**Online Shopping Cart**

A Python Project Submitted  
  
In

Computer Science  
  
  
by

Kunwar Shaurya Pratap Singh (2401330120086)

Akshita Singh (2401330120019)

Ankur Mehrotra (2401330120026)  
  
  
  
Under the Supervision of

Mr. Ajay Kumar

Assistant Professor, Computer Science



**Computer Science**

**NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY, GREATER NOIDA**

**(An Autonomous Institute)**

**Affiliated to**

**DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW**

**June, 2024-25**

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| Serial No. | Content | Page.no | Remarks |
| 1 | Certificate | 3 |  |
| 2 | Acknowledgement | 4 |  |
| 3 | Introduction | 5 |  |
| 4 | Objectives | 6 |  |
| 5 | Literature Review | 7 |  |
| 6 | Methodology | 8 |  |
| 7 | Project Plan | 9 |  |
| 8 | Result and Discussion | 10 |  |
| 9 | Conclusion and Recommendation | 12 |  |
| 10 | References | 13 |  |
| 11 | Appendices | 14 |  |

**Certificate**

This is to certify that *Kunwar Shaurya Pratap Singh, Akshita Singh, Ankur Mehrotra* have successfully completed project in Advance Python[BCSE0252] at Noida Institute of Engineering and Technology

Date:

Name:

Signature:

Designation:

**Acknowledgement**

We would like to express my sincere gratitude to Noida Institute of Engineering and Technology for allowing us to work on Advance Python[BCSE0252]. We are grateful to our *Mentor Mr. Ajay Kumar* for their guidance and support throughout the project. We also acknowledge the contributions of colleagues who helped us complete this project.

Thank you all for your support and encouragement.

Sincerely,

Kunwar Shaurya Pratap Singh

Ankur Mehrotra

Akshita Singh

**Introduction**

This project titled **“Online Shopping Cart System Using Python,”** simulates the functionality of an e-commerce shopping cart through a command-line interface. In the modern digital era, online shopping is becoming increasingly common, and shopping carts are a critical component of these platforms. This project is designed to provide a basic yet functional shopping cart system that enables adding, removing, and updating products using object-oriented programming in Python. which enhances the modularity, reusability, and scalability of the system. The four fundamental concepts of OOP—**Encapsulation**, **Abstraction**, **Inheritance**, and **Polymorphism** are effectively applied in the development of this system:

* **Encapsulation** is used to bundle the data (such as product details and cart items) and the methods that operate on that data into single units called classes. This approach restricts direct access to internal object details and promotes data protection by using methods to interact with the data.
* **Abstraction** is implemented by exposing only the essential functionalities to the user while hiding the internal implementation details. For instance, users can add or remove products from the cart without needing to know how the product list is maintained internally.
* **Inheritance** is demonstrated through the creation of classes that inherit attributes and behaviours from parent classes. This avoids redundancy and allows common functionality to be reused across multiple components of the system. For example, a base class for generic items may be extended by more specific product classes.
* **Polymorphism** allows objects to be treated as instances of their parent class rather than their actual class. This enables flexibility in method implementation and supports method overriding, allowing different classes to define the same method in a way that's appropriate for their specific behaviour.

By integrating these OOP principles, the system achieves a clean and organized code structure, making it easier to manage and extend in future developments. This project not only serves as a practical example of applying object-oriented programming in Python but also simulates a real-world application relevant to today’s digital commerce environment.

**Objectives**

* To understand and apply object-oriented programming concepts.
* To simulate a real-world shopping cart scenario.
* To practice file handling and persistent data storage using JSON.
* To design a menu-driven console application.

This project demonstrates the power of Python in building real-world applications and helps understand the working model of a small-scale e-commerce system. It also lays a strong foundation for future development in web applications.

**Literature Review**

The idea of a digital shopping cart originates from early online commerce models in the late 1990s. Over the years, e-commerce platforms like Amazon and Flipkart have optimized the concept to allow seamless product selection and checkout processes.

The traditional cart includes features like:

* Add to cart
* Remove from cart
* Update quantity
* Display total and cart summary
* Session management

Our project draws inspiration from such features, adapted to a local command-line interface with Python. Prior works in this space primarily use web-based interfaces, but our goal was to implement the backend logic without relying on frontend technologies.

**Methodology**

**Design Approach:**

The system is built using object-oriented principles.

Three main classes were created:

* *Product* (with its subclasses)
* *CartItem*, and
* *ShoppingCart.*

**Data Handling:**

Products and cart data are stored using Python dictionaries and JSON for persistent storage.

The catalog and cart state can be saved and loaded through JSON files.

**Workflow:**

* User starts the CLI-based system.
* They can choose from options such as adding a product, removing it, or checking out
* Each product is identified by a unique product ID.

**Project Plan**

**The development was planned over four stages:**

* Requirement Analysis and Feature Listing – Deciding core features like *add/remove/update/display*.
* Design – Building the Product class hierarchy and *ShoppingCart* logic.
* Implementation – Writing and testing the code.
* Documentation and Reporting – Screenshots, testing output, and report preparation.

