**Inventory Management System Documentation**

**Introduction**

This documentation provides an overview of an Inventory Management System implemented using Python and SQLite. The system allows users to register, log in, and manage inventory by adding, viewing, updating, and deleting items.

**Features**

* User Registration
* User Login
* Add Inventory
* View Inventory
* Update Inventory
* Delete Inventory

**Database Schema**

The system uses SQLite as the database, with two tables:

**inventory Table**

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| ItemID | INTEGER (Primary Key) | Auto-incremented unique item ID |
| Date | TEXT | Date the item was added |
| ItemName | TEXT | Name of the inventory item |
| ItemDescription | TEXT | Description of the item |
| UnitPrice | REAL | Price per unit of the item |
| Quantity | REAL | Quantity available in inventory |

**users Table**

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| UserID | INTEGER (Primary Key) | Auto-incremented unique user ID |
| Username | TEXT (Unique) | Unique username for login |
| Password | TEXT | User password |

**Functionality**

**1. Connecting to Database**

The connect\_db() function establishes a connection to inventory.db and creates the required tables if they do not exist.

**2. User Registration**

The register(con, cursor) function allows users to create an account with a unique username and password.

**3. User Login**

The login(cursor) function checks user credentials and grants access upon successful authentication.

**4. Add Inventory**

The add\_inventory(con, cursor) function enables users to add new inventory items with details such as name, description, price, and quantity.

**5. View Inventory**

The view\_inventory(cursor) function retrieves and displays all inventory records in a tabular format.

**6. Update Inventory**

The update\_inventory(con, cursor) function allows users to modify existing inventory records.

**7. Delete Inventory**

The delete\_inventory(con, cursor) function removes an inventory item based on its ItemID.

**Usage Guide**

1. Run the script to start the application.
2. Choose to register a new user or log in if already registered.
3. After login, access the inventory management menu.
4. Perform operations like adding, viewing, updating, or deleting inventory.
5. Log out when done.

**Conclusion**

This Inventory Management System is a simple yet functional application for tracking inventory items. By implementing the suggested improvements, the security and robustness of the system can be enhanced.

import sqlite3  
import datetime  
  
def connect\_db():  
 con=sqlite3.connect('inventory.db')  
 cursor=con.cursor()  
  
 cursor.execute('''CREATE TABLE IF NOT EXISTS inventory(  
 ItemID INTEGER PRIMARY KEY AUTOINCREMENT,  
 Date TEXT,  
 ItemName TEXT,  
 ItemDescription TEXT,  
 UnitPrice REAL,  
 Quantity REAL)''')  
 cursor.execute('''CREATE TABLE IF NOT EXISTS users(  
 UserID INTEGER PRIMARY KEY AUTOINCREMENT,  
 Username TEXT UNIQUE,  
 Password TEXT)''')  
 con.commit()  
 return con,cursor  
def register(con,cursor):  
 username=input("ENTER YOUR USERNAME : ")  
 password=input("ENTER YOUR PASSWORD : ")  
  
 try:  
 cursor.execute("INSERT INTO users(username.password) VALUES?(?,?)",(username,password))  
 con.commit()  
 print("USER REGISTRATION SUCCESSFUL | YOU CAN LOGIN")  
  
 except sqlite3.IntegrityError:  
 print("USERNAME ALREADY EXISTS ! TRY AGAIN")  
  
def login(cursor):  
 username=input("ENTER YOUR REGISTERED USERNAME : ")  
 password=input("ENTER YOUR REGISTERED PASSWORD : ")  
  
 cursor.execute("SELECT \* FROM user WHERE username=?,password=?",(username,password))  
 user=cursor.fetchone()  
  
 if user:  
 print(f"WELCOME ,{username}")  
 return True  
 else:  
 print("LOGIN FAILED ! TRY AGAIN")  
 return False  
  
  
def add\_inventory(con,cursor):  
 date=datetime.date.today()  
 itemname=input("ENTER THE INVENTORY NAME : ")  
 itemdescription=input("ENTER THE ITEM DESCRIPTION : ")  
 unitprice=float(input("ENTER THE PRICE OF ONE UNIT : "))  
 quantity=int(input("ENTER THE TOTAL UNIT : "))  
  
 cursor.execute("INSERT INTO inventory(Date,ItemName,ItemDescription,UnitPrice,Quantity) VALUES(?,?,?,?,?)",  
 (date,itemname,itemdescription,unitprice,quantity))  
 con.commit()  
 print("\_\_\_INVENTORY ADDED SUCCESSFULLY\_\_\_")  
  
def view\_inventory(cursor):  
 cursor.execute("SELECT \* FROM inventory")  
 inventory=cursor.fetchall()  
 if not inventory:  
 print("NO INVENTORY FOUND.\n")  
 return  
 print("ItemID | Date | ItemName | ItemDescription | UnitPrice | Quantity")  
 print("\_" \* 60)  
 for inventory in inventory:  
 print(f"{inventory[0]} | {inventory[1]} | {inventory[2]} | {inventory[3]} | {inventory[4]} | {inventory[5]}")  
 print("\n")  
  
def update\_inventory(con,cursor):  
 view\_inventory(cursor)  
 inventory\_id=int(input("ENTER ITEM ID TO UPDATE : "))  
 itemname = input("ENTER THE NEW INVENTORY NAME : ")  
 itemdescription = input("ENTER THE NEW ITEM DESCRIPTION : ")  
 unitprice = float(input("ENTER THE NEW PRICE OF ONE UNIT : "))  
 quantity = int(input("ENTER THE NEW TOTAL UNIT : "))  
  
 cursor.execute("UPDATE inventory SET ItemName=?,ItemDescription=?,UnitPrice=?,Quantity=? WHERE ItemID=?",  
 (itemname,itemdescription,unitprice,quantity,inventory\_id))  
 con.commit()  
 print("\_\_\_INVENTORY UPDATED SUCCESSFULLY\_\_\_")  
  
def delete\_inventory(con,cursor):  
 view\_inventory(cursor)  
 inventory\_id=int(input("ENTER THE ITEM ID TO DELETE : "))  
 cursor.execute("DELETE FROM inventory WHERE ItemID=?",(inventory\_id,))  
 con.commit()  
 print("\_\_\_INVENTORY DELETED SUCCESSFULLY\_\_\_")  
  
def main():  
 con, cursor = connect\_db()  
 while True:  
 print("\_\_\_LOGIN MENU\_\_\_")  
 print("1.REGISTER USER")  
 print("2.LOGIN")  
 print("3.EXIT")  
  
 choice=input("ENTER YOUR CHOICE : ")  
 if choice=='1':  
 register(con,cursor)  
 elif choice=='2':  
 if login(cursor):  
 while True:  
 print("\n\_\_\_INVENTORY MANAGER\_\_\_")  
 print("1. ADD INVENTORY")  
 print("2. VIEW INVENTORY")  
 print("3. UPDATE INVENTORY")  
 print("4. DELETE INVENTORY")  
 print("5.LOGOUT")  
 choice = input("ENTER YOUR CHOICE: ")  
 if choice == '1':  
 add\_inventory(con, cursor)  
 elif choice == '2':  
 view\_inventory(cursor)  
 elif choice == '3':  
 update\_inventory(con, cursor)  
 elif choice == '4':  
 delete\_inventory(con, cursor)  
 elif choice=='5':  
 print("LOGGING OUT ...")  
 break  
 else:  
 print("INVALID CHOICE")  
 elif choice=='3':  
 print("EXCITING GOOD BYE")  
 break  
 else:  
 print("INVALID CHOICE")  
 con.close()  
  
main()