Team: NDY corp.

Midterm 2 Report **Date: April 16, 2023**

Amangeldin Erasyl	**Amanzhol Nursultan**	**Shulanbay Darkhan**
id:200103252	**id:200103444**	**id:200103038**

4) Filtering items based on price, rating
5) Giving rating for items
6) Commenting items

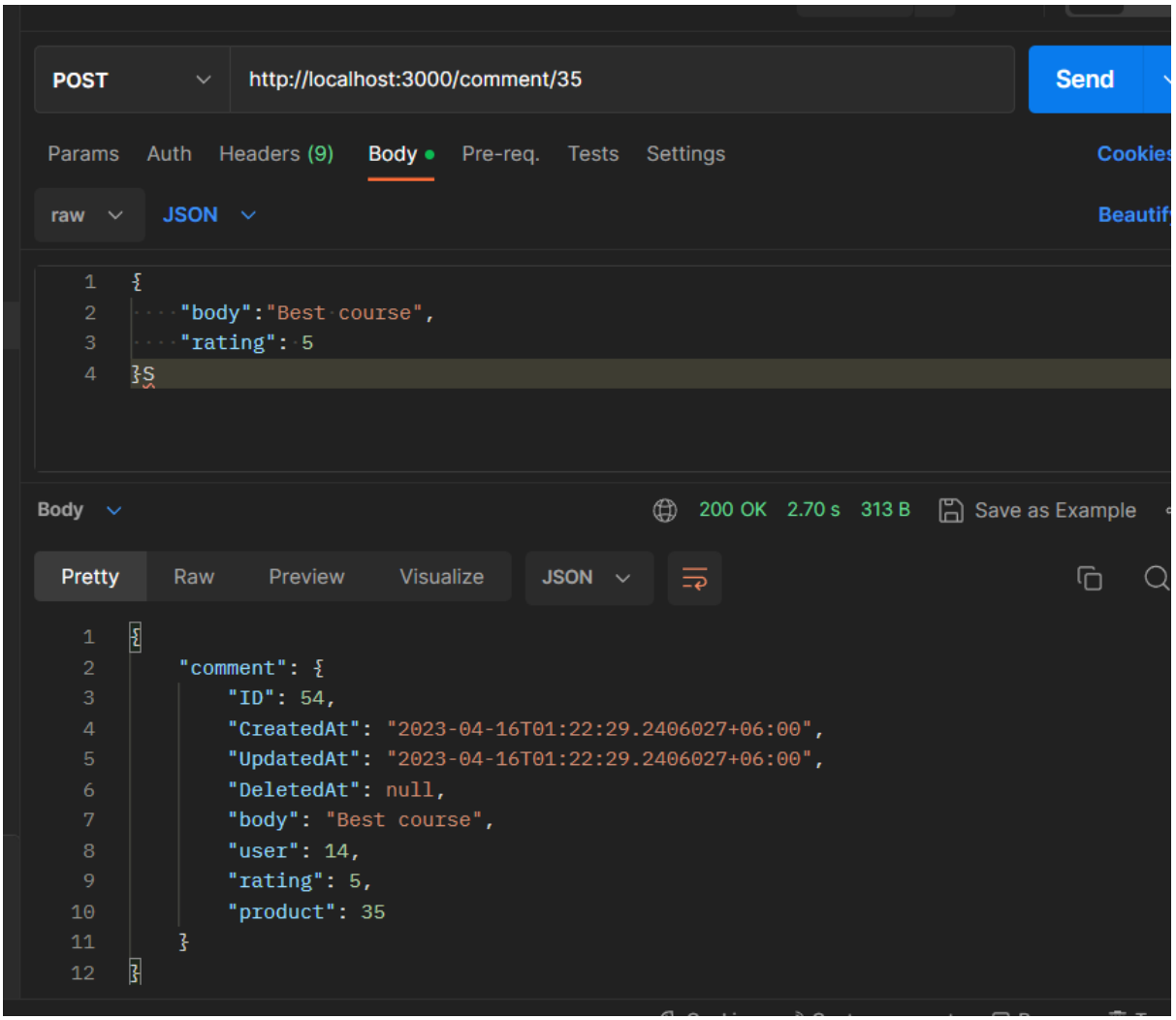
Link to the video~ https://youtu.be/7ekyqqGmjuc

