Module Title

Principles of Programming

Assessment Weightage & Type

50% of Coursework & Regular Year

2025

Student Name: NIRVIK K.C.

UWE ID: 25024649

Assignment Submission Date: January 18, 2026

Submission details

Module title and code: UFCFHS-30-1 Principles of Programming Individual Assignment

Assessment type: Individual Assignment – Software Development Portfolio

Assessment title: Application for Car Parts and Accessories Shop

Assessment weighting: 50% of total module mark

Module learning outcomes assessed by this task:

This assignment assesses the following module learning outcomes:

1. Acquire and apply knowledge of a range of essential programming language concepts and assess their effectiveness for a given problem.

2. Apply knowledge of basic concepts and principles of object-orientation such as objects and classes, encapsulation, object state and modularity.

3. Apply good programming style and interpret the impact of style of developing, testing, debugging and maintaining programs.

4. Evaluate the appropriateness of different programming paradigms for a given application.

Completing your assessment

What am I required to do on this assessment?

The assignment requires you to review the ideas of superclass and subclass relationship and provide opportunities to examine your level of knowledge in basic hierarchical class design and object-oriented programming using Python. Overall, you are required to design, implement and test an application based on the given scenario and document the outcomes.

You should complete your assignment considering the following two steps.

You only need to submit the work completed in Step 2 (which includes Step 1).

Please note that understanding the scenario and requirements is part of your assessment. Your tutors will assist you only when further clarification is required on certain parts of the assignment. Therefore, you are advised to read and understand the assessment requirements very carefully

TABLE OF CONTENT

Contents

INTRODUCTION

The Application for the Car Parts and Accessories Shop is a console-based inventory management system designed for car parts retailers. It uses object-oriented programming concept to organizes stock items through a modular class structure and uses DB Browser (SQLite) for data storage.

Figure 1: Application for Car Parts and Accessories Shop (Freepik)

The system helps in effective tracking and updating of car accessories, including navigation systems. The system is the way to replace manual record-keeping by providing the structured and accurate information . This system is a practical solution for small to medium-sized automotive and mechanics accessory shops.

Key Features of Application for Car Parts and Accessories Management System

The system features several important and optional features which are helpful for the shop. The system is built on the foundation of the StockItem base class, which contains all car accessories as well as provide attributes (private) for stock code, quantity, and price. The StockItem base class provided public methods for increasing the quantities of stock, selling of stocks, set new or change the price of items, and calculating price including VAT.

The sub-class NavSys inherits from StockItem, which then overrides several methods in the base class. This shows that both inheritance and polymorphism properties were applied in the system. The SQLite database helps in saving, updating, and reading items from the list accurately which shows that data persistence is provided effectively.

Other features include the ability to export the data to a CSV file for business data management, increase efficiency, and seamless information exchange between software applications. Python library “matplotlib” was used to generate bar chart to compare available accessories in the shop. The system also generates alert messages for low stock levels and completely out of stock items during sales. Finally, the system has input validation and handle any erros that may occur due to invalid inputs (e.g. negative values).

OBJECTIVE

- To implement a console-based inventory management system for car parts and accessories shop, enabling efficient creation, updating, and tracking of stock items.
- To provided specialized functionalities for different types of items available on the shop using inheritance and polymorphism.
- To integrate SQLite database for data persistent storage and retrieval of records. This ensures that the data remain available across many executions of program.
- To perform some basic error handling, such as checking for valid input to prevent blank or null entries and restrict invalid formats.
- To ensure the system can give fast and reliable responses even with large datasets
- To demonstrate programming concepts learned during the course, including classes, methods, file handling, exception handling, etc. through a real-world scenario.

SYSTEM DESIGN

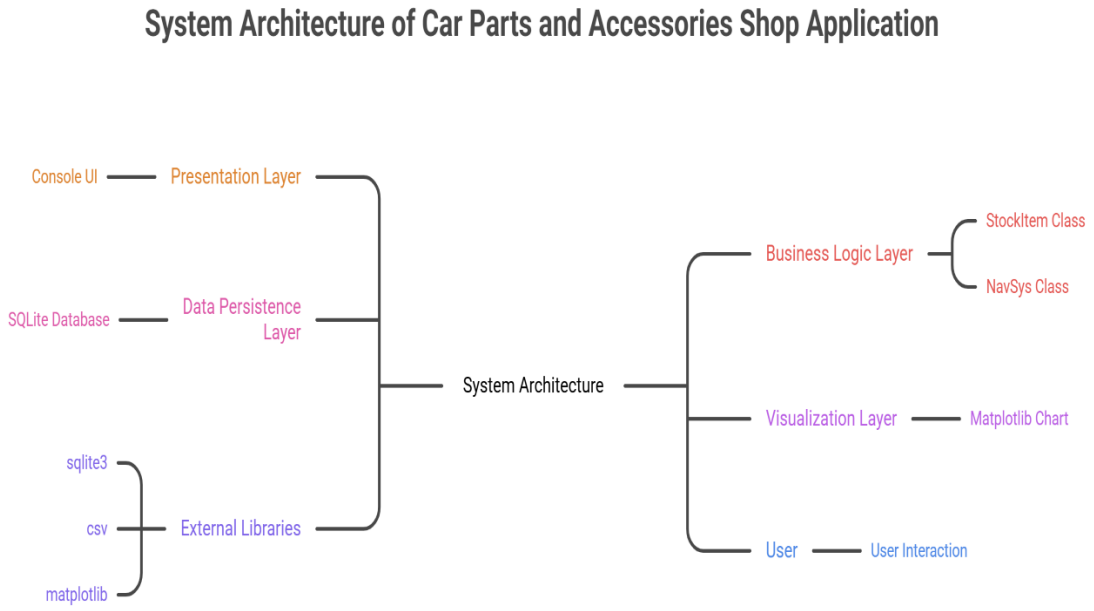


Figure 2 : System Design for the Car Parts and Accessories Management System (Napkin AI)

