University of Warwick

Warwick Manufacturing Group (WMG)

e-Business Management

Digital Development with Python

Project 4: Customer Loyalty Management System

You must develop a Customer Loyalty Management System using Python to manage a customer loyalty register within an e-business purchase environment. The project aims to enhance customer engagement and retention by fostering a rewarding experience for loyal customers. The key objectives include establishing a customer database and implementing functions for loyalty management. This system enables the calculation of loyalty points, offers rewards for accumulated points, and provides insights into customer loyalty status.

For this purpose, the system must contain the following key features:

1. **Key features:**

- Customer Database: Implement a customer database to store information the user provides, such as customer name, contact details, and loyalty points.
- Functionality: Create functions to perform operations on the inventory, including:
  - Loyalty Tracking Functions:
    - Recording loyalty points earned per transaction per customer. You do not need to record the transaction, but the points earned when a transaction occurs.
    - Calculate and display the total loyalty points per transaction.
    - Displaying the total loyalty points for each customer.
  - Customer Loyalty Management Functions:
    - Offering loyalty rewards based on accumulated points.
    - Displaying customer loyalty status.
    - Resetting loyalty points after redemption.
- Loyalty Reporting System: Create a reporting system that provides insights into customer loyalty, including the total loyalty points earned per customer and loyalty points redemption history.

2. **Minimum requirements:**

- Python Functions: the project must rely on the use of functions to modularise code and enhance reusability.
- Dictionaries: utilise dictionaries as the primary data structure for representing the customer database and storing information about loyalty points, rewards, and redemption history.
- User Interface: create a simple text-based user interface to interact with the inventory system. This includes recording loyalty points per purchase transaction, managing customer loyalty, and accessing loyalty reports.

- Error Handling: implement error-handling mechanisms to deal with invalid inputs. For example, recording purchases for non-existent customers, offering rewards without sufficient points, etc.

3. Submission guidelines:

- Code Submission: Submit a Python script (*.py* or *.ipynb* file) containing the complete code for the Customer Loyalty Management System.
- Comments: write comments in your code explaining the logic behind each function and clearly specify the chunk of code that implements each project feature.
- Documentation: Write a brief documentation file with the structure of the product database and key decisions made during the development process.
- Testing: Provide a separate section in the documentation that outlines how the system was tested. Include input sample scenarios and expected outcomes.

4. Evaluation Criteria:

- Functionality (40%): All key features and minimum requirements must be implemented and should work as expected.
- Code Structure (20%): The code should be well-organized, with clear and modular functions. Comments should be added to explain the computational thinking.
- Use of Dictionaries (15%): Dictionaries should appropriately represent the customer database and store information about loyalty points and rewards.
- User Interface (15%): The user interface should be intuitive, allowing users to interact with the system easily.
- Error Handling (10%): The system should handle invalid inputs and potential errors.