



اَوْنُوْ سَيِّدِيْ تِيْكَوْلُوْ كِيْ مَارَا
UNIVERSITI
TEKNOLOGI
MARA

UNIVERSITY TECHNOLOGY MARA (UiTM) KEDAH BRANCH
COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

DIPLOMA IN LIBRARY INFORMATICS (**IM144**)

PROGRAMMING IN LIBRARY (**IML208**)

TITLE: **MALINJA JEWELRY**

PREPARED FROM:

MUHAMMAD HANIF AIMAN BIN KAMARUZAMAN (2022871792)

MUHAMMAD AMMAR BIN KHAMISAN (2022895812)

MUHAMMAD FARIS IZZAT BIN TARMIZI (2022818042)

JOHAN ISKANDAR BIN AHMAD TAMIMI (2022610854)

GROUP: KCDIM 1443B

PREPARED FOR:

Sir AIRUL SHAZWAN BIN NORSHAHIMI

SUBMISSION DATE: **17TH JANUARY 2024 (WEEK 14)**

MALINJA JEWELRY

MUHAMMAD HANIF AIMAN BIN KAMARUZAMAN (2022871792)

MUHAMMAD AMMAR BIN KHAMISAN (2022895812)

MUHAMMAD FARIS IZZAT BIN TARMIZI (2022818042)

JOHAN ISKANDAR BIN AHMAD TAMIMI (2022610854)

DIPLOMA IN LIBRARY INFORMATICS (IM144)

UNIVERSITY TECHNOLOGY MARA (UiTM) KEDAH BRANCH

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

17TH JANUARY 2024 (WEEK 14)



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 and responsible 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 representation to 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 any manner, violate this pledge.

Name: MUHAMMAD HANIF AIMAN BIN KAMARUZAMAN

Matric Number: 2022871792

Course Code: IML207

Programme Code: CDIM144

Faculty / Campus: UiTM Kampus Sungai Petani



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 and responsible manner. I will not engage or tolerate acts of academic dishonesty, academic misconduct, or academic fraud including but not limited to:

- f. **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.
- g. **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.
- h. **Fabrication:** Falsifying data, information, or citations in any academic assessment and evaluation.
- i. **Deception:** Providing false information with intend to deceive an instructor concerning any academic assessment and evaluation.
- j. **Furnishing false information:** Providing false information or false representation to 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 any manner, violate this pledge.

Name: JOHAN ISKANDAR BIN AHMAD TAMIMI

Matric Number: 2022610854

Course Code: IML207

Programme Code: CDIM144

Faculty / Campus: UiTM Kampus Sungai Petani



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 and responsible 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 representation to 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 any manner, violate this pledge.

Name: MUHAMMAD AMMAR BIN KHAMISAN

Matric Number: 2022895812

Course Code: IML207

Programme Code: CDIM144

Faculty / Campus: UiTM Kampus Sungai Petani



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 and responsible manner. I will not engage or tolerate acts of academic dishonesty, academic misconduct, or academic fraud including but not limited to:

- f. **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.
- g. **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.
- h. **Fabrication:** Falsifying data, information, or citations in any academic assessment and evaluation.
- i. **Deception:** Providing false information with intend to deceive an instructor concerning any academic assessment and evaluation.
- j. **Furnishing false information:** Providing false information or false representation to 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 any manner, violate this pledge.

Name: MUHAMMAD FARIS IZZAT BIN HJ MUHAMMAD TARMIZI

Matric Number: 2022818042

Course Code: IML207

Programme Code: CDIM144

Faculty / Campus: UiTM Kampus Sungai Petani

Content

Contents

1.0 INTRODUCTION	8
2.0 PROBLEM STATEMENT	9
3.0 Flowchart	10
4.0 GUI	11
5.0 Show code	13
6.0 Database	20
7.0 Conclusion	24
Reference	25

1.0 INTRODUCTION

A comprehensive programming system developed utilising Python and supplied with a user-friendly Graphical User Interface (GUI). This cutting-edge technology offers businesses an easy-to-use and visually appealing platform for effectively managing their stores, marking a major advancement in the field of retail applications. The Malinja Jewel Shop system uses Python as its foundation and takes advantage of the language's flexibility. The GUI adds a level of user-friendliness, making it a great option for companies trying to improve their retail operations.

The Malinja Jewel Shop system is perfect for Python's well-known readability and simplicity, which makes it easy for developers to design feature-rich and durable retail applications. The incorporation of a graphical user interface (GUI) improves user experience by offering an intuitive interface that makes navigation and interaction easier. Businesses can optimise the retail process by using Python and GUI together to speed sales monitoring, inventory management, and customer interactions.

