Assumptions :

1. Admin creates discount settings with discount amount, min transactions, and discount code.
2. Once the admin creates such a discount setting, it is called a “current or active discount setting”.
3. There is only one active discount setting at a given time.
4. As soon as a customer completes minimum transactions required, he/she earns the discount of the code and amount mentioned in the current discount setting.
5. For every subsequent set of minimum number of transactions required for the “current discount setting”, the customer earns the discount again per “current discount setting”.

(For ex. If the active discount setting is :[discountCode = “ttq”, minTransactionsRequired: 3, discountAmount=200] and customer with id 8 completes 3 transaction, he/she earns a discount of $200 with code “ttq”.

If (s)he completes 3 more purchases (s)he earns one more discount of $200 with code “ttq”.

If, before the customer completes 3rd transaction, admin changes the active discount settings to :[discountCode = “tpq”, minTransactionsRequired: 5, discountAmount=300], the customer has to complete 3 more purchases to reach to 5 and earn $300 discount of code “tpq”)

1. Admin can check total discount given to all the customers by making a get call to the /api/totaldiscounts and he/she can also make a get call to /api/totaldiscounts/{discountCode} to find how much total discount was given with a certain discountCode.

Usage:

1. Git clone and open the store spring boot application in any IDE.
2. Use spring-boot:run to run the app. I use IntelliJ
3. Use Postman to test the app.

APIs: Swagger : http://localhost:8080/swagger-ui.html#/

1. POST /api/purchase: Customer creates a Purchase transaction.

JSON i/p for purchase without any discount:

*{*

*"customerId": "8",*

*"cost":200*

*}*

JSON i/p for purchase with a discount (after customer earns discount):

*{*

*"customerId": "8",*

*"cost":200,*

*"discountCode":"ttq",*

*"discountAmountUsed":200*

*}*

1. POST /api/discount: Admin creates a discount setting.

JSON i/p for purchase without any discount:

*{*

*"discountCode": "ttq",*

*"discountAmount":200,*

*“minTransactionsRequired”:3*

*}*

1. *GET /api/totaldiscounts: Admin reads total discounts given to all the customers*
2. *GET /api/totaldiscounts/{discountCode}: Admin reads total discounts given to all the customers for any discountCode*
3. *GET /api/discounts/{customerId}: }: Admin reads total discounts earned by a customer.*

Tests : (Customers can start purchases before admin creates discount settings.)

1. Customer 8 makes a purchase of $200 3 times.
   * Make 3 a POST requests to <http://localhost:8080/api/purchase> with the following json:

*{*

*"customerId": "8",*

*"cost":200*

*}*

* + Expected output: You see a message : Your purchase is successful. With every purchase.

1. Check if the customer 8 has earned any discount
   * Make a GET request to <http://localhost:8080/api/discounts/8>
   * Customer 8 has not earned any discount yet because admin has not set a settings for discount.
   * Expected output: You should see the following message
   * *{*

*"status": "OK",*

*"message": "This customer has not earned any discount",*

*"debugMessage": ""*

*}*

1. Admin now creates a discount setting.
   * Make a POST request to <http://localhost:8080/api/discount> with the following json body:

*{*

*"discountCode": "ttq",*

*"discountAmount":300,*

*"minTransactionsRequired":3*

*}*

* + The above POST request means, admin is creating discount with code as “ttq”, amount as $300. Customer has to complete minimum 3 transactions to get this discount.
  + Expected output: You should see the following message

“Discount Created”

* + Check again if customer 8 has earned any discount by making a get request to /api/discounts/8 as shown in #2 above. Since customer 8 has completed 3 purchases, he has now earned a discount as per admin’s settings. So the get request in #2 above should show the following:
  + {

"customerId": "8",

"discountList": [

    {

            "discountCode": "ttq",

            "discountAmount": 300

       }

    ]

}

1. Customer 8 makes 3 more purchases and he now earns more discount of code “ttq” .
2. Make a GET request to /api/discounts/8 and see that customer 8 has total of $600 discount of code “ttq”.
3. Admin can now make a get request to <http://localhost:8080/api/totaldiscounts/ttq> to find that the system has given out total discount of $600 with discount code = ttq
4. Now another customer with id “2” makes 4 purchases of $200 each. (Please check #1 above to learn how to make purchase requests.)
5. Since the customer 2 has made 4 requests, he earns discount of $300 of code “ttq”. You can double check this by making a GET request /api/discounts/2
6. The customer 2 now makes 2 more purchases. Since he has made total of 6 purchases, he has total of $600 discount with code “ttq” .
7. Customer 8 now makes purchases using the discount. To do so, make a POST request to /api/purchase with the following JSON

*{*

*"customerId": "8",*

*"cost":200,*

*"discountCode":"ttq",*

*"discountAmountUsed":200*

*}*

1. After the above purchase, check how much discount customer 8 now has by making GET request to /api/discounts/8 similar to #2 above. As he used $200 discount out of 600, he now has $400

Expected Output:

*{*

*"customerId": "8",*

*"discountList": [*

*{*

*"discountCode": "ttq",*

*"discountAmount": 400*

*}*

*]*

*}*

1. You can try this with customer 2 as well.
2. Check the total discount the system has given so far by making a GET request to <http://localhost:8080/api/totaldiscounts>.

Expected output: Total discount offered: 1200