



اُنِيْوَرْسِيْٓتِيْ تِيْكَنُوْلُوْجِيْ مَآرَا  
UNIVERSITI  
TEKNOLOGI  
MARA

**COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS**

**UNIVERSITI TEKNOLOGI MARA**

**MERBOK, KEDAH**

**DIPLOMA IN LIBRARY INFORMATICS**

**(IM144)**

**PROGRAMMING FOR LIBRARIES**

**(IML208)**

**PROJECT NAME: STORE INVENTORY SYSTEM**

**PREPARED BY:**

**ANIS SYAZWANI BINTI ROZAINI**

**GROUP: CDIM1443D**

**PREPARED FOR:**

**SIR MOHD FIRDAUS BIN MOHD HELMI**

**SUBMISSION DATE: 12<sup>th</sup> DECEMBER 2023**

**PROJECT NAME:** Stationery Store Inventory System

**FILE NAME:** inventory.py

**PROMPT DATA:**

1. **ID:** for the item Id
2. **Name:** name of the item
3. **Price:** the cost for one item
4. **Quantity:** how many items of one product

**FUNCTION**

1. **CREATE** data
2. **READ** data
3. **UPDATE** data
4. **DELETE** data

**CONDITIONAL STATEMENT:** YES

If-else statement

```
if itemId == "" or itemId == " ":
    print("Error Inserting Id")
if itemName == "" or itemName == " ":
    print("Error Inserting Name")
if itemPrice == "" or itemPrice == " ":
    print("Error Inserting Price")
if itemQuantity == "" or itemQuantity == " ":
    print("Error Inserting Quantity")
else:
    insert(str(itemId),str(itemName),str(itemPrice),str(itemQuantity))
```

**GUI:** YES

The screenshot shows a GUI application window titled "Inventory System" with a subtitle "Stationery Store 24/7". On the left, there are four input fields labeled "ID", "Name", "Price", and "Quantity". Below these fields are three buttons: "Enter" (blue), "Update" (yellow), and "Delete" (red). On the right, there is a table with four columns: "ID", "Name", "Price", and "Quantity". The table is currently empty.

## RESULT:

The screenshot shows a Tkinter window titled "Inventory System" with a subtitle "Stationery Store 24/7". On the left, there is a form with four input fields: "ID" (containing "6002"), "Name" (containing "pilot"), "Price" (containing "6.00"), and "Quantity" (containing "10"). Below these fields are three buttons: "Enter" (blue), "Update" (yellow), and "Delete" (red). On the right, there is a table displaying the current inventory:

ID	Name	Price	Quantity
6002	pilot	6.00	10
6001	fabercastell	2.00	6
6000	stabilo	1.30	5

## STRENGTH:

- **User-Friendly Interface:** The graphical user interface (GUI) built with Tkinter allows users to interact with the inventory system in a simple and user-friendly manner.
- **Basic CRUD Operations:** Users can insert, view, update, and delete inventory items using the system's basic CRUD (Create, Read, Update, Delete) operations.
- **SQLite Database Integration:** Using a SQLite database to store data is appropriate for small-scale applications. SQLite is a lightweight database that is simple to set up and configure.
- **Treeview for Data Display:** The Tkinter library's Treeview widget is used to display inventory data in a tabular format, allowing users to easily view and manage the data.
- **Error Handling:** Some basic error handling is included in the code, such as checking for empty or whitespace input when inserting data.
- Can list all the items available.
- Able to check the current prices of an item.
- The quantity left on the shelves and the quantity restocked.

## KAIZEN (ROOM FOR IMPROVEMENT):

- **Validation of Input:** Increase the strength of input validation to ensure that only valid data is entered. Examine the data types, numerical ranges, and lengths.
- **Modularity and structure of the code:** Divide the code into modular functions with distinct responsibilities. This makes the code more maintainable and understandable.

- **User Reactions:** Provide informative messages or alerts for successful and unsuccessful operations to improve user feedback. Think about using pop-up messages or status bars.
- **Security Procedures:** Implement basic security measures such as input sanitization and SQL injection protection. Consider using parameterized queries for database operations rather than string concatenation.
- **Logging:** Implement logging to keep track of important system events, errors, and changes. This can help with system debugging and monitoring.
- **Dialogs for Confirmation:** Before performing critical operations such as deletion or updates, add confirmation dialogs. This can aid in the prevention of unintentional data loss.