**ADVANCED INVENTORY MANAGEMENT SYSTEM**

**GROUP 1 SIA**

AMARILLE, CHENEMEI

IMONG, IAN JAY OREJAS

TAMAYO, JURIEL

RANUCO, LOVELYN

GOLANDRINA, MAY MARTINO

OLAYVAR, RIZA MAE JEAN

import sqlite3

import csv

DB\_FILE = "inventory.db"

def initialize\_database():

with sqlite3.connect(DB\_FILE) as conn:

cursor = conn.cursor()

cursor.execute("""

CREATE TABLE IF NOT EXISTS inventory (

id INTEGER PRIMARY KEY AUTOINCREMENT,

name TEXT NOT NULL,

category TEXT NOT NULL,

quantity INTEGER NOT NULL,

price REAL NOT NULL

)

""")

print("Database initialized.")

def add\_item():

name = input("Enter item name: ").strip()

category = input("Enter item category: ").strip()

try:

quantity = int(input("Enter quantity: "))

price = float(input("Enter price: "))

with sqlite3.connect(DB\_FILE) as conn:

cursor = conn.cursor()

cursor.execute("""

INSERT INTO inventory (name, category, quantity, price)

VALUES (?, ?, ?, ?)

""", (name, category, quantity, price))

conn.commit()

print(f"Item '{name}' added successfully!")

except ValueError:

print("Invalid input. Quantity must be a number, and price must be a decimal.")

def view\_inventory(order\_by="id", ascending=True):

with sqlite3.connect(DB\_FILE) as conn:

cursor = conn.cursor()

order = "ASC" if ascending else "DESC"

cursor.execute(f"SELECT \* FROM inventory ORDER BY {order\_by} {order}")

inventory = cursor.fetchall()

if not inventory:

print("The inventory is empty.")

else:

print("\nInventory List:")

print(f"{'ID':<5} {'Item Name':<20} {'Category':<15} {'Quantity':<10} {'Price':<10}")

print("-" \* 60)

for item in inventory:

print(f"{item[0]:<5} {item[1]:<20} {item[2]:<15} {item[3]:<10} ${item[4]:<10.2f}")

print()

def update\_item():

try:

item\_id = int(input("Enter the ID of the item to update: "))

with sqlite3.connect(DB\_FILE) as conn:

cursor = conn.cursor()

cursor.execute("SELECT \* FROM inventory WHERE id = ?", (item\_id,))

item = cursor.fetchone()

if not item:

print("Item not found.")

return

name = input(f"Enter new name (current: {item[1]}): ").strip() or item[1]

category = input(f"Enter new category (current: {item[2]}): ").strip() or item[2]

quantity = input(f"Enter new quantity (current: {item[3]}): ").strip() or item[3]

price = input(f"Enter new price (current: ${item[4]:.2f}): ").strip() or item[4]

cursor.execute("""

UPDATE inventory

SET name = ?, category = ?, quantity = ?, price = ?

WHERE id = ?

""", (name, category, int(quantity), float(price), item\_id))

conn.commit()

print(f"Item '{name}' updated successfully!")

except ValueError:

print("Invalid input. Quantity must be a number, and price must be a decimal.")

def delete\_item():

try:

item\_id = int(input("Enter the ID of the item to delete: "))

with sqlite3.connect(DB\_FILE) as conn:

cursor = conn.cursor()

cursor.execute("DELETE FROM inventory WHERE id = ?", (item\_id,))

conn.commit()

if cursor.rowcount == 0:

print("Item not found.")

else:

print("Item deleted successfully!")

except ValueError:

print("Invalid input. Please enter a valid ID.")

def search\_inventory():

query = input("Enter the name or category to search: ").strip().lower()

with sqlite3.connect(DB\_FILE) as conn:

cursor = conn.cursor()

cursor.execute("""

SELECT \* FROM inventory

WHERE LOWER(name) LIKE ? OR LOWER(category) LIKE ?

""", (f"%{query}%", f"%{query}%"))

results = cursor.fetchall()

if not results:

print("No items found matching your search.")

else:

print("\nSearch Results:")

print(f"{'ID':<5} {'Item Name':<20} {'Category':<15} {'Quantity':<10} {'Price':<10}")

print("-" \* 60)

for item in results:

print(f"{item[0]:<5} {item[1]:<20} {item[2]:<15} {item[3]:<10} ${item[4]:<10.2f}")

print()

def export\_inventory\_to\_csv():

with sqlite3.connect(DB\_FILE) as conn:

cursor = conn.cursor()

cursor.execute("SELECT \* FROM inventory")

inventory = cursor.fetchall()

if not inventory:

print("The inventory is empty. Nothing to export.")

return

with open("inventory\_export.csv", "w", newline="") as csvfile:

writer = csv.writer(csvfile)

writer.writerow(["ID", "Name", "Category", "Quantity", "Price"])

writer.writerows(inventory)

print(f"Inventory exported successfully to 'inventory\_export.csv'.")

def generate\_report():

print("\nReports:")

print("1. Low Stock Items (Quantity < 10)")

print("2. Items by Category")

choice = input("Choose a report: ").strip()

with sqlite3.connect(DB\_FILE) as conn:

cursor = conn.cursor()

if choice == "1":

cursor.execute("SELECT \* FROM inventory WHERE quantity < 10")

results = cursor.fetchall()

print("\nLow Stock Items:")

elif choice == "2":

category = input("Enter category: ").strip()

cursor.execute("SELECT \* FROM inventory WHERE category = ?", (category,))

results = cursor.fetchall()

print(f"\nItems in Category '{category}':")

else:

print("Invalid choice.")

return

if not results:

print("No items found for the selected report.")

else:

print(f"{'ID':<5} {'Item Name':<20} {'Category':<15} {'Quantity':<10} {'Price':<10}")

print("-" \* 60)

for item in results:

print(f"{item[0]:<5} {item[1]:<20} {item[2]:<15} {item[3]:<10} ${item[4]:<10.2f}")

print()

def main():

initialize\_database()

while True:

print("\n--- Advanced Inventory Management System ---")

print("1. View Inventory")

print("2. Add Item")

print("3. Update Item")

print("4. Delete Item")

print("5. Search Inventory")

print("6. Export Inventory to CSV")

print("7. Generate Reports")

print("8. Exit")

choice = input("Enter your choice (1-8): ").strip()

if choice == "1":

view\_inventory()

elif choice == "2":

add\_item()

elif choice == "3":

update\_item()

elif choice == "4":

delete\_item()

elif choice == "5":

search\_inventory()

elif choice == "6":

export\_inventory\_to\_csv()

elif choice == "7":

generate\_report()

elif choice == "8":

print("Exiting the system. Goodbye!")

break

else:

print("Invalid choice. Please choose between 1 and 8.")

if \_name\_ == "\_main\_":

main





