TsetCases-Capstone1

Entity Insertion Order (for Testing and Data Initialization):

Order	Entity	Reason
1️⃣	**Category**	Products require a valid `categoryID`, so categories must be inserted first.
2️⃣	**Product**	Each product references a category, so categories must exist beforehand.
3️⃣	**Merchant**	MerchantStock needs a valid `merchantId`, so merchants come next.
4️⃣	**MerchantStock**	Depends on both `productId` and `merchantId`, which should already be present.
5️⃣	**User**	Required for purchase operations. Can be tested after the above entities exist.

Category → Product → Merchant → MerchantStock → User

TsetCases-Capstone1

Fake Product Data (JSON) :

```
[
  { "id": "p1", "name": "Wireless Mouse", "price": 89.99, "categoryID": "c1" },
  { "id": "p2", "name": "Bluetooth Headphones", "price": 149.50, "categoryID": "c1" },
  { "id": "p3", "name": "Smartphone", "price": 2999.00, "categoryID": "c1" },

  { "id": "p4", "name": "Java Programming Book", "price": 120.00, "categoryID": "c2" },
  { "id": "p5", "name": "Spring Boot Guide", "price": 95.00, "categoryID": "c2" },
  { "id": "p6", "name": "Data Structures Book", "price": 150.00, "categoryID": "c2" },

  { "id": "p7", "name": "Men T-Shirt", "price": 60.00, "categoryID": "c3" },
  { "id": "p8", "name": "Women Dress", "price": 180.00, "categoryID": "c3" },
  { "id": "p9", "name": "Sneakers", "price": 250.00, "categoryID": "c3" },

  { "id": "p10", "name": "Microwave Oven", "price": 450.00, "categoryID": "c4" },
```

TsetCases-Capstone1

{ "id": "p11", "name": "Air Conditioner", "price": 2200.00, "categoryID": "c4" },

{ "id": "p12", "name": "Refrigerator", "price": 3100.00, "categoryID": "c4" },

{ "id": "p13", "name": "Vitamin C Serum", "price": 80.00, "categoryID": "c5" },

{ "id": "p14", "name": "Hair Dryer", "price": 150.00, "categoryID": "c5" },

{ "id": "p15", "name": "Electric Toothbrush", "price": 200.00, "categoryID": "c5" },

{ "id": "p16", "name": "LEGO Set", "price": 350.00, "categoryID": "c6" },

{ "id": "p17", "name": "Puzzle 1000 Pieces", "price": 90.00, "categoryID": "c6" },

{ "id": "p18", "name": "RC Car", "price": 220.00, "categoryID": "c6" },

{ "id": "p19", "name": "Rice 5kg", "price": 70.00, "categoryID": "c7" },

{ "id": "p20", "name": "Olive Oil 1L", "price": 40.00, "categoryID": "c7" },

{ "id": "p21", "name": "Coffee Beans 500g", "price": 60.00, "categoryID": "c7" },

{ "id": "p22", "name": "Car Battery", "price": 450.00, "categoryID": "c8" },

{ "id": "p23", "name": "Car Wax", "price": 55.00, "categoryID": "c8" },

{ "id": "p24", "name": "Engine Oil", "price": 120.00, "categoryID": "c8" },

{ "id": "p25", "name": "Football", "price": 110.00, "categoryID": "c9" },

{ "id": "p26", "name": "Yoga Mat", "price": 85.00, "categoryID": "c9" },

{ "id": "p27", "name": "Tennis Racket", "price": 320.00, "categoryID": "c9" },

{ "id": "p28", "name": "Office Chair", "price": 600.00, "categoryID": "c10" },

{ "id": "p29", "name": "Desk Lamp", "price": 90.00, "categoryID": "c10" },

{ "id": "p30", "name": "Filing Cabinet", "price": 400.00, "categoryID": "c10" }

]

TsetCases-Capstone1

Fake Category Data (JSON) :

```json
[
 { "id": "c1", "name": "Electronics" },
 { "id": "c2", "name": "Books" },
 { "id": "c3", "name": "Fashion" },
 { "id": "c4", "name": "Home Appliances" },
 { "id": "c5", "name": "Health & Beauty" },
 { "id": "c6", "name": "Toys & Games" },
 { "id": "c7", "name": "Groceries" },
 { "id": "c8", "name": "Automotive" },
 { "id": "c9", "name": "Sports & Outdoors" },
 { "id": "c10", "name": "Office Supplies" }
]
```