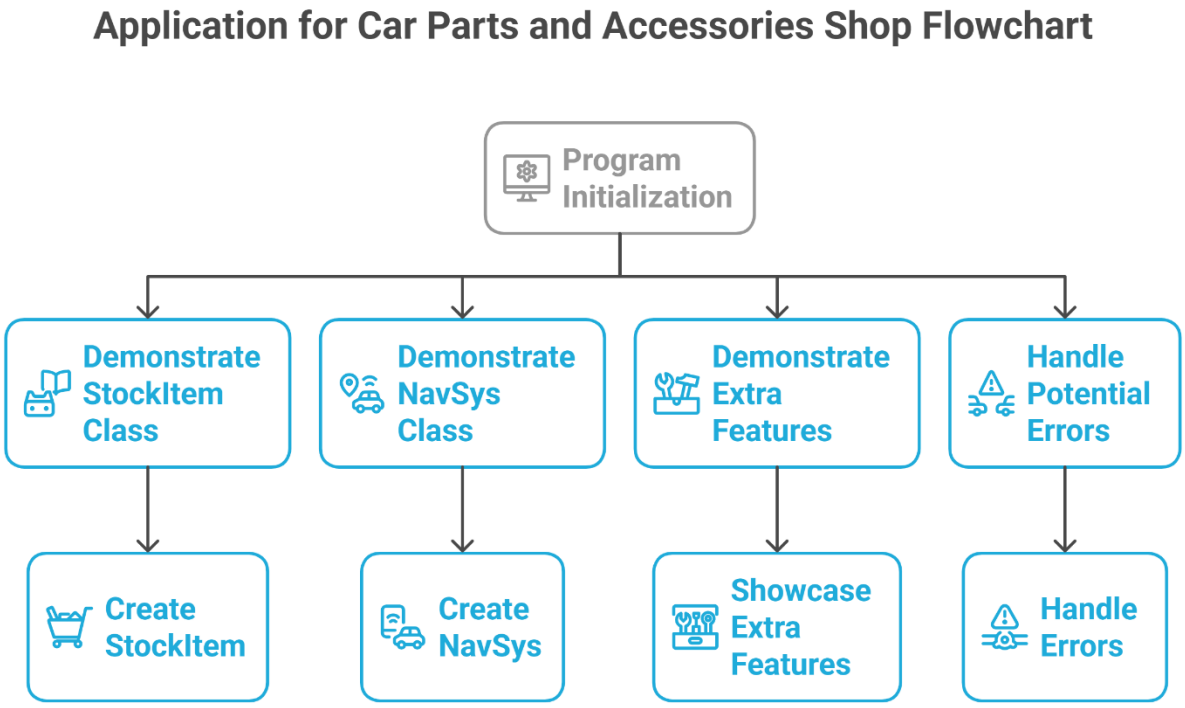


Figure 3 : Flowchart for Car Parts and Accessories Management System (Napkin AI)

IMPLEMENTATION

Python can be executed using the command line or integrated environment (IDE) for writing and execution of code. So, I am using Visual Studio Code for writing the code.

