Team: NDY corp.

Progress Report 8 **Date: April 09, 2023**

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

In this progress report we added:

4) Filtering items based on price, rating

*Here in our product model, we've added our **price.***

```
type Product struct {
    gorm.Model
    Title       string  `json:"title"`
    Description string  `json:"description"`
    Rating      float32 `json:"rating"`
    Price       int     `json:"price"`
}
```

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 we just added *Price.*

```go
    }

    func CreateProduct(c *gin.Context) {
        var body struct {
            Title       string `json:"title,omitempty"`
            Description string `json:"description,omitempty"`
            Price       int    `json:"price,omitempty"`
        }
```

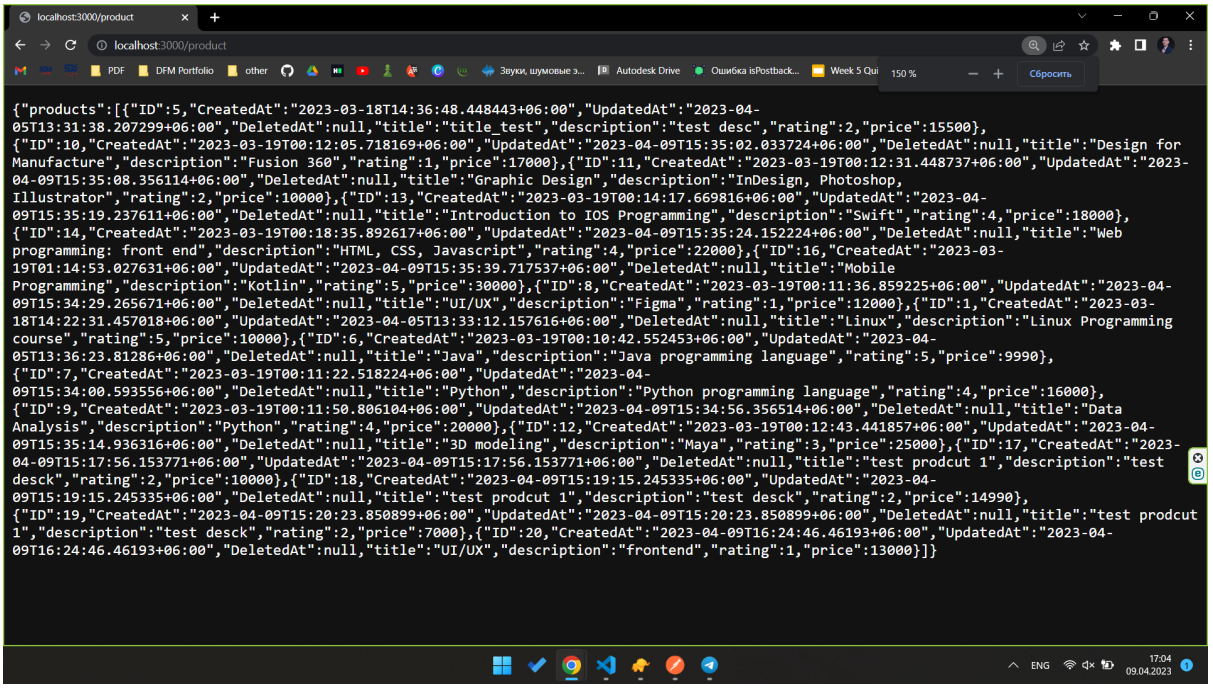
We created a product.

After that we enter our data into the database.

