

Midterm Report 2

Hello and welcome to the report 2 of our online shop project that is named All-in-One. Our website is about the online shop that includes buying products. There is a website that has main page, filter on main page that filters products, review page, adding item page, authorisation and registration page. Also, we improved and updated the design of all pages.

Members:

Tlepbergen Abdussamad 200103400: Responsible for creating a new table in the database. Made comments and ratings. Design of the review page

Moldabayev Almat 200103324: I created filtration function based on price redesign some elements in HTML. Let me describe how filtration works, you choose the range of the price you want to filter (min, max) and it shows you the items between this range.

Tabyldiyev Rauan 200103412: Responsible for fixing errors on the project such as: commenting and rating, giving right queries on database tables. Also, responsible for managing the reports.

Bulatov Maxat 200103033: Responsible for working with HTML and CSS, created a carousel of items with the most popular products, created a search engine to make it easier for users to use the site. I wrote the code for the footer, in which I placed all the available

information about us, working hours, contacts, as well as social networks, so that users could contact us.

How to run code: First you have to run MAMP, for the database. Create in phpMyAdmin a structure called "golang". Inside this structure create two tables: "article" and "login". Inside "article" add such rows as "id", "title", "anons", "price". Inside "login" add such rows as "id", "name", "email", "password". Then open our code and run main.go with the command "go run main.go". Then through the browser we open "localhost:8080"

First picture that we see on the website is the searching window, it is very useful for searching products that you need.



