UNIVERSITY PREPARATION

**Coursework Coversheet**

| | | | |
|---|---|---|---|
| **Module:** | **Computer Science** | **Term:** | **Third Term** |
| **Date:** | **07/05/2025** | **Weighting:** | **30%** |
| **Student First Name:** | **Riadh** | | |
| **Student Last name:** | **Moudir** | | |
| **Course Tutor:** | **CLAUDIA PAI** | | |
| **Class:** | **CS1** | | |

**Marks Obtained (out of 100):**

**Grade:**

***Declaration that the work submitted is my own. I:***

**✓ declare that this submission is entirely written in my own words and no part has been generated by AI software.**

**✓ declare that this submission has been written with contributions from AI software.**

**I acknowledge that-**

- ✓ **used AI for suggestions, generate ideas or understand core concepts as a preparatory activity**
- ✓ **used AI to write, rephrase, or paraphrase part of the essay**
- ✓ **used QuillBot, Grammarly or other software to review language**

NAME: **RIADH MOUDIR**

CLASS: CS1

## Problem Definition and Analysis

When thinking about what project to choose, I decided to create an Inventory Management System. I picked this idea because managing stock can be difficult, especially for small shops that still do everything manually. I thought it would be helpful to build a system that makes it easier to keep track of products, prices, and stock levels.

I chose to use Python for this project because it's the language I have practiced most during my course, and I feel more comfortable with it. I also used SQLite for the database because it’s simple to connect with Python and doesn't require any complicated setup. The project is designed mainly for small businesses who need a simple tool to manage their inventory without spending too much money on expensive systems.

I planned the project by first designing the basic structure: the database, the main menu, and the core functions like adding products, searching, and updating stock. I also checked some tutorials and looked at Python’s official documentation to better understand how to use the sqlite3 library. The system needed to be easy to use, work in a simple console environment, and handle basic errors like wrong inputs.

---

## Documented Design

**System Overview:**
The system follows a very simple flow:
The user opens the program ➔ sees the menu ➔ selects what they want to do (add/search/update stock) ➔ completes the action ➔ returns to the menu until they choose to exit.

I wanted the design to be simple so anyone using it for the first time would immediately understand how to work with it.

---

**Database Design:**
The system uses a single database called inventory.db where all the products are saved.
There is one table called products with the following fields:

- id: an integer that acts as a unique product ID
- name: the name of the product (text)
- category: the category of the product (text)
- price: the price of the product (real number)
- stock: the quantity of the product in stock (integer)

---

**Main Parts of the Program:**

- **main_menu()**:
  Displays the options for the user to choose from (Add Product, Search Product, Update Stock, Exit).
- **add_product()**:
  Allows the user to input a new product's details. It saves the information into the database. It also checks if the product ID already exists and shows an error if it does.
- **search_product()**:
  Lets the user search for products by ID, Name, or Category. It fetches the matching product(s) from the database and displays them.
- **update_stock()**:
  Lets the user select a product by ID and either add stock (for a new delivery) or subtract stock (after a sale). It updates the stock quantity directly in the database.

---

**Data Types and Structures:**

All products are stored in an SQLite table, not in Python lists.
Inputs like ID and Stock are integers, Name and Category are text, and Price is a floating-point number.

---

**Algorithms:**

*Add Product:*

1. Ask user for ID, Name, Category, Price, and Stock.
2. Insert the data into the products table.
3. Handle errors like duplicate IDs.