**Resources Used:**

* Python 3.x
* VS Code (IDE)
* Windows OS
* json module
* typing module

**Results and Discussion**

*Code Architecture Breakdown*  
  
 **Product Class (Base Class)**: Defines the basic attributes of a product:

* *product\_id*
* *name*
* *price*
* *quantity\_available*Also includes methods to:
* increase/decrease stock
* display product details
* convert product data to dictionary format

**DigitalProduct and PhysicalProduct (Subclasses)**

*DigitalProduct* adds a *download\_link*. *PhysicalProduc*t adds a weight attribute. They override the *display\_details()* method to show additional info.

***CartItem Class*** Stores:

product (object)  
quantity selected by user  
Includes a method to calculate the subtotal (price × quantity).

**ShoppingCart Class:** Maintains a dictionary of items (*product\_id → CartItem*).  
  
Key functions:

* *add\_item (product\_id, quantity)*
* *remove\_item(product\_id)*
* *update\_quantity (product\_id, new\_quantity)*
* *get\_total () and display\_cart ()*
* *save\_cart\_state() and load\_cart\_state() via JSON*

**JSON Data Handling**  
 Two files are used:

* *catalog.json*: stores product data
* *cart\_state.json*: stores shopping cart contents persistently

**Main Loop & Menu**  
The user interacts through a terminal interface with options to:

* View products
* Add to cart
* Remove/update
* View cart
* Exit and save session

**Conclusion and Recommendations**

This project successfully simulates an online shopping cart system using Python and demonstrates the practical application of object-oriented programming. Through the use of custom classes, JSON handling, and CLI interaction, the project offers a foundational backend system for a shopping interface.