The Malinja Jewel Shop system is a coding look that goes beyond conventional retail management systems as we examine its possibilities. Businesses can now harness the power of Python through an intuitive graphical user interface (GUI), opening the door to a new era of smooth and user-friendly retail experiences. This is achieved by putting an emphasis on user-centric design and efficient coding methods. The Malinja Jewel Shop system is an example of how technology and retail can work together, providing a complete solution for companies looking to succeed in the cutthroat world of contemporary a company.

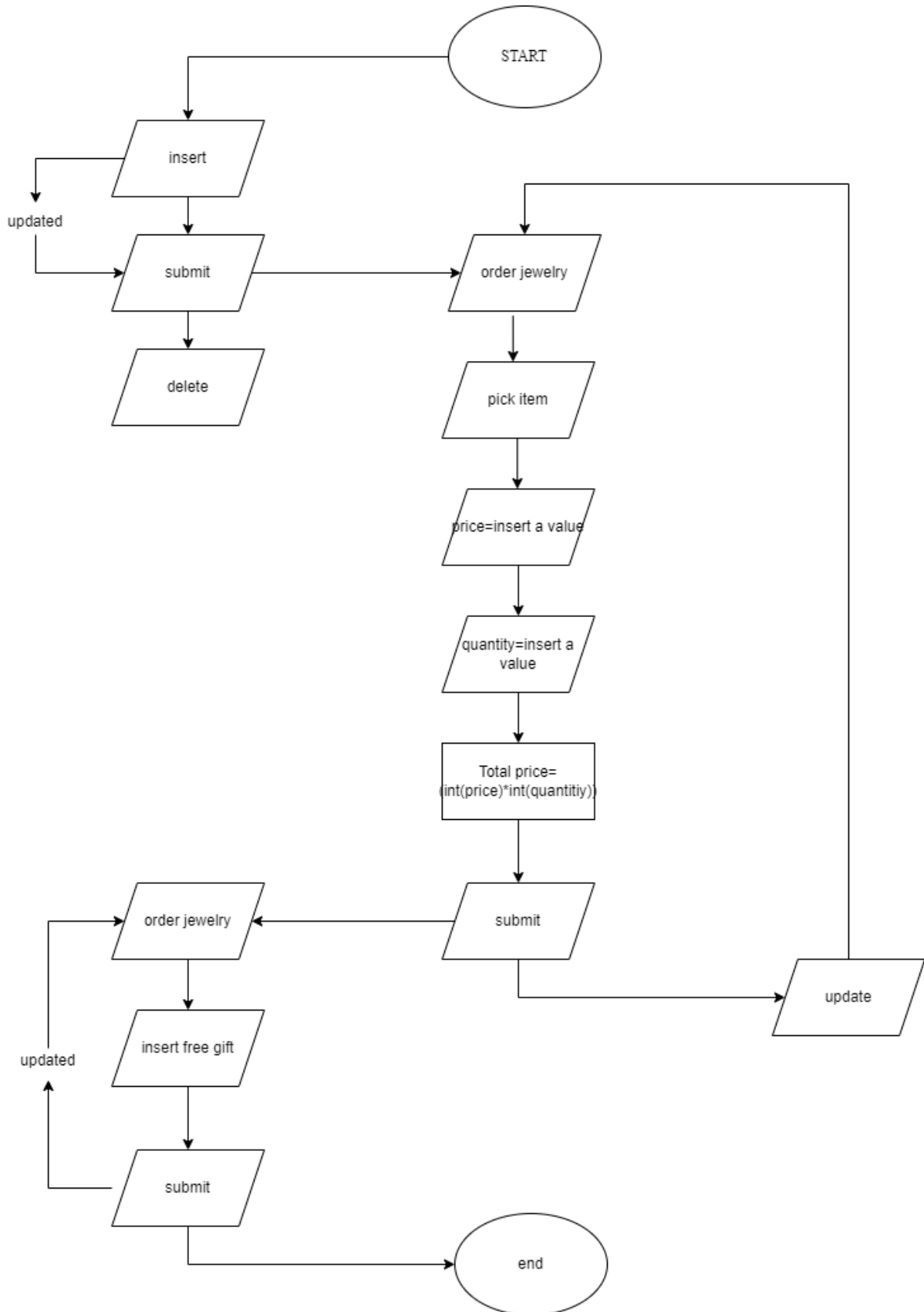
2.0 PROBLEM STATEMENT

Malinja Jewellery Shop is like a jewelry store that sells and receive orders to produce unique and beautiful bracelets, necklaces and rings. The way this store operates is by accepting advance orders from customers through walk-in to the store and discussing the jewelry chosen and the quantity they wanted.

The problem is this shop doesn't have a proper system to store their customer order data's, so they just use papers and physical file to store the customers order. This is not convenience and not practical because this will be increasing the usage of paper and makes the data storage space even fuller because there are many physical files and makes it difficult for the staff to find and review the order.

