")
        prices_text.insert(tk.END, "badminton racket: RM20\n\n")
        prices_text.insert(tk.END, "tennis racket:    RM30\n\n")
        prices_text.insert(tk.END, "pingpong racket:  RM20\n\n")
        prices_text.insert(tk.END, "bicycle:          RM200\n\n")
        prices_text.insert(tk.END, "soccer ball:      RM40\n\n")
        prices_text.insert(tk.END, "netball:          RM40\n\n")
        prices_text.insert(tk.END, "basketball:       RM40\n\n")
        prices_text.insert(tk.END, "rugby:            RM50\n\n")
        prices_text.configure(state='disabled')

        # Item Type Dropdown
        self.item_var = tk.StringVar(self)
        self.item_var.set("Select Item")
        trip_dropdown = tk.OptionMenu(self, self.item_var, "badminton racket", "tennis racket", "pingpong racket", "bicycle", "soccer ball"
        trip_dropdown.config(bg='tan', fg='black')
        trip_dropdown.pack(pady=10)

        # Total damaged item spinbox
        item_label = tk.Label(self, text="Total damage:", bg='tan', fg='black', font=("Georgia", 8))
        item_label.pack()
        self.damaged_item_entry = tk.Spinbox(self, from_=0, to=100, bg='tan', fg='black', font=("Georgia", 8) )
        self.damaged_item_entry.pack(pady=10)

```



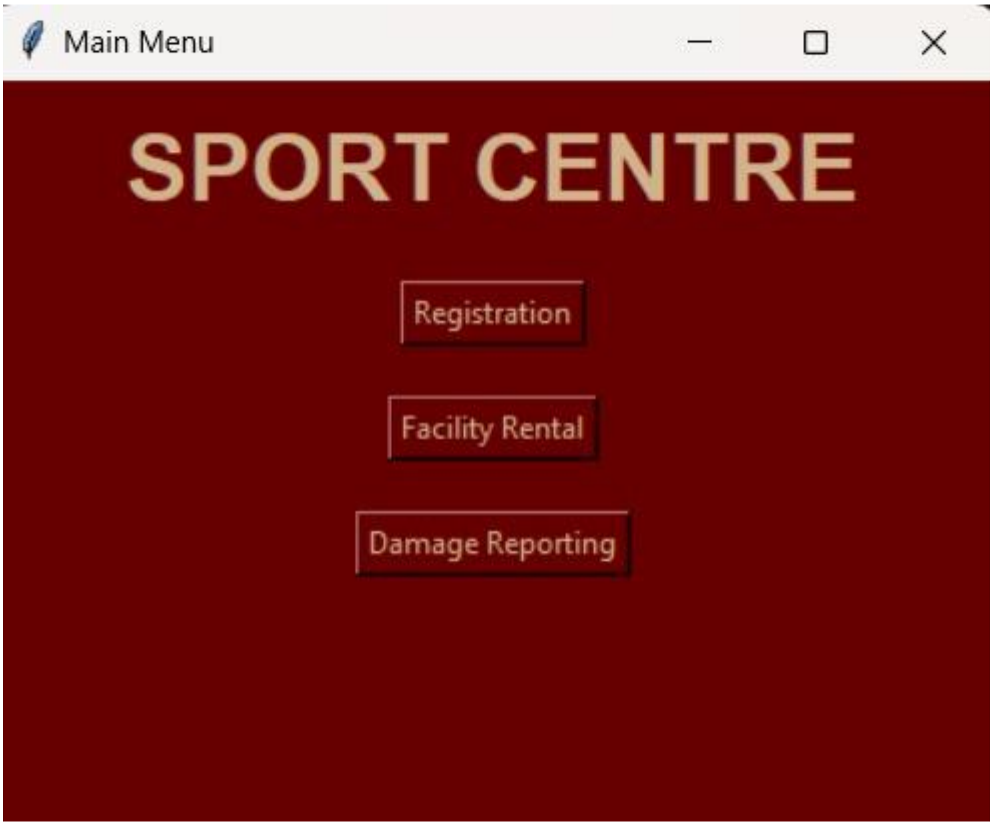
```
        # Save Button
        save_button = tk.Button(self, text="Calculate", command=self.collect_data, bg='tan', fg='black', font=("Georgia", 8))
        save_button.pack(pady=10)