Here is the code of the Searching Function.

```

func searchHandler(w http.ResponseWriter, r *http.Request) {
    query := r.FormValue("query")
    if query == "" {
        http.Redirect(w, r, "/", http.StatusSeeOther)
    } else {
        t, err := template.ParseFiles("templates/index2.html", "templates/header.html", "templates/footer.html")

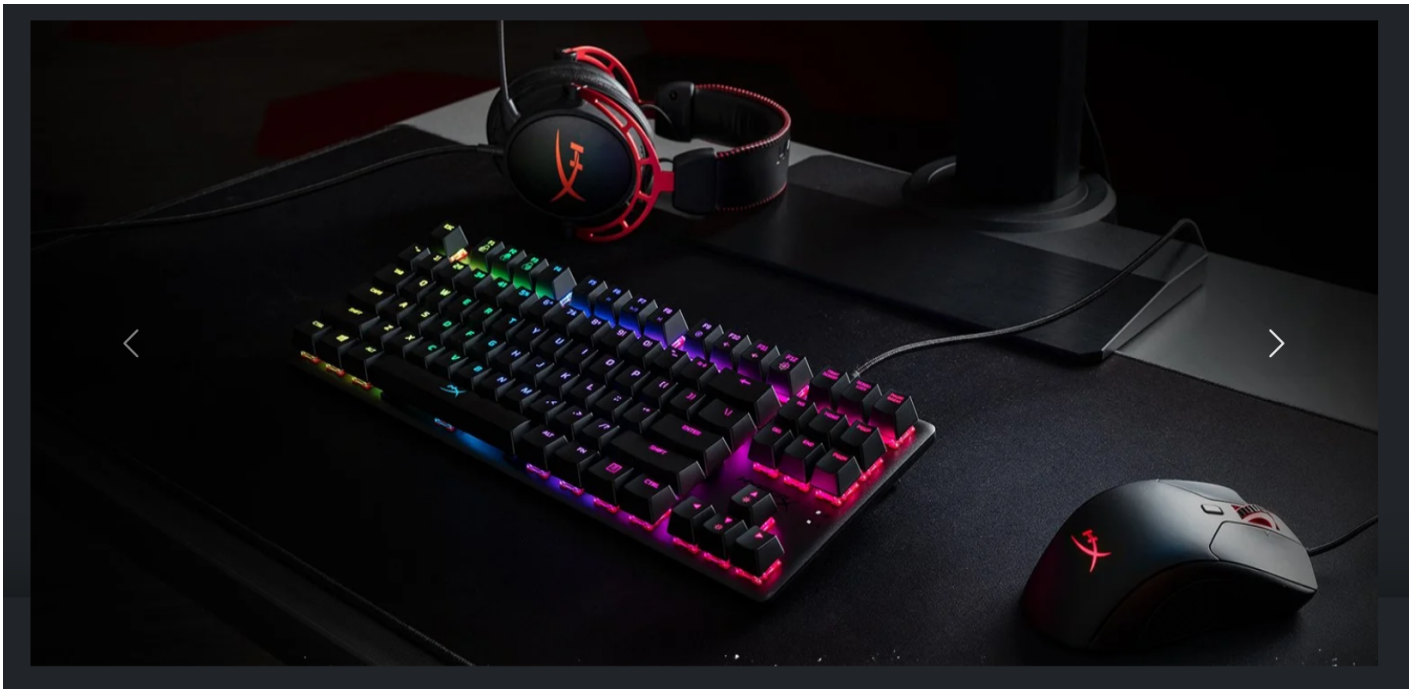
        if err != nil {
            fmt.Fprint(w, err.Error())
        }

        db, err := sql.Open("mysql", "root:root@tcp(127.0.0.1:3306)/golang")
        if err != nil {
            panic(err)
        }
        defer db.Close()

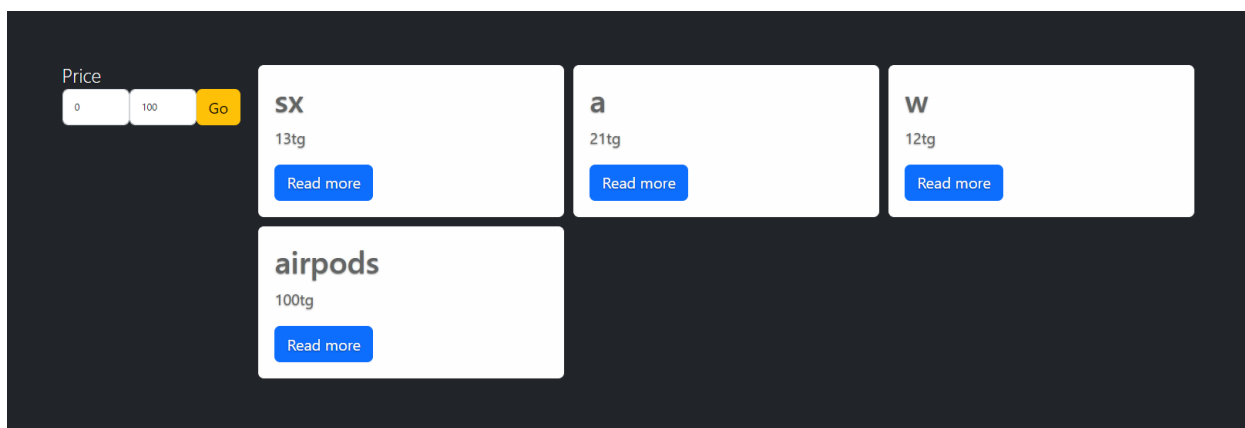
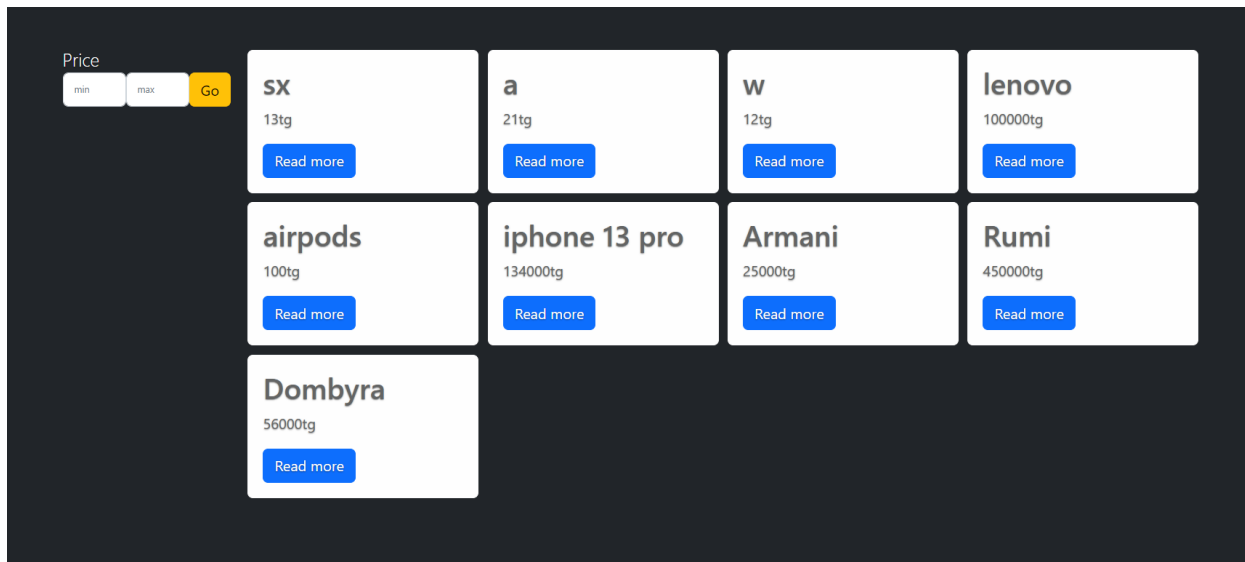
        res, err := db.Query(fmt.Sprintf("SELECT * FROM `article` WHERE `title` = '%s'", query))
        if err != nil {
            panic(err)
        }
    }
}