*Search Product:*

1. Ask user how they want to search (ID, Name, or Category).
2. Perform a database query to find the matching product.

*Update Stock:*

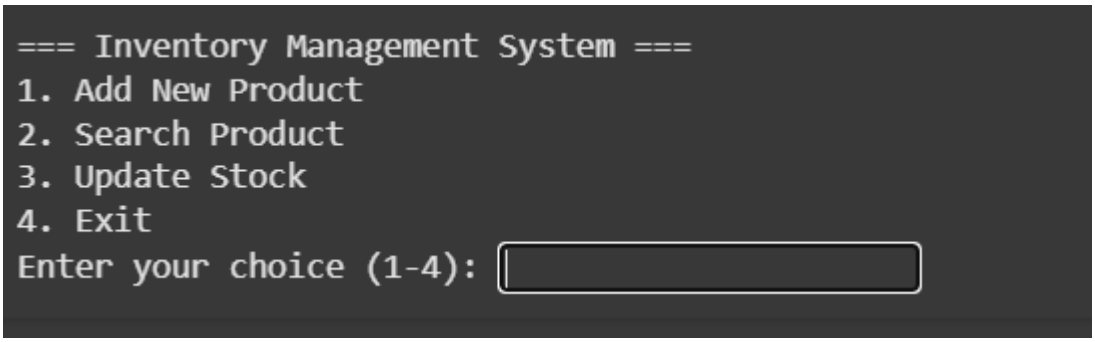
1. Find the product by ID.
2. Ask if user wants to add or subtract stock.
3. Update the stock quantity in the database.

---

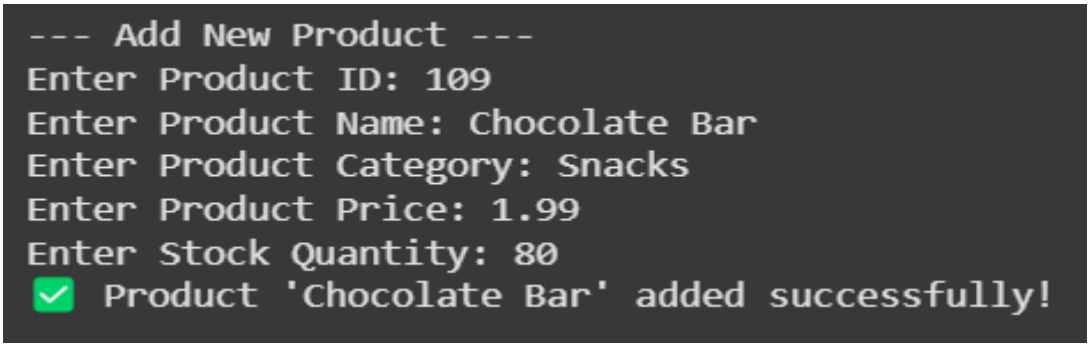
**User Interface (UI):**

## Screenshots of the System in Action:

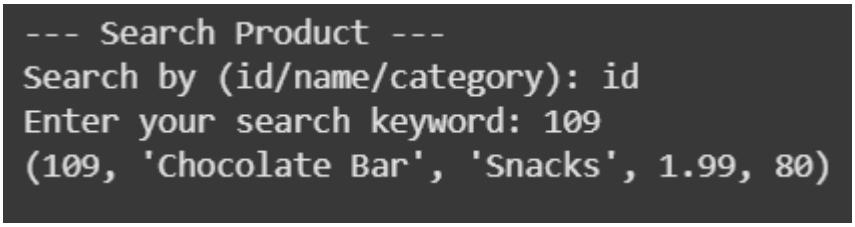
- **Figure 1: Main Menu in Google Colab**




- **Figure 2: Adding a Product**




- **Figure 3: Searching a Product**



The program runs in a simple console.
Instructions are very clear (like "Enter Product Name:") and messages are displayed if the user does something wrong (e.g., entering text instead of a number).

---

**Libraries Used:**

I used the built-in Python sqlite3 library to connect to the database with No extra installations needed, which made the setup very simple.

---

## Testing and Evaluation

**Testing:**

After finishing the main functions, I tested the system by adding several products like "Apple," "Milk," and "Bread."…..
I tested searching for products by both ID and Name to make sure the search function works correctly.
I also tested updating the stock by adding and subtracting quantities.

To check error handling, I tried entering wrong inputs like typing letters where numbers were expected. The program caught the errors and showed friendly messages without crashing.

**Evaluation:**

The main goal was to create a simple inventory system that allows users to add, search, and update products, while saving the data permanently.
I think the project meets these objectives well. The system is simple, easy to understand, and reliable for basic inventory management.

If I had more time, I would add extra features like the ability to delete a product or maybe generate reports.
But for the time and skills I had available, I am happy with how the project turned out, and I hope that this project will please you .