

Visoka škola strukovnih studija
za informacione i komunikacione tehnologije

CombatZone
<https://combat-zone.kibonga.space>

Smer: Internet tehnologije
Modul: Web programiranje
Predmet: Praktikum PHP

Pavle Đurđić 7/19

Sadržaj

1 UVOD	5
1.1 Korišćeni programski jezici	5
Korišćene tehnologije:	5
1.2 Opis funkcionalnosti.....	5
1.3 Template.....	6
2. Organizacija.....	6
2.1 Organizaciona šema I šema baze podataka.....	6
2.2 Sitemap	9
2.2.1 Robots.txt.....	11
2.3 Opis stranica.....	12
2.3.1 index.php.....	12
2.3.2 about.php	13
2.3.3 contact.php	14
2.3.4 shop.php.....	15
2.3.5 glove.php.....	16
2.3.6 cart.php	17
2.3.7 author.php.....	18
2.3.8 login.php	18
2.3.9 register.php	19
2.3.10 admin-dashboard.php	20
2.3.11 admin-orders.php	20
2.3.12 admin-user.php	22
2.3.13 admin-order.php	23
2.3.14 admin-gloves.php	23
2.3.15 gloves.php	25
2.3.16 admin-users.php	27

2.3.18 admin-message.php.....	30
3. Struktura projekta	31
4. Kodovi	34
4.1 PHP.....	34
4.1.1 index.php.....	34
4.1.2 config/config.php.....	37
4.1.3. config/connection.php.....	37
4.1.4 config/routes.php	38
4.1.5 models/auth/Login.php	39
4.1.6 models/auth/Logout.php.....	43
4.1.7 models/auth/Register.php	45
4.1.8 models/public/SendMessage.php	48
4.1.9 models/users/Users.php.....	50
4.1.10 models/users/admin/Admin.php	53
4.1.11 models/users/admin/gloves/ActiveGloves.php.....	53
4.1.12 models/users/admin/gloves/Category.php	58
4.1.13 models/users/admin/gloves/Edit.php	59
4.1.14 models/users/admin/gloves/Insert.php.....	65
4.1.15 models/users/admin/gloves/Remove.php	69
4.1.16 models/users/admin/orders/Display.php	70
4.1.17 models/users/admin/orders/Excel.php.....	72
4.1.18 models/users/admin/users/Unlcok.php.....	73
4.1.19 models/users/customers/Customer.php	75
4.1.20 models/users/customers/Purchase.php	76
4.1.21 models/Filter.php	79
4.1.22 models/Helpers.php	81
4.1.23 models/Patterns.php	101
4.1.24 models/Word.php	101
4.1.25 views/fixed/admin-table.php.....	102
4.1.26 views/fixed/admin.php	104
4.1.27 views/fixed/footer.php	105

4.1.28 views/fixed/head.php	107
4.1.29 views/fixed/nav.php	108
4.1.30 views/fixed/components/order-detail.component.php	109
4.1.31 views/pages/auth/login.php	110
4.1.32 views/pages/auth/register.php.....	111
4.1.33 views/pages/public/about.php	114
4.1.34 views/pages/public/author.php	116
4.1.35 views/pages/public/contact.php.....	117
4.1.36 views/pages/public/contact.php.....	119
4.1.37 views/pages/public/index.php	121
4.1.38 views/pages/public/shop.php	123
4.1.39 views/pages/users/cart.php	127
4.1.40 views/pages/users/admin/admin-dashboard.php.....	129
4.1.41 views/pages/users/admin/admin-gloves.php	130
4.1.42 views/pages/users/admin/admin-message.php.....	134
4.1.43 views/pages/users/admin/admin-messages.php	136
4.1.44 views/pages/users/admin/order.php.....	136
4.1.45 views/pages/users/admin/admin-orders.php	139
4.1.46 views/pages/users/admin/user.php	142
4.1.47 views/pages/users/admin/admin-users.php	146
4.1.48 views/pages/users/admin/mode/gloves.php.....	146
4.2 CSS	153
4.2.1 style.css	153
4.3 JS	156
4.3.1 main.js	156

1 UVOD

1.1 Korišćeni programski jezici

Korišćene tehnologije:

- HTML5
- CSS3
- Javascript
- PHP
- Material Design Icons
- Bootstrap 5
- JQuery
- JQuery Validate
- PHP klase

1.2 Opis funkcionalnosti

- Navigacija se dinamički učitava iz baze (sve vrste navigacije)
- Proizvodi se dinamički učitavaju filtriranje/sortiranje/paginacija/straničenje (sve serverski)
- Svi naslovi, podnaslovi, navigacija, metapodaci, ikonice se za svaku stranicu učitavaju dinamički iz baze
- Obrada grešaka
- Čuvanje podataka korpe na klijentskoj i serverskoj strani, svaki korisnik ima svoju posebnu korpu koja se čuva u bazi i nakon logout-a
- Registracija/Logovanje (klijent/server)
- Zaštita stranica od neovlašćenog pristupa
- Prikaz poseta sajtu izražen u %
- CRUD operacije nad proizvodima (menjanje veličine, slike)
- Podaci o proizvodima i porudžbinama su istorijski, detalj porudžbine kao i njene stavke se ne menjaju u zavisnosti od promene stanja nekog proizvoda
- Detalji o korisnicima/porudžbinama/porukama
- Prikaz korisnika, njihovi osnovni podaci, porudžbine, detalji porudžbine, stavke porudžbine...
- Zaključavanje profila korisnika posle 5 neuspelih pokušaja logovanje u roku od 5min
- Prikaz poseta sajtu poslednjih 24h, brisanje zastarelih poseta
- Excel implementiran kao download detalja poruzbine
- Porudžbina se čuva kao istorijski podatak
- Word download je u footer-u

1.3 Template

Nije korišćen templejt

Napomena:

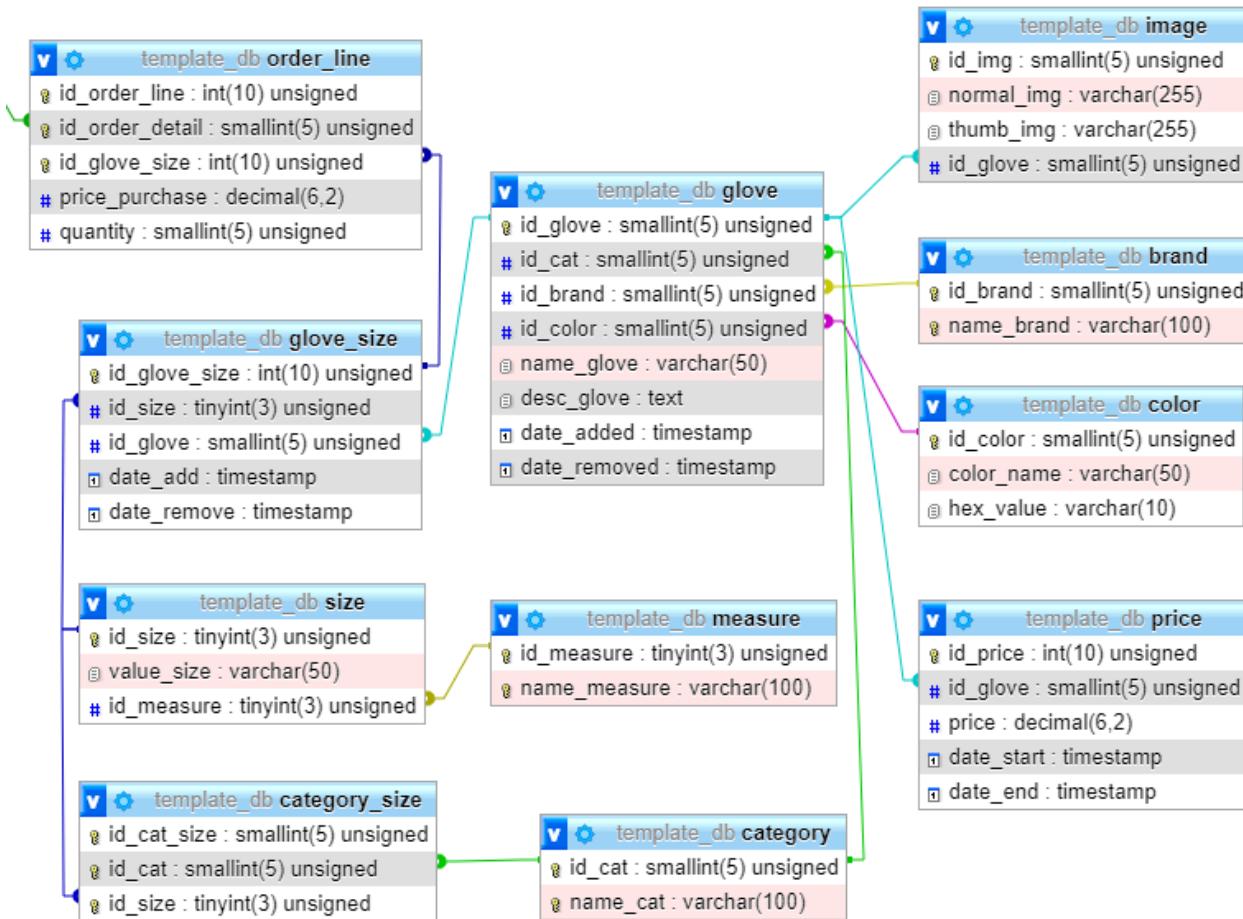
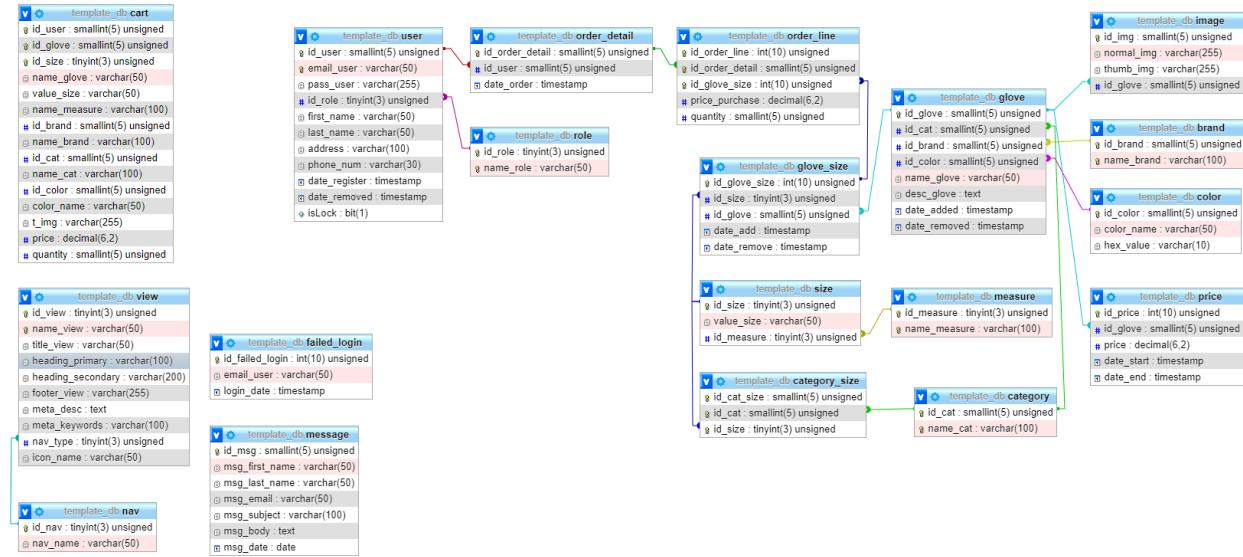
Određene komponente PHP-a jednostavno nisu kompatibilne sa Linux serverom

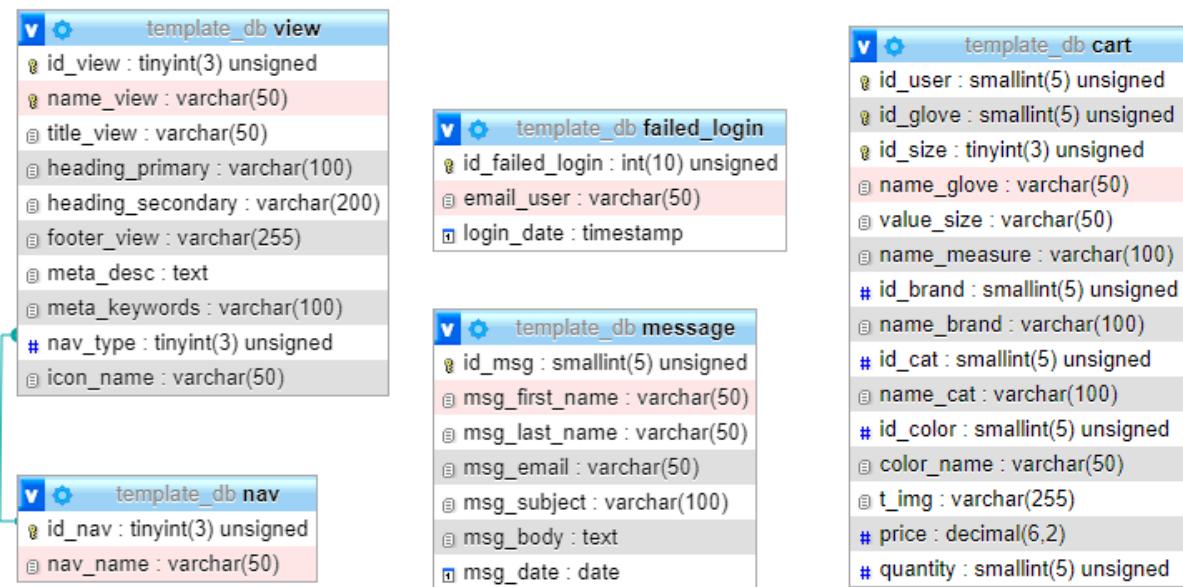
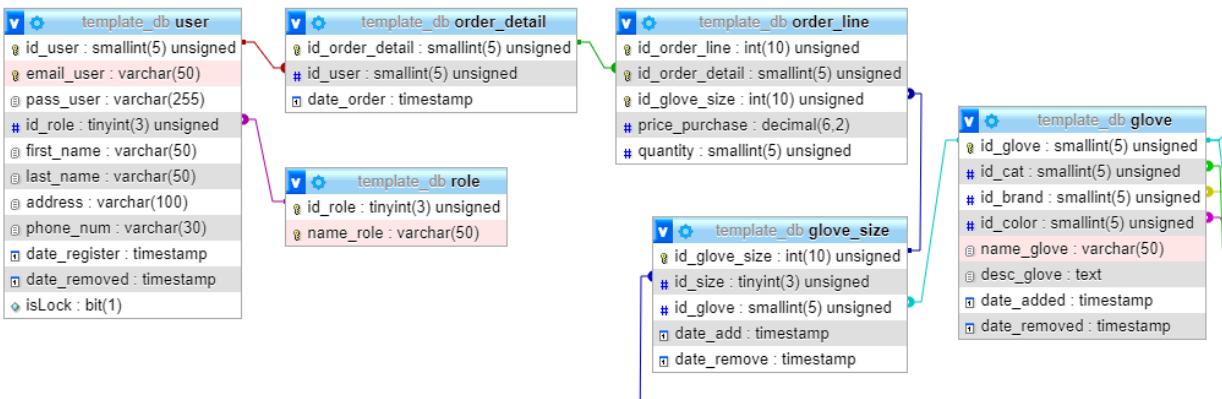
1. Download Word dokumenta (COM klasa podržava samo Windows)
2. Serverska Validation klasa (<https://github.com/davidecesarano/Validation>)
 - sve ove komponente rade u lokalnu ali ne i na Linux serveru

