# 📦 Inventory Control System

**Platform:** Desktop Application  
**Language:** Python 3.x  
**Libraries:** Tkinter, SQLite3  
**Filename:** full\_program.py  
**Database:** inventory.db (auto-generated)

## 🎯 Objective

To develop a desktop-based Inventory Control System with a user-friendly GUI for managing product inventory, including login authentication, product management, and stock updates.

## 🔧 Technologies Used

| **Module** | **Purpose** |
| --- | --- |
| tkinter | GUI development |
| sqlite3 | Embedded database for data persistence |
| os.path | File check for auto DB creation |

## 📁 Project Structure

pgsql

Inventory-Control-System/

│

├── full\_program.py # Main application script

├── inventory.db # SQLite database (auto-created on first run)

## 🗂️ Database Schema

### 1. admin table

Stores login credentials.

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| admin\_id | INTEGER | Primary Key (AutoIncrement) |
| username | TEXT | Admin username |
| password | TEXT | Admin password |

**Default Admin Credentials:**  
Username: admin  
Password: admin

### 2. product table

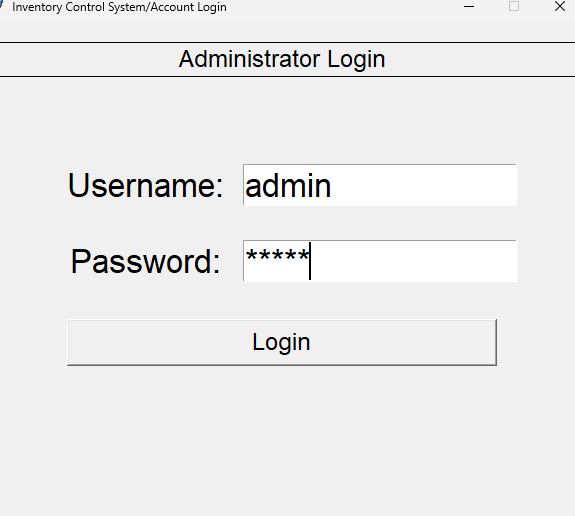
Stores all product records.

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| product\_id | INTEGER | Primary Key (AutoIncrement) |
| product\_name | TEXT | Name of the product |
| product\_qty | INTEGER | Current stock quantity |
| product\_price | REAL | Price of a single product unit |

## 🖼️ GUI Overview

### 1. Login Window

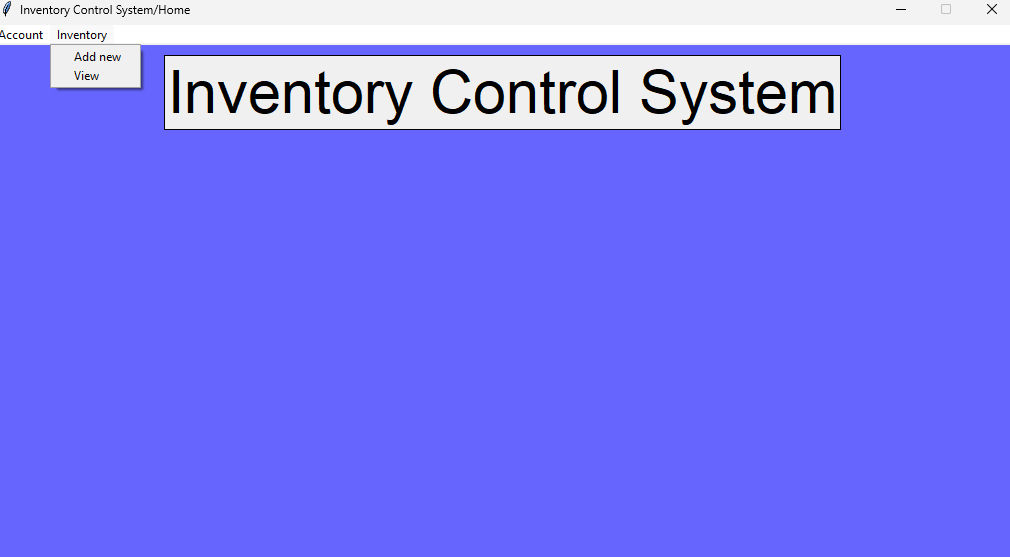
* Prompts user for username and password.
* Verifies credentials from admin table.
* On success, redirects to **Dashboard**.