Here is our filter based on ***inputTitle*** and ***inputSort***.

```go
func ListProducts(c *gin.Context) {
    qs := c.Request.URL.Query()
    inputTitle := qs.Get("title")
    inputSort := qs.Get("sort")
    var products []models.Product
    if inputTitle != "" && inputSort != "" {
        if inputSort[0] == '-' {
            _ = initializers.DB.Order(fmt.Sprint(inputSort[1:])).Find(&products, "title = ?", inputTitle)
        } else {
            _ = initializers.DB.Order(fmt.Sprint(inputSort, " desc")).Find(&products, "title = ?", inputTitle)
        }
    } else if inputTitle != "" {
        _ = initializers.DB.Find(&products, "title = ?", inputTitle)
    } else if inputSort != "" {
        if inputSort[0] == '-' {
            _ = initializers.DB.Order(fmt.Sprint(inputSort[1:])).Find(&products)
        } else {
            _ = initializers.DB.Order(fmt.Sprint(inputSort, " desc")).Find(&products)
        }
    } else {
        _ = initializers.DB.Find(&products)
    }

    c.JSON(http.StatusOK, gin.H{"products": products})
}
    Darkha003, 3 weeks ago • Add Search by title
```

Here is giving *Rating and Commenting* to our *Products*.



Here is shown how our rating is calculating.

So it takes sum of rating, for example max Value which user can give is 5. So let's say like:

1st User give 5 rating.

2nd User give 1 rating.

3rd User give 4 rating.

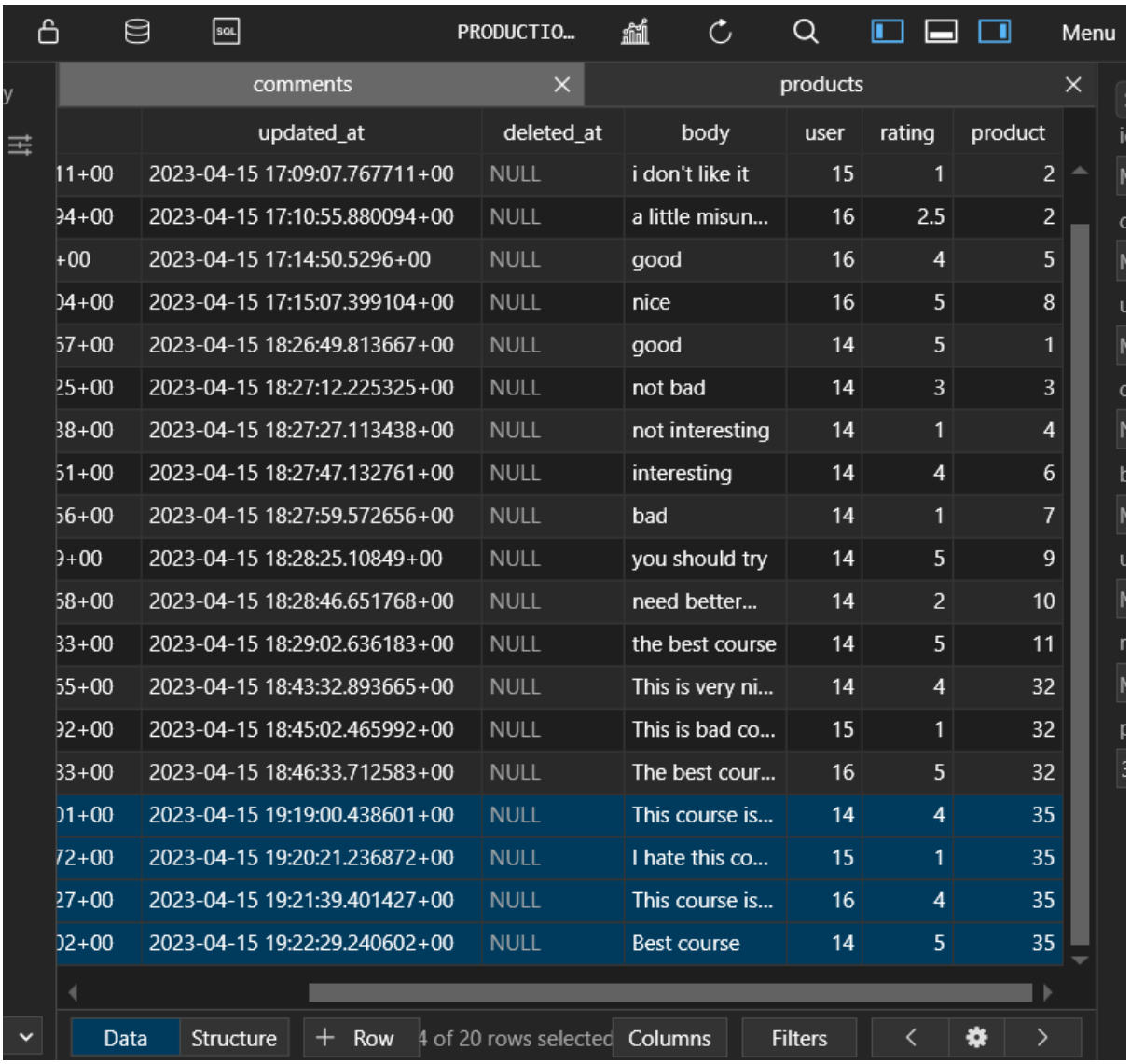
So it calculate like **5+1+4 / 3** divided my count of rating. So the result will be **3.3333333.**

```
        return
    }

    initializers.DB.Table("comments").Where("product = ?", id).Select("sum(rating) as sm").Scan(&sum)
    initializers.DB.Table("comments").Where("product = ?", id).Select("count(*) as cnt").Scan(&cnt)
    avg := sum / cnt
    initializers.DB.Model(&product).Update("rating", avg)

    c.JSON(http.StatusOK, gin.H{"comment": comment})
}
```