**Recommendations for future improvements:**

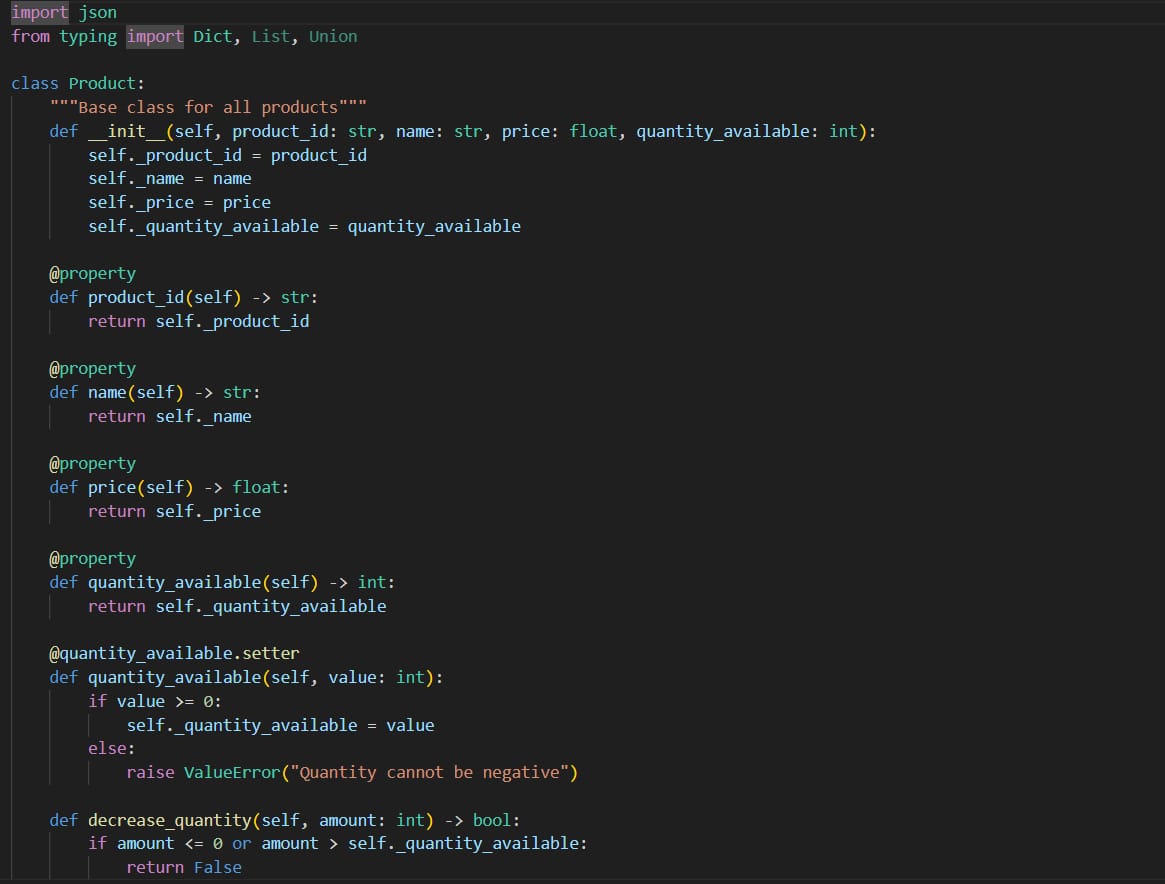
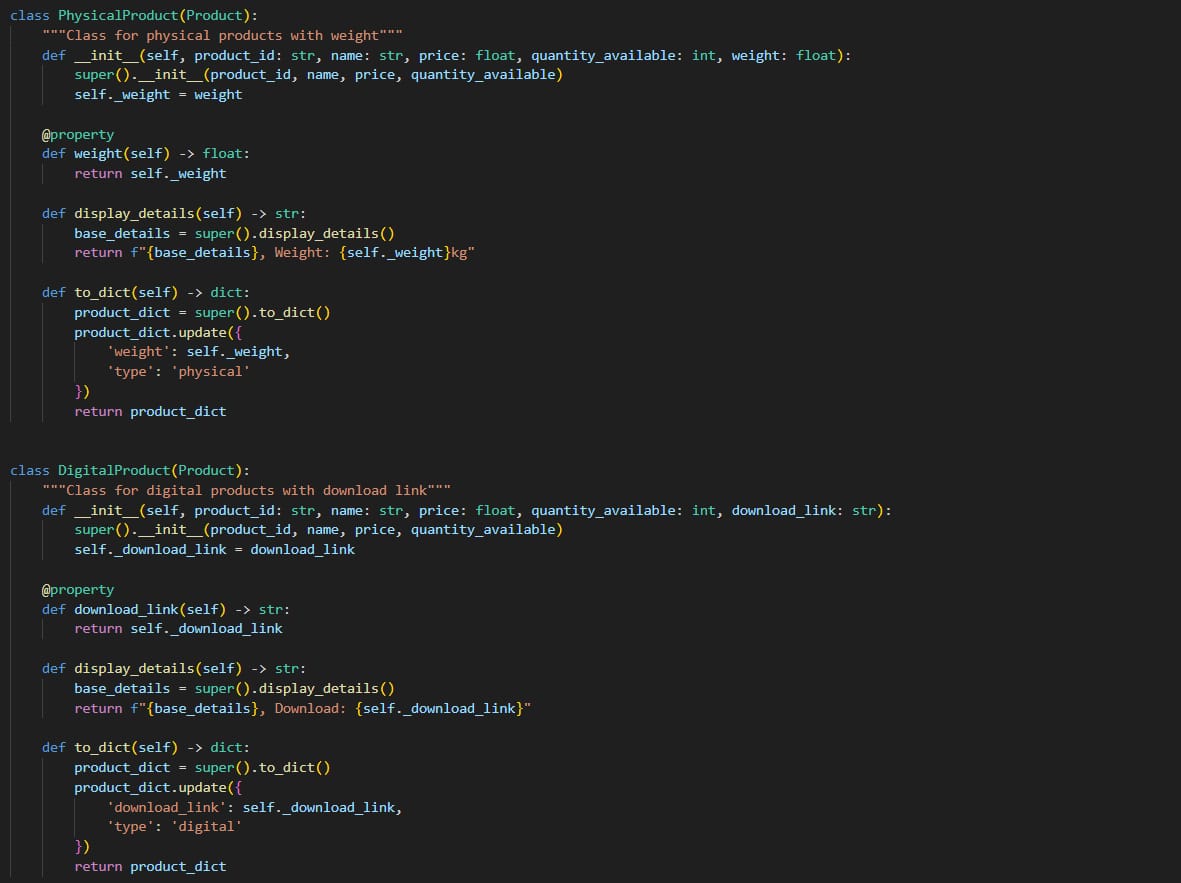
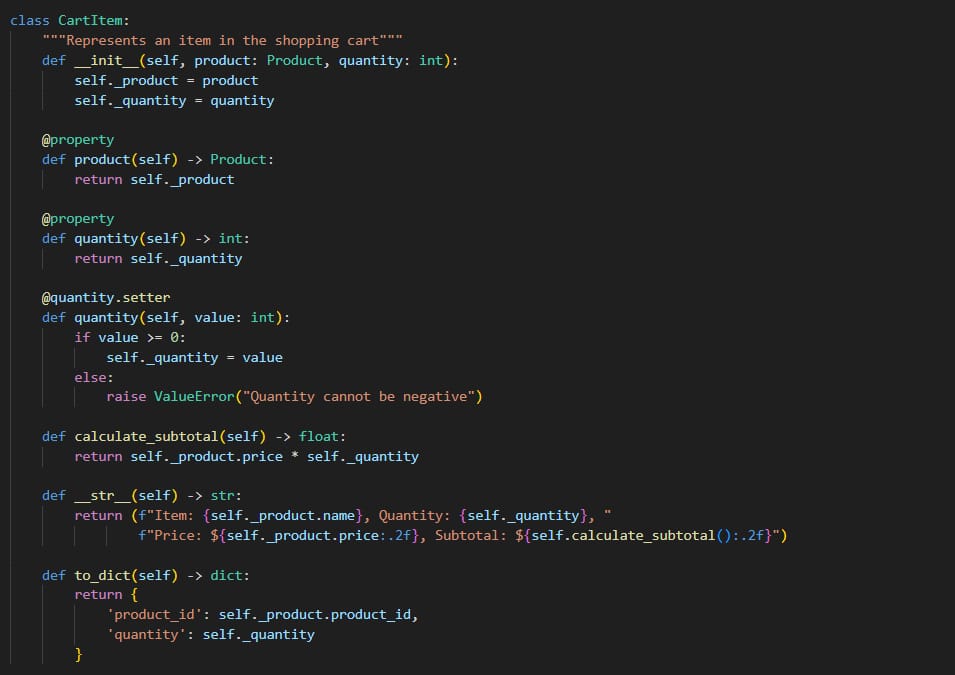
* Implement a GUI using Tkinter.
* Add user authentication.
* Connect to a database like SQLite for multi-user support.
* Introduce categories, product filtering, and search.

