

Implementing Comments part in project:

```
func saveComment(w http.ResponseWriter, r *http.Request) { 1 usage  ± Dmitry Pan *
    product_id := r.FormValue( key: "product_id")
    nickname := r.FormValue( key: "nickname")
    comment_text := r.FormValue( key: "comment_text")

    if nickname == "" || comment_text == "" {
        fmt.Fprintf(w, format: "You didnt write something")
    } else {
        db, err := sql.Open( driverName: "mysql", dataSourceName: "root:root@tcp(127.0.0.1:8889)/golang")
        if err != nil : err *
        defer db.Close()

        insert, err := db.Query(fmt.Sprintf( format: "INSERT INTO `products_comments` (`product_id`,`comment`,`User_nickname`) VALUES('%s','%s'"
        if err != nil : err *
        defer insert.Close()

        http.Redirect(w, r, url: "/product/"+product_id+"?", http.StatusSeeOther)
    }
}
```

Implementing rating part:

```
func add_rating(w http.ResponseWriter, r *http.Request) { 1 usage  new *
    rating := r.FormValue( key: "rating")
    product_id := r.FormValue( key: "product_id")
    if rating == "" {
        fmt.Fprintf(w, format: "You didnt write something")
    } else {
        db, err := sql.Open( driverName: "mysql", dataSourceName: "root:root@tcp(127.0.0.1:8889)/golang")
        if err != nil {
            panic(err)
        }
        defer db.Close()
        insert, err := db.Query(fmt.Sprintf( format: "UPDATE `products` SET `rating_count` = `rating_count`+1, `rating_sum` = `rating_sum`+"
        if err != nil {
            panic(err)
        }
        defer insert.Close()

        http.Redirect(w, r, url: "/product/"+product_id+"?", http.StatusSeeOther)
    }
}
```

Implementing filter part on catalog page

```
func catalog(w http.ResponseWriter, r *http.Request) { 1 usage  ± Bekkhan +1 *
    typeFilter := r.URL.Query().Get( key: "type")
    brandFilter := r.URL.Query().Get( key: "brand")
    sortValue := r.URL.Query().Get( key: "sorting")

    t, err := template.ParseFiles( filenames...: "templates/header.html", "templates/footer.html", "templates/smallcard.html", "templates/
    if err != nil {
        fmt.Fprintf(w, err.Error())
    }

    db, err := sql.Open( driverName: "mysql", dataSourceName: "root:root@tcp(127.0.0.1:8889)/golang")
    if err != nil : err *
    defer db.Close()

    query := "SELECT * FROM `products`"
    if typeFilter != "" {
        query += " WHERE `type`='" + typeFilter + "'"
    }
    if brandFilter != "" {
        if typeFilter == "" {
            query += " WHERE "
        } else {
            query += " AND "
        }
        query += "`brand`='" + brandFilter + "'"
    }
    if sortValue != "" {
        query += " ORDER BY `" + sortValue + "` DESC"
    }

    rows, err := db.Query(query)
    if err != nil : err *
    defer rows.Close()
}
```