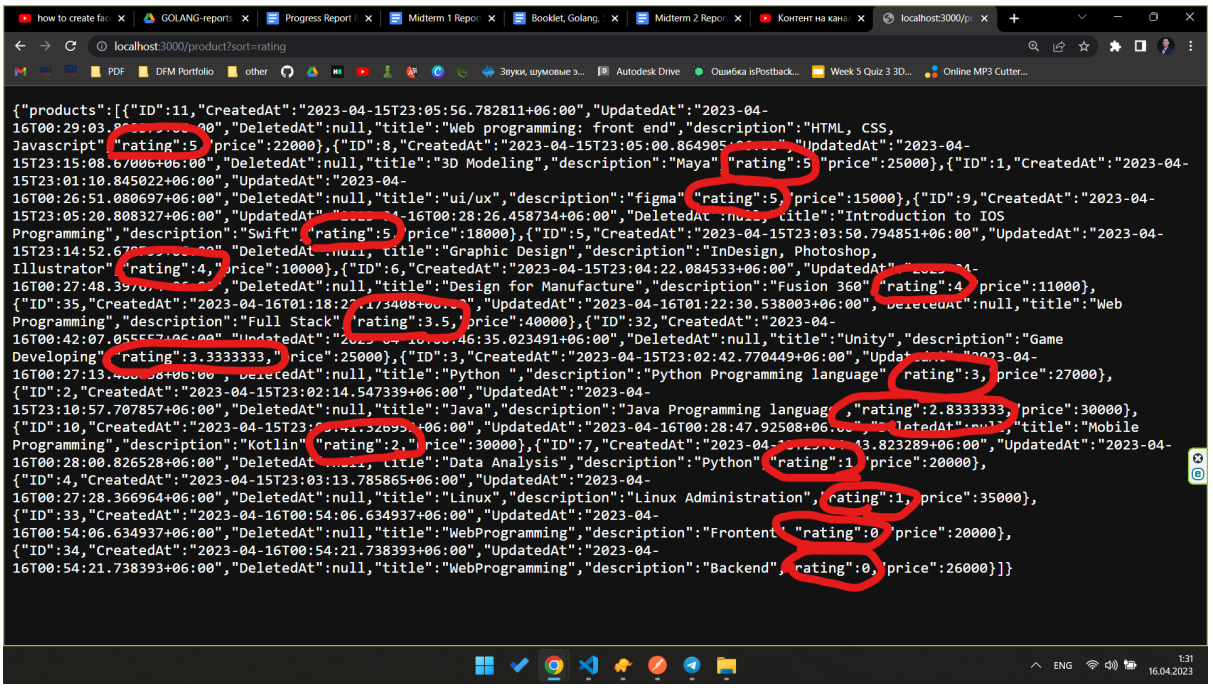
Here our comments.