        # Output
        label = tk.Label(self, text='Total price:', bg='tan', fg='black', font=("Georgia", 8))
        label.pack(ipadx=10, ipady=10)
        self.output_label = tk.Label(self, text="", bg='tan', fg='black', font=("Georgia", 8))
        self.output_label.pack()

    def collect_data(self):
        global Total_Price
        Item = self.item_var.get()
        Total_Damaged = int(self.damaged_item_entry.get())

        # prices dictionary
        prices = {
            "badminton racket": 20,
            "tennis racket": 30,
            "pingpong racket": 20,
            "bicycle": 200,
            "soccer ball": 40,
            "netball": 40,
            "basketball": 40,
            "rugby": 50,
        }

        Item_Price = prices[Item]
```

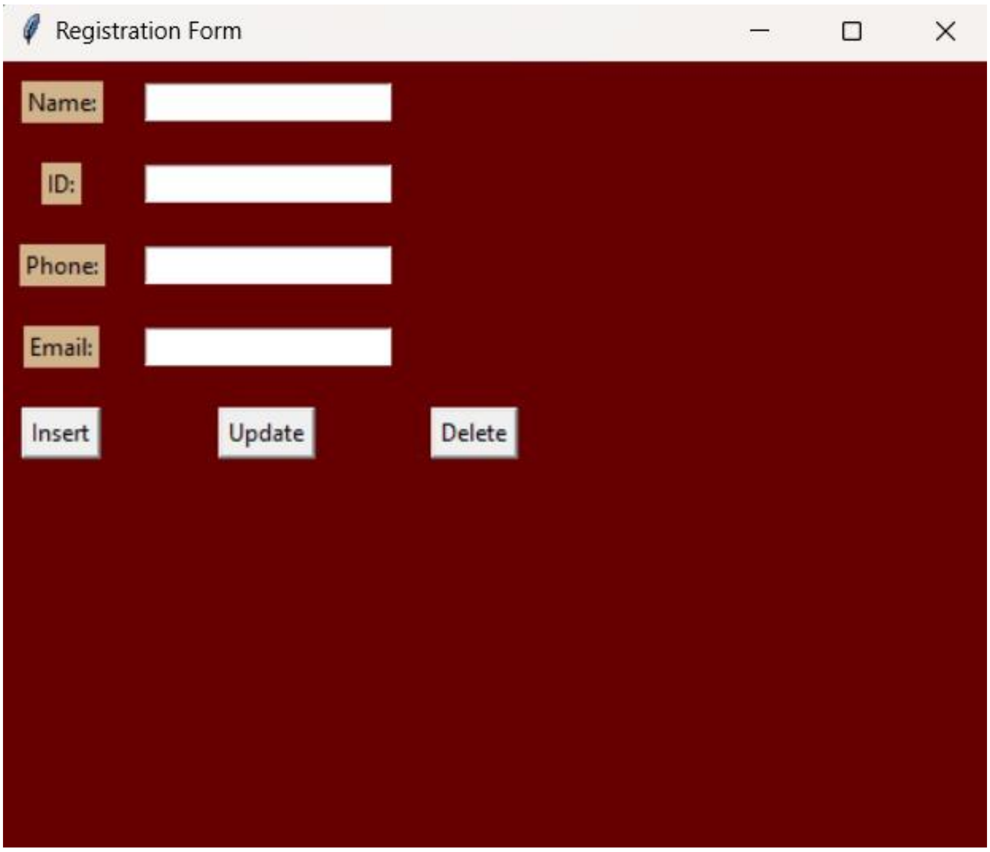


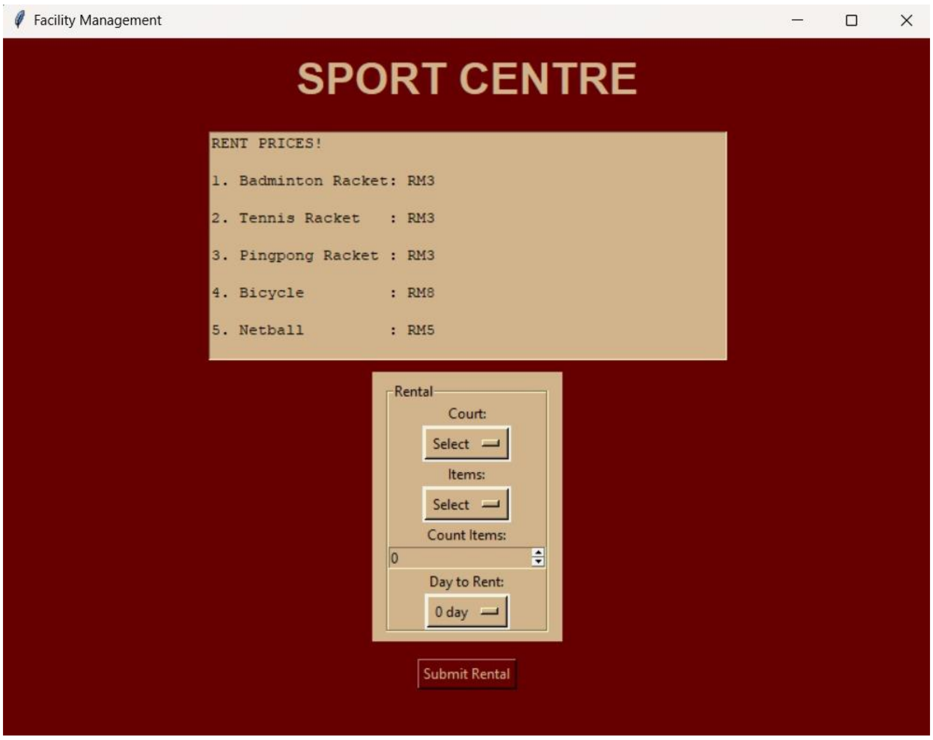
```
            "netball": 40,
            "basketball": 40,
            "rugby": 50,
        }