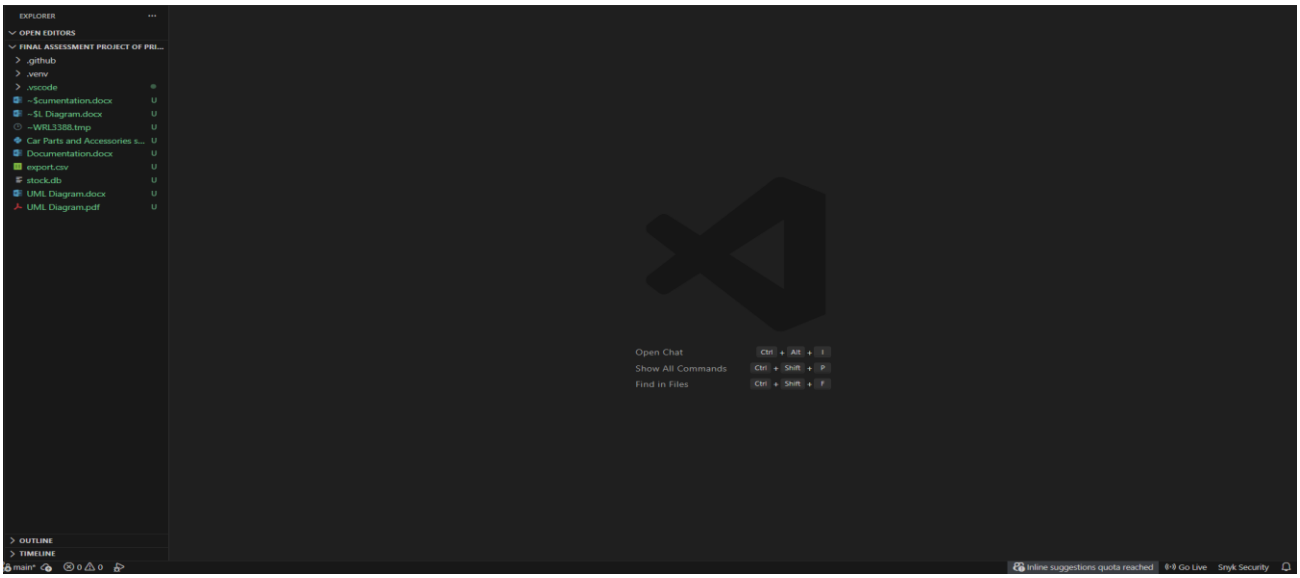


Figure 4 :Visual Studio Code used for writing Python program and use libraries

SQLite is a lightweight, file-based relational database engine that is built into Python (via sqlite3 module) and does not require any separate server installation. Here, SQLite was chosen by me because it reliable, serverless way to save and retrieve stock data persistently using a single file.

