**Customer Rewards Program - Spring Boot Application**

This Spring Boot application calculates reward points for customers based on their monthly transactions. The application follows a RESTful architecture and includes a fully functional backend service.

**Table of Contents**

1. Project Overview

2. Technologies Used

3. Reward Points Calculation Logic

4. Project Structure

5. Endpoints

6. Sample Data and Response

7. Testing the Application

**1. Project Overview**

The Customer Rewards Program calculates reward points for customers based on their purchases. The rewards are calculated as:

1 point for every dollar spent between $50 and $100.

2 points for every dollar spent over $100.

Example:

A purchase of $120 will earn:

1 point for $50 (from $50 to $100) = 50 points

2 points for $20 (over $100) = 40 points

Total = 90 points

**2. Technologies Used**

Java 17

Spring Boot 3.x

Maven

REST API

Postman (for testing)

IDE (IntelliJ, Eclipse, etc.)

**3. Reward Points Calculation Logic**

The points calculation is done based on each transaction:

For amounts greater than $100:

Points = (Amount - 100) \* 2 + 50

For amounts between $50 and $100:

Points = (Amount - 50)

**4. Project Structure**

css

Copy code

RewardsApp

└── src

└── main

└── java

└── com.example.rewards

├── RewardsApplication.java

├── controller

│ └── RewardsController.java

├── model

│ └── Transaction.java

└── service

└── RewardsService.java

└── resources

└── application.properties

**5 Endpoints**

HTTP Method Endpoint Description

GET /customers/rewards Get reward points for all customers

GET /customers/{customerId}/transactions Get transactions for a specific customer

POST /customers/{customerId}/transactions Add a new transaction

PUT /customers/{customerId}/transactions Update an existing transaction

DELETE /customers/{customerId}/transactions/{month} Delete a transaction for a specific month

Calculates reward points for each customer per month based on their transactions.

Request Body (JSON):

json

Copy code

[

{"customerId": "C001", "month": "January", "amount": 120},

{"customerId": "C001", "month": "January", "amount": 75},

{"customerId": "C002", "month": "February", "amount": 200},

{"customerId": "C002", "month": "February", "amount": 50},

{"customerId": "C001", "month": "March", "amount": 110},

{"customerId": "C003", "month": "March", "amount": 90}

]

Response (JSON):

json

Copy code

{

"C001": {

"January": 90,

"March": 70

},

"C002": {

"February": 250

},

"C003": {

"March": 40

}

}

**6. Sample Data and Response**

Sample Transaction Data

Customer ID Month Amount ($)

C001 January 120

C001 January 75

C002 February 200

C002 February 50

C001 March 110

C003 March 90

Sample Response

json

Copy code

{

"C001": {

"January": 90,

"March": 70

},

"C002": {

"February": 250

},

"C003": {

"March": 40

}

}

**7. Testing the Application**

Test Cases

Scenario Input Amount ($) Expected Points

Below $50 40 0

Between $50 and $100 75 25

Above $100 120 90

Multiple transactions, same month 50, 200 300

Example Input:

json

Copy code

[

{"customerId": "C001", "month": "January", "amount": 40},

{"customerId": "C001", "month": "January", "amount": 75},

{"customerId": "C002", "month": "February", "amount": 120}

]

Expected Output:

json

Copy code

{

"C001": {

"January": 25

},

"C002": {

"February": 90

}

}