Project Report: Inventory Management Tool

Introduction and Objective

For this assignment, I opted to develop an Inventory Management Tool tailored to assist small business owners in managing their product stock. Manual inventory handling often results in disorganization, especially in smaller retail environments. This program simplifies the management of products, pricing, and stock data through digital tracking.

Python was chosen for its user-friendliness and extensive documentation. SQLite was selected for data storage because it integrates well with Python and requires minimal configuration.

The project targets users in need of a basic, cost-effective solution for inventory tasks. My initial step was to outline the application's structure, which includes setting up a database, designing a console-based menu, and implementing essential functions such as adding, locating, and modifying product details.

System Architecture

The system operates through a straightforward interface. Upon launching the application, the user is presented with a list of options. The chosen task is executed, and control returns to the menu until the user exits the application.

Database Structure

The inventory is managed using a single SQLite database file named 'inventory.db'. It includes a table titled 'products' with the following columns:
- id (INTEGER): Unique identifier
- name (TEXT): Product name
- category (TEXT): Classification of the product
- price (REAL): Cost of the product
- stock (INTEGER): Quantity available

Key Features

- main_menu(): Provides the user with action choices
- add_product(): Takes product details and stores them in the database after validation
- search_product(): Locates products based on various criteria
- update_stock(): Alters stock quantity based on user input

Data Handling

Data entries are exclusively handled within the SQLite database. Product IDs and stock quantities are integers, names and categories are text-based, and pricing is stored as a floating-point number.

Algorithm Design

- Adding Items: Collect input and store after checking for duplicate IDs
- Product Search: Allow look-up by ID, name, or category and display results
- Modify Stock: Update quantity based on user-specified action

User Interaction

The system is built to run in a text-based interface with clear prompts and error messages for invalid entries, ensuring ease of use even for beginners.

Libraries Utilized

The project makes use of Python's built-in 'sqlite3' library, ensuring minimal setup complexity.

Testing Phase

The functions were verified using various test cases such as adding products , conducting search operations, and adjusting stock levels. Inputs were intentionally varied to include both valid and invalid formats to evaluate error handling.

Project Assessment

The system met its intended objectives by providing essential features for managing inventory with a simple, intuitive design. Future upgrades could involve the ability to delete entries or export records.