# Test Cases for Rewards Calculation

## Test Case: calculateRewardPoints

**Description: This test case checks the calculation of reward points based on transaction amounts.**

Test Cases:

1. Transaction amount is 0

- Input: 0

- Expected Output: 0

2. Transaction amount is 75

- Input: 75

- Expected Output: 25

3. Transaction amount is 50

- Input: 50

- Expected Output: 0

4. Transaction amount is 150

- Input: 150

- Expected Output: 150

5. Transaction amount is 100.2

- Input: 100.2

- Expected Output: 50

6. Transaction amount is 100.5

- Input: 100.5

- Expected Output: 51

## Test Case: calculateTotalReward

**Description: This test case checks the calculation of rewards for multiple transactions of a single customer.**

Test Cases:

1. Transactions:

- Input: [{ customerName: 'Alice', price: 60 }]

- Expected Output: {Alice: 10}

**Description: This test case checks the calculation of rewards for multiple customers.**

Test Cases:

1. Transactions:

- Input: [{ customerName: 'Alice', price: 60 }, { customerName: 'Bob', price: 150 }]

- Expected Output: { Alice: 10, Bob: 150 }

## Test Case: calculateMonthlyRewards

**Description: This test case checks the calculation of montly rewards for a single customer.**

Test Cases:

1. Transactions:

- Input: [{customerName: 'Alice', purchaseDate: '2024-06-15', price: 60, customerId: '1'}, { customerName: 'Alice', purchaseDate: '2024-06-20', price: 150, customerId: '1' }]

- Expected Output: {'Alice-6-2024': { customerName: 'Alice', customerId: '1', month: 6, year: 2024, rewardPoints: 160 }}

**Description: This test case checks the calculation of montly rewards for a muliple customers.**

Test Cases:

1. Transactions:

- Input: [{customerName: 'Alice', purchaseDate: '2024-06-15', price: 60, customerId: '1'}, { customerName: ‘Bob’, purchaseDate: '2024-06-20', price: 150, customerId: '2' }]

- Expected Output: {'Alice-6-2024': { customerName: 'Alice', customerId: '1', month: 6, year: 2024, rewardPoints: 10 }, 'Bob-6-2024': { customerName: 'Bob', customerId: '2', month: 6, year: 2024, rewardPoints: 150 }}

**Description: This test case checks the calculation of montly rewards for a muliple months.**

Test Cases:

1. Transactions:

- Input: [{customerName: 'Alice', purchaseDate: '2024-06-15', price: 60, customerId: '1'}, { customerName: ‘Alice’, purchaseDate: '2024-07-20', price: 150, customerId: '1' }]

- Expected Output: {'Alice-6-2024': { customerName: 'Alice', customerId: '1', month: 6, year: 2024, rewardPoints: 10 }, 'Alice-7-2024': { customerName: 'Alice', customerId: '1', month: 6, year: 2024, rewardPoints: 150 }}

**Test Cases for RewardsPointsCalculator**

This document describes the test cases for the RewardPointsCalculator using the Jest and React Testing Library. The RewardPointsCalculator fetches transaction data, displays the montly reaward, transaction table and total rewatds table. The test cases ensure the component's functionality and error handling.

# Test Case 1: Renders Rewards for Customers

**Description: Verifies that the component renders all child components and transaction data.**

\*\*Mocks Used\*\*:

1. `fetchTransactions`: Returns a mock list of transactions.

2. `calculateTotalRewards`: Returns a mock monhlty rewards.

3. `calculateMontlyRewards`: Returns a mock total rewards.

\*\*Steps\*\*:

1. Render the RewardPointsCalculator .

2. Wait for the `fetchTransactions` function to be called.

3. Check if the customer IDs and total points are rendered.

4. Check if the monthly points are rendered.

5. Check if transaction fetching get’s failed then render error state

\*\*Expected Results\*\*:

1. The component should render customer IDs and their total points.

2. The component should render the monthly points for each customer.

3. The component should render “Failed to load data”

# Test Case 2: Displays Error Message on API Failure

**Description: Verifies that the component displays an error message when the API call fails.**

\*\*Mocks Used\*\*:

1. `fetchTransactions`: Rejects with an error to simulate a failed API call.

\*\*Steps\*\*:

1. Mock `fetchTransactions` to reject with an error.

2. Render the RewardPointsCalculator Component.

3. Wait for the `fetchTransactions` function to be called.

4. Check if the error message is displayed.

\*\*Expected Results\*\*:

1. The component should display an error message: "Failed to load data."