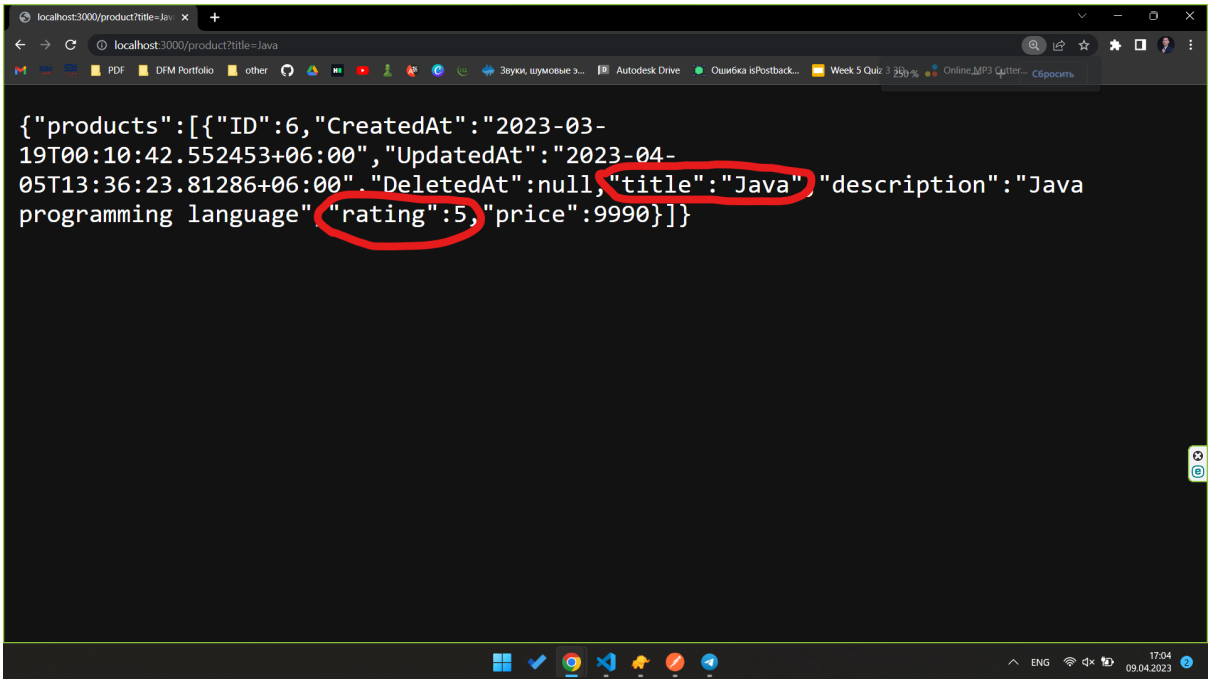
```go
    product := models.Product{Title: body.Title, Description: body.Description, Price: body.Price}
    result := initializers.DB.Create(&product)

    if result.Error != nil {
        c.JSON(http.StatusBadRequest, gin.H{
            "error": "Failed to create product",
        })

        return
    }

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