        Item_Price = prices[Item]
```

```
        Total_Price = Total_Damaged * Item_Price

        sql = "INSERT INTO `damage` (Item, Total_Damaged, Item_Price, Total_Price) VALUES (%s, %s, %s, %s)"
        val = (Item, Total_Damaged, Item_Price, Total_Price)
        print(f"Executing SQL: {sql} with values {val}")
        cursor.execute(sql, val)
        mydb.commit()

        self.output_label.config(text=f"Item: {Item}, Total damage: {Total_Damaged}\n\n Total Price: RM{Total_Price}")

if __name__ == "__main__":
    app = MainMenu()
    app.mainloop()
```

4.0 GUI


Main Menu
SPORT CENTRE
Registration
Facility Rental
Damage Reporting


Registration Form
Name:
ID:
Phone:
Email:
Insert
Update
Delete



SPORT CENTRE

```
RENT PRICES!
1. Badminton Racket: RM3
2. Tennis Racket    : RM3
3. Pingpong Racket  : RM3
4. Bicycle          : RM8
5. Netball          : RM5
```

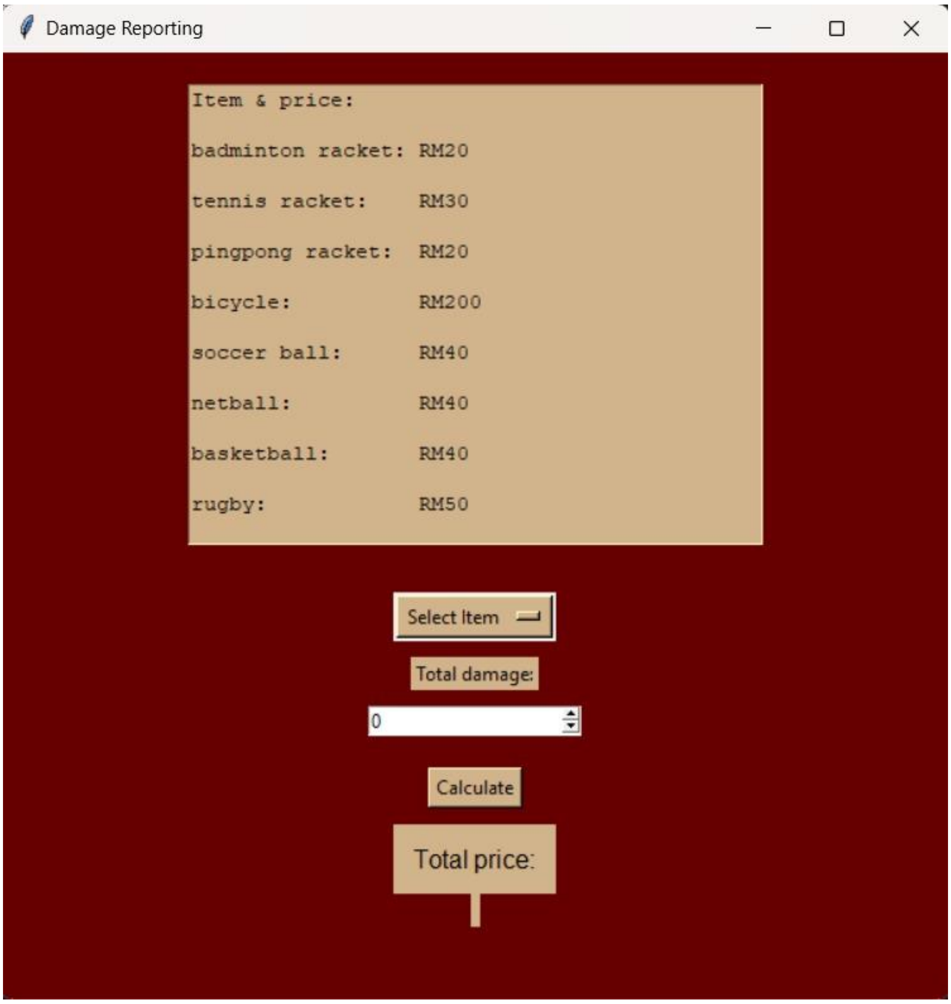
Rental

Court: Select

Items: Select

Count Items: 0

Day to Rent: 0 day



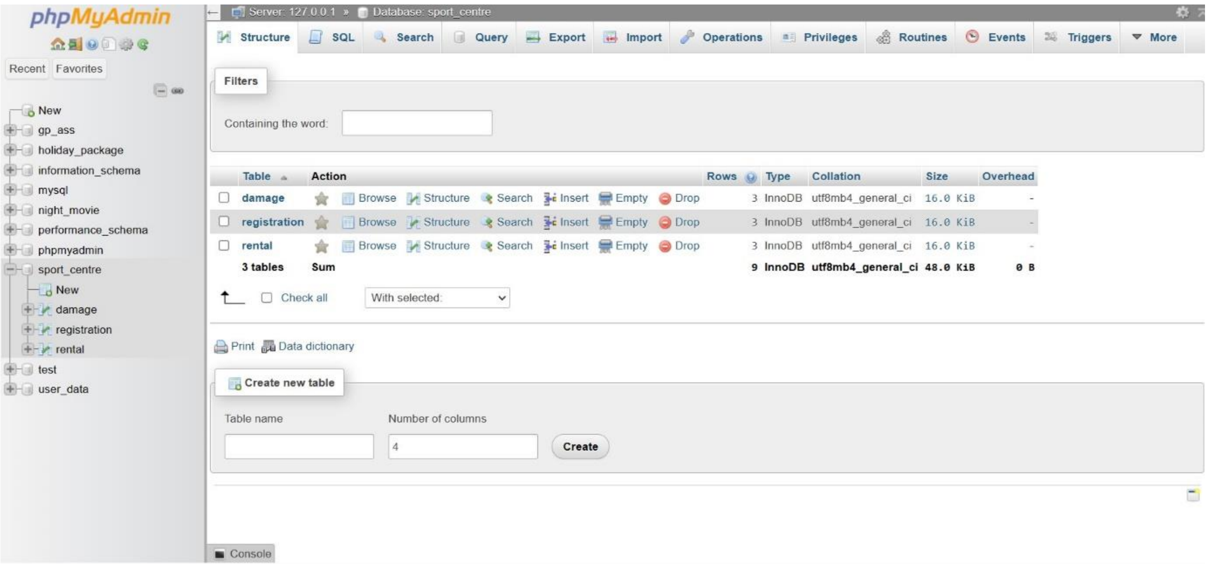
```
Item & price:
badminton racket: RM20