	updated_at	deleted_at	body	user	rating	product
11+00	2023-04-15 17:09:07.767711+00	NULL	i don't like it	15	1	2
94+00	2023-04-15 17:10:55.880094+00	NULL	a little misun...	16	2.5	2
+00	2023-04-15 17:14:50.5296+00	NULL	good	16	4	5
04+00	2023-04-15 17:15:07.399104+00	NULL	nice	16	5	8
57+00	2023-04-15 18:26:49.813667+00	NULL	good	14	5	1
25+00	2023-04-15 18:27:12.225325+00	NULL	not bad	14	3	3
38+00	2023-04-15 18:27:27.113438+00	NULL	not interesting	14	1	4
51+00	2023-04-15 18:27:47.132761+00	NULL	interesting	14	4	6
56+00	2023-04-15 18:27:59.572656+00	NULL	bad	14	1	7
9+00	2023-04-15 18:28:25.10849+00	NULL	you should try	14	5	9
58+00	2023-04-15 18:28:46.651768+00	NULL	need better...	14	2	10
33+00	2023-04-15 18:29:02.636183+00	NULL	the best course	14	5	11
55+00	2023-04-15 18:43:32.893665+00	NULL	This is very ni...	14	4	32
92+00	2023-04-15 18:45:02.465992+00	NULL	This is bad co...	15	1	32
33+00	2023-04-15 18:46:33.712583+00	NULL	The best cour...	16	5	32
01+00	2023-04-15 19:19:00.438601+00	NULL	This course is...	14	4	35
72+00	2023-04-15 19:20:21.236872+00	NULL	I hate this co...	15	1	35
27+00	2023-04-15 19:21:39.401427+00	NULL	This course is...	16	4	35
02+00	2023-04-15 19:22:29.240602+00	NULL	Best course	14	5	35

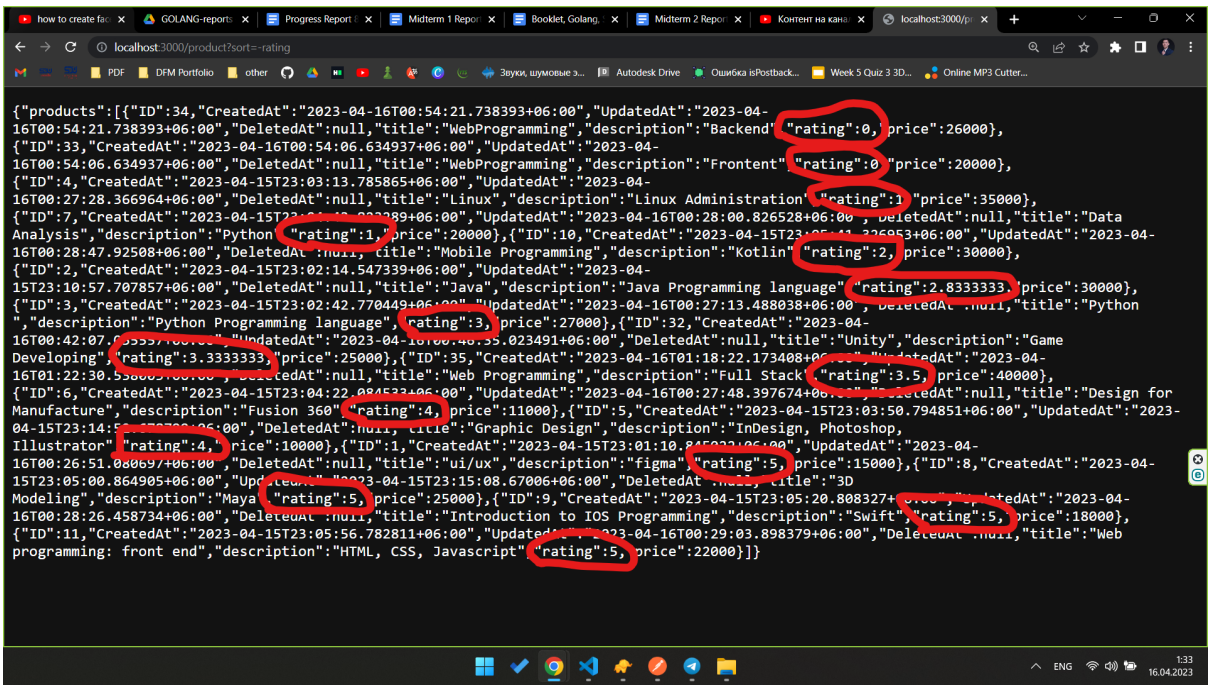
1.http://localhost:3000/product?sort=rating - Sorting products by rating from the largest to the smallest.

We have 5 rating types. As you can see it is sorted from *Largest 5* **to** *Smallest 1*.