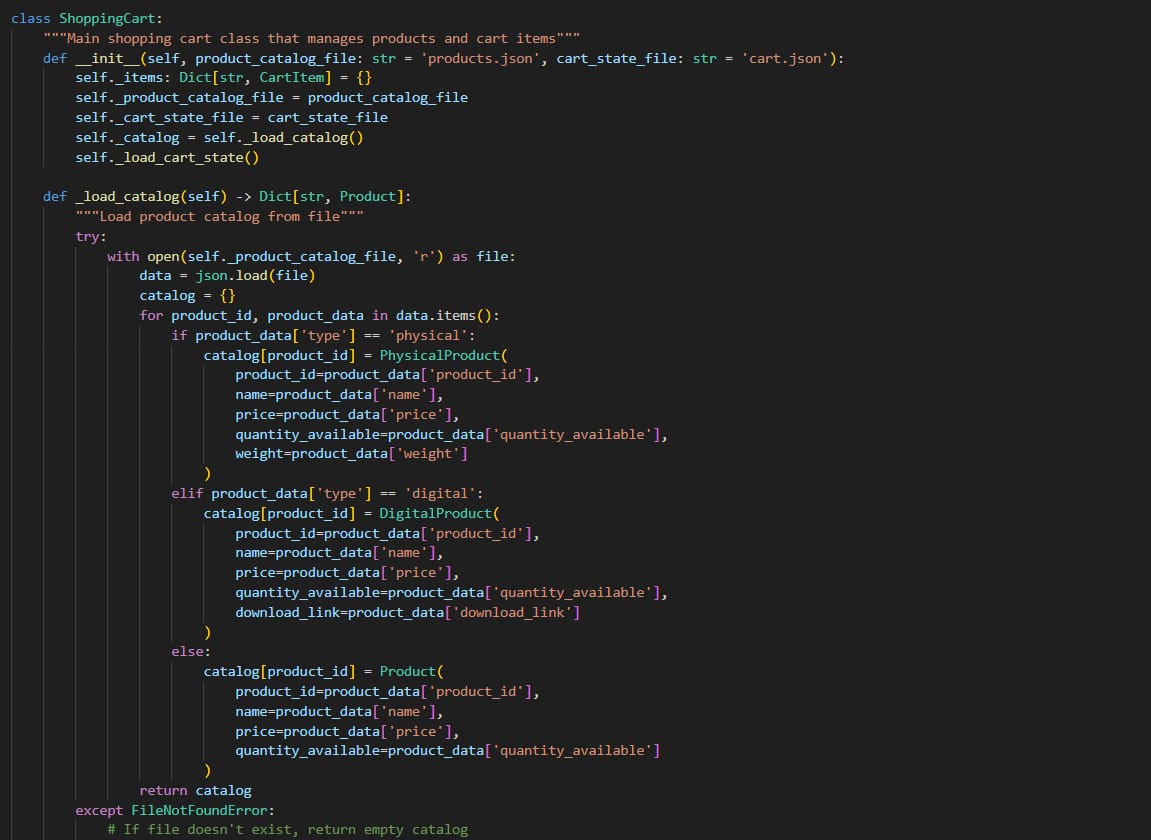
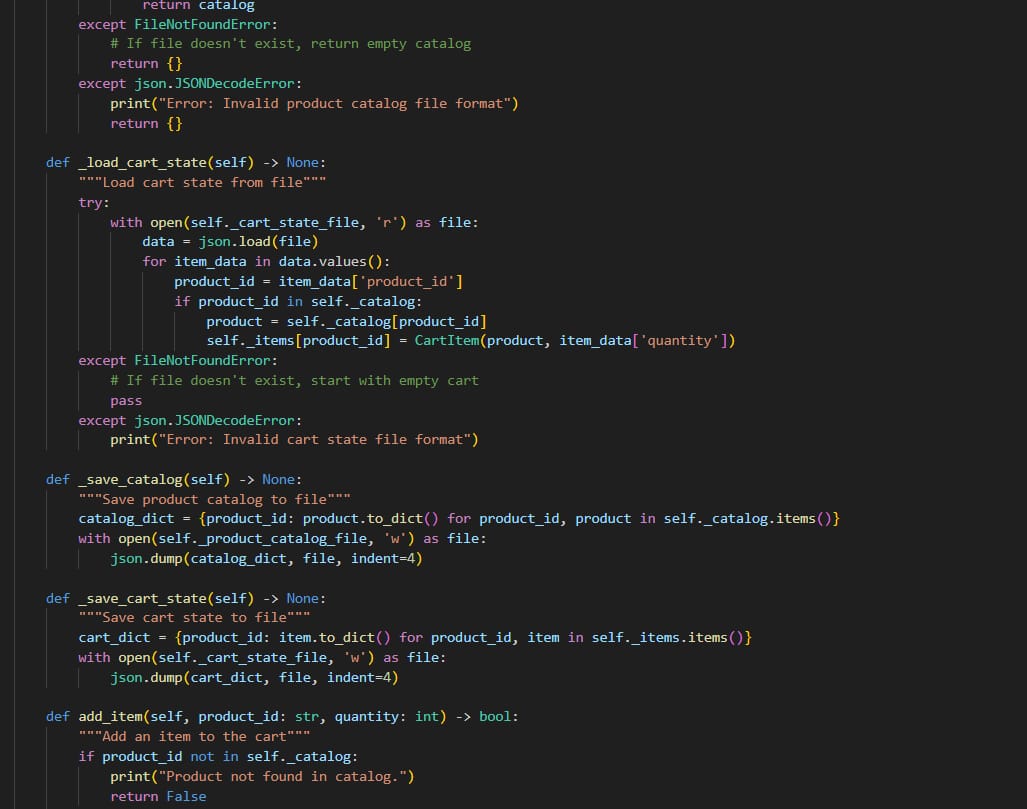