So our filter works based on this urls:

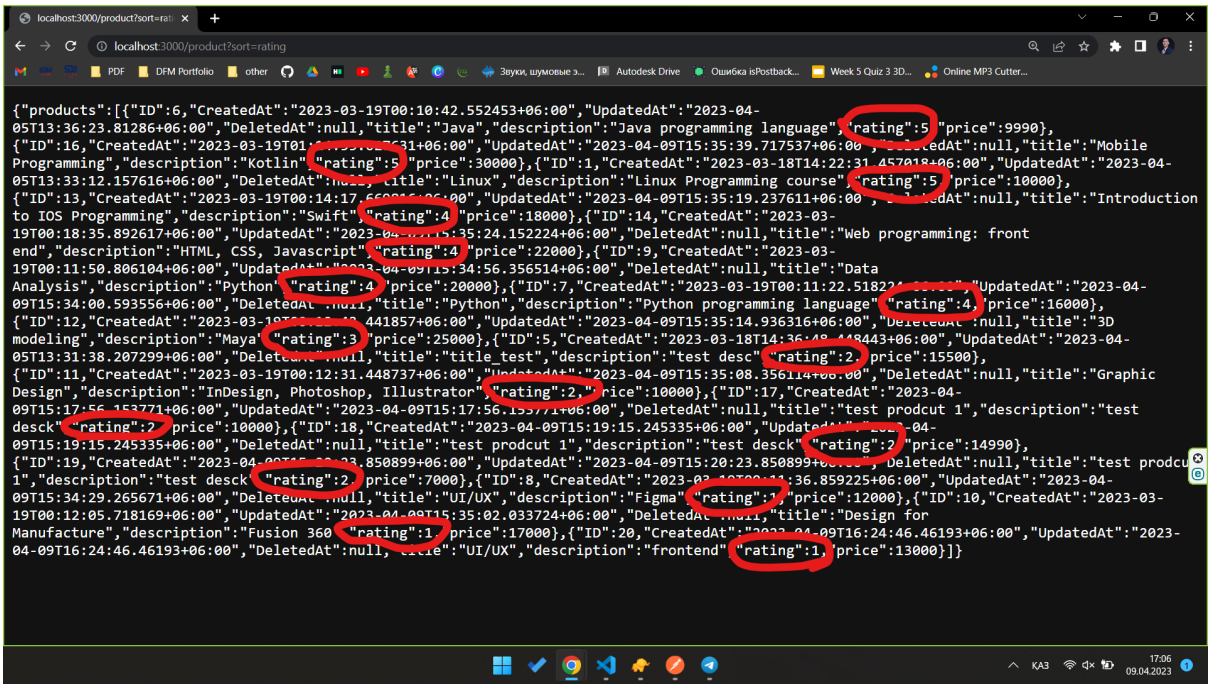
1) http://localhost:3000/product - list of all products.



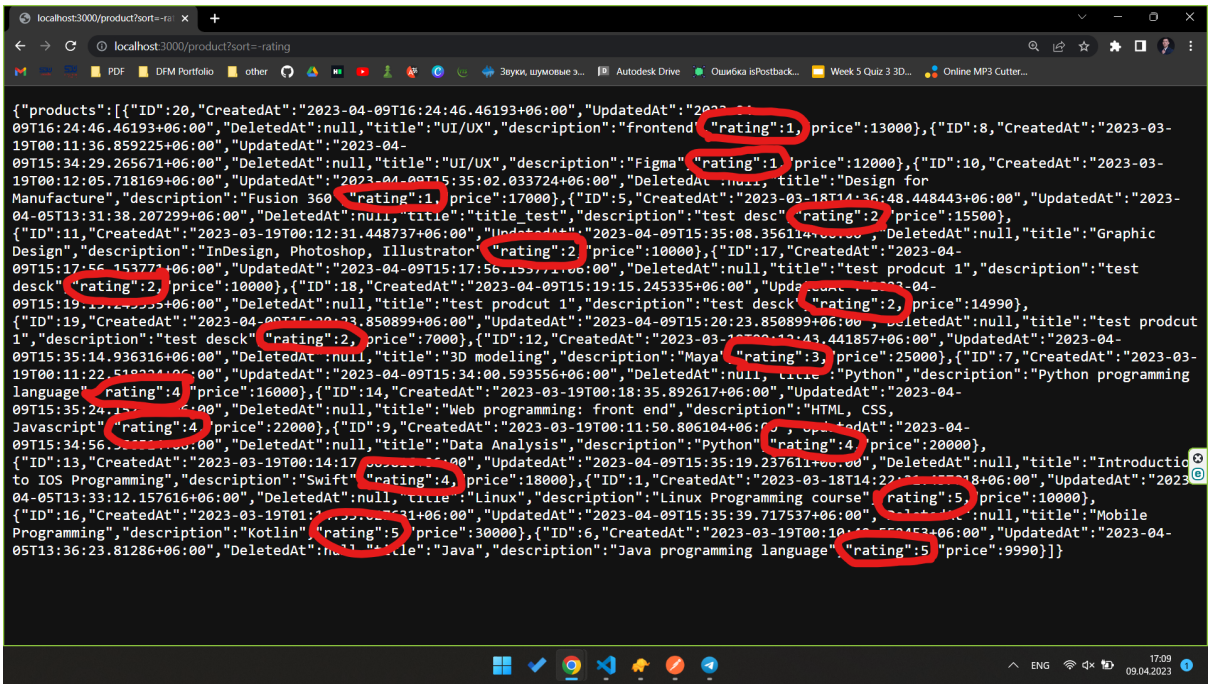
2) http://localhost:3000/product?title=Java - based on title (search)



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*.

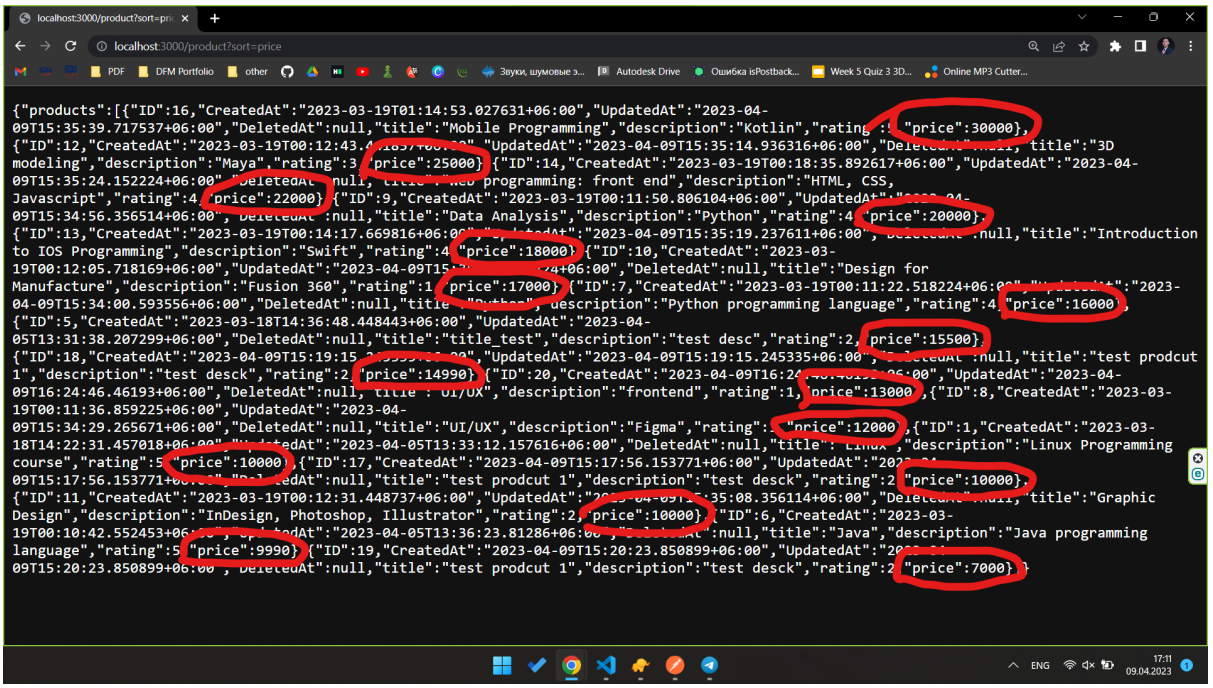