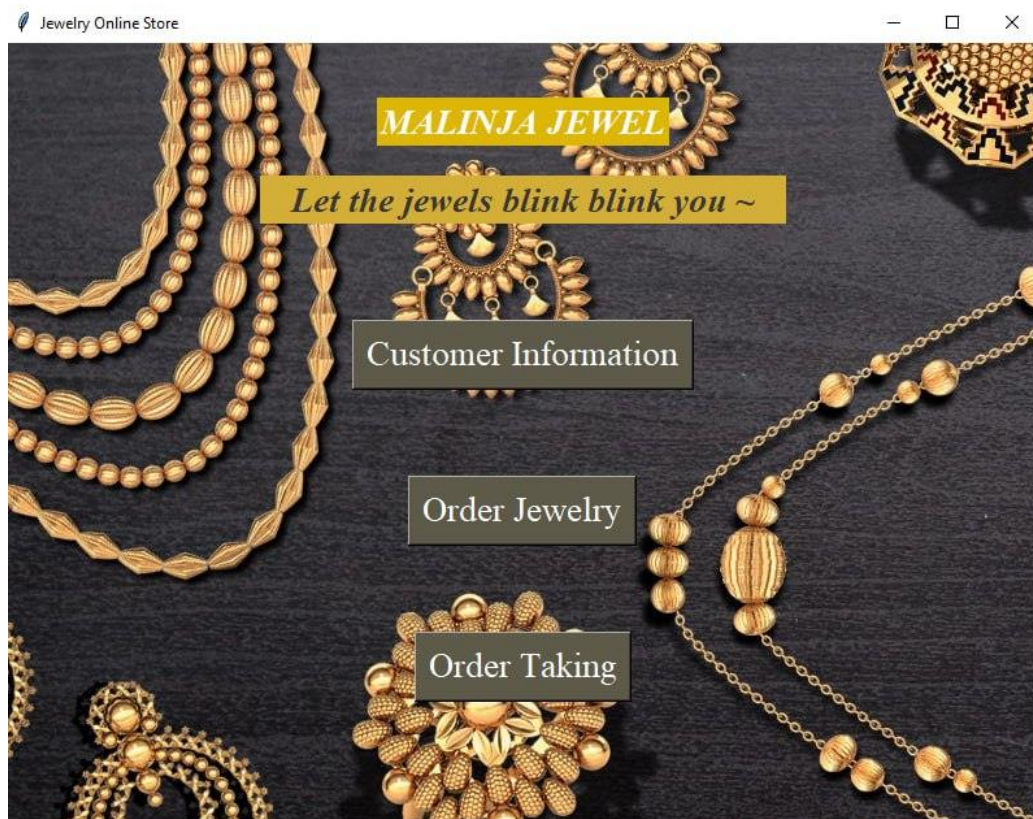
This is where a system and database are needed to facilitate and improve the efficiency of Malinja Jewelry's operations. So, all the data can be store and check back with easily and faster.

3.0 Flowchart



4.0 GUI

This is the main gui



Customer information GUI

The screenshot shows a window titled "Customer Information" with a black background. It contains a form with the following fields and values:

Name:	lee jung kuk	Title:	Tan Sri
Age:	53	E-Mail:	izzudin@gmail.com
Phone Number:	0124553678	Address:	taman permai indah
Gender:	Male		
Nationality:	Chinese		

At the bottom of the form is a yellow **Submit** button.

Order jewelry gui

Order Jewelry

JEWELRY ITEM:

fish Bracelets : Bracelets: Price:RM350

meteor bracelets : Bracelets: Price:RM600

flower bracelets : Bracelets: Price:RM1190

shark necklace : necklace : Price:RM6086

dino necklace : necklace : Price:RM4093

cat necklace : necklace : Price:RM7183

diamond Ring : Ring : Price:RM10023

gold ring : Ring : Price:RM6028

silver ring : Ring : Price:RM1297

ITEM	shark necklace
JEWELRY PRICE	6086
QUANTITY	27

Submit

Order taking

Order Taking

THANK YOU FOR PURCHASING
PLEASE SELECT FREE GIFT

Free Gift:	Beg
Place to Claim (Branch):	Alor Gajah, Melaka (014-35288
Date Claim:	1/17/24
Additional Notes:	thank you