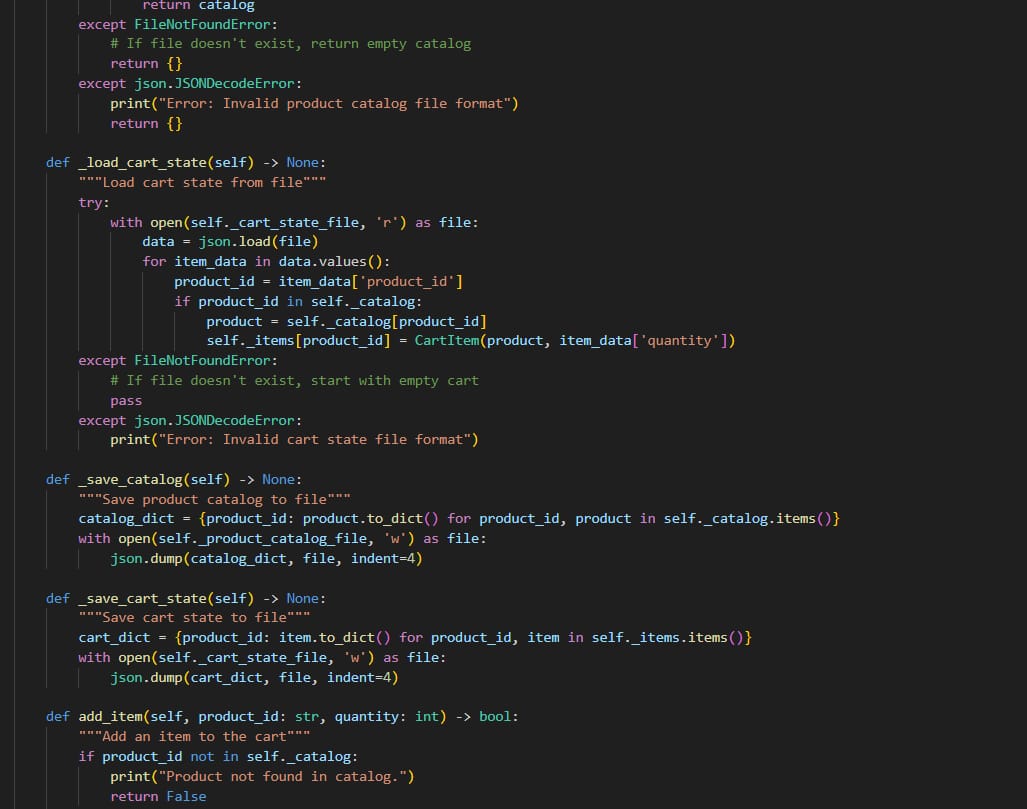
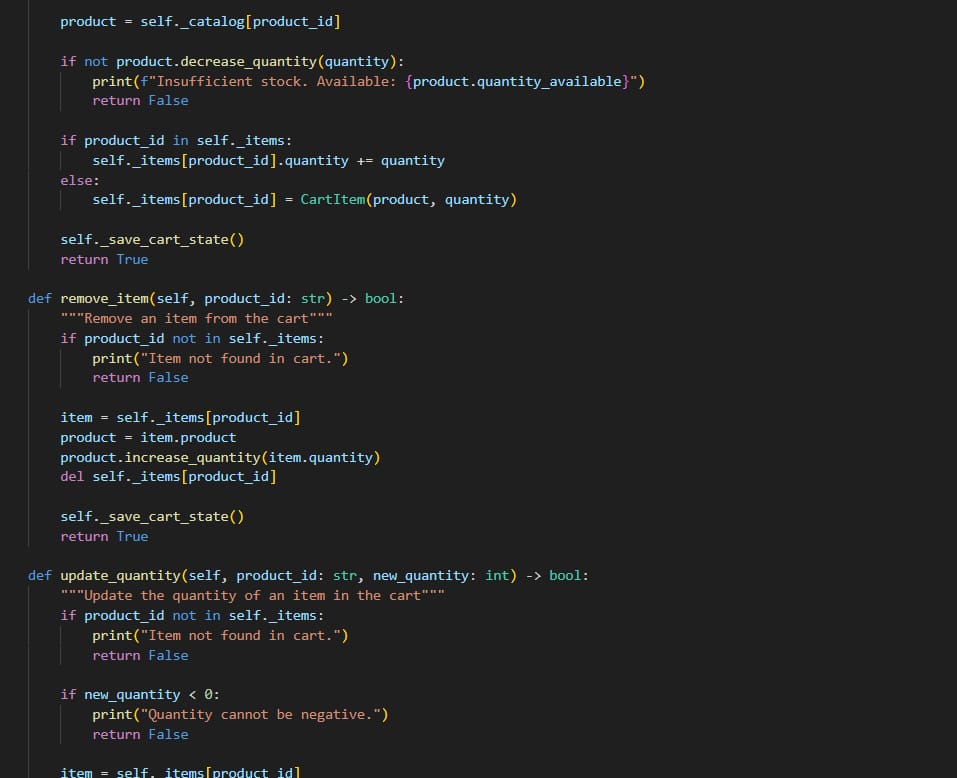
**References**