2. Organizacija

2.1 Organizaciona šema I šema baze podataka

Sve stranice su međusobno povezane i autorizovane





2.2 Sitemap

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset>
  xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
  <url>
    <loc>https://combat-zone.kibonga.space/index.php?page=index</loc>
    <lastmod>2021-06-09</lastmod>
    <changefreq>daily</changefreq>
    <priority>1.00</priority>
  </url>
  <url>
    <loc>https://combat-zone.kibonga.space/index.php?page=login</loc>
    <lastmod>2021-06-09</lastmod>
    <changefreq>daily</changefreq>
    <priority>0.90</priority>
  </url>
  <url>
    <loc>https://combat-zone.kibonga.space/index.php?page=register</loc>
    <lastmod>2021-06-09</lastmod>
    <changefreq>weekly</changefreq>
    <priority>0.80</priority>
  </url>
  <url>
    <loc>https://combat-zone.kibonga.space/index.php?page=customer</loc>
    <lastmod>2021-06-09</lastmod>
    <changefreq>weekly</changefreq>
    <priority>0.80</priority>
  </url>
  <url>
    <loc>https://combat-zone.kibonga.space/index.php?page=admin-
dashboard</loc>
    <lastmod>2021-06-09</lastmod>
    <changefreq>weekly</changefreq>
    <priority>0.80</priority>
  </url>
  <url>
    <loc>https://combat-zone.kibonga.space/index.php?page=admin-orders</loc>
    <lastmod>2021-06-09</lastmod>
    <changefreq>weekly</changefreq>
    <priority>0.80</priority>
  </url>
  <url>
    <loc>https://combat-zone.kibonga.space/index.php?page=admin-gloves</loc>
    <lastmod>2021-06-09</lastmod>
    <changefreq>weekly</changefreq>
```

```
    <priority>0.80</priority>
  </url>
  <url>
    <loc>https://combat-zone.kibonga.space/index.php?page=glove</loc>
    <lastmod>2021-06-09</lastmod>
    <changefreq>weekly</changefreq>
    <priority>0.80</priority>
  </url>
  <url>
    <loc>https://combat-zone.kibonga.space/index.php?page=admin-users</loc>
    <lastmod>2021-06-09</lastmod>
    <changefreq>weekly</changefreq>
    <priority>0.80</priority>
  </url>
  <url>
    <loc>https://combat-zone.kibonga.space/index.php?page=admin-messages</loc>
    <lastmod>2021-06-09</lastmod>
    <changefreq>weekly</changefreq>
    <priority>0.80</priority>
  </url>
  <url>
    <loc>https://combat-zone.kibonga.space/index.php?page=admin-message</loc>
    <lastmod>2021-06-09</lastmod>
    <changefreq>weekly</changefreq>
    <priority>0.80</priority>
  </url>
  <url>
    <loc>https://combat-zone.kibonga.space/index.php?page=admin-order</loc>
    <lastmod>2021-06-09</lastmod>
    <changefreq>weekly</changefreq>
    <priority>0.80</priority>
  </url>
  <url>
    <loc>https://combat-zone.kibonga.space/index.php?page=admin-user</loc>
    <lastmod>2021-06-09</lastmod>
    <changefreq>weekly</changefreq>
    <priority>0.80</priority>
  </url>
  <url>
    <loc>https://combat-zone.kibonga.space/index.php?page=cart</loc>
    <lastmod>2021-06-09</lastmod>
    <changefreq>weekly</changefreq>
    <priority>0.80</priority>
  </url>
```