Submit

5.0 Show code

```
import tkinter as tk
from tkinter import ttk
from PIL import Image, ImageTk
from tkinter import messagebox
from tkcalendar import DateEntry
import mysql.connector

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

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

def open_customer_info():
    customer_info_window = tk.Toplevel(root)
    customer_info_window.title("Customer Information")
    customer_info_window.geometry("500x250")
    customer_info_window.configure(bg="#030303")

    # Customer Information Form
    tk.Label(customer_info_window, text=" Name: ", font=("Times New Roman",
12), bg="#D4AF37").grid(row=0, column=0, sticky="ew", padx=5, pady=5)
    customer_name_entry = tk.Entry(customer_info_window)
    customer_name_entry.grid(row=0, column=1, padx=5, pady=5, sticky="ew")

    tk.Label(customer_info_window, text=" Title: ", font=("Times New Roman",
12), bg="#D4AF37").grid(row=0, column=2, padx=5, pady=5, sticky='ew')
    title_entry = ttk.Combobox(customer_info_window, values=["Dr.", "Mr.",
"Mrs.", "Dato", "Datin", "Tan Sri"])
    title_entry.grid(row=0, column=3, padx=10, pady=10, sticky="ew")

    tk.Label(customer_info_window, text=" Age: ", font=("Times New Roman",
12), bg="#D4AF37").grid(row=1, column=0, sticky="ew", padx=5, pady=5)
    age_entry = tk.Entry(customer_info_window)
    age_entry.grid(row=1, column=1, padx=5, pady=5, sticky="ew")

    tk.Label(customer_info_window, text="E-Mail:", font=("Times New Roman",
12), bg="#D4AF37").grid(row=1, column=2, sticky="ew", padx=5, pady=5)
    email_entry = tk.Entry(customer_info_window)
    email_entry.grid(row=1, column=3, padx=5, pady=5, sticky="ew")
```

```

    tk.Label(customer_info_window, text="Phone Number:", font=("Times New Roman", 12), bg="#D4AF37").grid(row=2, column=0, sticky="ew", padx=5, pady=5)
    phone_entry = tk.Entry(customer_info_window)
    phone_entry.grid(row=2, column=1, padx=5, pady=5, sticky="ew")

    tk.Label(customer_info_window, text="Address:", font=("Times New Roman", 12), bg="#D4AF37").grid(row=2, column=2, sticky="ew", padx=5, pady=5)
    address_entry = tk.Entry(customer_info_window)
    address_entry.grid(row=2, column=3, padx=5, pady=5, sticky="ew")

    tk.Label(customer_info_window, text="Gender:", font=("Times New Roman", 12), bg="#D4AF37").grid(row=3, column=0, sticky="ew", padx=5, pady=5)
    gender_combobox = ttk.Combobox(customer_info_window, values=["Male", "Female"])
    gender_combobox.grid(row=3, column=1, padx=5, pady=5, sticky="ew")

    tk.Label(customer_info_window, text="Nationality:", font=("Times New Roman", 12), bg="#D4AF37").grid(row=4, column=0, sticky="ew", padx=5, pady=5)
    nationality_entry = ttk.Combobox(customer_info_window, values=["Malay", "Indian", "Chinese", "Iban", "Kadazan"])
    nationality_entry.grid(row=4, column=1, padx=5, pady=5, sticky="ew")

def submit_customer_info():
    customer_name = customer_name_entry.get()
    customer_age = age_entry.get()
    title = title_entry.get()
    customer_address = address_entry.get()
    customer_phone = phone_entry.get()
    customer_email = email_entry.get()
    customer_gender = gender_combobox.get()
    customer_nationality = nationality_entry.get()

    # You can process the collected data here
    messagebox.showinfo("Customer Information",
                        f"Name: {customer_name}\nAge: {customer_age}\nPhone: {customer_phone}\nGender: {customer_gender}\nNationality: {customer_nationality}\nTitle: {title}\nAddress: {customer_address}\nEmail: {customer_email}")

    customer_info_window.destroy()

    # Update the customer information in the database
    update_sql = "UPDATE customer_information SET customer_name = %s, customer_age = %s, title = %s, customer_address = %s, customer_phone = %s, customer_email = %s, customer_gender = %s, customer_nationality = %s WHERE customer_name = %s"