```

Then there is the carousel of most popular products on our website



Let's move onto the next function. It is the filtering function based on the price of the products.



The code of the filter function:

```

func filterHandler(w http.ResponseWriter, r *http.Request) {
    query1 := r.FormValue("query1")
    query2 := r.FormValue("query2")

    if query1 == "" || query2 == "" {
        http.Redirect(w, r, "/", http.StatusSeeOther)
    } else {
        t, err := template.ParseFiles("templates/index2.html", "templates/header.html", "templates/footer.html")

        if err != nil {
            fmt.Fprint(w, err.Error())
        }

        db, err := sql.Open("mysql", "root:root@tcp(127.0.0.1:3306)/golang")
        if err != nil {
            panic(err)
        }
        defer db.Close()

```

```

    res, err := db.Query(fmt.Sprintf("SELECT * FROM `article` WHERE `price` >= '%s' AND `price` <= '%s'", query1, query2))
    if err != nil {
        panic(err)
    }

    posts = []Article{}
    for res.Next() {
        var post Article
        err = res.Scan(&post.Id, &post.Title, &post.Anons, &post.Price)
        if err != nil {
            panic(err)
        }

        posts = append(posts, post)
    }

    t.ExecuteTemplate(w, "index2", posts)

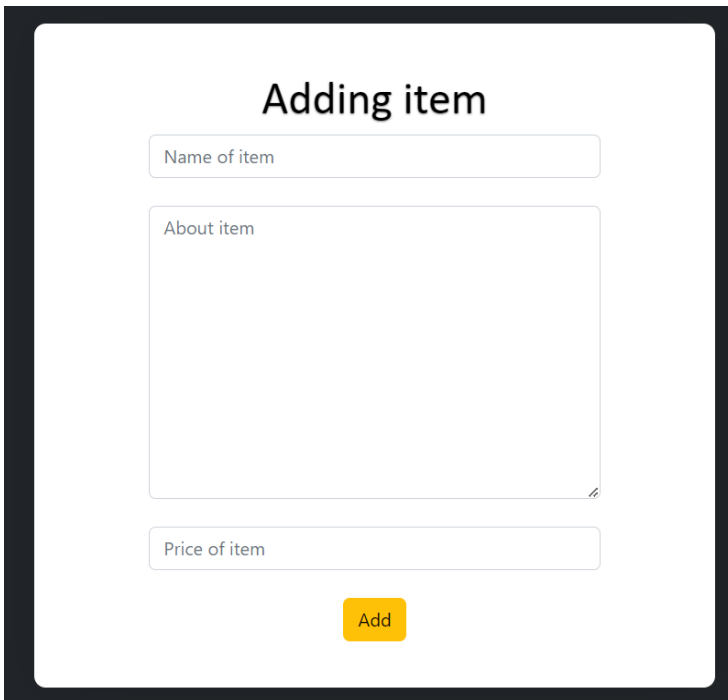
```