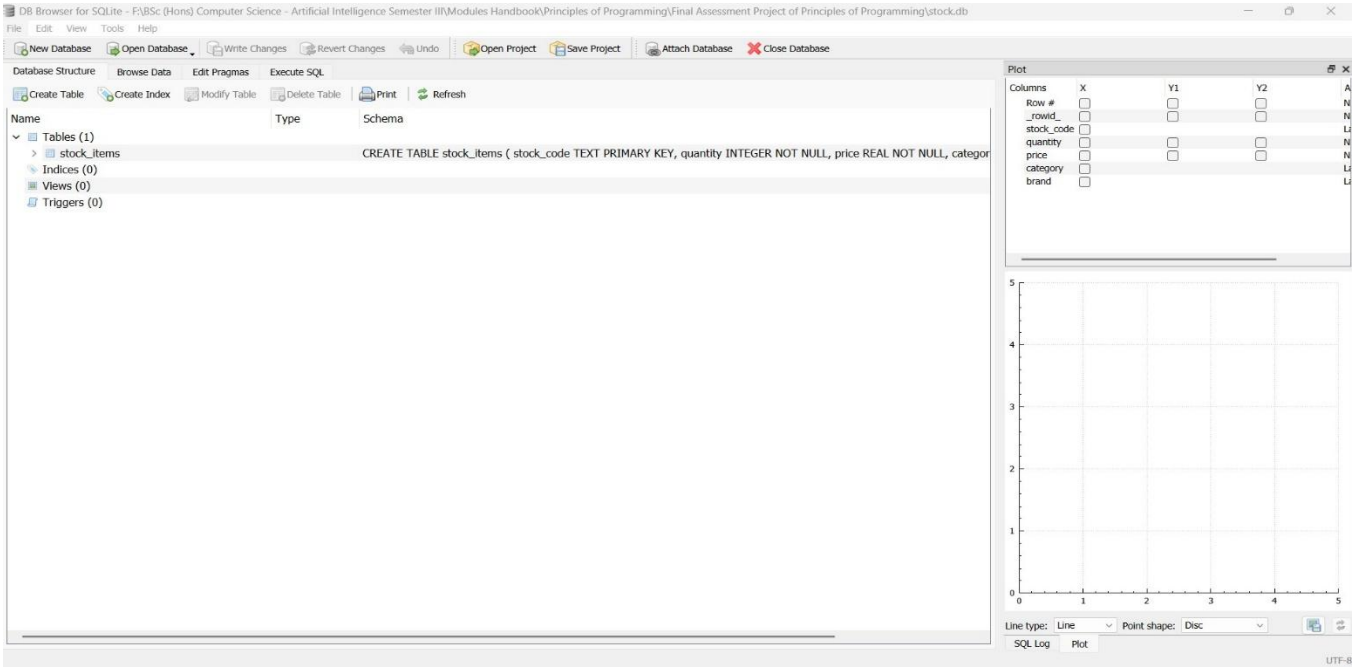


Figure 5: SQLite interface used for Car Parts and Accessories Management System

SAMPLE OUTPUT

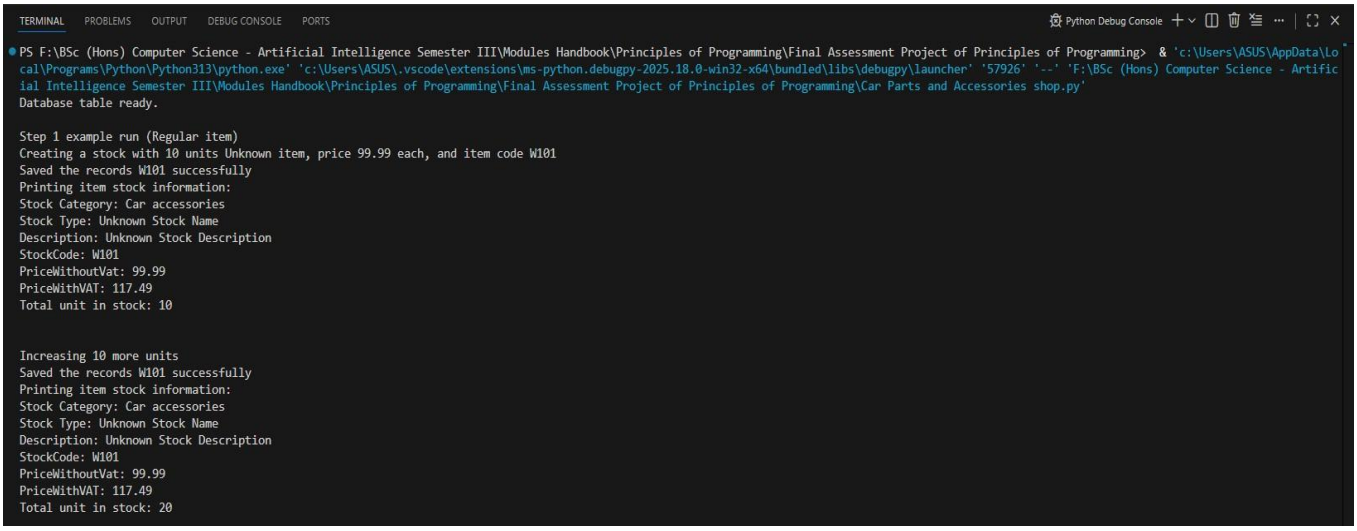


Figure 6 : Console based run of the program (Part 1)

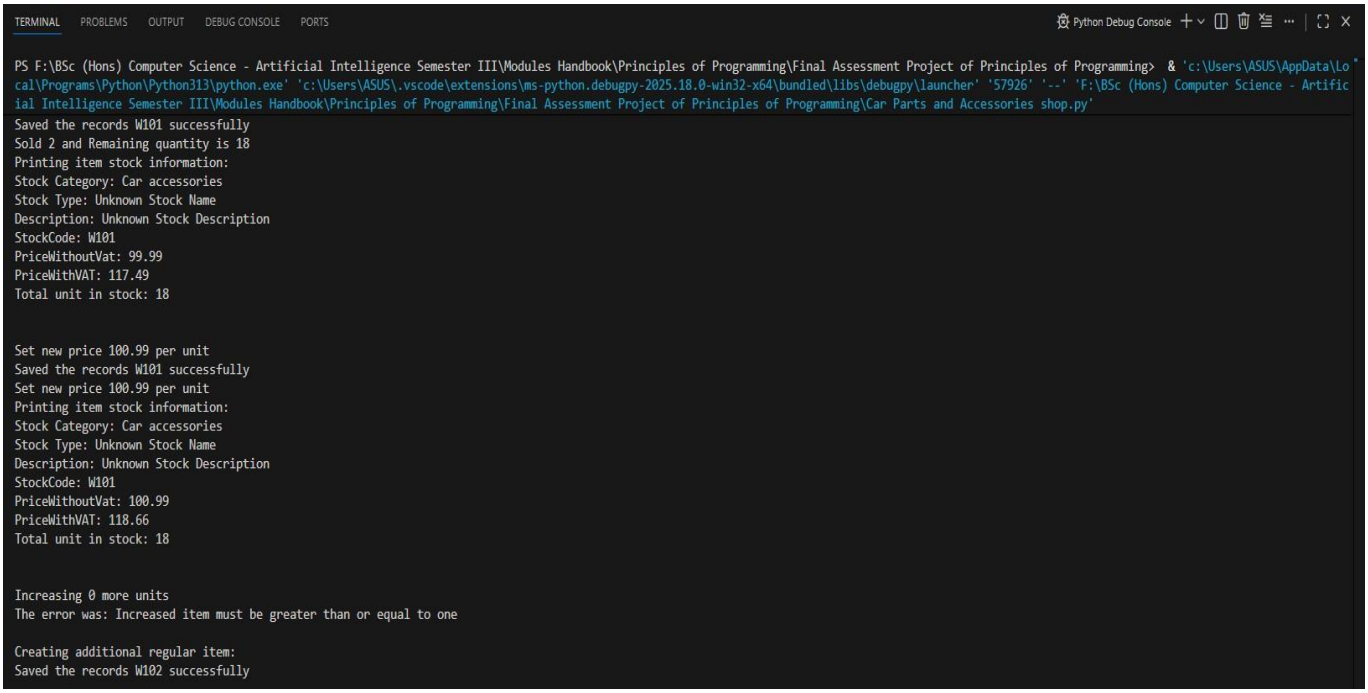


Figure 7: Console based run of the program (Part 2)