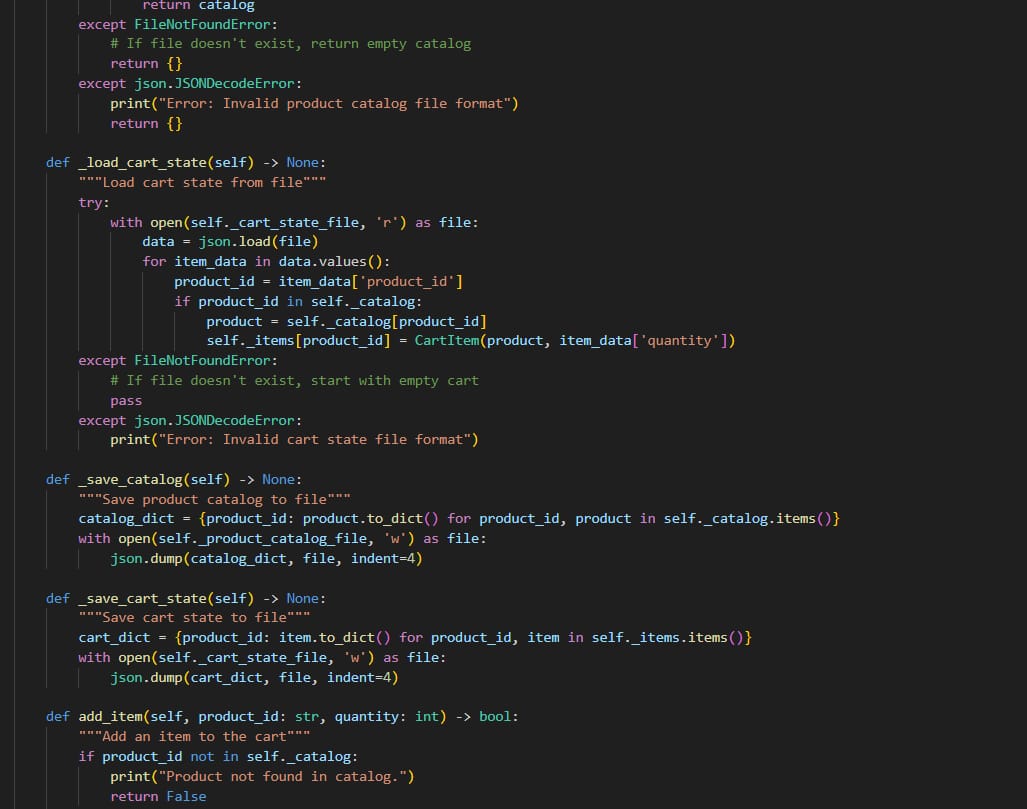
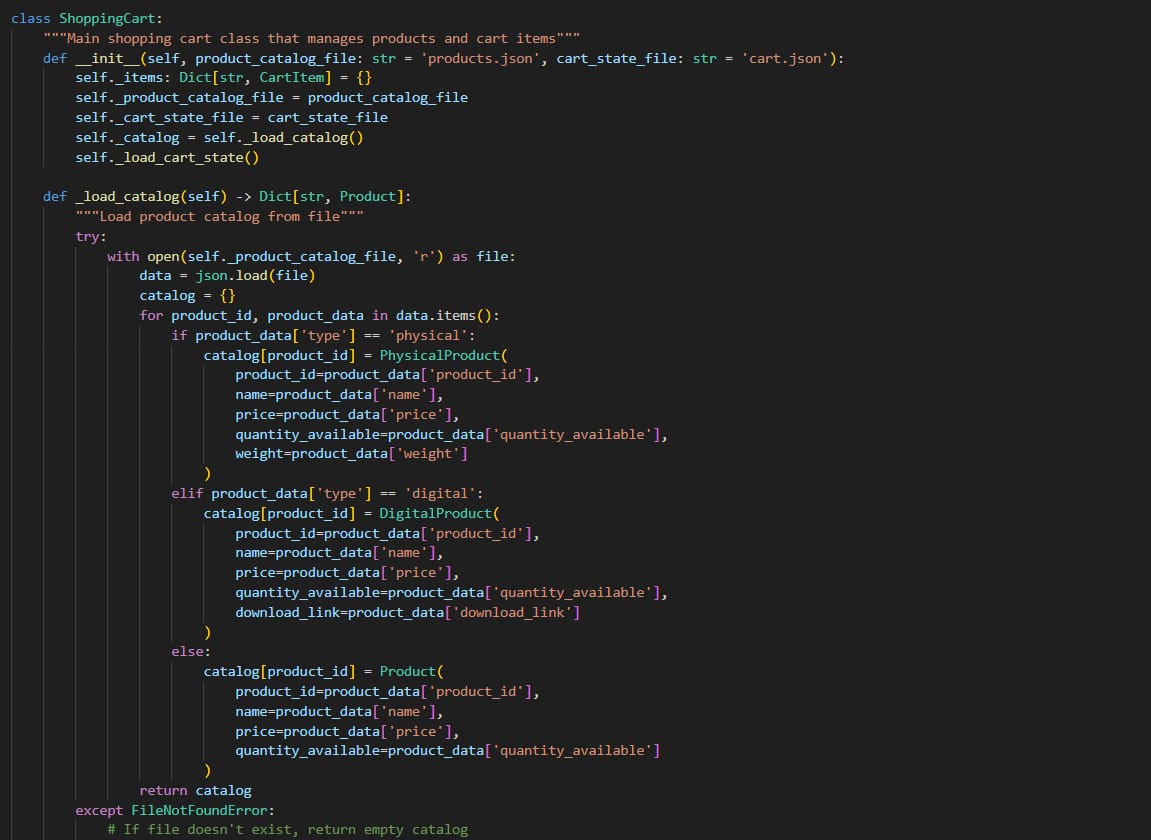
* Python Documentation: *https://docs.python.org*
* GeeksforGeeks: *https://geeksforgeeks.org*
* Stack Overflow: *https://stackoverflow.com*
* Class notes and lectures