```

```

        update_val = (customer_name, customer_age, title, customer_address,
customer_phone, customer_email, customer_gender,customer_nationality,
customer_name)
        mycursor.execute(update_sql, update_val)
        mydb.commit()

        messagebox.showinfo("Customer Information", f"Customer information
updated for {customer_name}")
        customer_info_window.destroy()
        sql = "INSERT INTO customer_information (customer_name, customer_age,
title, customer_address, customer_phone, customer_email, customer_gender,
customer_nationality) VALUES (%s, %s, %s, %s, %s, %s, %s, %s)"
        val = (customer_name, customer_age, title, customer_address,
customer_phone, customer_email, customer_gender, customer_nationality)
        mycursor.execute(sql, val)
        mydb.commit()

        update_button = tk.Button(customer_info_window,
text="Update",bg="#FAF49E", command= update_sql)
        update_button.grid(row=8, column=0,
columnspan=5,sticky="ew", pady=10)

        submit_button = tk.Button(customer_info_window,
text="Submit",bg="#FAF49E", command=submit_customer_info)
        submit_button.grid(row=8, column=0, columnspan=5,sticky="ew", pady=10)

def open_order_jewelry():
    order_jewelry_window = tk.Toplevel(root)
    order_jewelry_window.title("Order Jewelry")
    order_jewelry_window.geometry("650x550")
    order_jewelry_window.configure(bg="#030303")

    price_text=tk.Text(order_jewelry_window, bg="#ffc953",
fg="#030303",height=20 , width=50)
    price_text.grid(row=0, column=1, padx=20, pady=20)

    price_text.insert(tk.END, "JEWELRY ITEM:\n\n")
    price_text.insert(tk.END, "fish Bracelets : Bracelets: Price:RM350\n\n")
    price_text.insert(tk.END, "meteor bracelets : Bracelets: Price:RM600\n\n")
    price_text.insert(tk.END, "flower bracelets : Bracelets:
Price:RM1190\n\n")
    price_text.insert(tk.END, "shark necklace : necklace : Price:RM6086\n\n")
    price_text.insert(tk.END, "dino necklace : necklace : Price:RM4093\n\n")
    price_text.insert(tk.END, "cat necklace : necklace : Price:RM7183\n\n")
    price_text.insert(tk.END, "diamond Ring : Ring : Price:RM10023\n\n")
    price_text.insert(tk.END, "gold ring : Ring : Price:RM6028\n\n")
    price_text.insert(tk.END, "silver ring : Ring : Price:RM1297\n\n")

```



```

    item = tk.Label(order_jewelry_window, text="ITEM", font=("Felix
Titling",), bg="#D5A970", fg="#030303" )
    item.grid(row=1, column=0, padx=10, pady=10, sticky='ew')
    item_combobox = ttk.Combobox(order_jewelry_window, values=[" fish
bracelets", "meteor bracelets", "flower bracelets", "shark necklace", "dino
necklace", "cat necklace", "diamond ring", "gold ring", "silver ring"])
    item_combobox.grid(row=1, column=1, padx=10, pady=10, sticky='ew')

    product = tk.Label(order_jewelry_window, text="JEWELRY PRICE", font=("Felix
Titling",), bg="#D5A970", fg="#030303")
    product.grid(row=3, column=0, padx=10, pady=10, sticky='ew')
    product_combobox = tk.Entry(order_jewelry_window)
    product_combobox.grid(row=3, column=1, padx=10, pady=10, sticky='ew')

    quantity = tk.Label(order_jewelry_window, text="QUANTITY", font=("Felix
Titling",), bg="#D5A970", fg="#030303")
    quantity.grid(row=5, column=0, padx=10, pady=10, sticky='ew')
    quantity_entry = tk.Entry(order_jewelry_window)
    quantity_entry.grid(row=5, column=1, padx=10, pady=10, sticky='ew')

    def calculate():
        item_value = item_combobox.get()
        product_value = product_combobox.get()
        quantity_value = quantity_entry.get()

        # Check if the quantity is a valid integer
        try:
            quantity_value = int(quantity_value)
        except ValueError:
            messagebox.showerror("Error", "Please enter a valid quantity
(numeric value).")
            return

        # Check if the product price is a valid integer
        try:
            product_value = int(product_value)
        except ValueError:
            messagebox.showerror("Error", "Please enter a valid product price
(numeric value).")
            return

        total_price = product_value * quantity_value

        output_label = tk.Label(order_jewelry_window)
        output_label.grid(row=8, column=0)

        output_label.config(text=f"Total Price: RM{total_price}")