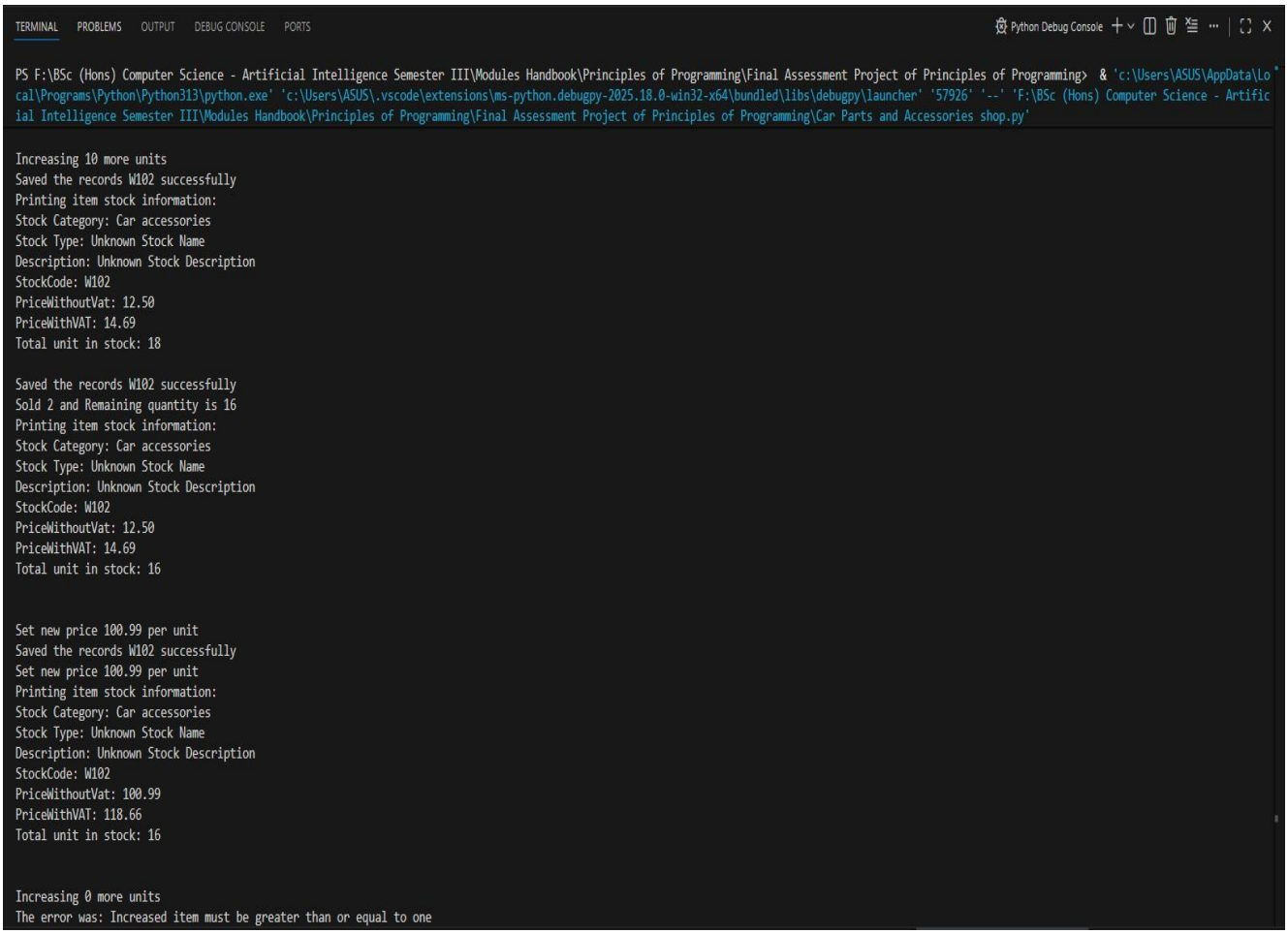


Figure 8: Console based run of the program (Part 3)

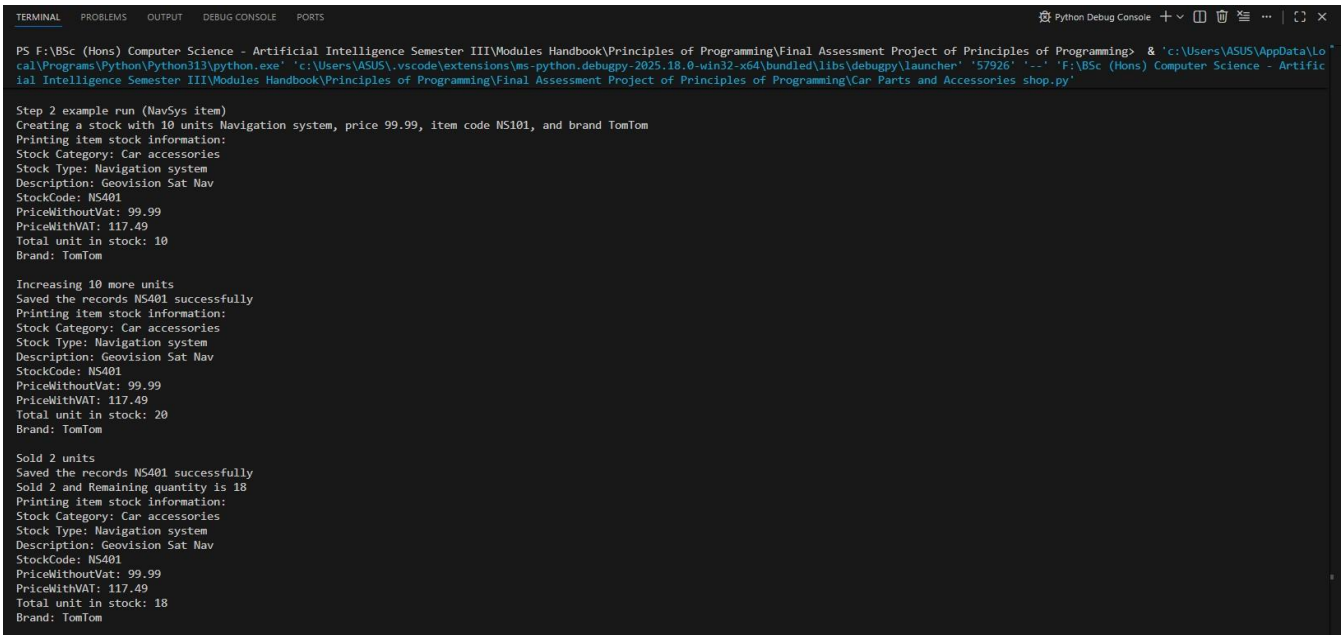


Figure 9: Console based run of the program (Part 4)