### 2. Dashboard (Main Menu)

Provides access to the following modules:

* ➕ **Add Product**
* 📋 **View Products**
* 🔓 **Logout / Exit**

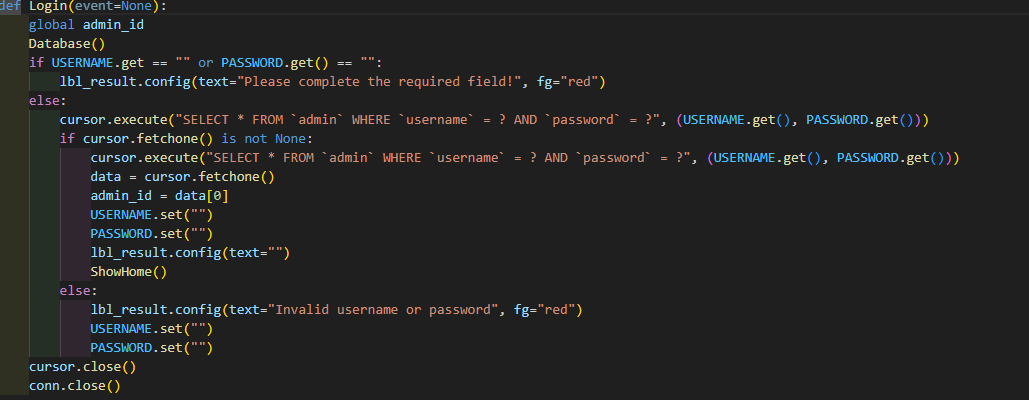


## 🧰 Functional Modules

### 1. ****Login Authentication****

* Validates credentials against the admin table.
* Displays error message on incorrect login.
* Opens Dashboard upon successful login.

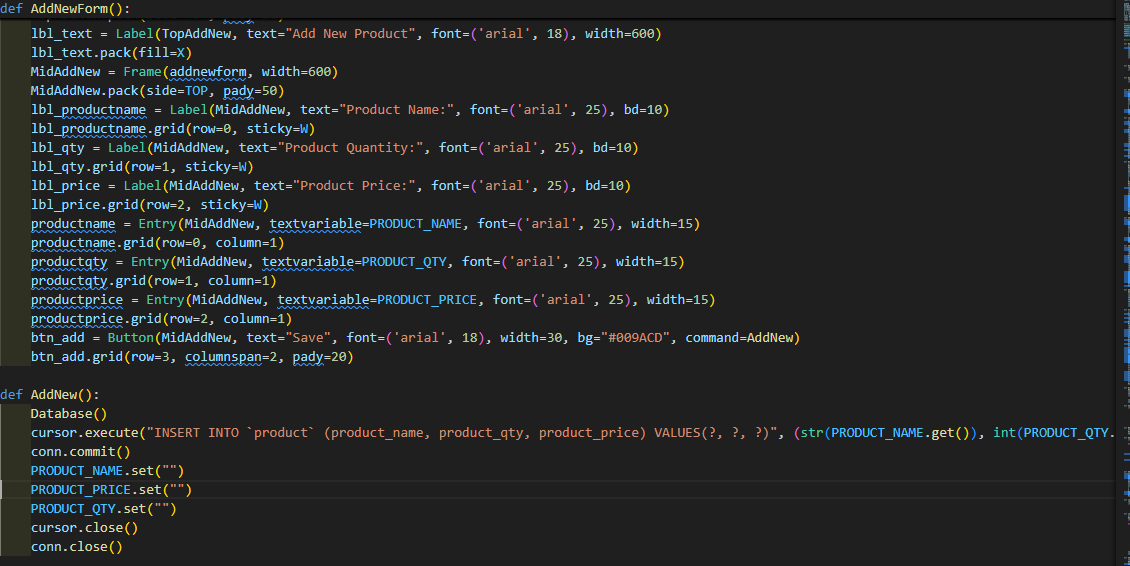
Sample code:



### 2. ****Add Product****

* Opens a form with fields:
  + Product Name
  + Quantity
  + Price
* Saves the data to the product table upon submission.
* Validates input (non-empty, valid numbers).