```



```

        messagebox.showinfo("Jewelry Info", f"Item: {item_value}\nProduct
Price: RM{product_value}\nQuantity: {quantity_value}\nTotal Price:
RM{total_price}")

    order_jewelry_window.destroy()

    sql = "INSERT INTO order_jewelry (item_value, product_value,
quantity_value, total_price) VALUES (%s, %s, %s, %s)"
    val = (item_value, product_value, quantity_value, total_price)
    mycursor.execute(sql, val)
    mydb.commit()

    submit_button = tk.Button(order_jewelry_window, text="Submit",
bg="#FAF49E", fg="#030303", command=calculate)
    submit_button.grid(row=7, column=0, columnspan=2, pady=10, sticky="ew")

def open_order_taking():
    order_taking_window = tk.Toplevel(root)
    order_taking_window.title("Order Taking")
    order_taking_window.geometry("480x400")
    order_taking_window.configure(bg="#D4AF37")

    # Order Taking Form
    note1 = tk.Label(order_taking_window, font=("Black Burger
Rough",20), text="THANK YOU FOR PURCHASING \n please select free gift ",
bg="#D4AF37", fg="#fefefe")
    note1.grid(row=0, columnspan=2, padx=30, pady=30)

    free_gift = tk.Label(order_taking_window, text="Free Gift:",
font=("Elephant",), bg="#282622", fg="#ffbf34" )
    free_gift.grid(row=1, column=0, sticky="ew", padx=5, pady=5)
    free_gift_combobox = ttk.Combobox(order_taking_window, values=["Perfume",
"Umbrella", "Beg"])
    free_gift_combobox.grid(row=1, column=1, padx=5, pady=5, sticky="ew")

    tk.Label(order_taking_window, text="Place to Claim (Branch):",
font=("Elephant",), bg="#282622", fg="#ffbf34").grid(row=2, column=0,
sticky="Ew", padx=5, pady=5)
    place_combobox = ttk.Combobox(order_taking_window, values=["Hulu
Teregganu,Terengganu (014-9307134)", "Batu Pahat,Johor(014-7726817)",
"Petaling Jaya,Selangor (014-8726615)", "Alor Gajah, Melaka (014-3528817)) "])
    place_combobox.grid(row=2, column=1, padx=5, pady=5, sticky='ew')

    tk.Label(order_taking_window, text="Date Claim:", font=("Elephant",),
bg="#282622", fg="#ffbf34").grid(row=3, column=0, sticky="ew", padx=5, pady=5)
    date_calender = DateEntry(order_taking_window, width=12,
background='darkblue', foreground='white', borderwidth=2)

```

```

date_calender.grid(row=3, column=1, padx=5, pady=5, sticky='ew')

tk.Label(order_taking_window, text="Additional Notes:",
font=("Elephant",), bg="#282622", fg="#ffbf34").grid(row=4, column=0,
sticky="ew", padx=5, pady=5)
notes_entry = tk.Entry(order_taking_window)
notes_entry.grid(row=4, column=1, padx=5, pady=5, sticky='ew')

def entry. grid():
    free_gift = free_gift_combobox.get()
    claim_place = place_combobox.get()
    claim_date = date_calender.get_date()
    additional_notes = notes_entry.get()

    # You can process the collected data here
    messagebox.showinfo("Order Taking", f"Free Gift: {free_gift}\nPlace to
Claim: {claim_place}\nDate Claim: {claim_date}\nAdditional Notes:
{additional_notes}")

    order_taking_window.destroy()

    sql = "INSERT INTO order_tracking (free_gift, claim_place, claim_date,
additional_notes) VALUES (%s, %s, %s, %s)"
    val = (free_gift, claim_place, claim_date, additional_notes)
    mycursor.execute(sql, val)
    mydb.commit()

    submit_button = tk.Button(order_taking_window, text="Submit",
command=entry. grid)
    submit_button.grid(row=5, column=0, columnspan=2, pady=10, sticky='e')

root = tk.Tk()
root.title("Jewelry Online Store")
root.geometry("800x600")

# Load background image
bg_image = Image.open("gambar jewels.jpg") # Replace with your image file
bg_photo = ImageTk.PhotoImage(bg_image)
bg_label = tk.Label(root, image=bg_photo)
bg_label.place(relwidth=1, relheight=1)

# Buttons to navigate to different interfaces
malinja_jewel = tk.Label(root, text="MALINJA JEWEL", font=("times new roman",
20, "bold", "italic"), bg="#DEB706", fg="FFFFFF")
malinja_jewel.place(relx=0.5, rely=0.1, anchor="center")

note = tk.Label(root, text="    Let the jewels blink blink you ~    ",
font=("times new roman", 20, "bold", "italic"), bg="#D4AF37", fg="#3A3536")