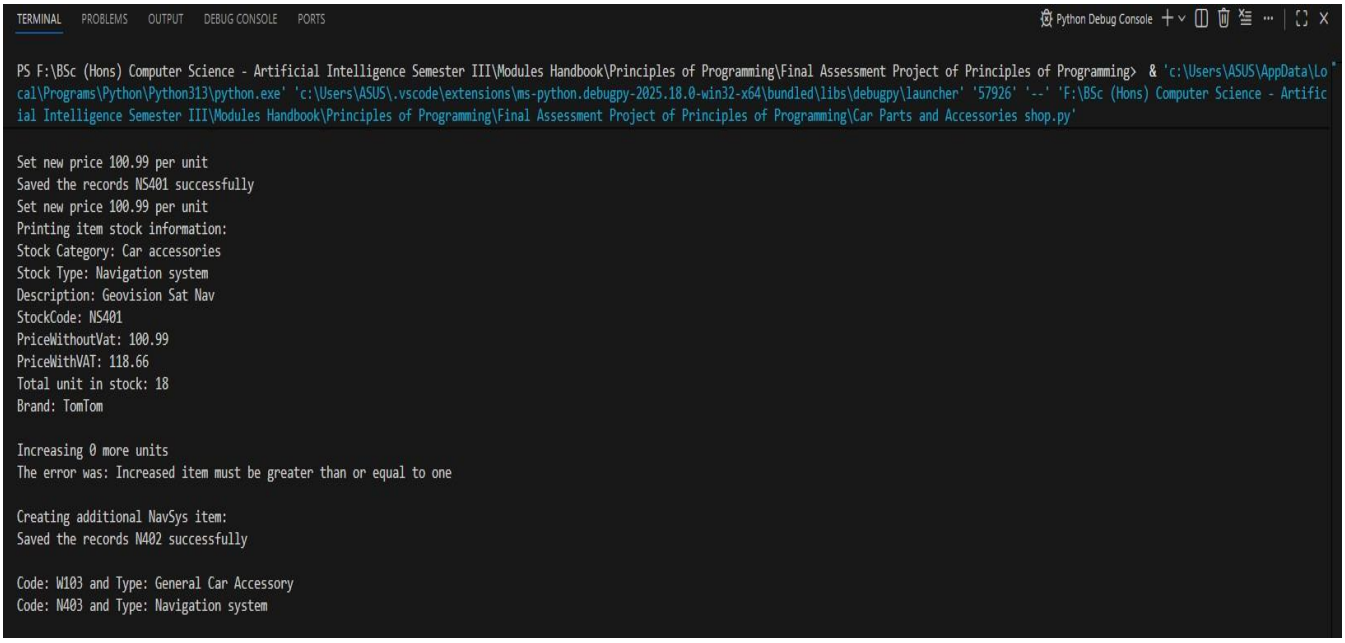


Figure 10: Console based run of the program (Part 5)