tennis racket:    RM30
pingpong racket:  RM20
bicycle:          RM200
soccer ball:      RM40
netball:          RM40
basketball:       RM40
rugby:            RM50
```

Select Item

Total damage: 0

Total price:

5.0 DATABASE



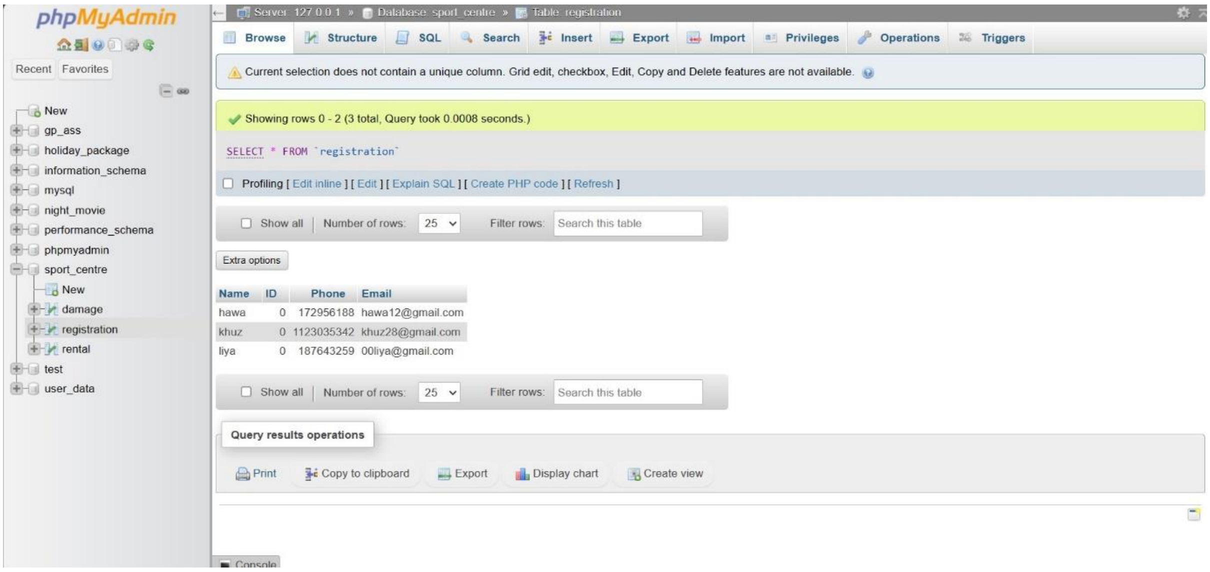
Server: 127.0.0.1 » Database: sport_centre

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
damage	☆ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
registration	☆ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
rental	☆ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
3 tables	Sum	9	InnoDB	utf8mb4_general_ci	48.0 KiB	0 B

Table name:

Number of columns: 4



Server: 127.0.0.1 » Database: sport_centre » Table: registration

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 0.0008 seconds.)

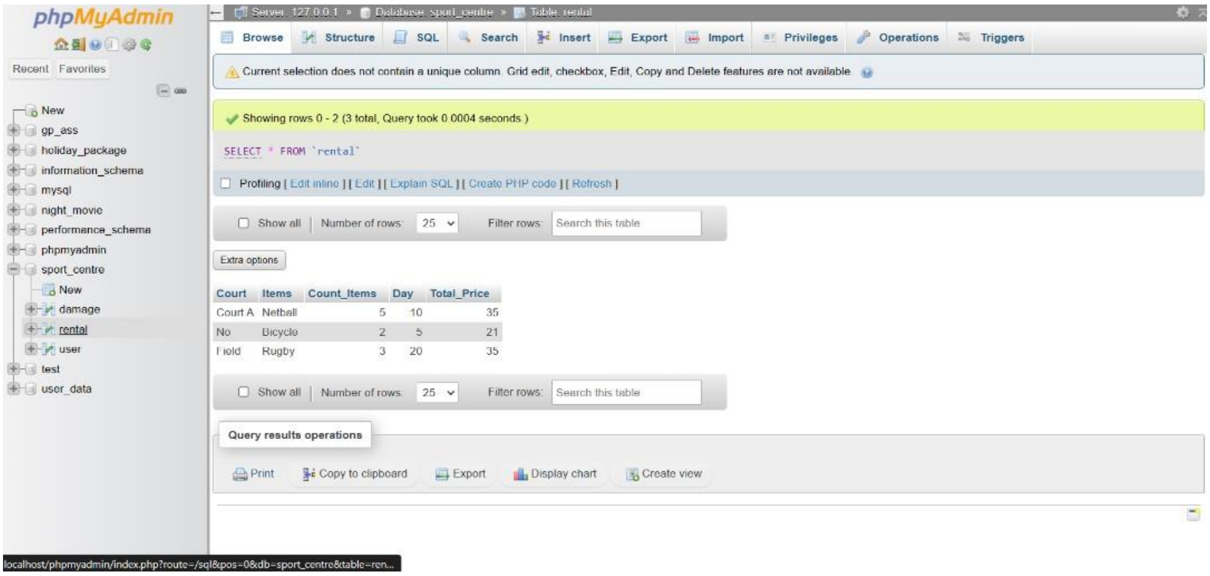
```
SELECT * FROM `registration`
```

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

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

Name	ID	Phone	Email
hawa	0	172956188	hawa12@gmail.com
khuz	0	1123035342	khuz28@gmail.com
liya	0	187643259	00liya@gmail.com