```
<url>
  <loc>https://combat-zone.kibonga.space/index.php?page=shop</loc>
  <lastmod>2021-06-09</lastmod>
  <changefreq>weekly</changefreq>
  <priority>0.80</priority>
</url>
<url>
  <loc>https://combat-zone.kibonga.space/index.php?page=about</loc>
  <lastmod>2021-06-09</lastmod>
  <changefreq>weekly</changefreq>
  <priority>0.80</priority>
</url>
<url>
  <loc>https://combat-zone.kibonga.space/index.php?page=contact</loc>
  <lastmod>2021-06-09</lastmod>
  <changefreq>weekly</changefreq>
  <priority>0.80</priority>
</url>
<url>
  <loc>https://combat-zone.kibonga.space/index.php?page=author</loc>
  <lastmod>2021-06-09</lastmod>
  <changefreq>weekly</changefreq>
  <priority>0.80</priority>
</url>
</urlset>
```

2.2.1 Robots.txt

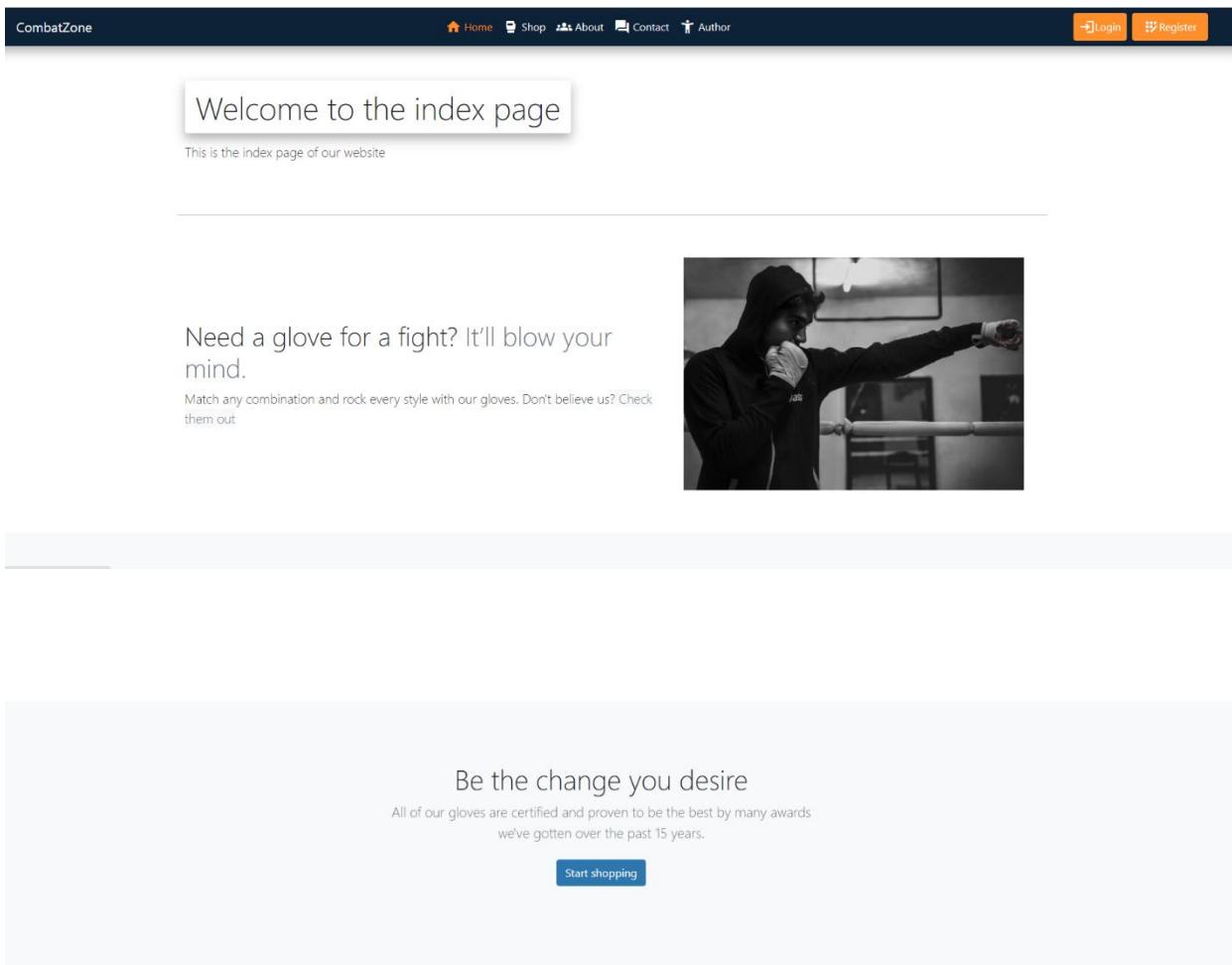
User-agent:*

Disallow:/admin/

2.3 Opis stranica

Na svim stranicama se razni navigacioni meniji, ikonice kao i meta podaci, dinamički učitavaju iz baze

2.3.1 index.php



The screenshot shows the homepage of a website named 'CombatZone'. The header includes a navigation bar with links for Home, Shop, About, Contact, Author, Login, and Register. The main content area features a large text box with the heading 'Welcome to the index page' and the subtext 'This is the index page of our website'. Below this, there is a promotional message: 'Need a glove for a fight? It'll blow your mind.' followed by a smaller note: 'Match any combination and rock every style with our gloves. Don't believe us? Check them out.' To the right of the text is a black and white photograph of a person in a hoodie performing a boxing or kickboxing move. Further down the page is another promotional section with the slogan 'Be the change you desire' and a 'Start shopping' button, accompanied by a black and white image of two fighters in a ring.



Oh yeah, it's that good. See for yourself.

Quality and design are our base principles and only the finest is the best. Our products are made entirely of eco-friendly materials.

FOOTER CONTENT
This is the footer of the index page

NAVIGATION

- [Home](#)
- [Shop](#)
- [About](#)
- [Contact](#)
- [Author](#)

USEFUL

- Docs
- Word
- JS
- LinkedIn
- Github
- Portfolio

Made by Kibonga

2.3.2 about.php

Statička stranica sa dinamički učitanim podrazumevanim sadržajem

CombatZone

Home Shop About Contact Author Login Register

About us

Here you can learn more about us

Who are we



We are a shop specialized in selling gloves for combat sports. Our products have been praised by many around the world.



Our top notch materials make for the best quality by which we are recognized. Our gloves have gathered many awards for their pristine design.



Combat zone has a stong base of satisfied customers, and we tend to keep it that way. Our quality of service is our number one moto.

What others think about us



Ben Askren



Jorge Masvidal



Kamaru Usman

I've been a fan of Combat Zone from day one. I'm really impressed with the quality and design, not to mention the prices.

If you are like me and like to change gloves frequently I would suggest looking into their newest Venum collection, its awesome.

They have gloves for all the seasons, doesn't matter if its boxing or MMA or even karate. They rock with everything.

FOOTER CONTENT

This is the footer of the about page

NAVIGATION

- [Home](#)
- [Shop](#)
- [About](#)
- [Contact](#)
- [Author](#)

USEFUL

- Docs
- Word
- JS
- LinkedIn
- Github
- Portfolio

Made by Kibonga

2.3.3 contact.php

Forma za slanje poruke adminu (validacija klijent/server)

Home Shop About Contact Author Login Register

Contact us

Here you can contact us



Have any questions? Send us a message.

Don't hesitate to contact us if you have any questions, we are here to help.

First name Last name

Email

Subject

Message

Send

2.3.4 shop.php

Prikaz svih dostupnih proizvoda

Filteri: kategorije, brendovi, boje, cena, naziv, sortiranje, straničenje, paginacija (sve serverski)

Proizvodi mogu imati različite veličine kao i tip veličine izražene kroz obim ili kroz unce (circumference/OZ)

CombatZone

Home Shop About Contact Author Login Register

Welcome to the shop

Here you can view and buy any glove

Search for glove

Search

Glove type

Boxing
 Karate
 MMA

Brand

Everlast
 Leone
 Venum

Color

Red
 Black
 White
 Blue
 Yellow

Available gloves: 13

Gloves per page:

Sort:

Select


\$ 89.99
Leone Pink Fight
10, 14, 16 (OZ)
Pink
View


\$ 89.99
Venum Blood Red
10, 12, 14 (OZ)
Red
View


\$ 59.99
Leone Bellator Bout
M, L
Black
View


\$ 99.99
Leone Glory Champion
8, 10 (OZ)
White
View

Search for glove

C

Glove type

Boxing
 Karate
 MMA

Brand

Everlast
 Leone
 Venum

Color

Red
 Black
 White
 Blue
 Pink

Price range

Available gloves: 4

Gloves per page:

Sort:

Select


\$ 99.99
Leone Glory Champion
10 (OZ)
White
View


\$ 34.99
Leone Classic Yellow
8, 10, 12, 14, 16 (OZ)
Yellow
View


\$ 69.99
Leone Classic Red
12, 14 (OZ)
Red
View


\$ 79.99
Leone Canelo White
12, 14, 16 (OZ)
White
View

2.3.5 glove.php

Prikaz jednog proizvoda

Glove page

Here you can view and purchase this glove

[← Shop](#)



Leone

Leone Pink Fight

\$89.99

Pink

10 OZ

14 OZ

16 OZ

Unleash the fighter within with newest Leone Fight collection

Glove page

Here you can view and purchase this glove

[← Shop](#)



Leone

Leone Pink Fight

\$89.99

Pink

10 OZ

14 OZ

16 OZ

Unleash the fighter within with newest Leone Fight collection

1 

2.3.6 cart.php

Samo korisnik može da kupi proizvod

Korpa se čuva na klijentu i serveru, za svakog korisnika, tako da se pamti kada se korisnik uloguje/izloguje

Cart page

Here you can confirm your purchase

Items in cart: 1

No.	Image	Name	Category	Brand	Color	Size	Price	Quantity	Total	Remove
1		Leone Pink Fight	Boxing	Leone	Pink	100Z	89.99	<input type="text" value="1"/>	\$ 89.99	

[Make Purchase](#) [Continue shopping](#)

 Free Delivery within 1-2 weeks

Order
Price:
\$89.99

Total:
\$89.99

FOOTER CONTENT

This is the cart page footer

NAVIGATION

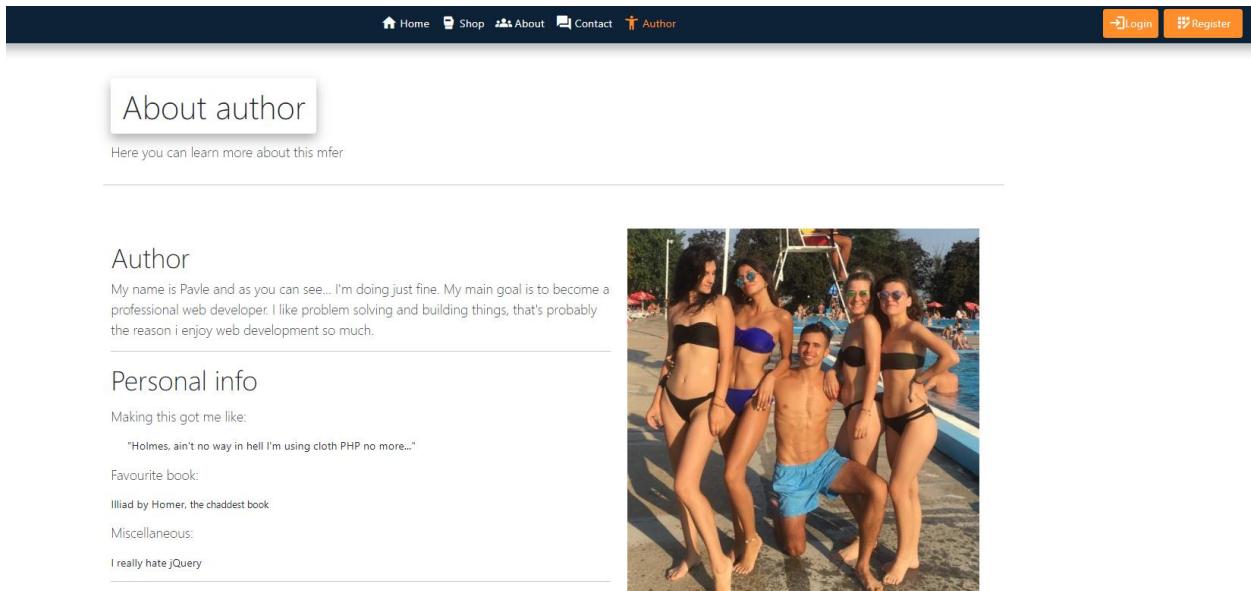
[Home](#)
[Shop](#)
[About](#)

USEFUL

[Docs](#)
[Word](#)
[JS](#)
[LinkedIn](#)
[GitHub](#)

2.3.7 author.php

Statička stranica o autoru



The screenshot shows a website's navigation bar at the top with links for Home, Shop, About, Contact, Author, Login, and Register. The main content area has a title 'About author' and a sub-section 'Author'. It includes a bio about the author, a quote, and a list of interests. To the right of the text is a photograph of five people in swimwear standing on a beach.

Author
My name is Pavle and as you can see... I'm doing just fine. My main goal is to become professional web developer. I like problem solving and building things, that's probably the reason i enjoy web development so much.

Making this got me like:
"Holmes, ain't no way in hell I'm using cloth PHP no more..."

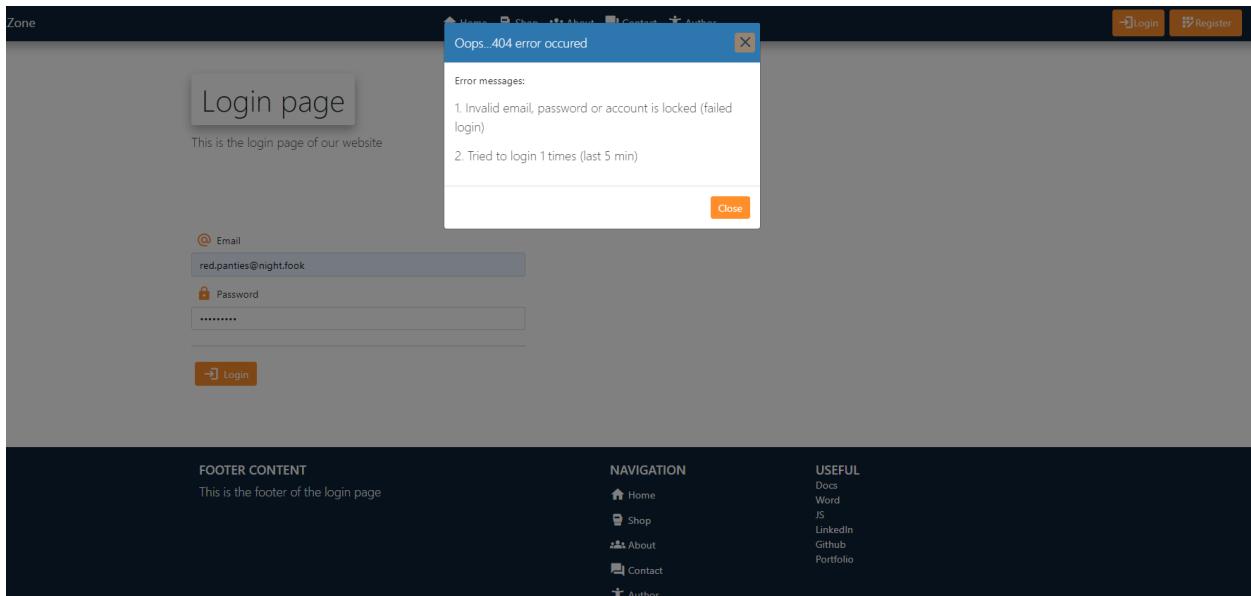
Favourite book:
Illiad by Homer, the chaddest book

Miscellaneous:
I really hate jQuery

2.3.8 login.php

Validacija (klijent/server)

Ako korisnik, koji već postoji u bazi(validan mejl) pokuša da se uloguje 5 puta za redom u intervalu od 5min(bez uspeha), zaključava mu se profil koji može da otključa samo admin



The screenshot shows a login page with fields for Email and Password. A modal window titled 'Oops.. 404 error occured' displays two error messages: '1. Invalid email, password or account is locked (failed login)' and '2. Tried to login 1 times (last 5 min)'. Below the modal is a 'Close' button. At the bottom of the page, there is footer content, a navigation menu, and a 'USEFUL' section with links to Docs, Word, JS, LinkedIn, Github, and Portfolio.

Zone

Login page
This is the login page of our website

Oops.. 404 error occured

Error messages:

1. Invalid email, password or account is locked (failed login)
2. Tried to login 1 times (last 5 min)

Close

Email: red.panties@night.fook
Password:

Login

FOOTER CONTENT
This is the footer of the login page

NAVIGATION
Home
Shop
About
Contact
Author

USEFUL
Docs
Word
JS
LinkedIn
Github
Portfolio

2.3.9 register.php

Validacija(klijent/server)



Register page

This is the register page of our website

Full name

First name

Last name

Email

Password

Confirm password

Contact phone

Address

Register

2.3.10 admin-dashboard.php

Izlistana statistika o posetama stranica za prethodni dan, podaci su sačuvani u log fajlu i svi podaci koji prekorače zadati interval se brišu i log fajla

Sve admin stranice imaju svoj zaseban tip dinamičke navigacije

The screenshot shows a dark-themed administrative dashboard. At the top, there's a header bar with links for Home, Shop, About, Contact, Author, a shopping cart icon, and a Logout button. Below the header, a large white box says "Welcome back". Underneath it, a smaller text says "This is the main dashboard". A blue navigation bar at the top of the main content area contains links for Dashboard, Orders, Gloves, Users, and Messages. Below this, a black banner displays the text "Total visits: 941". The main content area lists the most visited pages from the previous day:

Page	Visits (%)
admin-dashboard	4.68%
admin-orders	3.61%
admin-gloves	5.1%
admin-users	11.16%
admin-messages	4.78%

2.3.11 admin-orders.php

Listing svih dostupnih porudžbina

CRUD operacije ne uticu na detalje porudzbine jer se porudzbine cuvaju kao istorijski podaci
DELETE / INSERT ne brišu proizvod, samo mu dodeljuju nullable timestamp() cime se obezbeđuje da podaci budu istorijski

Manage orders

Here you can manage all of the orders

[Dashboard](#) [Orders](#) [Gloves](#) [Users](#) [Messages](#)

Search for order

 Search

Number of orders: 13

Orders per page:

4

Sort:

Select

All orders

No.	Customer	Items	Total	Date	More details
1	Mystic Mac	2	\$189.98	2021-06-05 18:51:24	
2	Mystic Mac	3	\$399.94	2021-06-05 18:52:54	
3	Dustin Poirier	2	\$149.98	2021-06-06 22:36:29	
4	Derrick Lewis	1	\$89.99	2021-06-07 16:14:29	

[1](#) [2](#) [3](#) [4](#)

2.3.12 admin-user.php

Podaci o konkretnom korisniku i svim njegovim porudžbinama

User page

View user's info

Dashboard Orders Gloves Users Messages

All Users

User

Full name

Mystic Mac

Email

red.panties@night.fook

Address

Double Champ 155/45

Phone number

0601234567

Number of orders: 5

Orders per page:

4

Sort:

Select

Customer's orders

No.	Customer	Items	Total	Date	More details
1	Mystic Mac	2	\$189.98	2021-06-05 18:51:24	
2	Mystic Mac	3	\$399.94	2021-06-05 18:52:54	
3	Mystic Mac	1	\$59.99	2021-06-10 14:18:52	
4	Mystic Mac	1	\$269.97	2021-06-10 15:17:59	

2.3.13 admin-order.php

Prikaz konkretnje porudžbine za konkretnog korisnika, kao i download porudžbine u vidu Excela

[Stari verziji](#)

[More info about order](#)

[Dashboard](#) [Orders](#) [Gloves](#) [Users](#) [Messages](#)

[Mystic's account](#)

Order

[Excel](#)

[Order ID](#)

30

[Customer](#)

Mystic Mac

[Date of purchase](#)

2021-06-05 18:51:24

[Number of items](#)

2

[Total](#)

\$ 189.98

All order lines

No.	Image	Name	Category	Brand	Color	Size	Price	Quantity	Total
1		Leone Glory Champion	Boxing	Leone	<input type="checkbox"/> White	10 OZ	\$99.99	1	\$99.99
2		Leone Pink Fight	Boxing	Leone	<input checked="" type="checkbox"/> Pink	14 OZ	\$89.99	1	\$89.99

2.3.14 admin-gloves.php

CRUD operacije kao i podaci vezani za konkretnje proizvode

Delete opcija je zapravo UPDATE koji koristi nullable timestamp da bi se osiguralo da podatak bude istorijski

INSERT dodaje novi proizvod

UPDATE dodaje novi proizvod(ne azurira postojeci) da bi se osiguralo da porudzbina korisnika bude istorijski podatak, prethodnom proizvodu se dodaje nullable timestamp cime se osigurava da se samo jedan aktivran proizvod nalazi u korpi, ali vise tih proizvoda moze biti u razlicitim porudzbinama

Manage gloves

Here you can manage all of the gloves

Dashboard Orders Gloves Users Messages

 Add new glove

Search for glove

 Search

Available gloves: 13

Gloves per page:

4

Categories:

Select

Brands:

Select

Colors:

Select

Sort:

Select

Price range

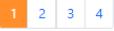
 \$100

No.	Image	Name	Category	Brand	Color	Sizes	Measure	Date	Edit	Delete
1		Leone Pink Fight	Boxing	Leone	Pink	10, 14, 16	OZ	2021-06-02 14:11:01		

Price range



No.	Image	Name	Category	Brand	Color	Sizes	Measure	Date	Edit	Delete
1		Leone Pink Fight	Boxing	Leone	Pink	10, 14, 16	OZ	2021-06-02 14:11:01	 	
2		Venum Blood Red	Boxing	Venum	Red	10, 12, 14	OZ	2021-06-02 14:12:10	 	
3		Leone Bellator Bout	MMA	Leone	Black	M, L	Circumference	2021-06-02 14:14:37	 	
4		Leone Glory Champion	Boxing	Leone	White	8, 10	OZ	2021-06-02 16:39:14	 	



2.13.15 gloves.php

Stranica za insert/update proizvoda u zavisnosti od izabranog moda

INSERT / UPDATE su zapravo oba insert-i cime s tim da edit dodaje datum uklanjanja prethodnog proizvoda a na njegovo mesto dodaje novi(istorijski podatak)

Manage gloves

Here you can manage all of the gloves

 Dashboard  Orders  Gloves  Users  Messages

 Back to gloves

 Brand

Select brand

 Category

Select category

 Sizes

 Colors

Select color

 Product name

Leone Pink Fight

 Image

No file chosen



 Price

59.99

 Description

Enter description

 Insert glove

Brand

Category

Sizes

8 OZ
 10 OZ
 12 OZ
 14 OZ
 16 OZ

Colors

Product name

Image

Choose File No file chosen



Price

Description

Update glove

2.3.16 admin-users.php

Izlistani svi korisnici kao i status njihovog naloga, koji može biti zaključan, na ovoj stranici se korisnik otključava

Manage users

Here you can manage registered users

Dashboard Orders Gloves Users Messages

Search for order

Search

Number of users: 5

Users per page:

4

Sort:

Select

No.	Customer	Email	Orders	Total spent	More details	Account status
1	Mystic Mac	red.panties@night.fook	5	\$979.87		
2	Dustin Poirier	pressure.makes.diamonds@clavicle.ko	1	\$149.98		
3	Derrick Lewis	knock.out@artist.ko	2	\$374.93		
4	Jorge Masvidal	steet.jesus@baptising.fools	4	\$634.92		

1 2

2.3.17 admin-messages.php

Izlistane sve poruke za svakog korisnika

Manage messages

Here you can manage messages

[Dashboard](#) [Orders](#) [Gloves](#) [Users](#) [Messages](#)

Search for order

Search

Number of messages: 7

Messages per page:

4

Sort:

Select

All messages

No.	Full name	Email	Subject	Date	More details
1	Mystic Mac	red.panties@night.fook	Fookin mate	2021-06-08	
2	Mystic Mac	red.panties@night.fook	Apology	2021-06-08	
3	Dustin Poirier	pressure.makes.diamonds@clavicle.ko	Job	2021-06-08	
4	Dustin Poirier	pressure.makes.diamonds@clavicle.ko	Injury update	2021-06-08	

1

2

FOOTER CONTENT

NAVIGATION

USEFUL

2.3.18 admin-message.php

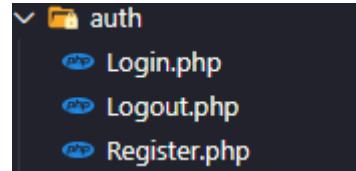
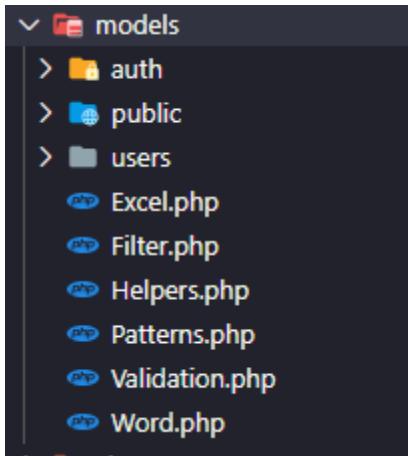
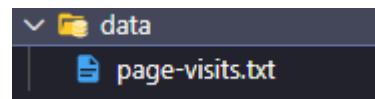
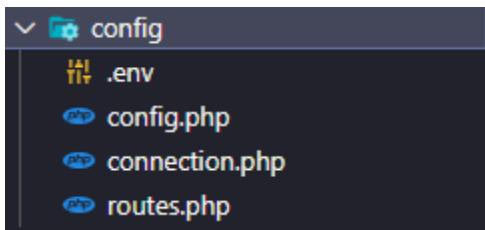
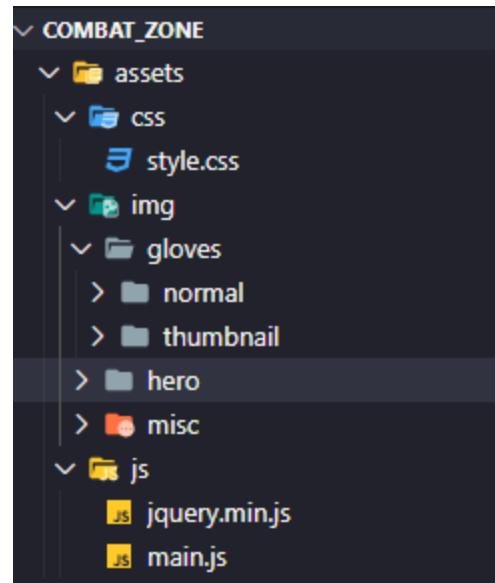
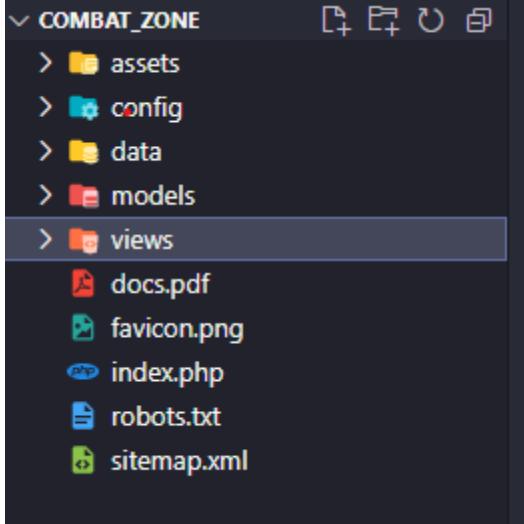
Prikaz konkretne poruke

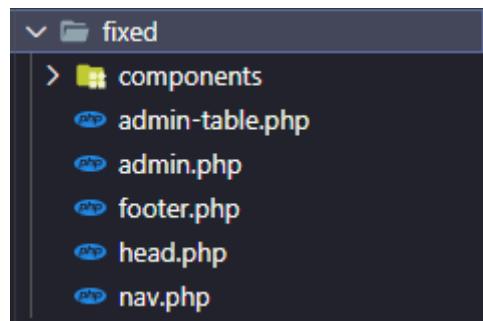
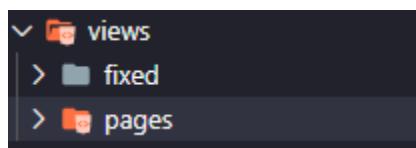
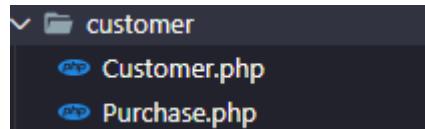
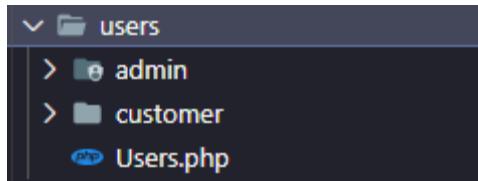
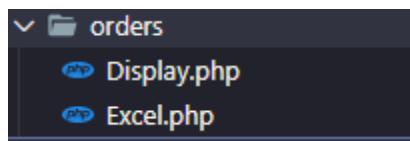
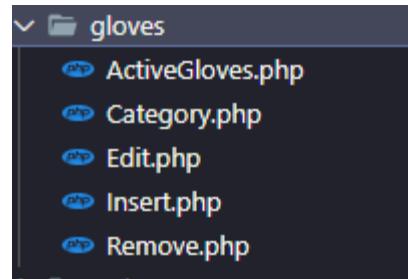
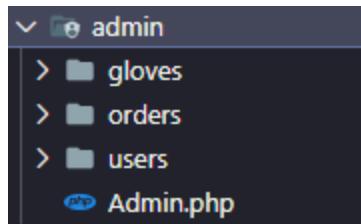
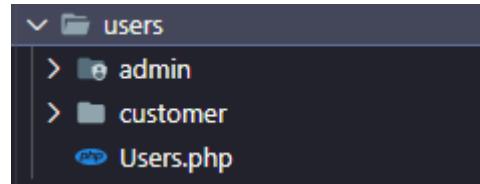
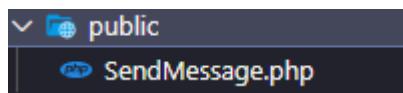
The screenshot shows a web application interface for managing messages. At the top, there is a dark header bar with navigation links: Home, Shop, About, Contact, and Author. Below this is a blue navigation bar with links: Dashboard, Orders, Gloves, Users, and Messages. A back button labeled "All Messages" is visible. The main content area has a title "Message detail" and a link "View more about the message". The message details are listed in a table-like format:

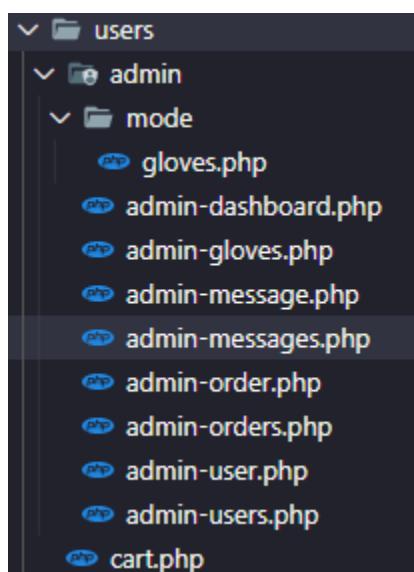
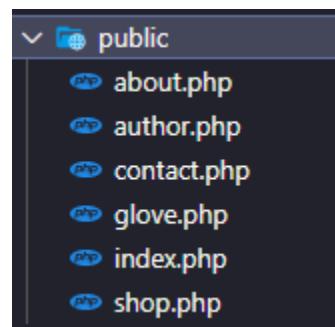
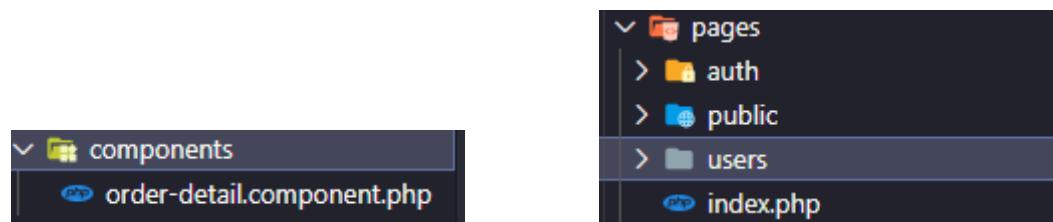
Sender	Mystic Mac
Date received	2021-06-08
Subject	Apology
Message body	And I'd just like to say from the bottom of my heart, I'd like to take this chance to apologise... to absolutely nobody. The double champ does what the fuck he wants!

At the bottom of the page is a footer section with three columns: "FOOTER CONTENT" (This is the footer of the message page), "NAVIGATION" (Home, Shop), and "USEFUL" (Docs, Word, JS, LinkedIn).

3. Struktura projektu







4. Kodovi

4.1 PHP

4.1.1 index.php