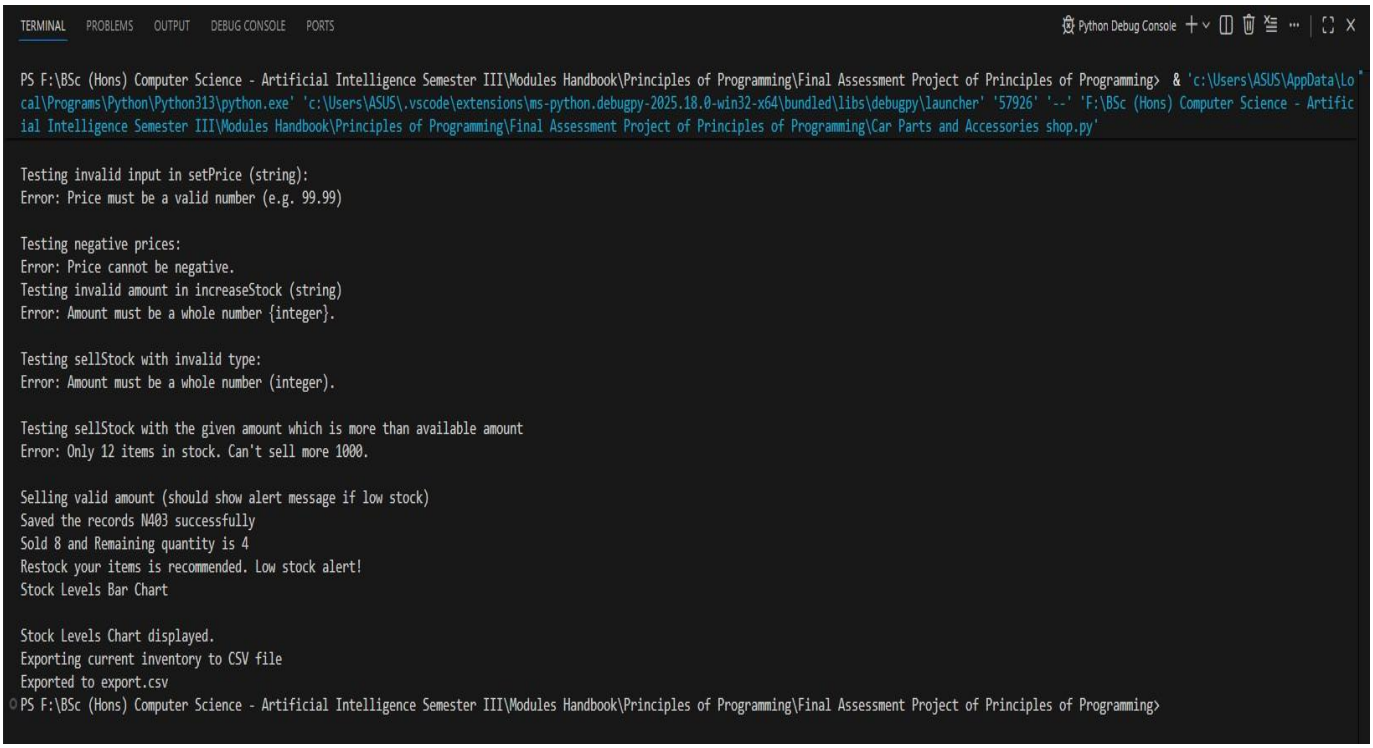
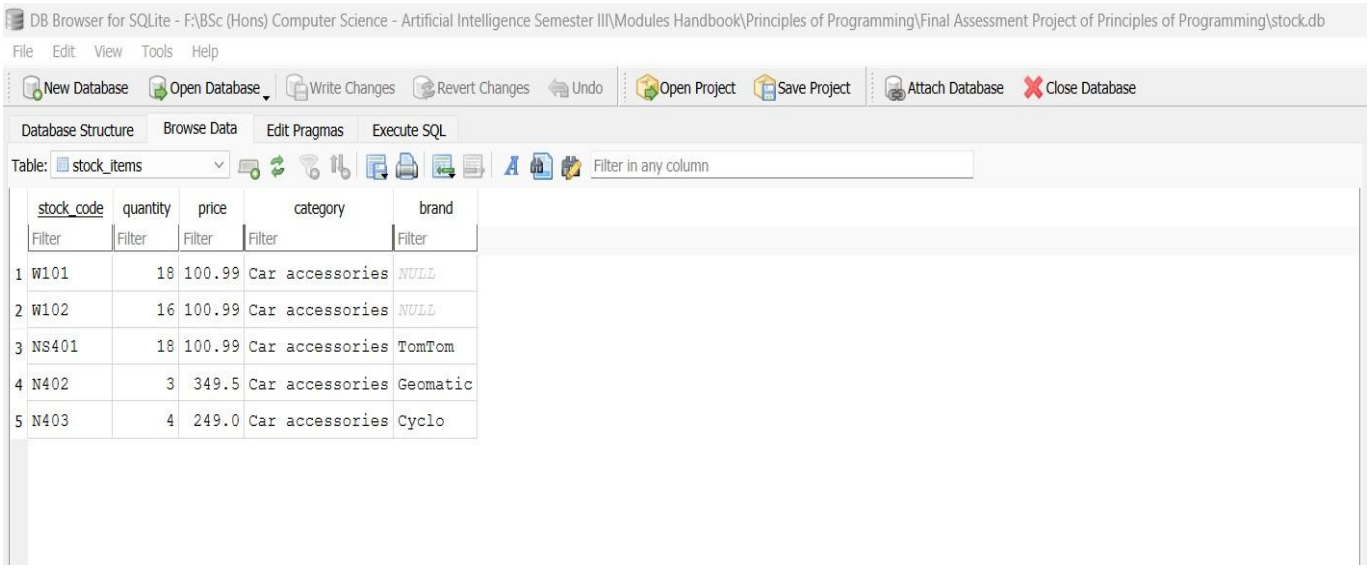


Figure 11: Console based run of the program (Part 6)



	stock_code	quantity	price	category	brand
	Filter	Filter	Filter	Filter	Filter
1	W101	18	100.99	Car accessories	NULL
2	W102	16	100.99	Car accessories	NULL
3	NS401	18	100.99	Car accessories	TomTom
4	N402	3	349.5	Car accessories	Geomatic
5	N403	4	249.0	Car accessories	Cyclo