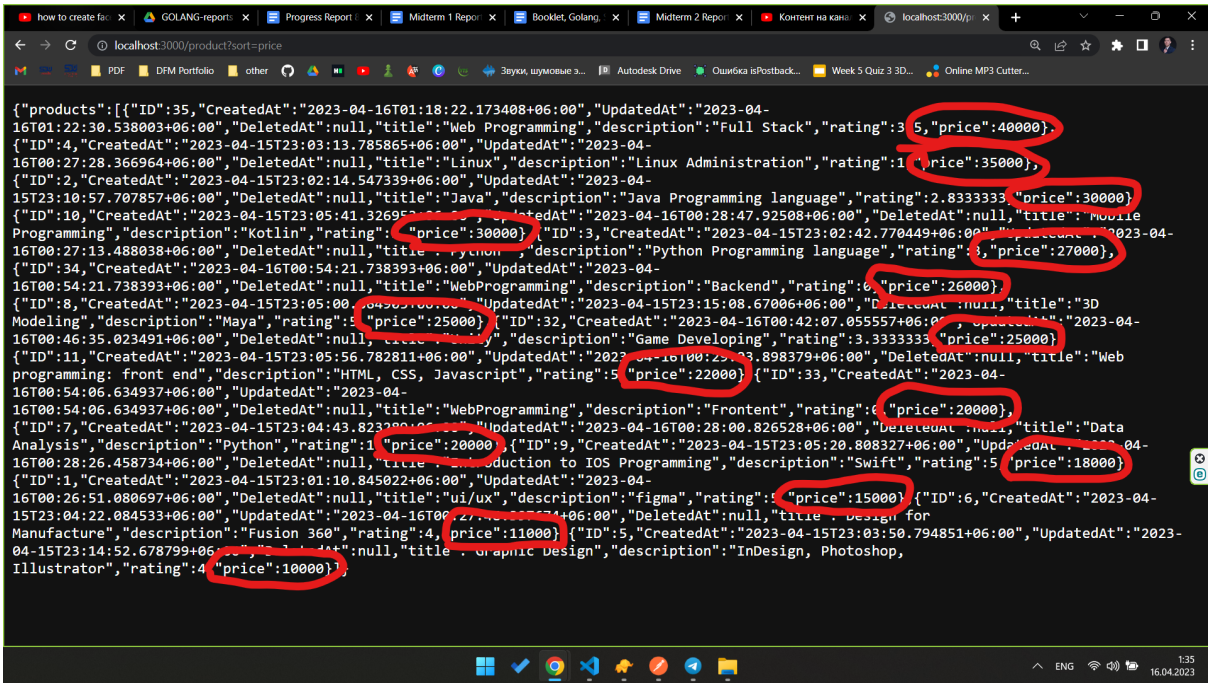
http://localhost:3000/product?sort=-rating - Sorting products by rating from the smallest to the largest.

it is sorted from *Smallest 1* **to** *Largest 5*.



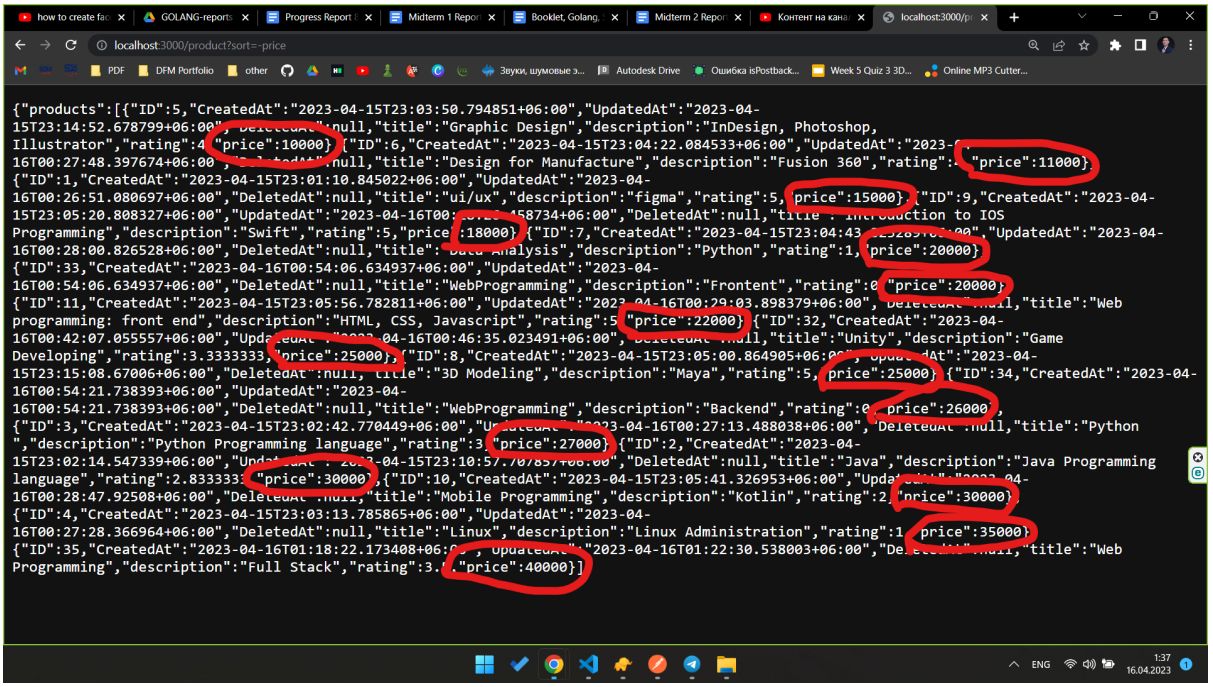
http://localhost:3000/product?sort=price - Sorting of products by price from the largest to the smallest.