Description of code:

1. Function retrieves two query parameters named "query1" and "query2". These two parameters represent the lower and upper bounds of the price range that the user wants to filter the items by.
2. The function checks if both "query1" and "query2" are not empty. If one of them is empty, the function redirects the user to the homepage using the "Redirect" method of the "http" package.

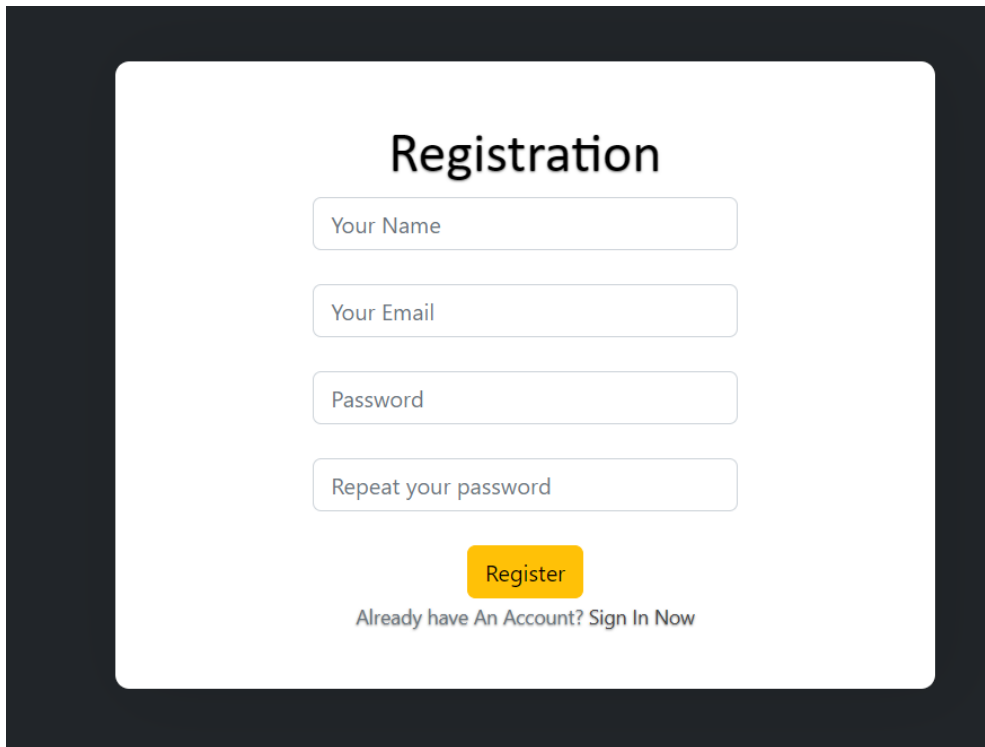
3. The function constructs a SQL query string using the "fmt.Sprintf" method of the "fmt" package. This query selects all the columns from the "article" table where the "price" column is greater than or equal to "query1" and less than or equal to "query2".
4. The function iterates over the result set returned by the SQL query using a "for" loop and retrieves the values of each column using the "res.Scan" method. These values are then stored in a struct named "Article" and appended to a slice named "posts"

Next is the adding item, registration, authorisation function that we already added at the Midterm 1 report.



The image shows a web form titled "Adding item" enclosed in a dark border. The form has a white background and contains three input fields and one button. The first input field is labeled "Name of item" and is a single-line text box. The second input field is labeled "About item" and is a larger text area. The third input field is labeled "Price of item" and is a single-line text box. Below these fields is a yellow button with the text "Add".

