**Flexcar - Take Home Assignment - E-commerce**

**Problem Statement:**

You have been tasked with designing an inventory and promotions engine for an e-commerce software platform. As a business-to-business software platform, customers of your product can use the promotions engine to create promotions for items in their inventory.

**Item Requirements:**

Items can be sold by weight or quantity.

Items can be grouped into categories

Items can have a brand

More than one item of each type of item can be added to the cart

Items do not have taxes for this problem.

**Promotion Requirements:**

**Promotions can have the following types:**

Flat fee discount (ex: 20 dollars off on an item)

Percentage discount (ex: 10% of off an item)

Buy X Get Y discount (ex: Buy 1 get one free, buy 3 get 1 50% off)

Weight threshold discounts (ex: buy more than 100 grams and get 50% off the item)

Promotions can be valid for individual items, or categories (ex: 50% off on all products of X category)

Promotions must have a start time

Promotions may have an end time

Multiple promotions can be applied to a cart if they both have a valid set of items

Each item is only valid for one promotion

Only one instance of a promotion can be applied at a time

**Your solution must contain:**

The ability to add items to a cart.

The ability to remove and view items in the cart

Every time an item is added to the cart the best possible price given the valid promotions must be shown.

General Guidelines

**Clarity and Readability:**

Write clear, concise, and readable code.

Use meaningful variable and method names.

Comment your code where necessary, but avoid over-commenting.

**Structure and Organization:**

Organize your code logically into classes and modules.

Follow the principles of Object-Oriented Programming (OOP).

Adhere to the Single Responsibility Principle (SRP) to ensure each class and method has one clear purpose.

**Conventions and Style:**

Follow Ruby community style guidelines. We recommend using the .

Consistently format your code. Consider using tools like RuboCop to enforce style conventions.

Technical Guidelines

**Dependencies:**

Use Bundler for managing gem dependencies. Include a Gemfile with the required gems.

Ensure all dependencies are documented and included in your submission.

**Code Execution:**

Provide clear instructions on how to run your code. Include a README file with setup and execution steps.

Ensure your code runs correctly and meets the assignment requirements.

**Error Handling:**

Implement proper error handling to manage unexpected inputs or issues gracefully.

Avoid using rescue clauses for control flow.

**Performance:**

Optimize your code for performance where applicable, but not at the expense of readability.

Consider the time and space complexity of your solutions.

Submission Guidelines

**Documentation:**

Include a README file that explains your approach, how to set up and run the project, and any other relevant information.

Document any assumptions you made and explain any decisions that might not be immediately clear.

**Code Quality:**

Ensure your code is free from syntax errors and warnings.

Clean up any debugging statements or commented-out code before submission.

**Submission Guidelines**

* Submit a **clean project** with:
* No **errors**
* No **warnings**
* No **console debug logs**
* Push the source code in your own public git repository and share the url with us
* Include a well-structured **README** file that covers:
* Project setup and installation instructions
* How to run the application and tests
* Any design decisions or assumptions
* Please attach this take-home assignment file in the root directory of the application

**Your submission will be evaluated based on the following criteria:**

Correctness: Does your code solve the problem correctly?

Code Quality: Is your code clean, readable, and well-organized?

Testing: Are your tests comprehensive and passing?

Documentation: Is your documentation clear and helpful?

Efficiency: Is your code optimized for performance where applicable?

We look forward to reviewing your submission. Good luck!