We have our price. It is sorted from *Largest 40000* **to** *Smallest 10000*.



http://localhost:3000/product?sort=-price - Sorting of products by price from the smallest to the largest.

We have our price. It is sorted from *Smallest 10000* **to** *Largest 40000*.

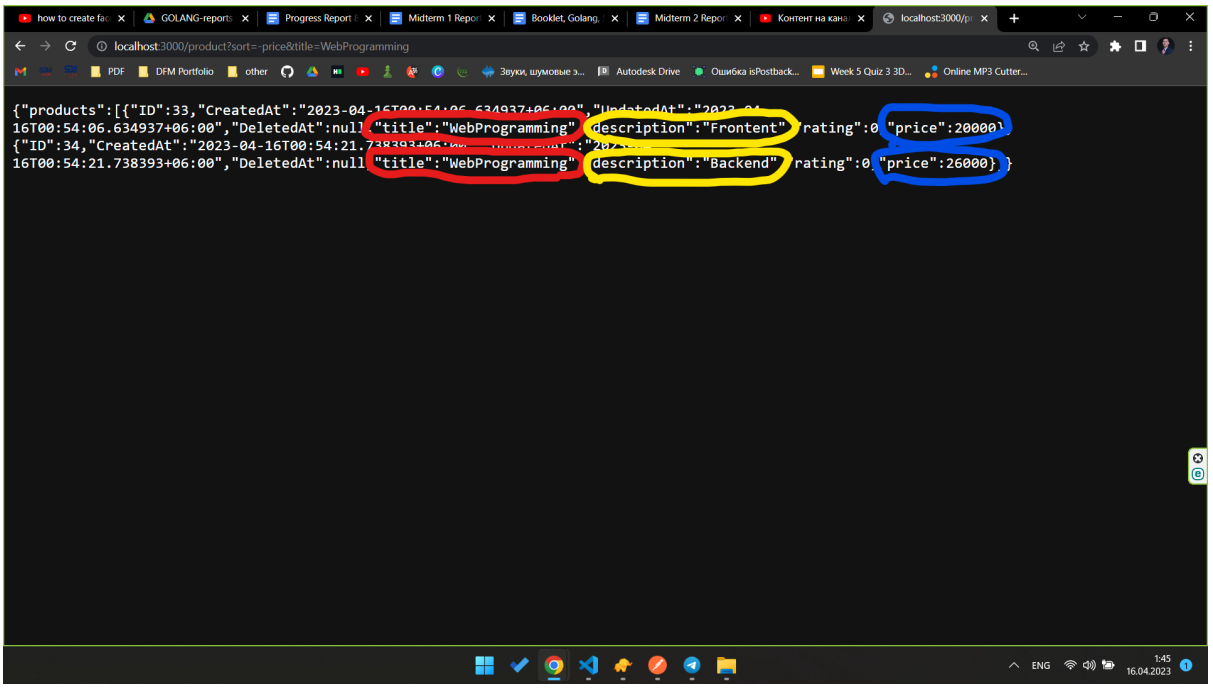


http://localhost:3000/product?sort=-price&title=WebProgramming - Sorting products and searching at the same time.

Here we have two similar title: “WebProgramming”. And this is by title and

price from smallest to largest

Here it is *“price”:20000* **to** *“price”:26000.*



INF 368	The Go programming language	**Teacher:** *Azamat Serek*	**Project type:** *Midterm 1*