Redesigned the registration and authorization pages. More beautiful, how do you think?

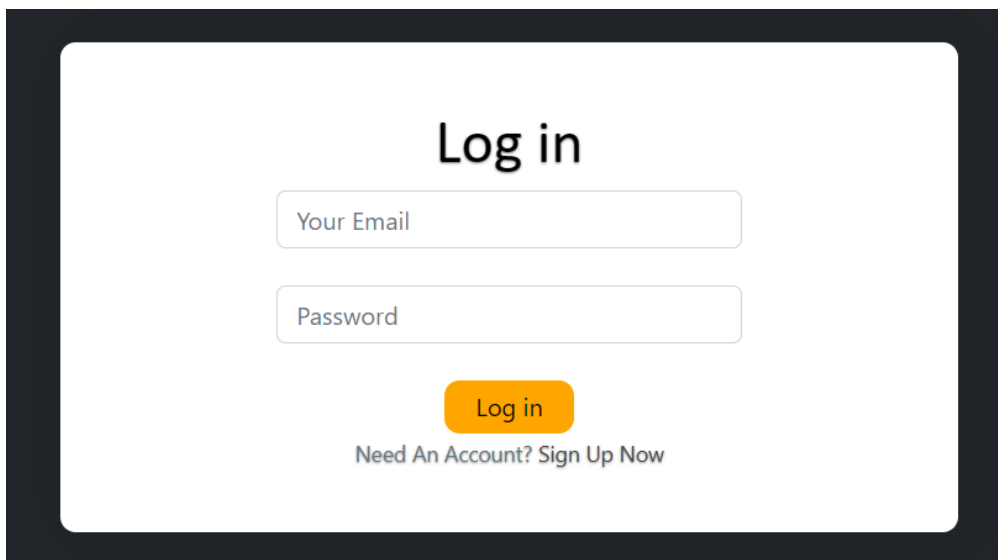


The registration form is centered on a white background with a dark gray border. It features a title 'Registration' in a large, bold, black font. Below the title are four input fields, each with a light gray border and rounded corners. The first field is labeled 'Your Name', the second 'Your Email', the third 'Password', and the fourth 'Repeat your password'. Below these fields is a yellow button with the text 'Register' in black. At the bottom, there is a link that says 'Already have An Account? Sign In Now' in a smaller, gray font.

Registration

Register

Already have An Account? Sign In Now



The log in form is centered on a white background with a dark gray border. It features a title 'Log in' in a large, bold, black font. Below the title are two input fields, each with a light gray border and rounded corners. The first field is labeled 'Your Email' and the second 'Password'. Below these fields is a yellow button with the text 'Log in' in black. At the bottom, there is a link that says 'Need An Account? Sign Up Now' in a smaller, gray font.

Log in

Log in

Need An Account? Sign Up Now

And finally, the last function is the function of commenting and giving a rate of the products, we call it review. Here we can text the name of user, feedback and rate (0-10)

Add your review

Your name

Your opinion

Give rating from 0 to 10

Add review

Reviews

Samat

jsdnjsadjnsadjsdna

8/10

ijdfjdf

djfnjdjndjnf

9/10

Abdussamad

project bazar jok eken, tema

10/10

Tulpar

bazar jok

8/10

Abdussamad

wqeqwfsdfsdfsfsd











9/10

Baital

qwertyuiodjnfjdnfds

6/10

+ Options

				id	name	text	rating
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	Samat	jsdnjsadjnsadjsdna	8
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	ijdfjdf	djfnjdjnfjdjnf	9
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	Abdussamad	project bazar jok eken, tema	10
<input type="checkbox"/>	 Edit	 Copy	 Delete	5	Tulpar	bazar jok	8
<input type="checkbox"/>	 Edit	 Copy	 Delete	6	Abdussamad	wqeqwfsdfsdfsfsd	9

The database table of the review.