```

```
note.place(relx=0.5, rely=0.2, anchor="center")

customer_info_button = tk.Button(root, text="Customer Information",
font=("times new roman", 20), bg="#5D5A4A", fg="FFFFFF",
command=open_customer_info)
customer_info_button.place(relx=0.5, rely=0.4, anchor="center")

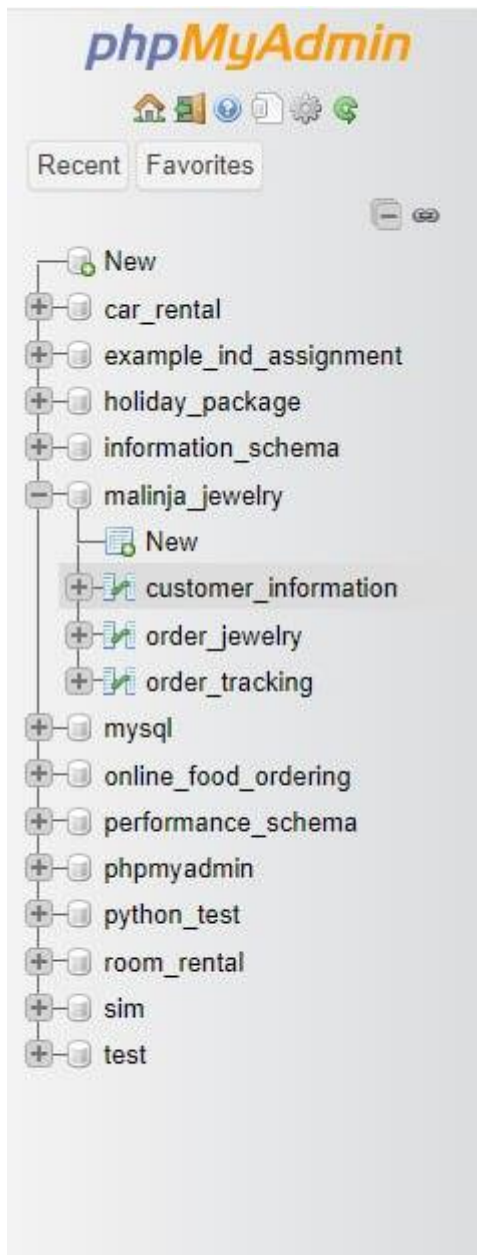
order_jewelry_button = tk.Button(root, text="Order Jewelry", font=("times new
roman", 20),bg="#5D5A4A", fg="FFFFFF", command=open_order_jewelry)
order_jewelry_button.place(relx=0.5, rely=0.6, anchor="center")

order_taking_button = tk.Button(root, text="Order Taking", font=("times new
roman", 20),bg="#5D5A4A", fg="FFFFFF", command=open_order_taking)
order_taking_button.place(relx=0.5, rely=0.8, anchor="center")

root.mainloop()
```

6.0 Database

This is the database that have three table that consist of customer_information, order_jewelry,



The structure of database

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> customer_information		12	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> order_jewelry		10	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> order_tracking		10	InnoDB	latin1_swedish_ci	16.0 KiB	-
3 tables	Sum	32	InnoDB	latin1_swedish_ci	48.0 KiB	0 B

☐ Check all With selected: ▼

The structure pf customer information structure

Server: 127.0.0.1 » Database: malinja_jewelry » Table: customer_information

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[Privileges](#)
[Operations](#)
[Tracking](#)
[Triggers](#)

[Table structure](#)
[Relation view](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	customer_name	text	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	2	customer_age	int(3)		No	None			Change Drop More
<input type="checkbox"/>	3	title	varchar(10)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	4	customer_address	varchar(300)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	5	customer_phone	varchar(12)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	6	customer_email	varchar(50)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	7	customer_gender	text	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	8	customer_nationality	text	latin1_swedish_ci	No	None			Change Drop More

☐ Check all
 With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#) [Spatial](#) [Fulltext](#)

[Add to central columns](#)
[Remove from central columns](#)

[Print](#)
[Propose table structure](#)
[Track table](#)
[Move columns](#)
[Normalize](#)

[Add](#) 1 column(s) after customer_nationality [Go](#)

This is the browser of the customer browser

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

Extra options

customer_name	customer_age	title	customer_address	customer_phone	customer_email	customer_gender	customer_nationality
MUHAMMAD HANIF AIMAN BIN KAMARUZAMAN	23	Mr.	kolej malinja	01121955708	hanifaيمان2004@gmail.com	Male	Malay
danish	0	Datin	5u3	wrgewtjie7	4i4	Male	Malay
hakim	56	Dato		01265483235		Male	
	0						
MUHAMMAD AMMAR	21	Dato	kampung sungai petani	0149306123	ammarr@gmail.com	Male	Malay
IZZAT ALI	24	Dato	jalan 15, taman permai	0124356786	ALI@gmail.com	Male	Indian
Johan iskandar	33	Mr.		0147864675	jo@gmail.com	Male	Malay
HANIF AIMAN KAMARUZAMAN	56	Dr.	taman hulu terengganu	0123456758	aiman@gmail.com	Male	Iban
fakhrul mirza	21	Mr.		0198776745	mirza@gmail.com	Male	Kadazan
lee jung kuk	53	Tan Sri	taman permai indah	0124553678	izzudin@gmail.com	Male	Chinese
muhammad rajesh	32	Mr.	jalan jaya diri	0173425667	rajesh@gmail.com	Male	Indian
nurul aisyah	32	Datin	taman megah indah	0167846752	aisyah@gmail.com	Female	Malay

The structure of order jewelry

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[Privileges](#)
[Operations](#)
[Tracking](#)
[Triggers](#)

[Table structure](#)
[Relation view](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	item_value	varchar(200)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	2	product_value	varchar(200)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	3	quantity_value	varchar(200)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	4	total_price	varchar(200)	latin1_swedish_ci	No	None			Change Drop More

