

## Report 6

Team members:

Dias Galikhanov 200103092

Yermukhamed Tilek 200103094

Baijuman Ali 200103514

Daurbek Sanakuly 200103453

Zhumadil Kenzhebay 200103245

For this week we have done:

1. Added new 'Reviews' table to our database.
2. Filtering products by price and rating.
3. List reviews of one product.

We added new table “Reviews” which contains user\_id, product\_id, comment, rating fields. And we added new avg\_rating field to our products table. We create new table with constraint for unique combination of product and user, which means that one user can only write one review for one product. But if he wants he can change it.

Details for filtering products by price and rating will be taken from query parameters.

startPrice - for starting price.

endPrice - for end price.

startRating - for starting rating.

endRating - for end rating.

Here how we take and store them. And if they are not given there will be empty string.

```
func ListProducts(c *fiber.Ctx) error {
    queries := make(map[string]string, 5)
    queries["title"] = c.Query("name")
    queries["startPrice"] = c.Query("StartPrice")
    queries["endPrice"] = c.Query("EndPrice")
    queries["startRating"] = c.Query("StartRating")
    queries["endRating"] = c.Query("EndRating")
}
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12

And here how we check and fix them

```
if queries["startPrice"] == "" {
    queries["startPrice"] = "0"
}
if queries["endPrice"] == "" {
    queries["endPrice"] = "10000000"
}

if queries["startRating"] == "" {
    queries["startRating"] = "0"
}

if queries["endRating"] == "" {
    queries["endRating"] = "5"
}

query := `SELECT p.product_id, p.product_title,
                u.user_id, u.username, u.email,
                c.category_id, c.category_title,
                p.price, p.description, p.avg_rating
            FROM products p
            JOIN users u ON p.seller_id = u.user_id
            JOIN categories c ON p.category_id = c.category_id
            WHERE LOWER(p.product_title) LIKE $1
            AND p.price >= $2 AND p.price <= $3
            AND p.avg_rating >= $4
            AND p.avg_rating <= $5`
```

Example:

<http://127.0.0.1:8080/api/v1/product?startPrice=14&endPrice=20&startRating=2&endRating=4>

List Reviews:

<http://127.0.0.1:8080/api/v1/product/1/review>

For the next week:

We will make all create, detail, delete, update methods for reviews. And we will work on how to make orders.