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



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.)

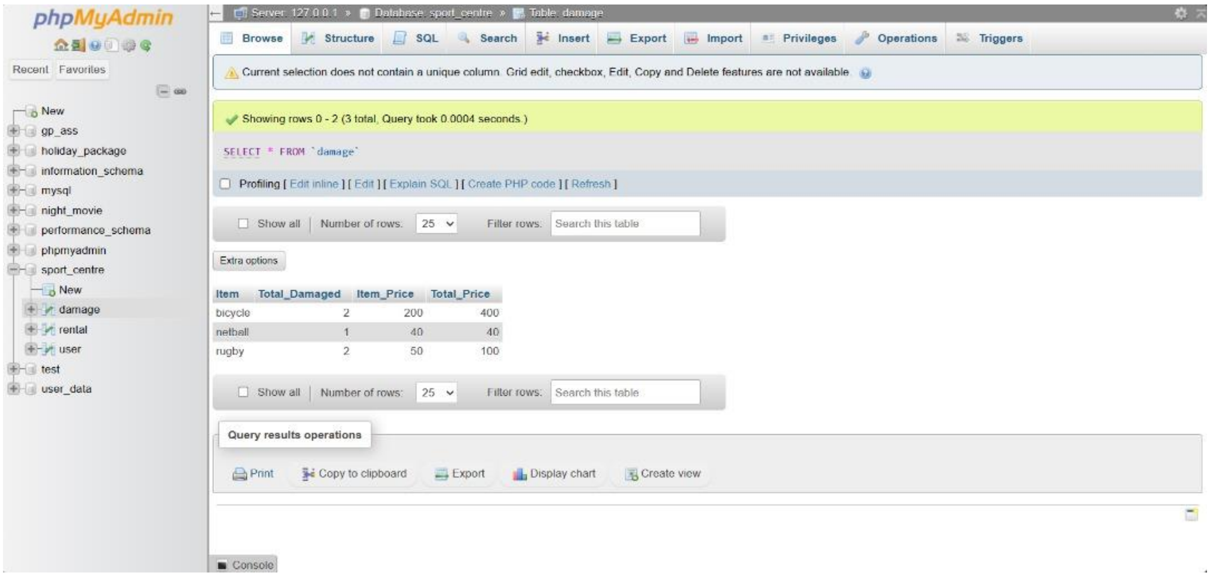
```
SELECT * FROM `rental`
```

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

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

Court	Items	Count_Items	Day	Total_Price
Court A	Netball	5	10	35
No	Bicycle	2	5	21
Field	Rugby	3	20	35

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



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 `damage`
```

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

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

Item	Total_Damaged	Item_Price	Total_Price
bicycle	2	200	400
netball	1	40	40
rugby	2	50	100

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

6.0 CONCLUSION

In conclusion, this system is straightforward, so it is easy to use and implement in the management. The form and attributes can be changed according to the institutions' needs and requirements. In brief, it is adaptable and flexible to meet the needs of individuals or organizations. Besides, the design and implementation are also changeable, depending on the preferences. Hence, the simple features facilitate the implementation of the system.

In addition, the output will be displayed in the form to inform the users about the information and data they have inserted. If any error occurs, users could change the data. This feature would make an easy correction to ensure the data is accurate. The information recorded would be used to confirm the booking of the item or court so that there would be no unauthorized users. Similarly, reporting damaged items needs to be accurate to ensure the broken item is being separated. Thus ensuring the safety of the users from being hurt when using the equipment.

At last, the implementation that are in this system not only is modernizes and streamlines management processess but also accounted to the overall success of the Sport Centre. By making the adaptibilty, user-friendly, and accurate data management as a priority, this system become a valuable tool for enhacing both operational efficiency and user satisfication.