Capstone 1

for

Web Application Development by JAVA

Add Review Endpoint

"Allowing users to provide feedback on products."

What does the endpoint do?

Enables users to leave reviews for products they've interacted with.

A review consists of:

Product ID: The product being reviewed.

Rating: A score from 1 to 5.

Comment: User feedback or thoughts about the product.

Product Service class

1-Rating Validation

2-Product Validation

3-Review Creation

```
52    //Extra endpoint 1 User can add review on the product
53        public String addReview(String productId, int rating, String comment) {  1 usage
54            if (rating < 1 || rating > 5) {
55                return "invalid rating";
56            }
57            for (Product p : products) {
58                if (p.getId().equals(productId)) {
59                    String review = String.format("Product: %s | Rating: %d | Comment: %s", productId, rating, comment);
60                    reviews.add(review);
61                    return "true";
62
63                }
64            }return "product id not found";
65        }
66
```

Calculate Average Rating Endpoint

"Determining the overall customer satisfaction for products."

What does the endpoint do?

- Computes the average rating of a specific product based on user reviews.

To Calculate Average Rating Need:

Product ID: The product being reviewed.

Product Service class

1- Initialize Variables

2-Iterate Through Reviews

3- Extract Ratings

4-Calculate Total and Count

```java
//Extra endpoint 2 Calculate Average rating on tha product
public double CalculateAvgRating(String productId){  2 usages
    int totalRating=0;
    int count=0;

    for(String review: reviews){
        if(review.contains("Product: "+productId)){
            String [] review_parts =review.split( regex: "\\|");
            int rating =Integer.parseInt(review_parts[1].trim().split( regex: ": ")[1]);
            totalRating=totalRating+rating;
            count++;
        }
    }
    if(count==0.0){
        return 0.0;
    }
    return (double) totalRating/count;
```

Admin Apply Discount on Product

"Enabling admins to adjust product prices with discounts."

What does the endpoint do?

- Allows an admin to apply a percentage-based discount on a product.
- Calculates the new price after applying the discount.
- Ensures only authorized admins can perform this action.

To perform the endpoint need:

Admin ID

Product ID

Percent

User Service class

1- Check if the user exists .

2- Verify the user's role is Admin

3-Locate the product using product ID

4-Calculate the new price

5-Return the updated price

```
111    //Extra endpoint 3 Admin can Apply discount on product
112    public String ApplyDiscount(String adminId,String productId,double percent){  2 usages
113        for(User user: users){
114            if(user.getId().equals(adminId)){
115                if(user.getRole().equals("Admin")){
116                    for(Product p: productService.getProducts()){
117                        if(p.getId().equals(productId)){
118                            double newPrice = p.getPrice()*(1-percent/100);
119                            return "Discount applied successfully, new Price: "+newPrice;
120                        }
121                    }return "C product not found";
122                }return "A not admin";
123            }
124        }return "B User not found";
125
```

Reset Password Endpoint

"Enabling users to securely reset their account password."

What does this endpoint do?

- Allows users to reset their account passwords securely.
- Ensures the old password is verified before setting a new one.
- Validates the new password to meet predefined security standards.

To perform the endpoint need:

User ID

Old Password

New Password

User Service class

1-Verify that the user ID exists in the system.

2-Confirm the old password matches the user's current password.

3-Validate the new password using a helper function (isValidPassword).

4-Update the password if all checks pass.

```
//Extra Endpoint 4 User can reset password
public String resetPassword(String userId, String oldPass ,String newPass){  1 usage
        for(User user : users){
            if(user.getId().equals(userId)){
                if(user.getPassword().equals(oldPass)){
                    if(isValidPassword(newPass)){
                    user.setPassword(newPass);
                    return "G good";}
                    else return "C New password is not valid";
                }return "B User old password doesn't matches the user's current password.";
            }
        }return "A user not found";
}
```

Return Product to Stock Endpoint

"Enabling users to return purchased products and manage refunds."

/10

What does this endpoint do?

- Allows users to return products they purchased.
- Ensures that the returned product meets specific conditions (e.g., refundable status).
- Restocks the product to the merchant inventory.
- Refunds the purchase amount to the user's account.

To perform the endpoint need:

User ID

Product ID

Merchant Stock ID

User Service class


```
//Extra endpoint 5 user can return product to stock
public String returnProduct(String userId,String productId,String merchantStockId){  1 usage
for(User user:users){
    if(user.getId().equals(userId)){
        for(Product product: productService.getProducts()){
            if(product.getId().equals(productId)){
                if(product.getStatus().equals("Refundable")&&product.isSold()) {
                    merchantStockService.addToStock(productId, merchantStockId,  amount: 1);
                    user.setBalance(user.getBalance() + product.getPrice());
                    return"Good";
                }else return "C Product status does not allow for return.";
            }
        }return "B product not found";
    }
}return "A user not found";
}
```

1-Verify that the user exists in the system.
2-Check that the product is valid and was purchased by the user.
3-Ensure the product's status is Refundable and marked as sold.
4-Restock the product by incrementing its stock in the merchant inventory.
5-Refund the product's price to the user's balance.
6-Return appropriate feedback for success or failure.

Thank You