☐ Check all
 With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#) [Spatial](#) [Fulltext](#)

[Add to central columns](#)
[Remove from central columns](#)

[Print](#)
[Propose table structure](#)
[Track table](#)
[Move columns](#)
[Normalize](#)

[Add](#) 1 column(s) after total_price [Go](#)

This is the browser of the order jewelry browser

✓ Showing rows 0 - 9 (10 total, Query took 0.0004 seconds.)

```
SELECT * FROM `order_jewelry`
```

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

☐ Show all | Number of rows: | Filter rows:

Extra options

item_value	product_value	quantity_value	total_price
fish bracelets	350	4	1400
meteor bracelets	600	4	2400
flower bracelets	1190	4	4760
dino necklace	4093	5	20465
flower bracelets	600	3	1800
diamond ring	10023	7	70161
cat necklace	7183	9	64647
shark necklace	6086	27	164322
flower bracelets	1190	3	3570
cat necklace	7183	4	28732

The structure of order taking

Server: 127.0.0.1 » Database: malinja_jewelry » Table: order_tracking

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[Privileges](#)
[Operations](#)
[Tracking](#)
[Trigg](#)

[Table structure](#)
[Relation view](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	free_gift	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 2	claim_place	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 3	claim_date	date			No	None			Change Drop More
<input type="checkbox"/> 4	additional_notes	varchar(200)	latin1_swedish_ci		No	None			Change Drop More

☐ Check all | With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#) [Spatial](#) [Fulltext](#)

[Add to central columns](#) [Remove from central columns](#)

[Print](#) [Propose table structure](#) [Track table](#) [Move columns](#) [Normalize](#)

[Add](#) column(s) [Go](#)

Indexes [+](#)

This is the browser of the order taking

Extra options

free_gift	claim_place	claim_date	additional_notes
Perfume	Hulu Teregganu, Terengganu (014-9307134)	0000-00-00	give me a nice quality
Beg	Batu Pahat, Johor(014-7726817)	2024-01-31	hope have nice quality
		2024-01-17	
Umbrella	Hulu Teregganu, Terengganu (014-9307134)	2024-01-17	thank you
Beg	Petaling Jaya, Selangor (014-8726615)	2024-01-17	terima kasih
Beg	Hulu Teregganu, Terengganu (014-9307134)	2024-01-17	buat cantik cantik ya
Perfume	Petaling Jaya, Selangor (014-8726615)	2024-01-17	okayyy
Beg	Alor Gajah, Melaka (014-3528817))	2024-01-17	thank youu
Perfume	Petaling Jaya, Selangor (014-8726615)	2024-01-31	-
Umbrella	Alor Gajah, Melaka (014-3528817))	2024-01-22	terima kasihh

☐ Show all | Number of rows: 25 ▼ | Filter rows:

7.0 Conclusion

As far as Python coding goes, the Malinja Jewel project has been a brilliant conclusion that fits the issue statement perfectly. It became evident as we worked our way through the complexities of this programming language that Python is more than just a tool; rather, it's a multifaceted gem that can solve a wide range of problems. We were able to precisely craft solutions that addressed the intricacies mentioned in the problem statement thanks to its clean syntax and clarity.

The expertise in Python coding acquired throughout the Malinja diamond project is a priceless asset that never goes away, much like a diamond that gains value with time. The flexibility of the language, together with a wide range of libraries and frameworks, enables programmers to create solutions that are scalable, maintainable, and effective. Our experience with Python coding has been life-changing, equipping us with a skill set that goes beyond the current project and puts us in a confident position to take on future challenges.

To sum up, the Malinja Jewel project is evidence of Python's continuing importance in the world of coding. It emphasises how the language can shine across a variety of fields and leave a lasting impression on our capacity for problem-solving. As we come to the end of our coding adventure, the Malinja Jewel serves as a reminder of the tenacity and intelligence that come with learning Python—a talent that will only increase in value as we continue to explore the exciting and rapidly changing field of technology.

Reference

freeCodeCamp.org. (2022, August 9). *Python for Beginners – Full course [Programming Tutorial]* [Video]. YouTube. <https://www.youtube.com/watch?v=eWRfhZUzrAc>

Sen Gideons. (2023, January 22). *Adding a Pop-Up Calendar to a Python Tkinter Form using TkCalendar to pick Dates* [Video]. YouTube. <https://www.youtube.com/watch?v=jU-LVFjBD3g>

Programming with Mosh. (2019, March 20). *MySQL Tutorial for Beginners [Full Course]* [Video]. YouTube. https://www.youtube.com/watch?v=7S_tz1z_5bA

MySQL Tutorial. (n.d.). <https://www.w3schools.com/MySQL/default.asp>

Python Functions. (n.d.). https://www.w3schools.com/python/python_functions.asp