TsetCases-Capstone1

Fake Merchant   Data (JSON) :

```
[
  { "id": "m1", "name": "TechZone },
  { "id": "m2", "name": "SmartMart },
  { "id": "m3", "name": "DailyNeeds}
]
```

Fake MerchantStock Data (JSON):

```
[
  { "id": "s1", "merchantId": "m1", "productId": "p1", "stock": 30 },
  { "id": "s2", "merchantId": "m1", "productId": "p2", "stock": 30 },
  { "id": "s3", "merchantId": "m1", "productId": "p3", "stock": 30 },
  { "id": "s4", "merchantId": "m1", "productId": "p4", "stock": 30 },
  { "id": "s5", "merchantId": "m1", "productId": "p5", "stock": 30 },
  { "id": "s6", "merchantId": "m1", "productId": "p6", "stock": 30 },

  { "id": "s7", "merchantId": "m2", "productId": "p1", "stock": 30 },
  { "id": "s8", "merchantId": "m2", "productId": "p2", "stock": 30 },
  { "id": "s9", "merchantId": "m2", "productId": "p3", "stock": 30 },
  { "id": "s10", "merchantId": "m2", "productId": "p4", "stock": 30 },
  { "id": "s11", "merchantId": "m2", "productId": "p5", "stock": 30 },
  { "id": "s12", "merchantId": "m2", "productId": "p6", "stock": 30 },

  { "id": "s13", "merchantId": "m3", "productId": "p1", "stock": 30 },
  { "id": "s14", "merchantId": "m3", "productId": "p2", "stock": 30 },
  { "id": "s15", "merchantId": "m3", "productId": "p3", "stock": 30 },
  { "id": "s16", "merchantId": "m3", "productId": "p4", "stock": 30 },
  { "id": "s17", "merchantId": "m3", "productId": "p5", "stock": 30 },
  { "id": "s18", "merchantId": "m3", "productId": "p6", "stock": 30 }
]
```

TsetCases-Capstone1

Fake User Data (JSON) :

```
[
 { "id": "u1", "username": "ahmed", "password": "pass123", "email": "ahmed@gmail.com", "role": "Customer", "balance": 500.0 },
 { "id": "u2", "username": "sara", "password": "sara456", "email": "sara@gmail.com", "role": "Customer", "balance": 350.0 },
 { "id": "u3", "username": "khalid", "password": "khalid789", "email": "khalid@gmail.com", "role": "Customer", "balance": 420.0 },
 { "id": "u4", "username": "mona", "password": "mona321", "email": "mona@gmail.com", "role": "Customer", "balance": 150.0 },
 { "id": "u5", "username": "faisal", "password": "faisal123", "email": "faisal@gmail.com", "role": "Customer", "balance": 800.0 },
 { "id": "u6", "username": "laila", "password": "laila321", "email": "laila@gmail.com", "role": "Customer", "balance": 275.0 },
 { "id": "u7", "username": "osama", "password": "osama111", "email": "osama@gmail.com", "role": "Customer", "balance": 600.0 },
 { "id": "u8", "username": "noura", "password": "noura999", "email": "noura@gmail.com", "role": "Admin", "balance": 1000.0 },
 { "id": "u9", "username": "tariq", "password": "tariq000", "email": "tariq@gmail.com", "role": "Customer", "balance": 90.0 },
 { "id": "u10", "username": "rana", "password": "rana555", "email": "rana@gmail.com", "role": "Admin", "balance": 700.0 }
]
```

TsetCases-Capstone1

First Endpoint: Get Available Merchants for Product

- **URL:** `GET /api/v1/merchant/stock/available-merchants/{productId}`
- **Method:** `GET`
- **Path Variable:**
    - `productId` *(String)* – ID of the product to search for.

📃 Description:

This endpoint retrieves a list of merchants who currently have stock available for a specific product. It allows customers or system users to see which merchants are ready to fulfill orders for that product.

💡 Use Case Scenario:

A user is browsing a product and wants to know from which merchants they can purchase it. Upon selecting a product, the system calls this endpoint and displays the merchants who currently have the item in stock. If no stock is available, the user is informed accordingly.

TsetCases-Capstone1

🔄 Success Response (200 OK):

Returns a list of merchants (as JSON) who have the product in stock.

```
 [
  {
    "id": "m1",
    "name": "TechZone"
  },
  {
    "id": "m4",
    "name": "SmartMart"
  }
]
```

⚠ Error Response (404 NOT FOUND):

```
{
  "message": "No available merchants found for the given product ID"
}
```

Second Endpoint: Get Purchase History

- **URL:** `GET /api/v1/user/purchase-history`
- **Method:** `GET`
- **Path Variable:** *None*

📃 Description:
This endpoint retrieves a full list of all successful purchase transactions made in the system. It is useful for administrators or users who want to audit or review which products were purchased, by whom, and from which merchants.