```
<?php
session_start();
// session_destroy();
ob_start();
// Routes will be available in every View
include "./config/routes.php";
include CONFIG . "/connection.php";
include MODELS . "./Helpers.php";
// Includes
defineNav();
// Switch
if (isset($_GET["page"])) && isset($_GET["page"]) && isView($_GET["page"])) {
    getViewData($_GET['page']);
    $user = "Unknown user";
    if (isset($_SESSION['loggedUser'])) {
        switch ($_SESSION['loggedUser']->name_role) {

            case "regular_user":
                $user = $_SESSION['loggedUser']->email_user;
                break;
            case "admin":
                $user = "admin";
                break;
            default:
                $user = "Unknown user";
        }
    }
    pageVisitsWrite($_GET["page"], $user);
    include FIXED . "./head.php";
    include FIXED . "./nav.php";
    switch ($_GET["page"]) {

        // IMPORTANT
        // 1. Define view in database
        // 2. Define view in array of allowed views Helpers.php -> isView()
        // 3. Define switch case "page" : (SERVER SIDE)
        // 4. Define switch case "page" : (CLIENT SIDE)
    }
}
```

```
// GENERAL
case "customer":
    include V_CUSTOMER . "/customer.php";
    break;
case "author":
    include V_PUBLIC . "/author.php";
    break;
case "contact":
    include V_PUBLIC . "/contact.php";
    break;
case "shop":
    include V_PUBLIC . "/shop.php";
    break;
case "about":
    include V_PUBLIC . "/about.php";
    break;
case "glove":
    include V_PUBLIC . "/glove.php";
    break;
case "cart":
    // Even though it's general, only users can access it (customers can
    // actually only buy)
    include V_USERS . '/cart.php';
    break;

// AUTH
case "register":
    include V_AUTH . "./register.php";
    break;
case "login":
    include V_AUTH . "./login.php";
    break;

// CUSTOMER

// ADMIN
case "admin-dashboard":
    include FIXED . "./admin.php";
    include V_ADMIN . "/admin-dashboard.php";
    break;
case "admin-orders":
    include FIXED . "./admin.php";
```

```
    include V_ADMIN . "/admin-orders.php";
    break;
case "admin-gloves":
    include FIXED . "./admin.php";
    include V_ADMIN . "/admin-gloves.php";
    break;
case "admin-users":
    include FIXED . "./admin.php";
    include V_ADMIN . "/admin-users.php";
    break;
case "admin-messages":
    include FIXED . "./admin.php";
    include V_ADMIN . "/admin-messages.php";
    break;
case "admin-message":
    include FIXED . "./admin.php";
    include V_ADMIN . "/admin-message.php";
    break;
case "admin-order":
    include FIXED . "./admin.php";
    include V_ADMIN . "/admin-order.php";
    break;
case "admin-user":
    include FIXED . "./admin.php";
    include V_ADMIN . "/admin-user.php";
    break;

    // Index
case "index":
    include V_PUBLIC . "/index.php";
    break;
}
} else {
    getViewData("index");
    include FIXED . "./head.php";
    include FIXED . "./nav.php";
    include V_PUBLIC . "/index.php";
}

// Footer
include FIXED . "./footer.php";
ob_flush();
```

4.1.2 config/config.php

```
<?php
// Configuration constants
define("SERVER", env("SERVER"));
define("DATABASE", env("DATABASE"));
define("USERNAME", env("USERNAME"));
define("PASSWORD", env("PASSWORD"));
define("WEBSITE", env("WEBSITE"));

// View data constants
// define("TITLE", viewData(""));

// Env function
function env($param)
{
    $rows = file(CONFIG . "./.env");
    $const = null;

    foreach ($rows as $i => $r) {
        $r = trim($r);
        list($name, $val) = explode("=", $r);
        if ($name == $param) {
            $const = $val;
            break;
        }
    }
    return $const;
}
```

4.1.3. config/connection.php

```
<?php
// Includes
include "config.php";

// Try making connection
try {
    $conn = new PDO("mysql:host=" . SERVER . ";dbname=" . DATABASE . ";charset=utf8", USERNAME, PASSWORD);
    $conn->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_OBJ);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $ex) {
```

```
    echo $ex->getMessage();
}
```

4.1.4 config/routes.php

```
<?php
// ROOT LEVEL
define("ROOT", $_SERVER["DOCUMENT_ROOT"] . "/combat_zone");

// FIRST LEVEL DIRECTORIES
define("MODELS", ROOT . "/models");
define("VIEWS", ROOT . "/views");
define("DATA", ROOT . "/data");
define("CONFIG", ROOT . "/config");
define("ASSETS", ROOT . "/assets");

// SECOND LEVEL DIRECTORIES
// Views
define("FIXED", VIEWS . "/fixed");
define("PAGES", VIEWS . "/pages");
// Pages (inside views)
define("V_PRODUCTS", PAGES . "/products");
define("V_LOGS", PAGES . "/logs");
define("V_USERS", PAGES . "/users");
define("V_AUTH", PAGES . "/auth");
define("V_ADMIN", V_USERS . "/admin");
define("V_ADMIN_MODE", V_ADMIN . "/mode");
define("V_GLOVES", PAGES . "/gloves");
define("V_CUSTOMER", V_USERS . "/customer");
define("V_PUBLIC", PAGES . "/public");
// Models
define("M_USERS", MODELS . "/users");
define("M_PRODUCTS", MODELS . "/products");
define("M_LOGS", MODELS . "/logs");
define("M_AUTH", MODELS . "/auth");
// Made changes here
// define("M_ADMIN", MODELS . "/admin");
define("M_ADMIN", M_USERS . "/admin");
define("M_CUSTOMER", MODELS . "/customer");
// Static
define("S_CSS", ASSETS . "/css");
define("S_IMG", ASSETS . "/img");
define("S_JS", ASSETS . "/js");
```

```
define("S_IMG_GLOVES", S_IMG . '/gloves');
```

4.1.5 models/auth/Login.php

```
<?php
// Session
if (!isset($_SESSION)) {
    session_start();
}
// Includes (routes, helpers)
include_once "../config/routes.php";
include MODELS . "./Helpers.php";
// include M_USERS . "/Users.php";

// Form submitted
if (isset($_POST['btnLogin'])) {
    // Includes (connection, users model)
    include CONFIG . "./connection.php";
    include M_USERS . "/Users.php";
    include MODELS . "/Validation.php";
    include MODELS . "./Patterns.php";
    // Try
    try {
        $conn->beginTransaction();
        // Inputs
        $inEmail = $_POST['inEmail'];
        $inPass = $_POST['inPass'];

        // Input validation
        $valid = new Validation();
        $valid->name('Email')->value($inEmail)->customPattern($regEmail)-
>required();
        $valid->name('Password')->value($inPass)->customPattern($regPass)-
>required();

        $errs = [];
        $errCodes = [];
        $success = [];

        if ($valid->isSuccess()) {
            // Form is valid
```

```

$isEmail = isEmail($inEmail);
if ($isEmail) {
    // There is user with that email
    // Hash
    $passSha = hash("sha256", $inPass);
    $isUser = fetchUser($inEmail, $passSha);

    // Try getting the user
    if ($isUser) {
        $_SESSION['loggedUser'] = $isUser;
        if ($isUser->name_role == "admin") {
            // ADMIN logged in
            $success = "Admin logged in sucessfully";
        } else {
            // REGULAR USER logged in
            $isCartActive = isAvailable("cart", "id_user", $isUser->id_user);
            if (!$isCartActive) {
                // Successfull login, fetch the cart
                // Customer has items in cart
                $isCart = selectAllFromCartWhereId($isUser->id_user);
                if (empty($isCart)) {
                    // Customer logged in but failed to retreive data
                    // from cart
                    $errs[] = "Failed to retreive cart";
                    $errCodes[] = 422;
                } else {
                    $success[] = "Customer logged in and fetched cart
successfully";
                }
            } else {
                // Success
                // Customer doesn't have items in cart, success
                $success[] = "Customer logged in succesfully, no item
s in cart";
            }
        }
    } else {
        // We know the email is good but, we don't want to show that
        // We only log error to failed login in if we know that the u
ser
        // who is trying to login is registered
        $errs[] = "Invalid email, password or account is locked (fail
ed login)";
        $errCodes[] = 404;
    }
}

```

```

        // Log the failed attempt into database
        $isInsert = insertIntoFailedLogin($inEmail);
        if ($isInsert) {
            // Successfully inserterd into Failed Login
            $isLockCount = isLockUserCount($inEmail);

            $errs[] = "Tried to login $isLockCount times (last 5 min)";
        };
        $errCodes[] = 422;