**Appendices**

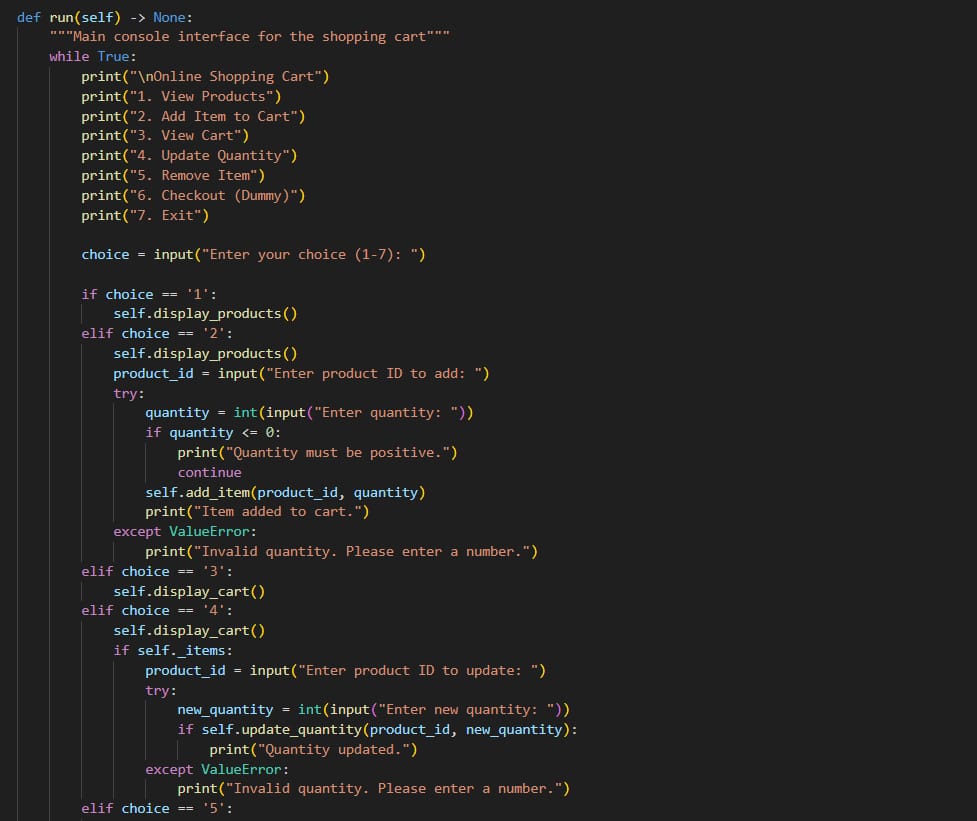
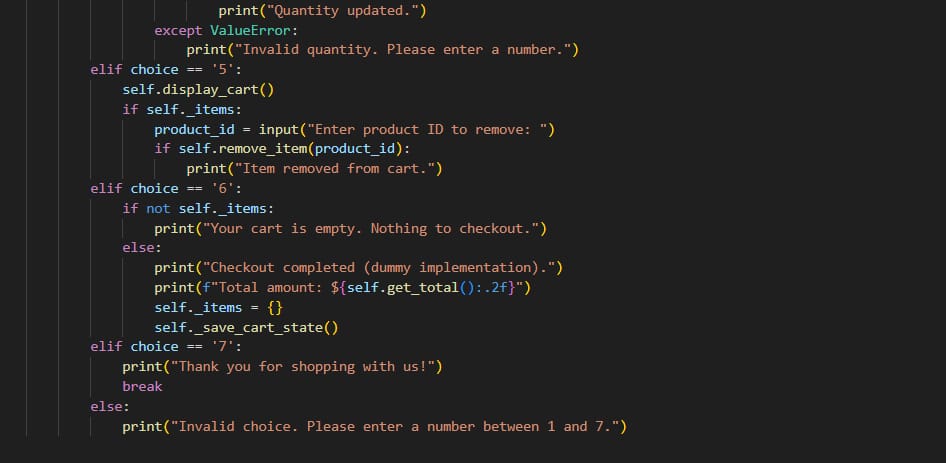
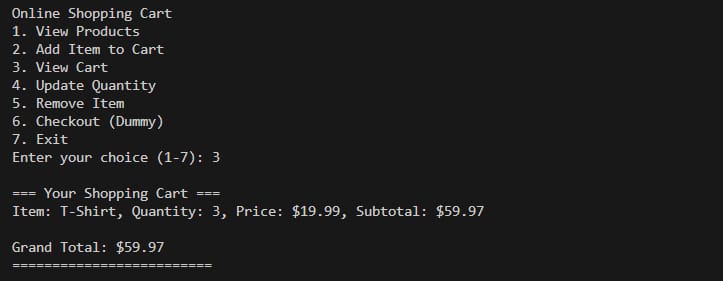
*Screenshot Output:*  
  