http://localhost:3000/product?sort=-rating - Sorting products by rating from the smallest to the largest.
it is sorted from *Smallest 1* to *Largest 5*.



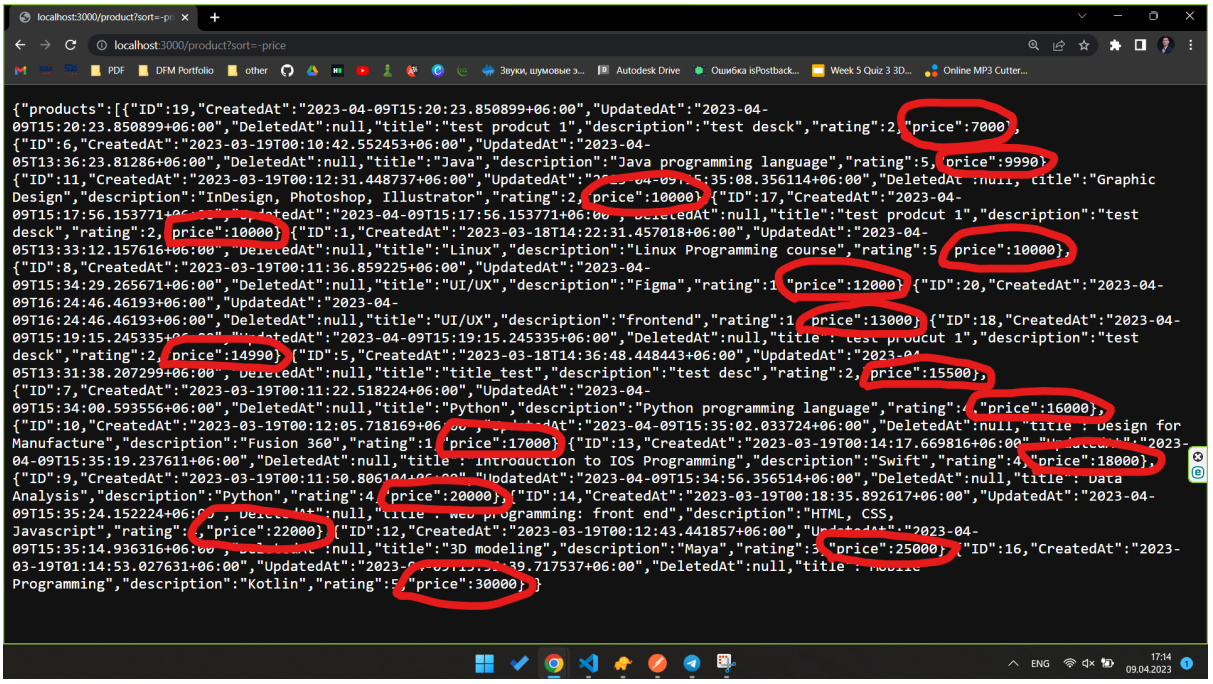
http://localhost:3000/product?sort=price - Sorting of products by price from the largest to the smallest.

We have our price. It is sorted from *Largest 30000* **to** *Smallest 7000*.

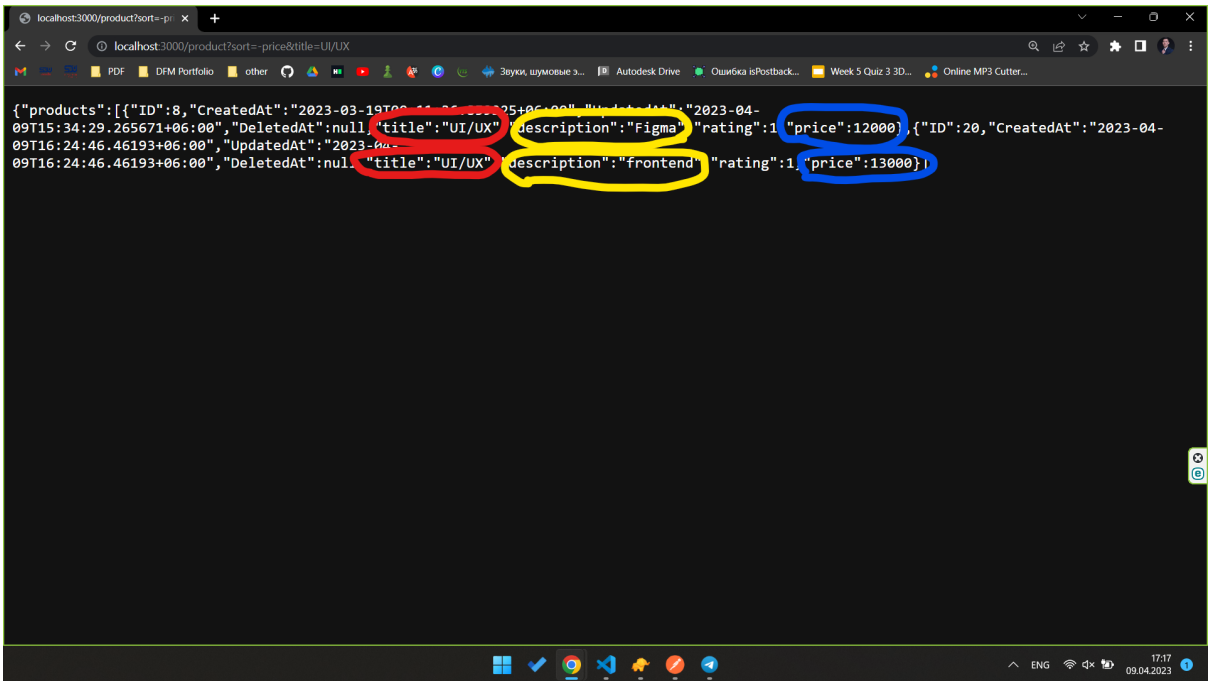


http://localhost:3000/product?sort=-price - Sorting of products by price from the smallest to the largest.

We have our price. It is sorted from *Smallest 7000* **to** *Largest 30000*.



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

Here we have two similar title: “UI/UX”. And this is by title and *price from smallest to largest*

Here it is *“price”:12000* **to** *“price”:13000.*