        // User tried to login more than 5 times in last 5 mins
        // User will be locked
        if ((int)$isLockCount >= 5) {
            $isLockUser = lockUser($inEmail);
            if (!$isLockUser) {
                // Failed to lock the user
                $errs[] = "Failed to lock the user";
                $errCodes[] = 422;
            } else {
                $errs[] = "Your account has been locked";
                $errCodes[] = 422;
                $conn->commit();
                echo json_encode($errs, JSON_FORCE_OBJECT);
                http_response_code($errCodes[0]);
                die();
            }
        }

        $conn->commit();
        echo json_encode($errs, JSON_FORCE_OBJECT);
        http_response_code($errCodes[0]);
        die();
    } else {
        // Failed to insert into Failed Login table
        $errs[] = "Failed to insert into failed login";
        $errCodes[] = 422;
    }
}

} else {
    // There is no such user with that email
    // But here we know that there is no user with that email, so we
    // don't log error into failed_login
    $errs[] = "Invalid email or password";
    $errCodes[] = 404;
}

```

```

} else {
    // Validation failed
    // Unprocessable entity
    $errs[] = $valid->errors;
    $errCodes[] = 422;
}

// Rollback or commit
// Check if there were any errors
if (!empty($errs)) {
    $conn->rollback();
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code($errCodes[0]);
} else {
    $conn->commit();
    if (empty($isCart)) {
        echo json_encode(["msg" => $success, "user" => $isUser-
>name_role], JSON_FORCE_OBJECT);
    } else {
        echo json_encode(["msg" => $success, "user" => $isUser-
>name_role, "cart" => $isCart], JSON_FORCE_OBJECT);
    }
    http_response_code(200);
}
}

// Catch err
catch (PDOException $ex) {
    // Try block failed
    $errs[] = $ex->getMessage();
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code(422);
}
} else {
    // Unauthorized access
    $errs[] = "Unauthorized access";
    $errCodes[] = 403;
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code($errCodes[0]);
}

```

4.1.6 models/auth/Logout.php

```
<?php
if (!isset($_SESSION)) {
    session_start();
}
include "../../config/routes.php";
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST) && isset($_SESSION['loggedUser'])) {
    // Includes
    include MODELS . "/Helpers.php";
    include CONFIG . "/connection.php";
    include M_USERS . "/Users.php";

    // Check user role
    if ($_SESSION['loggedUser']->name_role == "admin") {
        // Admin wants to logout
        unset($_SESSION['loggedUser']);
        echo json_encode("Successfully logged out", JSON_FORCE_OBJECT);
        http_response_code(201);
    } else if ($_SESSION['loggedUser']->name_role == "regular_user") {
        // We sending appended order inside Form Data
        // We stringified array of objects so it could be sent
        // Now we need to decode it
        // Important: $_POST is an array so it can be empty and still set
        if (isset($_POST['order'])) {
            $order = json_decode($_POST['order']);

            // Insert order into cart
            try {
                // Begin transaction
                $conn->beginTransaction();

                $errs = [];
                $errCodes = [];
                $success = [];

                $isUserIdAvailable = isAvailable("cart", "id_user", $order[0]-
>inUserID);
                if ($isUserIdAvailable) {
                    // That user id is not in the cart table
                    // INSERT
                    $isInserted = insertUserCart($order);
                    if ($isInserted) {
                        // Commit and then logout user
                    }
                }
            } catch (Exception $e) {
                // Handle exception
            }
        }
    }
}
```

```

                $success[] = "Successfully inserted into cart";
            } else {
                $errs[] = "Failed to insert data into cart for user with
Id: {$order[0]->inUserID}";
                $errCodes[] = 422;
            }
        } else {
            // There are rows with that users id
            // DELETE, INSERT
            $isDeleted = deleteAllFromTableWhere("cart", "id_user", $orde
r[0]->inUserID);
            if ($isDeleted) {
                // Successfully deleted user data from cart
                $isInserted = insertUserCart($order);
                if ($isInserted) {
                    // Commit and then logout user
                    $success[] = "Successfully inserted into cart";
                } else {
                    $errs[] = "Failed to insert data into cart for user w
ith Id: {$order[0]->inUserID}";
                    $errCodes[] = 422;
                }
            } else {
                $errs[] = "Failed to delete user data from cart";
                $errCodes[] = 404;
            }
        }

        if (!empty($errs)) {
            $conn->rollback();
            // Log all of the errors
            echo json_encode($errs, JSON_FORCE_OBJECT);
            http_response_code($errCodes[0]);
        } else {
            // There were no errors, so we can commit
            $conn->commit();
            unset($_SESSION['loggedUser']);
            echo json_encode("Successfully added items to cart and logged
out", JSON_FORCE_OBJECT);
            http_response_code(201);
        }
    } catch (PDOException $ex) {
        // Error, some exception thrown
        $errs[] = $ex->getMessage();
        echo json_encode($errs, JSON_FORCE_OBJECT);
    }
}

```

```

        http_response_code(422);
    }
} else {
    // Post is an empty array, which means the cart is empty
    // Post can be empty so nothing is being inserted
    // $errs[] = "No appropriate data was sent";
    // Just because order is not set doesn't mean that there is nothing in cart for that user
    // display($_SESSION['loggedUser']->id_user);
    $isUserIdAvailable = isAvailable("cart", "id_user", $_SESSION['loggedUser']->id_user);
    if (!$isUserIdAvailable) {
        // User is NOT available, aka there are items in cart"
        // We emptied the Local storage so we need to update the cart
        $isDeleted = deleteAllFromTableWhere("cart", "id_user", $_SESSION['loggedUser']->id_user);
        if (!$isDeleted) {
            $errs[] = "Failed to delete user data from cart";
            $errCodes[] = 404;
        }
    }

    unset($_SESSION['loggedUser']);
    echo json_encode("Successfully logged out", JSON_FORCE_OBJECT);
    http_response_code(201);
}
}
} else {
    // Unauthorized access: Not POST or it's not User(admin/customer)
    $errs[] = "Unauthorized access";
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code(403);
}

```

4.1.7 models/auth/Register.php

```

<?php
header('Content-Type: application/json');
// Session
if (!isset($_SESSION)) {
    session_start();
}
// Includes

```

```

include_once "../config/routes.php";
include MODELS . "/Helpers.php";
// Form submitted
if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST["btnRegister"])) {
    // Includes
    include CONFIG . "/connection.php";
    include M_USERS . "/Users.php";
    include MODELS . "/Validation.php";
    include MODELS . "./Patterns.php";

    // Inputs
    $inEmail = $_POST["inEmail"];
    $inFname = $_POST["inFname"];
    $inLname = $_POST["inLname"];
    $inPass = $_POST["inPass"];
    $inPassConf = $_POST["inPassConf"];
    $inPhone = $_POST["inPhone"];
    $inAddr = $_POST["inAddr"];

    // Input validation
    $valid = new Validation();
    $valid->name('Email')->value($inEmail)->customPattern($regEmail)->required();
    $valid->name('First name')->value($inFname)->customPattern($regName)-
>required();
    $valid->name('Last name')->value($inLname)->customPattern($regName)-
>required();
    $valid->name('Password')->value($inPass)->customPattern($regPass)-
>required();
    $valid->name('Address')->value($inAddr)->customPattern($regAddress)-
>required();
    // $valid->name('Phone number')->value($inPhone)->customPattern($regPhone)-
>required();

    $errs = [];
    $errCodes = [];
    $success = [];

    // Check if passwords match
    $pass_match = false;
    if ($inPass == $inPassConf) {
        $pass_match = true;
    } else {
        $errs[] = "Passwords don't match";
        $errCodes[] = 422;
    }
}

```

```

// Check if there were any errors
if ($valid->isSuccess() && $pass_match) {
    // Form is valid
    // Try
    try {
        $conn->beginTransaction();
        // Check if email is available
        $table = 'user';
        $where = 'email_user';
        $isEmailAvailable = isAvailable($table, $where, $inEmail);
        if ($isEmailAvailable) {
            // Hash password
            $passSha = hash("sha256", $inPass);

            // Try to register new user
            $loggedUser = registerUser($inEmail, $passSha, $inFname, $inLname
, $inAddr, $inPhone);
            if ($loggedUser) {
                // User successfully registered
                $success = "You have registered successfully";
                // $resp = ["msg" => "You have registered successfully"];
                // echo json_encode(["msg" => "You have registered successfully",
$loggedUser->name_role], JSON_FORCE_OBJECT);
            } else {
                // Log error
                $errs[] = "Failed to register a user";
                $errCodes[] = 422;
            }
        } else {
            // Email is not available
            $errs[] = "Email is already in use";
            $errCodes[] = 409;
        }
    }

    // Rollback or commit
    // Check if there were any errors
    if (!empty($errs)) {
        $conn->rollback();
        echo json_encode($errs, JSON_FORCE_OBJECT);
        http_response_code($errCodes[0]);
    } else {
        $conn->commit();
        $_SESSION['loggedUser'] = $loggedUser;
        echo json_encode(["msg" => $success, "user" => $loggedUser-
>name_role], JSON_FORCE_OBJECT);
    }
}

```

```

        http_response_code(201);
    }
} catch (PDOException $ex) {
    // Try block failed
    $errs[] = $ex->getMessage();
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code(422);
}
} else {
    // Form is not valid
    // Validation failed
    // Unprocessable entity
    $errs[] = $valid->errors;
    $errCodes[] = 422;
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code($errCodes[0]);
}
} else {
    // Unauthorized access
    $errs[] = "Unauthorized access";
    $errCodes[] = 403;
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code($errCodes[0]);
}
}

```

4.1.8 models/public/SendMessage.php

```

<?php
header('Content-Type: application/json');
include "../../config/routes.php";
if ($_SERVER["REQUEST_METHOD"] == "POST" && !empty($_POST)) {

    // Includes
    include MODELS . "/Helpers.php";
    include CONFIG . "./connection.php";
    include MODELS . "/Validation.php";
    include MODELS . "./Patterns.php";

    // Variables
    $msg_first_name = isset($_POST['inFname']) ? $_POST['inFname'] : null;
    $msg_last_name = isset($_POST['inLname']) ? $_POST['inLname'] : null;
    $msg_email = isset($_POST['inEmail']) ? $_POST['inEmail'] : null;
    $msg_subject = isset($_POST['inSubject']) ? $_POST['inSubject'] : null;
    $msg_body = isset($_POST['taMessage']) ? $_POST['taMessage'] : null;
}

```

```

// Validation
$valid = new Validation();
$valid->name("First name")->value($msg_first_name)->customPattern($regName)->required();
$valid->name("Last name")->value($msg_last_name)->customPattern($regName)->required();
$valid->name("Email")->value($msg_email)->customPattern($regEmail)->required();
$valid->name("Subject")->value($msg_subject)->customPattern($regSubject)->required();
$valid->name("Message body")->value($msg_body)->required();

// Check input fields
if (!$valid->isSuccess()) {
    // Validation failed
    $errs[] = $valid->errors;
    $errCodes[] = 422;
    // Form validation failed (checkboxes, radio buttons, input fileds, textarea)
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code(422);
}

// Try block
try {
    //Begin Transaction
    $conn->beginTransaction();

    $isInserted = insertMessage($msg_first_name, $msg_last_name, $msg_email,
$msg_subject, $msg_body);

    if (!$isInserted) {
        $errs[] = "Failed to send a message";
        $errCodes[] = 422;
    }

    // Rollback or commit
    if (!empty($errs)) {
        // Rollback
        $conn->rollback();
        echo json_encode($errs, JSON_FORCE_OBJECT);
        http_response_code($errCodes[0]);
    } else {
        // Commit
    }
}

```

```

        $conn->commit();
        echo json_encode(["msg" => "Message successfully sent"], JSON_FORCE_OBJECT);
    }
    http_response_code(201);
}
} catch (PDOException $ex) {
    // Try block failed
    $errs[] = $ex->getMessage();
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code(422);
}
} else {
    $errs[] = "Unauthorized access";
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code(403);
}

```

4.1.9 models/users/Users.php

```

<?php
// Login user
function fetchUser($email, $pass)
{
    // Connection
    global $conn;
    // Query
    $query = "SELECT * FROM user u JOIN role r ON u.id_role = r.id_role WHERE u.email_user = ? AND u.pass_user = ? AND u.isLock = 0";
    // Prepared Statement
    $prep = $conn->prepare($query);
    // Result
    if ($prep->execute([$email, $pass])) {
        $res = $prep->fetch();
        return count((array)$res) == 0 ? null : $res;
    }
}
// Register user
function registerUser($email, $passSha, $fname, $lname, $address, $phone)
{
    global $conn;

```

```

$query = "INSERT INTO user (email_user, pass_user, first_name, last_name, address, phone_num) VALUES(?, ?, ?, ?, ?, ?)";
$prep = $conn->prepare($query);
if ($prep->execute([$email, $passSha, $fname, $lname, $address, $phone])) {
    $id = $conn->lastInsertId();
    $query = "SELECT u.id_user, u.email_user, u.pass_user, u.id_role, u.first_name, u.last_name, u.address, u.phone_num, r.name_role FROM user u INNER JOIN role r ON u.id_role = r.id_role WHERE id_user = ?";
    $prep = $conn->prepare($query);
    if ($prep->execute([$id])) {
        return $prep->fetch();
    }
} else {
    return false;
}
// Insert user's cart into cart table
function insertUserCart($order)
{
    global $conn;
    $errs = [];

    foreach ($order as $i => $o) {
        $id_user = $o->inUserID;
        $id_glove = $o->inGloveID;
        $id_size = $o->raSize;
        $name_glove = $o->inGloveName;
        $value_size = $o->raSizeName;
        $name_measure = $o->inMeasure;
        $id_brand = $o->inBrandID;
        $name_brand = $o->inBrandName;
        $id_cat = $o->inCatID;
        $name_cat = $o->inCatName;
        $id_color = $o->inColorID;
        $color_name = $o->inColor;
        $t_img = $o->inImg;
        $price = $o->inPrice;
        $quantity = $o->inQuantity;

        $query = "INSERT INTO cart(id_user, id_glove, id_size, name_glove, value_size, name_measure, id_brand, name_brand, id_cat, name_cat, id_color, color_name, t_img, price, quantity) VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
        $prep = $conn->prepare($query);
        if (!$prep->execute([$id_user, $id_glove, $id_size, $value_size, $name_glove, $name_measure,

```

```

        $id_brand, $name_brand, $id_cat, $name_cat, $id_color, $color_name, $t_img, $price,
        $quantity])) {
            return true;
        }
    }

    if (empty($errs)) {
        return true;
    } else {
        return false;
    }
}

// Select all from cart WHERE
function selectAllFromCartWhereId($id)
{
    global $conn;
    $query = "SELECT id_user as inUserID, id_glove as inGloveID, id_size as raSize,
    name_glove as inGloveName, value_size as raSizeName, name_measure as inMeasure,
    id_brand as inBrandID, name_brand as inBrandName, id_cat as inCatID, name_cat as
    inCatName, id_color as inColorID, color_name as inColor, t_img as inImg, price
    as inPrice, quantity as inQuantity FROM cart WHERE id_user = ?";
    $prep = $conn->prepare($query);
    if ($prep->execute([$id])) {
        return $prep->fetchAll();
    }
}

// Insert into failed login
function insertIntoFailedLogin($email_user)
{
    global $conn;