**Class Implementations:** To Explain structure of classes (Product, DigitalProduct, PhysicalProduct, CartItem)

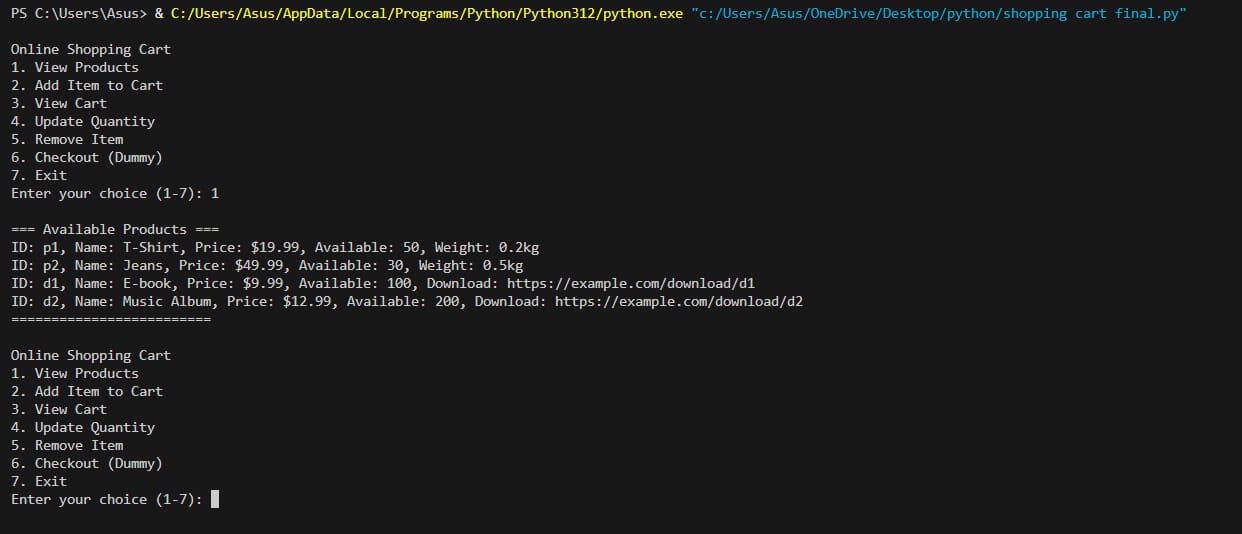
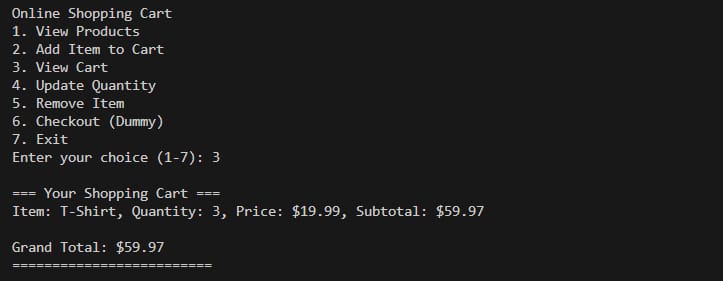
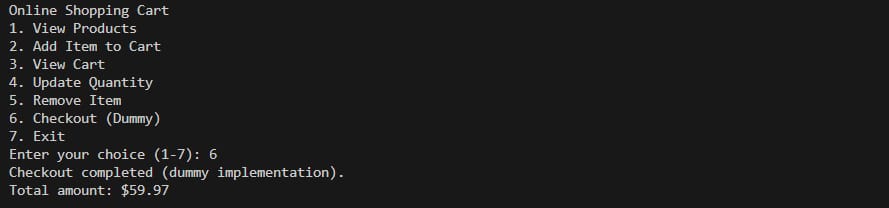
  
  
  
  
  


**Product Creation and Catalog**: Shows how products are loaded, stored, and shown to users.  
  
  
  
  
  
**Cart Operations (Add/Remove/Update)**:  
  
  
  


**Saving and Loading Data:** Persistent cart and catalog management using JSON  
  
  


**Cart Display and Total Price Calculation:**

**  
  
  
  
**

**Sample Execution Output:  
  
  
  
  
  
  
  
  
**