Figure 12: SQLite database table

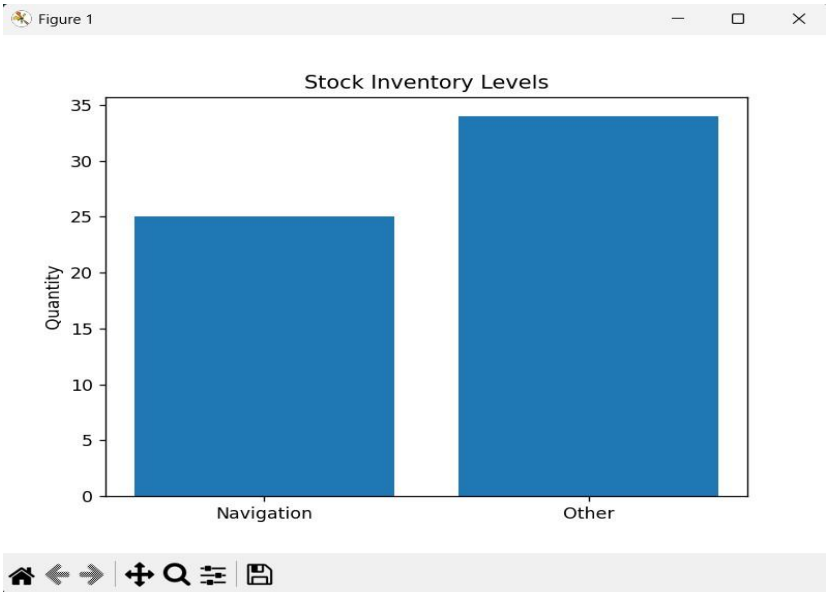


Figure 13: Displays bar chart with Navigation vs Other categories

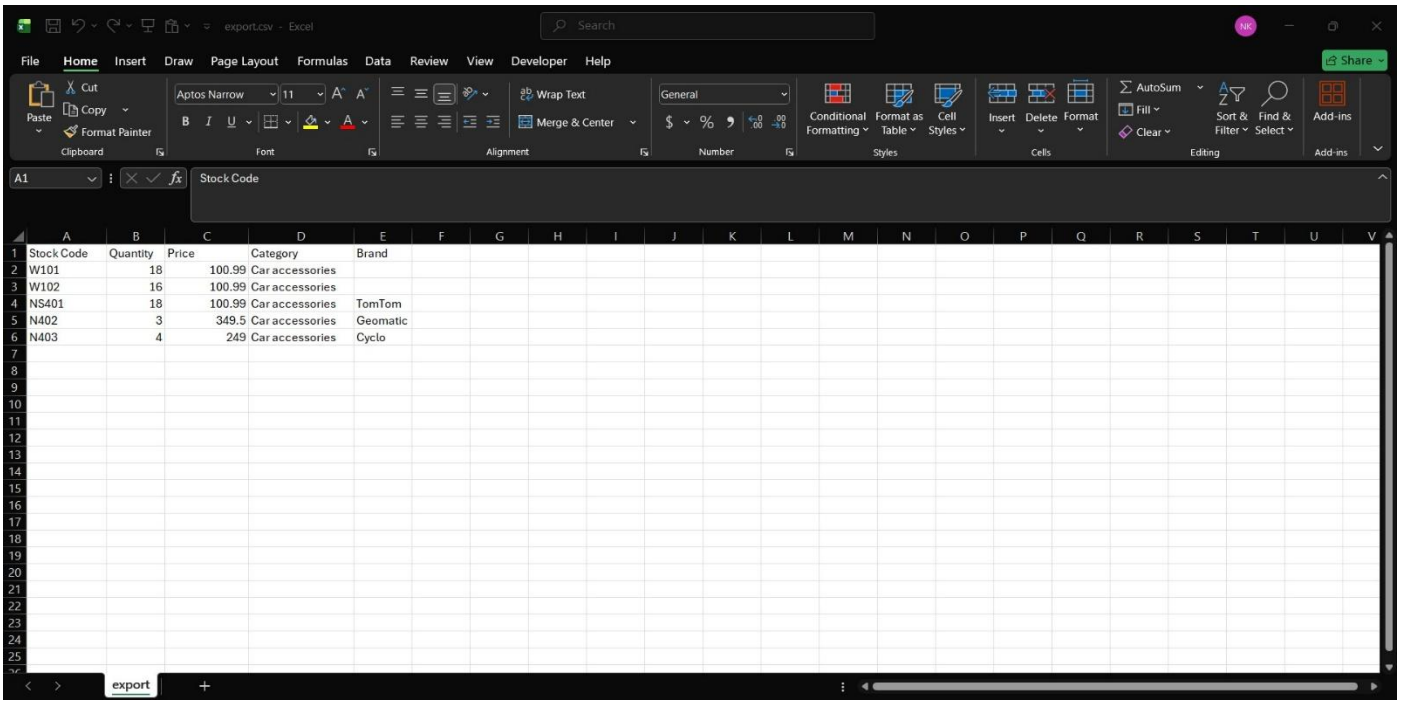


Figure 14: Creates custom CSV file with specified name

CHALLENGES FACED

- Difficulty in planning and system design.
- Understanding flow of the program.
- Creating an intuitive and user-friendly interface.
- To implement file handling with right file structure
- Validating checks for error input as part of error handling.
- Difficulty in making SQLite database connection.
- Problem of indentation and unreachable code warnings.

CONCLUSION

The inventory management system which I made for Car Parts and Accessories Shop project aims at improving stock tracking, sales processing, data management, and reporting for automative accessory retailers. The system is great example for replacing manual record-keeping with the help of digitalization, where accurate and reliable records of stocks and various item are possible.

The key features included in the system are object-oriented programming principles, SQLite database integration, CSV file exports for data management and reporting, and use of python library matplotlib for data visualization. The code implementation shows a clear class hierarchy that uses both inheritance and method overriding.

The project highlights the advantages of applying principles of object-orientation such as objects and classes, inheritance encapsulation, object state and modularity. Learning about testing and debugging programs is helpful for our future references

REFERENCES

Anon. (no date) Napkin AI - The visual AI for business storytelling. Napkin AI [online]. Available from: https://www.napkin.ai/ [Accessed 6 January 2026].

Anon. (no date) Python file open. W3schools.com [online]. Available from: https://www.w3schools.com/python/python_file_handling.asp [Accessed 6 January 2026c].

Anon. (2016) Python OOP concepts. GeeksforGeeks [online]. Available from: https://www.geeksforgeeks.org/python/python-oops-concepts/ [Accessed 6 January 2026].

Anon. (no date) [online]. fig. Available from: https://www.freepik.com/premium-ai-image/auto-parts-store-shelves-with-automotive-products-accessories_349332833.htm.

Anon. (2018) Introduction to SQLite. GeeksforGeeks [online]. Available from: https://www.geeksforgeeks.org/sql/introduction-to-sqlite/ [Accessed 8 January 2026].

Anon. (2021) Libraries in python. GeeksforGeeks [online]. Available from: https://www.geeksforgeeks.org/python/libraries-in-python/ [Accessed 12 January 2026].