    $query = "INSERT INTO failed_login(email_user) VALUES(?)";
    $prep = $conn->prepare($query);
    if ($prep->execute([$email_user])) {
        return true;
    }
}

// Lock user
function lockUser($email_user)
{
    global $conn;
    $query = "UPDATE user SET isLock=1 WHERE email_user = ? ";
    $prep = $conn->prepare($query);
}

```

```

    if ($prep->execute([$email_user])) {
        return true;
    }
}

// Unlock user
function unlockUser($id_user)
{
    global $conn;
    $query = "UPDATE user SET isLock=0 WHERE id_user = ? ";
    $prep = $conn->prepare($query);
    if ($prep->execute([$id_user])) {
        return true;
    }
}

// Delete login failed data for user
function deleteLoginFailedData($id_user, $email_user)
{
    global $conn;

    $query = "DELETE FROM failed_login WHERE email_user = ?";
    $prep = $conn->prepare($query);

    if ($prep->execute([$email_user])) {
        return true;
    }
}

```

4.1.10 models/users/admin/Admin.php

```

<?php
// INSERT/UPDATE/DELTE functions
// (everyone can read(some) but not everyone can execute(only admin))

// Insert glove in the Glove table
function insertGlove($cat, $brand, $color, $name, $desc)
{
    global $conn;
    $query = "INSERT INTO glove (id_cat, id_brand, id_color, name_glove, desc_glo
ve) VALUES (?, ?, ?, ?, ?)";
    $prep = $conn->prepare($query);

```

```

    if ($prep->execute([$cat, $brand, $color, $name, $desc])) {
        return $conn->lastInsertId();
    }
}

// Update glove in the Glove table
function updateGlove($id, $cat, $brand, $color, $name, $desc)
{
    global $conn;
    $query = "UPDATE glove SET id_cat=? , id_brand=? , id_color=? , name_glove=? , de sc_glove=? WHERE id_glove = ?";
    $prep = $conn->prepare($query);
    return $prep->execute([$cat, $brand, $color, $name, $desc, $id]);
}

// Insert sizes selected in checkboxes
function insertGloveSizes($sizes, $glove_id)
{
    global $conn;
    $query = "INSERT INTO glove_size (id_size, id_glove) VALUES(?, ?)";
    $prep = $conn->prepare($query);
    foreach ($sizes as $i => $s) {
        if (!$prep->execute([$s, $glove_id])) {
            return false;
        }
    }
    return true;
}

function deleteGloveSizes($id)
{
    global $conn;
    $time = date("Y-m-d H:i:s");
    $query = "UPDATE glove_size SET date_remove='$time' WHERE id_glove = ? AND da te_remove IS NULL";
    // display($query);
    $prep = $conn->prepare($query);
    return $prep->execute([$id]);
}

// function deleteGloveSizes($id)
// {
//     global $conn;
//     $query = "DELETE FROM glove_size WHERE id_glove = ?";
//     $prep = $conn->prepare($query);
//     return $prep->execute([$id]);
// }

// Insert glove images
function insertImages($normal_img, $thumb_img, $glove_id)

```

```

{
    global $conn;
    $query = "INSERT INTO image(normal_img, thumb_img, id_glove) VALUES(?, ?, ?)"
;
    $prep = $conn->prepare($query);
    return $prep->execute([$normal_img, $thumb_img, $glove_id]);
}
// Update glove images
function updateImages($normal_img, $thumb_img, $glove_id)
{
    global $conn;
    $query = "UPDATE image (id_size, id_glove) VALUES(?, ?)";
    $query = "UPDATE image SET normal_img=? , thumb_img=? WHERE id_glove = ?";
    $prep = $conn->prepare($query);
    return $prep->execute([$normal_img, $thumb_img, $glove_id]);
}
// Insert glove price
function insertGlovePrice($price, $id)
{
    global $conn;
    $query = "INSERT INTO price(id_glove, price) VALUES(?, ?)";
    $prep = $conn->prepare($query);
    return $prep->execute([$id, $price]);
}
// Update glove price
function updateGlovePrice($price, $id)
{
    global $conn;
    $query = "UPDATE price SET price=? WHERE id_glove = ?";
    $prep = $conn->prepare($query);
    return $prep->execute([$price, $id]);
}
// Remove glove from Database (SET date_removed)
function removeGlove($id) {
    global $conn;
    $time = date('Y-m-d H:i:s','1299762201428');
    $query = "UPDATE glove SET date_removed='$time' WHERE id_glove=?";
    $prep = $conn->prepare($query);
    if($prep->execute([$id])) {
        return true;
    }
}
// Selects all orders, displays it in admin-orders
function selectAllOrders() {
    global $conn;

```

```

        $query = "SELECT u.first_name, u.last_name, SUM(ol.price_purchase * ol.quantity) as total, COUNT(ol.id_order_line) FROM user u INNER JOIN order_detail od ON u.id_user = od.id_user INNER JOIN order_line ol ON od.id_order_detail = ol.id_order_detail GROUP BY od.id_order_detail";
        return $conn->query($query)->fetchAll();
    }

// Checks if data is available(no duplicates)
function isUserID($id_user)
{
    global $conn;
    // Table/Column name cannot be parameterized
    // display($email);
    $query = "SELECT * FROM user WHERE id_user = ?";
    $prep = $conn->prepare($query);
    // display($query);
    // display($conn->query($query)->fetch());

    // display($query);

    if ($prep->execute([$id_user])) {
        // Counts the number of rows
        // display($prep->fetch());
        return $prep->fetch();
    }
}

// Update sizes selected in checkboxes
// function updateGloveSizes($sizes, $glove_id)
// {
//     global $conn;
//     $query = "UPDATE glove_size SET id_size=? WHERE id_glove = ?";
//     $prep = $conn->prepare($query);
//     foreach ($sizes as $i => $s) {
//         if (!$prep->execute([$s, $glove_id, $glove_id])) {
//             return false;
//         }
//     }
//     return true;
// }

```

4.1.11 models/users/admin/gloves/ActiveGloves.php

```
<?php
```

```
include "../../config/routes.php";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Includes
    include_once MODELS . "/Helpers.php";
    include_once CONFIG . "/connection.php";
    include_once "../Admin.php";

    $errs = [];
    $errCodes = [];

    try {
        // Begin transaction
        $conn->beginTransaction();

        // Fetch all active gloves
        $gloves = selectAllActiveGloves();
        if (!count($gloves) > 0) {
            // Failed to retrieve sizes
            $errs[] = "Failed to retrieve gloves";
            $errCodes[] = 404;
        } else {
            $success = "Gloves fetched successfully";
        }

        // Rollback or commit
        if (!empty($errs)) {
            // Rollback
            $conn->rollback();
            echo json_encode($errs, JSON_FORCE_OBJECT);
            http_response_code($errCodes[0]);
        } else {
            // Commit
            $conn->commit();
            echo json_encode(["msg" => $success, "data" => $gloves], JSON_FORCE_OBJECT);
            http_response_code(201);
        }
    } catch (PDOException $ex) {
        // Try block failed
        $errs[] = $ex->getMessage();
        echo json_encode($errs, JSON_FORCE_OBJECT);
        http_response_code(422);
    }
} else {
    $errs[] = "Unauthorized access";
}
```

```
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code(403);
}
```

4.1.12 models/users/admin/gloves/Category.php

```
<?php
header('Content-Type: application/json');
include "../../config/routes.php";
if ($_SERVER["REQUEST_METHOD"] == "POST") && isset($_POST)) {
    // Includes
    include MODELS . "/Helpers.php";
    include CONFIG . "/connection.php";
    include "../Admin.php";

    // Variables
    $id_cat = $_POST['cat'];

    try {
        // Begin transaction
        $conn->beginTransaction();

        // Select available sizes
        $sizes = selectAvailableSizes($id_cat);
        if (!count($sizes) > 0) {
            // Failed to retrieve sizes
            $errs[] = "Failed to retrieve sizes";
            $errCodes[] = 404;
        } else {
            $success = "Sizes fetched successfully";
        }

        // Rollback or commit
        if (!empty($errs)) {
            // Rollback
            $conn->rollback();
            echo json_encode($errs, JSON_FORCE_OBJECT);
            http_response_code($errCodes[0]);
        } else {
            // Commit
            $conn->commit();
            echo json_encode(["msg" => $success, "data" => $sizes], JSON_FORCE_OBJECT);
            http_response_code(201);
        }
    } catch (Exception $e) {
        $errs[] = $e->getMessage();
        $errCodes[] = 500;
    }
}
```

```

        }
    } catch (PDOException $ex) {
        // Try block failed
        $errs[] = $ex->getMessage();
        echo json_encode($errs, JSON_FORCE_OBJECT);
        http_response_code(422);
    }
} else {
    $errs[] = "Unauthorized access";
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code(403);
}

```

4.1.13 models/users/admin/gloves/Edit.php

```

<?php
header('Content-Type: application/json');
include "../../config/routes.php";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Includes
    include MODELS . "/Helpers.php";
    include CONFIG . "./connection.php";
    include MODELS . "/Validation.php";
    include MODELS . "./Patterns.php";
    include "../Admin.php";

    // Variables
    $brand = $_POST['ddlBrand'];
    $cat = $_POST['ddlCat'];
    $color = $_POST['ddlColor'];
    $sizes = isset($_POST['cbSize']) ? $_POST['cbSize'] : null;
    $name = $_POST['inName'];
    $price = $_POST['inPrice'];
    $desc = $_POST['taDesc'];
    $id = $_POST['inHidden'];

    $errs = [];
    $errCodes = [];
    $success;

    if (isset($_FILES['fileImg']) && $_FILES['fileImg']['error'] == 0) {
        // Image wasn't uploaded

```

```

    $img = $_FILES['fileImg'];
}

// // Validation
$valid = new Validation();
$valid->name("Glove name")->value($name)->customPattern($regGloveName)->required();
$valid->name("Price")->value($price)->customPattern($regPrice)->required();
$valid->name("Description")->value($desc)->required();

if (!isset($brand) || empty($brand)) {
    // Brand must be selected
    $errs[] = "Must select a brand";
    $errCodes[] = 422;
}
if (!isset($cat) || empty($cat)) {
    // Category must be selected
    $errs[] = "Must select a category";
    $errCodes[] = 422;
}
if (!isset($color) || empty($color)) {
    // Color must be selected
    $errs[] = "Must select a color";
    $errCodes[] = 422;
}

// Check if dropdowns/sizes/form are all valid
if (!empty($errs)) {
    // Errors array is not empty, not all drop downs were selected
    $errs[] = "Every dropdown must be selected";
    $errCodes[] = 422;
}
if (!isset($sizes) || empty($sizes)) {
    // Size must be selected
    $errs[] = "At least one size must be selected";
    $errCodes[] = 422;
}
if (!$valid->isSuccess()) {
    // Validation failed
    // Unprocessable entity
    $errs[] = $valid->errors;
    $errCodes[] = 422;
}

```

```

if (empty($errs)) {
    // Validation success
    // Try
    try {
        //Begin Transaction
        $conn->beginTransaction();

        if (isset($img)) {
            // Image was uploaded
            // Image parts
            // This is checked only if image is set, because image is OPTIONAL
            // by default
            $n_name = time() . "__" . $img['name'];
            $n_type = $img['type'];
            $n_tmp = $img['tmp_name'];
            $n_size = $img['size'];

            // Allowed types
            $allowed_types = ['image/jpg', 'image/jpeg', 'image/png'];

            // Check for image errors
            if (!in_array($n_type, $allowed_types)) {
                $errs[] = "Image type is not valid";
                $errCodes[] = 422;
            }
            if ($n_size > 3000000) {
                $errs[] = "Uploaded image is too large";
                $errCodes[] = 422;
            }
        }
    }

    if (!count($errs)) {

        if (isset($img)) {
            // Image was uploaded, OPTIONAL
            // Creating thumbnail image
            list($n_w, $n_h) = getimagesize($n_tmp);
            $n_img = imagecreatefromstring(file_get_contents($n_tmp));
            $t_w = 150;
            $t_h = ($t_w / $n_w) * $n_h;
            $t_img = imagecreatetruecolor($t_w, $t_h);
            imagecopyresampled($t_img, $n_img, 0, 0, 0, 0, $t_w, $t_h, $n_w, $n_h);
            // Definig name/path

```

```

$t_name = "thumb_" . $n_name;
$t_path = "assets/img/gloves/thumbnail/{$t_name}";

// Upload thumbnail img
switch ($n_type) {
    case "image/png":
        imagepng($t_img, '../../../' . $t_path);
        break;
    default:
        imagejpeg($t_img, '../../../' . $t_path);
}

// Upload normal img
$n_path = "assets/img/gloves/normal/" . $n_name;

if (isset($img)) {
    $isUploaded = move_uploaded_file($n_tmp, '../../../' .
$n_path);
    if (!$isUploaded) {
        // Failed to upload image
        $errs[] = "Failed to upload the image";
        $errCodes[] = 422;
    }
}
}

// Is there a active glove with that id
$isGlove = isGloveAndActive($id);
if (!$isGlove) {
    $errs[] = "There is no active glove with that ID";
    $errCodes[] = 404;
} else {
    $glove = selectCurrentGloveValues($id);
    $removeID = $glove->id;
    $isGloveRemove = removeGlove($removeID);
    if (!$isGloveRemove) {
        // Failed to remove previous glove (DATE REMOVED to NULL)
        $errs[] = "Failed to insert glove";
        $errCodes[] = 422;
    } else {
        if (!isset($img)) {
            // Image is not set so we need to get current image v
alues
            $n_name = $glove->n_img;
        }
    }
}
}

```

```

        $t_name = $glove->t_img;
    }

    // Get the id of the newly inserted image
    $lastInsertedId = insertGlove($cat, $brand, $color, $name
, $desc);
    if (!$lastInsertedId) {
        // Couldn't get the lastInsertedId
        // Glove was not inserted
        $errs[] = "Failed to insert glove";
        $errCodes[] = 422;
    } else {
        // Got the ID of the inserted Glove
        $isSizeInserted = insertGloveSizes($sizes, $lastInser
tedId);
        if (!$isSizeInserted) {
            // Failed to insert sizes for glove
            $errs[] = "Failed to insert glove sizes";
            $errCodes[] = 422;
        } else {
            // Inserted sizes for glove
            $isPriceInserted = insertGlovePrice($price, $last
InsertedId);
            if (!$isPriceInserted) {
                // Failed to insert price for glove
                $errs[] = "Failed to insert glove price";
                $errCodes[] = 422;
            } else {
                // Price was inserted
                $isImageInsert = insertImages($n_name, $t_nam
e, $lastInsertedId);
                if (!$isImageInsert) {
                    // Failed to insert image
                    $errs[] = "Failed to insert image";
                    $errCodes[] = 422;
                } else {
                    // Image was inserted
                    // FINNISHED INSETING PRODUCT
                    $success = "Glove updated succesfully";
                }
            }
        }
    }
}

```