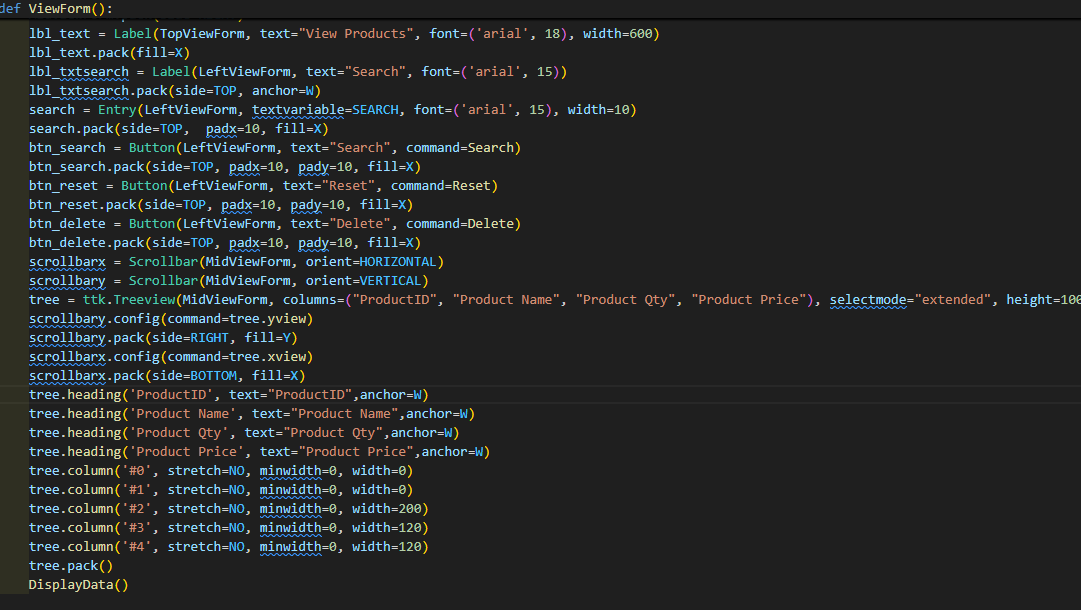
Sample code:



### 3. ****View Products****

* Opens a table view of all products.
* Displays columns: Product ID, Name, Quantity, Price.
* Table is sortable and scrollable.

Sample code:



## ▶️ How to Run the Application

1. Make sure Python 3.8 or later is installed.
2. Save the script as full\_program.py.
3. Open a terminal and run:

python full\_program.py

1. The application window will launch, and the database will be created automatically if not found.

## 🔮 Future Enhancements

Here are proposed features to implement in future versions:

| **Feature** | **Description** |
| --- | --- |
| 📤 Export to CSV | Allow users to export product data to CSV files |
| 👥 User Roles | Differentiate between Admin and Regular User functionalities |
| 🚨 Low Stock Alerts | Show alerts when quantity drops below a threshold |
| 🔍 Product Search & Filter | Add keyword search and filtering by category/price |
| 📊 Transaction Logs | Record every stock update for historical tracking |
| 📈 Charts & Reports | Visualize data via bar/pie charts |
| 🧾 PDF Export | Generate PDF reports of product or transaction summaries |

## 👨‍💻 Developer Notes

* Built using standard Python libraries.
* No external dependencies required.
* Ideal for small shops or internal company inventory.