TsetCases-Capstone1

💡 Use Case Scenario:

An admin wants to review all customer purchases made through the system to analyze buying trends, verify user transactions, or debug issues. The frontend calls this endpoint and displays the history of all purchases including the user ID, product ID, and merchant ID.

🔁 Success Response (200 OK):

Returns a list of string messages describing each purchase in a readable format.

```
 [
  "User [u1] bought product [p1] from merchant [m1]",
  "User [u3] bought product [p2] from merchant [m2]"
]
```

⚠ Error Response (200 OK with empty list):

If no purchases have been made yet, the response will be an empty list:

```
 []
```

Third Endpoint: Get Products Below Stock Threshold

- **URL:** `GET /api/v1/merchant/stock/low-stock/{threshold}`
- **Method:** GET
- **Path Variable:**
  - `threshold` (Integer) – the stock value under which a product is considered low in stock.

📃 Description:

This endpoint retrieves a list of products that have **stock below a given threshold** across all merchants. It allows admins or store managers to identify which products are running low and need to be restocked.

TsetCases-Capstone1

Use Case Scenario:

An inventory manager wants to generate a report of products that are **almost out of stock**. The system calls this endpoint with a specified threshold (e.g., 10), and it returns all products that have stock less than 10. This helps prevent stockouts and ensures timely replenishment.

Success Response (200 OK):

Returns a JSON array of `Product` objects that have stock below the specified threshold.

```
  [
   {
     "id": "p5",
     "name": "USB-C Charger",
     "price": 49.99,
     "categoryID": "c3"
   },
   {
     "id": "p7",
     "name": "HD Webcam",
     "price": 220.0,
     "categoryID": "c4"
   }
]
```

⚠ Error Response (404 NOT FOUND):

```
{
  "message": "No products found below the given stock threshold"
}
```

TsetCases-Capstone1

Fourth Endpoint: Get Product Sales Summary

- **URL:** `GET /api/v1/user/sales-summary`
- **Method:** `GET`
- **Path Variable:** *None*

🗏 Description:

This endpoint provides a summary report of all sold products, including the total number of units sold per product and the total revenue generated from those sales. It helps merchants or administrators monitor product performance and financial outcomes over time.

💡 Use Case Scenario:

An admin or system operator wants to evaluate which products are selling the most and how much revenue they generate. By calling this endpoint, the system retrieves a list of products that have been sold, the quantity of each, and the total earnings from those sales.

For example, if a product is sold 5 times at 100 SAR per unit, the endpoint will return that the product generated 500 SAR in revenue.

TsetCases-Capstone1

🔁 Success Response (200 OK):

Returns a list of product sales summary in the following format:

```
 [
  "Product [p1] - Sold: 3 times - Revenue: 600.0 SAR",
  "Product [p2] - Sold: 1 times - Revenue: 200.0 SAR"
]
```

⚠ Notes:

- If no products have been sold, the list will be empty.
- The system does not persist the sales data to a database. It’s stored in memory using a 2D array (`String[][]`), so restarting the application resets the sales record.

Fifth Endpoint: Get Products That Match User's Budget

◆ **URL:** `GET /api/v1/user/products-within-budget/{userId}`
◆ **Method:** `GET`
◆ **Path Variable:**

Name	Type	Description
userId	String	The ID of the user making the query

📃 Description

This endpoint retrieves all products that are priced **within the balance** of the specified user. It is useful for helping users explore only the products they can afford without going over budget.

TsetCases-Capstone1

💡 Use Case Scenario

A user wants to browse products they can afford based on their current wallet balance. When the user logs in, the system uses their `userId` to fetch a list of matching products. This improves the shopping experience and avoids frustration due to unaffordable items.

✅ Success Response: `200 OK`

Returns an array of products priced within the user's available balance.

Example:

```
json
CopyEdit
[
  {
    "id": "p1",
    "name": "Wireless Mouse",
    "price": 70.0,
    "categoryID": "c2"
  },
  {
    "id": "p3",
    "name": "USB-C Cable",
    "price": 35.0,
    "categoryID": "c1"
  }
]
```

⚠ Error Response: `404 NOT FOUND`

If the user does not exist or has insufficient balance for any available product:

```
json
CopyEdit
{
  "message": "User not found or no products within budget"
}
```

TsetCases-Capstone1