```

    }

    // Rollback or Commit
    if (!empty($errs)) {
        // Rollback
        $conn->rollback();
        echo json_encode($errs, JSON_FORCE_OBJECT);
        http_response_code($errCodes[0]);
    } else {
        // Commit
        $conn->commit();
        if (isset($t_img)) {
            // Send thumbnail image name, if exists
            echo json_encode(['msg' => $success, "img" => $t_name, 'id' =
> $lastInsertedId], JSON_FORCE_OBJECT);
        } else {
            // Image was not uploaded but is success
            echo json_encode(['msg' => $success, 'id' => $lastInsertedId]
, JSON_FORCE_OBJECT);
        }
        http_response_code(201);
    }
}

// Catch
catch (PDOException $ex) {
    // Try block failed
    $errs[] = $ex->getMessage();
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code(422);
}
} else {
    // Form validation failed (checkboxes, radio buttons, input fileds, text
area)
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code(422);
}
} else {
    // Unauthorized access
    $errs[] = "Unauthorized access";
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code(403);
}

```

4.1.14 models/users/admin/gloves/Insert.php

```
<?php
header('Content-Type: application/json');
include "../../config/routes.php";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Includes
    include MODELS . "/Helpers.php";
    include CONFIG . "./connection.php";
    include MODELS . "/Validation.php";
    include MODELS . "./Patterns.php";
    include "../Admin.php";

    // Variables
    $brand = $_POST['ddlBrand'];
    $cat = $_POST['ddlCat'];
    $color = $_POST['ddlColor'];
    $sizes = isset($_POST['cbSize']) ? $_POST['cbSize'] : null;
    $name = $_POST['inName'];
    $price = $_POST['inPrice'];
    $desc = $_POST['taDesc'];
    $img = $_FILES['fileImg'];

    // Validation
    $valid = new Validation();
    $valid->name("Glove name")->value($name)->customPattern($regGloveName)->required();
    $valid->name("Price")->value($price)->customPattern($regPrice)->required();
    $valid->name("Description")->value($desc)->required();

    // Errors array
    $errs = [];
    $errCodes = [];

    // Check dropdowns and checkboxes
    if (!isset($brand) || empty($brand)) {
        $errs[] = "Must select a brand";
        $errCodes[] = 422;
    };
    if (!isset($cat) || empty($cat)) {
        $errs[] = "Must select a category";
        $errCodes[] = 422;
    };
    if (!isset($color) || empty($color)) {
        $errs[] = "Must select a color";
    };
}
```

```

        $errCodes[] = 422;
    };
    if (empty($sizes)) {
        $errs[] = "Must select at least one size";
        $errCodes[] = 422;
    }
    // Check input fields
    if (!$valid->isSuccess()) {
        // Validation failed
        $errs[] = $valid->errors;
        $errCodes[] = 422;
    }

    if (empty($errs)) {
        // Validation success, errors array is empty
        // Try
        try {
            //Begin Transaction
            $conn->beginTransaction();

            // Image parts
            // Image is required
            $n_name = time() . "__" . $img['name'];
            $n_type = $img['type'];
            $n_tmp = $img['tmp_name'];
            $n_size = $img['size'];

            // Allowed types
            $allowed_types = ['image/jpg', 'image/jpeg', 'image/png'];

            // Check for image errors
            if (!in_array($n_type, $allowed_types)) {
                $errs[] = "Image type is not valid";
                $errCodes[] = 422;
            }
            if ($n_size > 3000000) {
                $errs[] = "Uploaded image is too large";
                $errCodes[] = 422;
            }

            if (!count($errs)) {
                // Creating thumbnail image
                list($n_w, $n_h) = getimagesize($n_tmp);
                $n_img = imagecreatefromstring(file_get_contents($n_tmp));
                $t_w = 150;

```

```

$t_h = ($t_w / $n_w) * $n_h;
$t_img = imagecreatetruecolor($t_w, $t_h);
imagecopyresampled($t_img, $n_img, 0, 0, 0, 0, $t_w, $t_h, $n_w,
$n_h);

// Definig name/path
$t_name = "thumb__" . $n_name;
$t_path = "assets/img/gloves/thumbnail/{$t_name}";

// Upload thumbnail img
switch ($n_type) {
    case "image/png":
        imagepng($t_img, '../../../../../' . $t_path);
        break;
    default:
        imagejpeg($t_img, '../../../../../' . $t_path);
}

// Upload normal img
$n_path = "assets/img/gloves/normal/" . $n_name;
if (move_uploaded_file($n_tmp, '../../../../../' . $n_path)) {
    // Image uploaded successfully
    $lastInsertedId = insertGlove($cat, $brand, $color, $name, $d
esc);
    if ($lastInsertedId) {
        // Glove is inserted
        $isInserted = insertGloveSizes($sizes, $lastInsertedId);
        if ($isInserted) {
            $isInserted = insertGlovePrice($price, $lastInsertedI
d);
            if ($isInserted) {
                // Price is inserted
                $isInserted = insertImages($n_name, $t_name, $la
tInsertedId);
                if ($isInserted) {
                    // Image is inserted
                    $success = "Glove addedd succesfully";
                } else {
                    $errs[] = "Failed to insert image";
                    $errCodes[] = 422;
                }
            } else {
                $errs[] = "Failed to insert glove price";
                $errCodes[] = 422;
            }
        } else {
            $errs[] = "Failed to insert image";
            $errCodes[] = 422;
        }
    } else {
        $errs[] = "Failed to insert glove price";
        $errCodes[] = 422;
    }
} else {

```

```

        $errs[] = "Failed to insert glove sizes";
        $errCodes[] = 422;
    }
} else {
    // Couldn't get the lastInsertedId
    // Glove was not inserted
    $errs[] = "Failed to insert glove";
    $errCodes[] = 422;
}
}

// Rollback or commit
if (!empty($errs)) {
    // Rollback
    $conn->rollback();
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code($errCodes[0]);
} else {
    // Commit
    $conn->commit();
    echo json_encode(["msg" => $success], JSON_FORCE_OBJECT);
    http_response_code(201);
}
}

// Catch
catch (PDOException $ex) {
    // Try block failed
    $errs[] = $ex->getMessage();
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code(422);
}
} else {
    // Form validation failed (checkboxes, radio buttons, input fileds, text area)
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code(422);
}
} else {
    $errs[] = "Unauthorized access";
    echo json_encode($errs, JSON_FORCE_OBJECT);
}

```

```
        http_response_code(403);
}
```

4.1.15 models/users/admin/gloves/Remove.php

```
<?php
header('Content-Type: application/json');
include "../../config/routes.php";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (isset($_POST['id'])) {
        // Id was set

        // Includes
        include MODELS . "/Helpers.php";
        include CONFIG . "./connection.php";
        include "../Admin.php";

        // Variables
        $id = $_POST['id'];
        $errs = [];
        $errCodes = [];

        try {
            //Begin Transaction
            $conn->beginTransaction();

            // Checks if there is a glove with that id
            $isGloveAndActive = isGloveAndActive($id);
            if (!$isGloveAndActive) {
                $errs[] = "There is no glove with that id";
                $errCodes[] = 404;
            } else {
                // Checks if the glove with that id is removed
                $isRemoved = removeGlove($id);
                if (!$isRemoved) {
                    $errs[] = "Failed to remove glove";
                    $errCodes[] = 422;
                } else {
                    $success = "Glove removed successfully";
                }
            }
        }
    }
}
```

```

    // Rollback or commit
    if (!empty($errs)) {
        // Rollback
        $conn->rollback();
        echo json_encode($errs, JSON_FORCE_OBJECT);
        http_response_code($errCodes[0]);
    } else {
        // Commit
        $conn->commit();
        echo json_encode(["msg" => $success], JSON_FORCE_OBJECT);
        http_response_code(201);
    }
} catch (PDOException $ex) {
    // Try block failed
    $resp = ["msg" => $ex->getMessage()];
    echo json_encode($resp, JSON_FORCE_OBJECT);
    http_response_code(422);
}
} else {
    // Id was not sent
    $errs[] = "Id was not sent";
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code(422);
}
} else {
    // Unauthorized access
    $errs[] = "Method is not POST which means you are a Terrorist";
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code(403);
}
}

```

4.1.16 models/users/admin/orders/Display.php

```

<?php
include "../../config/routes.php";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Includes
    include_once MODELS . "/Helpers.php";
    include_once CONFIG . "/connection.php";
    include_once "../Admin.php";

    $errs = [];

```

```

$errCodes = [];

try {
    // Begin transaction
    $conn->beginTransaction();

    // Fetch all active gloves
    $orders = selectAllOrders();
    display($orders);
    // if (!count($gloves) > 0) {
    //     // Failed to retrieve sizes
    //     $errs[] = "Failed to retrieve gloves";
    //     $errCodes[] = 404;
    // } else {
    //     $success = "Gloves fetched successfully";
    // }

    // // Rollback or commit
    // if (!empty($errs)) {
    //     // Rollback
    //     $conn->rollback();
    //     echo json_encode($errs, JSON_FORCE_OBJECT);
    //     http_response_code($errCodes[0]);
    // } else {
    //     // Commit
    //     $conn->commit();
    //     echo json_encode(["msg" => $success, "data" => $gloves], JSON_FORCE_OBJECT);
    //     http_response_code(201);
    // }
} catch (PDOException $ex) {
    // Try block failed
    $errs[] = $ex->getMessage();
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code(422);
}
} else {
    $errs[] = "Unauthorized access";
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code(403);
}

```

4.1.17 models/users/admin/orders/Excel.php

```
<?php
include "../../config/routes.php";
// Includes
include_once MODELS . "/Helpers.php";
include_once CONFIG . "/connection.php";
include_once "../Admin.php";

$orderID = isset($_GET['id']) ? $_GET['id'] : "";
if (!$orderID) {
    header("Location: index.php?page=index");
}

$isOrder = returnOrderDetailInfo($orderID);
if (!isset($isOrder) || empty($isOrder[0])) {
    header("Location: index.php?page=index");
}

list($order_lines, $u, $count) = $isOrder;
$order_lines = (array)$order_lines;
$u = (array)$u;
list($first_name, $last_name, $email_user, $id_user) = array_values($u);

$excel = "<div style='padding: 30px 50px'>";
$excel .= "<h3>Full name: ".$first_name. " " . $last_name ."</h3>";
$excel .= "<h3>Email: ".$email_user. "</h3>";
$excel .= "<h3>Order ID: ". $orderID . "</h3>";
$excel .= "<table style='width: 80%; border: 1px solid; margin: 0 auto;'>
<thead>
<tr>
    <td style='padding: 5px 8px'>Glove ID</td>
    <td style='padding: 5px 8px'>Name</td>
    <td style='padding: 5px 8px'>Category</td>
    <td style='padding: 5px 8px'>Brand</td>
    <td style='padding: 5px 8px'>Color</td>
    <td style='padding: 5px 8px'>Size</td>
    <td style='padding: 5px 8px'>Price</td>
    <td style='padding: 5px 8px'>Quantity</td>
    <td style='padding: 5px 8px'>Total</td>
</tr>
</thead>
<tbody>
```

```

";
foreach ($order_lines as $i => $ol) {
    $ol = (array)$ol;
    list($date, $name_glove, $id_glove, $name_brand, $name_cat, $price, $color_name, $img, $size, $measure, $quantity) = array_values($ol);
    $total = $quantity * $price;
    $size = $size . " " . ($measure == 'OZ' ? $measure : '');
    $excel .= "<tr>
        <td style='border: 1px solid; padding: 5px 8px'>" . $id_glove . "</td>
        <td style='border: 1px solid; padding: 5px 8px'>" . $name_glove . "</td>
        <td style='border: 1px solid; padding: 5px 8px'>" . $name_cat . "</td>
        <td style='border: 1px solid; padding: 5px 8px'>" . $name_brand . "</td>
        <td style='border: 1px solid; padding: 5px 8px'>" . $color_name . "</td>
        <td style='border: 1px solid; padding: 5px 8px'>" . $size . "</td>
        <td style='border: 1px solid; padding: 5px 8px'>" . $price . "</td>
        <td style='border: 1px solid; padding: 5px 8px'>" . $quantity . "</td>
        <td style='border: 1px solid; padding: 5px 8px'>" . $total . "</td>
    </tr>";
}
$excel .= "</tbody>
</table></div>";
header("Content-type: application/vnd.ms-excel");
header("Content-Disposition: attachment; filename=order-detail.xls");
display($excel);

```

4.1.18 models/users/admin/users/Unlcok.php

```

<?php
header('Content-Type: application/json');
include "../../config/routes.php";
if ($_SERVER["REQUEST_METHOD"] == "POST") {

    // Includes
    include MODELS . "/Helpers.php";
    include CONFIG . "./connection.php";
    include MODELS . "/Validation.php";
    include MODELS . "./Patterns.php";
    include "../Admin.php";
    include "../../Users.php";
    // display($_POST);

    $id_user = $_POST['userID'];

```

```

// $id_user = "666";

$errors = [];
$errCodes = [];
$success;

try {
    //Begin Transaction
    $conn->beginTransaction();

    // Check if it is registered user
    $isUser = isUserID($id_user);
    if ($isUser) {
        // User is registered
        $email_user = $isUser->email_user;
        $isDelete = deleteLoginFailedData($id_user, $email_user);

        if (!$isDelete) {
            // No such user
            $errors[] = "Failed to remove failed login data";
            $errCodes[] = 422;
        } else {
            // Update isLock column in user
            $isUpdate = unlockUser($id_user);
            if (!$isUpdate) {
                $errors[] = "Failed to unlock user";
                $errCodes[] = 422;
            } else {
                $success = "User successfully unlocked";
            }
        }
    } else {
        // No such user
        $errors[] = "There is no such user";
        $errCodes[] = 404;
    }
}

// Rollback or Commit
if (!empty($errors)) {
    // Rollback
    $conn->rollback();
    echo json_encode($errors, JSON_FORCE_OBJECT);
    http_response_code($errCodes[0]);
} else {

```

```

        // Commit
        $conn->commit();
        if (isset($t_img)) {
            // Send thumbnail image name, if exists
            echo json_encode(['msg' => $success, "img" => $t_name], JSON_FORCE_OBJECT);
        } else {
            // Image was not uploaded but is success
            echo json_encode(['msg' => $success], JSON_FORCE_OBJECT);
        }
        http_response_code(201);
    }
} catch (PDOException $ex) {
    // Try block failed
    $errs[] = $ex->getMessage();
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code(422);
}
} else {
    // Unauthorized access
    $errs[] = "Unauthorized access";
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code(403);
}

```

4.1.19 models/users/customers/Customer.php

```

<?php
// Insert new Order Detail
function insertOrderDetail($id)
{
    global $conn;

    $query = "INSERT INTO order_detail(id_user) VALUES(?)";
    $prep = $conn->prepare($query);
    if ($prep->execute([$id])) {
        return $conn->lastInsertId();
    }
}
// Return glove size from Glove size table
function returnGloveSize($id_size, $id_glove)
{

```

```

global $conn;

$query = "SELECT id_glove_size FROM glove_size WHERE id_size = ? AND id_glove
= ?";
$prep = $conn->prepare($query);
if ($prep->execute([$id_size, $id_glove])) {
    return $prep->fetch()->id_glove_size;
}
}

// Insert new Order Line
function insertOrderLine($id_order_detail, $id_glove_size, $price_purchase, $quantity)
{
    global $conn;

    $query = "INSERT INTO order_line(id_order_detail, id_glove_size, price_purchase,
    quantity) VALUES(?, ?, ?, ?)";
    $prep = $conn->prepare($query);
    if ($prep-
>execute([$id_order_detail, $id_glove_size, $price_purchase, $quantity])) {
        return true;
    }
}

```

4.1.20 models/users/customers/Purchase.php

```

<?php
header('Content-Type: application/json');
// Session
if (!isset($_SESSION)) {
    session_start();
}
// Includes (routes, helpers)
include_once "../../config/routes.php";
include MODELS . "./Helpers.php";
include CONFIG . "./connection.php";
include "./Customer.php";

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    if (isset($_POST)) {
        // Parse json data, and get array of objects

```

```

$data = json_decode($_POST['order']);

$errors = [];
$errCodes = [];

// $errors[] = 1;

try {
    $conn->beginTransaction();
    // Check if data is empty
    if (empty($data)) {
        $errors[] = "Data is empty";
        $errCodes[] = 422;
    } else {
        // Try to Insert new order detail
        $isOrderDetailId = insertOrderDetail($data[0]->id_user);
        if (!$isOrderDetailId) {
            // Failed to insert new Order detail
            $errors[] = "Failed to insert new Order detail";
            $errCodes[] = 422;
        } else {
            // Try to get glove_size from gloveSize table
            foreach ($data as $i => $o) {
                $isGloveSizeId = returnGloveSize($o->id_size, $o->id_glove);
                if (!$isGloveSizeId) {
                    // Invalid glove size combination
                    $errorsGloveSize[] = "Invalid glove size combination: Size: " . $o->name_size . " " . $o->measure . " and Glove: " . $o->name_glove;
                } else {
                    // Current glove size id is valid
                    $isInserted = insertOrderLine($isOrderDetailId, $isGloveSizeId, $o->price_purchase, $o->quantity);
                    if (!$isInserted) {
                        // Failed to insert new Order Line
                        $errorsOrderLine[] = "Failed to insert new order line for: Glove: " . $o->name_glove . " with size: " . $o->name_size . " " . $o->measure;
                    }
                }
            }
        }
    }
}

```

```

    if (isset($errsGloveSize) && !empty($errsGloveSize)) {
        $errs[] = $errsGloveSize;
        $errCodes[] = 404;
    }
    if (isset($errsOrderLine) && !empty($errsOrderLine)) {
        $errs[] = $errsOrderLine;
        $errCodes[] = 422;
    }

    // Rollback or Commit
    if (!empty($errs)) {
        // Rollback
        $conn->rollback();
        echo json_encode($errs, JSON_FORCE_OBJECT);
        http_response_code($errCodes[0]);
    } else {
        // Commit
        $conn->commit();
        // $conn->rollback();
        // Order and Order detail were inserted
        $success = "Order inserted successfully";
        echo json_encode(['msg' => $success], JSON_FORCE_OBJECT);
        http_response_code(201);
    }
} catch (PDOException $ex) {
    // Error, some exception thrown
    $errs[] = $ex->getMessage();
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code(422);
}
}

} else {
    // Unauthorized access
    $errs[] = "Unauthorized access";
    $errCodes[] = 403;
    echo json_encode($errs, JSON_FORCE_OBJECT);
    http_response_code($errCodes[0]);
}

```

4.1.21 models/Filter.php

```
<?php
include "../config/routes.php";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Includes
    include MODELS . "/Helpers.php";
    include CONFIG . "/connection.php";

    // Variable
    // display($_POST);
    $type = $_POST['type'];

    // Determine what needs filtering
    switch ($type) {
        case 'gloves':
            $categories = $_POST['categories'];
            $brands = $_POST['brands'];
            // display($brands);
            $colors = $_POST['colors'];
            $search = $_POST['search'];
            $sort = $_POST['sort'];
            $range = $_POST['range'];
            $perPage = $_POST['perPage'];    // select 6, 12, 24
            $page = $_POST["page"];          // 1,2,3...Last

            $gloves = filterActiveGlovesQuery($categories, $brands, $colors, $sea
rch, $sort, $range, $perPage, $page);
            // display($gloves);
            echo json_encode($gloves, JSON_FORCE_OBJECT);
            http_response_code(200);
            break;
        case "messages":
            // Variables
            $sort = $_POST['sort'];
            $perPage = $_POST['perPage'];    // select 6, 12, 24
            $page = $_POST["page"];          // 1,2,3...Last
            $search = $_POST['search'];

            // Get id if set
            $id = null;
            if (isset($_POST['id']) && !empty($_POST['id'])) {
                $id = $_POST['id'];
            }
    }
}
```

```

    // Filter messages
    $messages = filterMessagesQuery($search, $sort, $perPage, $page, $id)
;

    // Return result messages
    echo json_encode($messages, JSON_FORCE_OBJECT);
    http_response_code(200);
    break;
case "orders":
    $sort = $_POST['sort'];
    $perPage = $_POST['perPage'];    // select 6, 12, 24
    $page = $_POST["page"];          // 1,2,3...Last
    $search = isset($_POST['search']) ? $_POST['search'] : "";
    // Get id if set
    $id = null;
    if (isset($_POST['id']) && !empty($_POST['id'])) {
        $id = $_POST['id'];
    }

    // Filter orders
    $orders = filterOrdersQuery($search, $sort, $perPage, $page, $id);

    // Return result orders
    echo json_encode($orders, JSON_FORCE_OBJECT);
    http_response_code(200);
    break;
case "users":
    $sort = $_POST['sort'];
    $perPage = $_POST['perPage'];    // select 6, 12, 24
    $page = $_POST["page"];          // 1,2,3...Last
    $search = $_POST['search'];

    // Filter users
    $users = filterUsersQuery($sort, $search, $perPage, $page);

    // Return result users
    echo json_encode($users, JSON_FORCE_OBJECT);
    http_response_code(200);
    break;
}
} else {
    // Unauthorized access
    $errs[] = "Unauthorized access";
    echo json_encode($errs, JSON_FORCE_OBJECT);
}

```

```
    http_response_code(403);
}
```

4.1.22 models/Helpers.php

```
<?php

// Displaying data
function display($content)
{
    echo "<pre>";
    print_r($content);
    echo "</pre>";
}

// Getting all meta data
function getViewData($page)
{
    global $conn;
    $query = "SELECT * FROM view WHERE name_view = ? ";
    $prep = $conn->prepare($query);
    if ($prep->execute([$page])) {
        $res = $prep->fetch();
        define("HEADING_PRIMARY", $res->heading_primary);
        define("HEADING_SECONDARY", $res->heading_secondary);
        define("FOOTER", $res->footer_view);
        define("META_DESC", $res->meta_desc);
        define("META_KEYWORDS", $res->meta_keywords);
        define("TITLE", $res->title_view);
    }
}

// Define nav
function defineNav()
{
    define("GENERAL_NAV", getNav('general'));
    define("AUTH_NAV", getNav('auth'));
    define("ADMIN_NAV", getNav('admin'));
    define("CUSTOMER_NAV", getNav('customer'));
}

// Get nav
```

```

function getNav($name)
{
    global $conn;
    $query = "SELECT * FROM view v INNER JOIN nav n ON v.nav_type = n.id_nav WHERE n.nav_name = ?";
    $prep = $conn->prepare($query);
    if ($prep->execute([$name])) {
        // Constant cannot be defined with objects
        // Must use arrays
        return $prep->fetchAll(PDO::FETCH_ASSOC);
    }
}

// Select all from table
function selectAllFromTable($table)
{
    global $conn;
    $query = "SELECT * FROM $table";
    $res = $conn->query($query)->fetchAll();
    return $res;
}

// Delete all from table WHERE
function deleteAllFromTableWhere($table, $where, $param)
{
    global $conn;
    $query = "DELETE FROM $table WHERE $where = ?";
    $prep = $conn->prepare($query);
    if ($prep->execute([$param])) {
        return true;
    }
}

// Check if view is valid
function isView($view)
{
    $arr = [
        // Default
        "index",

        // Auth
        "login",
        "register",

        // Admin
    ];
}

```

```

    "admin-dashboard",
    "admin-orders",
    "admin-gloves",
    "admin-users",
    "admin-messages",
    "admin-message",
    "admin-order",
    "admin-user",

    // Customer
    "customer",

    // General
    "cart",
    "shop",
    "about",
    "contact",
    "author",
    "glove"
];
return in_array($view, $arr);
}

// Checks if data is available(no duplicates)
function isAvailable($table, $where, $param)
{
    global $conn;
    // Table/Column name cannot be parameterized
    $query = "SELECT " . $where . " as count FROM " . $table . " WHERE " . $where
. " = ?";
    $prep = $conn->prepare($query);

    if ($prep->execute([$param])) {
        // Counts the number of rows
        $res = $prep->rowCount();
        return !$res;
    }
}
// Checks if data is available(no duplicates)
function isEmail($email)
{
    global $conn;
    // Table/Column name cannot be parameterized
    // display($email);
    $query = "SELECT * FROM user WHERE email_user = ?";

```

```

$prep = $conn->prepare($query);
// display($query);
// display($conn->query($query)->fetch());

// display($query);

if ($prep->execute([$email])) {
    // Counts the number of rows
    // display($prep->fetch());
    return $prep->fetchColumn();
}
}

// Check if user should be disabled
function isLockUserCount($email)
{
    global $conn;
    $query = "SELECT COUNT(*) FROM failed_login WHERE email_user = ? AND login_date > DATE_SUB(NOW(), INTERVAL 5 MINUTE) AND email_user <> 'ko.bi.reko.da.je.mala01@gmail.com' ";
    $prep = $conn->prepare($query);
    if ($prep->execute([$email])) {
        // display($prep->fetchAll());
        return $prep->fetchColumn();
    }
}

///////////////////////////////
// Functions for gloves which everyone can use

// Count all active gloves
function countAllActiveGloves($search = "")
{
    global $conn;
    $where = "";
    if ($search) {
        $where = " AND (LOWER(name_glove) LIKE '%$search%') ";
        // $where = " AND name_glove = ?";
        // (LOWER(u.first_name) LIKE '%$search%'
    }
    $query = "SELECT COUNT(*) FROM glove WHERE date_removed IS NULL" . $where;
    // display($query);
}

```

```

// if ($search) {
//     $prep = $conn->prepare($query);
//     if ($prep->execute([$search])) {
//         return $prep->fetchColumn();
//     }
// } else {
//     return $conn->query($query)->fetchColumn();
// }

// Count all from table
function countAllFromTable($table)
{
    global $conn;
    $query = "SELECT COUNT(*) FROM $table";
    return $conn->query($query)->fetchColumn();
}

// Count all from table
function countAllOrdersUsersMessages($table, $search = "") {
    global $conn;
    // display($table);
    $where = "";
    if ($search) {
        $where = " WHERE (LOWER(u.first_name) LIKE '%$search%' OR LOWER(u.last_name) LIKE '%$search%')";
    }
    if ($table == "users") {
        // $query = "SELECT COUNT(DISTINCT od.id_user) FROM order_detail od INNER JOIN user u ON u.id_user = od.id_user" . $where;
        // display("Are we in here users");
        $query = "SELECT COUNT(DISTINCT od.id_user) FROM order_detail od INNER JOIN user u ON u.id_user = od.id_user" . $where;
    } else if ($table == "order_detail") {
        // display("Are we in here order");
        $query = "SELECT COUNT(od.id_user) FROM order_detail od INNER JOIN user u ON u.id_user = od.id_user" . $where;
    } else if ($table == "message") {
        // display("Are we in here message");
        $where2 = " WHERE (LOWER(msg_first_name) LIKE '%$search%' OR LOWER(msg_last_name) LIKE '%$search%')";
        $query = "SELECT COUNT(*) FROM message" . $where2;
    }
}

```

```

// display($query);
if ($search) {
    $prep = $conn->prepare($query);
    if ($prep->execute([$search])) {
        return $prep->fetchColumn();
    }
} else {
    return $conn->query($query)->fetchColumn();
}
}

// Count all from table where
function countAllFromTableWhere($table, $where, $param)
{
    global $conn;
    $query = "SELECT COUNT(*) FROM $table WHERE $where = ?";
    $prep = $conn->prepare($query);
    if ($prep->execute([$param])) {
        return $prep->fetchColumn();
    } else {
        return false;
    }
}

// Select all from table where
function selectAllFromTableWhere($table, $where, $param)
{
    global $conn;
    try {
        $isAvailable = isAvailable($table, $where, $param);
        if (!$isAvailable) {
            $query = "SELECT * FROM $table WHERE $where = ?";
            $prep = $conn->prepare($query);
            if ($prep->execute([$param])) {
                return $prep->fetch();
            }
        } else {
            return false;
        }
    } catch (PDOException $ex) {
        echo $ex->getMessage();
    }
}

// Select current glove values

```

```

function selectCurrentGloveValues($id)
{
    global $conn;
    $query = "SELECT g.id_glove as id, g.name_glove as name, g.desc_glove as description, g.date_added, clr.id_color as colorID, clr.color_name as color, c.id_cat as catID, c.name_cat as cat, b.id_brand as brandID, b.name_brand as brand, GROUP_CONCAT(s.id_size SEPARATOR ',') as sizesID, GROUP_CONCAT(s.value_size SEPARATOR ',') as sizes, m.name_measure as measure, p.price, i.normal_img as n_img, i.thumb_img as t_img FROM glove as g INNER JOIN color clr ON g.id_color = clr.id_color INNER JOIN brand b ON b.id_brand = g.id_brand INNER JOIN category c ON c.id_cat = g.id_cat INNER JOIN glove_size gs ON g.id_glove = gs.id_glove INNER JOIN size s ON s.id_size = gs.id_size INNER JOIN image i ON i.id_glove = g.id_glove INNER JOIN measure m ON m.id_measure = s.id_measure INNER JOIN price p ON p.id_glove = g.id_glove WHERE g.id_glove = ? AND g.date_removed IS NULL AND gs.date_remove IS NULL GROUP BY g.id_glove";
    $prep = $conn->prepare($query);
    if ($prep->execute([$id])) {
        return $prep->fetch();
    }
}

// Selects All available sizes
function selectAvailableSizes($id)
{
    global $conn;

    $query = "SELECT m.name_measure as measure, s.value_size as size, s.id_size as id FROM size s INNER JOIN category_size cs ON s.id_size = cs.id_size INNER JOIN category c ON c.id_cat = cs.id_cat INNER JOIN measure m ON m.id_measure = s.id_measure WHERE c.id_cat = ? ";
    $prep = $conn->prepare($query);

    if ($prep->execute([$id])) {
        $res = $prep->fetchAll();
        return $res;
    }
}

// Check if glove(row) exists for ID plus if date_deleted is NULL
function isGloveAndActive($id)
{
    global $conn;
    $query = "SELECT COUNT(*) FROM glove WHERE id_glove = ? AND date_removed IS NULL";
    $prep = $conn->prepare($query);
}

```

```

    if ($prep->execute([$id])) {
        return $prep->fetchColumn();
    }
}

// Select all active gloves
function selectAllActiveGloves()
{
    global $conn;
    $query = "SELECT g.id_glove as id, g.name_glove as name, g.desc_glove as description, g.date_added, clr.color_name as color, c.name_cat as cat, b.name_brand as brand, GROUP_CONCAT(s.value_size SEPARATOR ', ') as sizes, m.name_measure as measure, i.normal_img as n_img, i.thumb_img as t_img FROM glove as g INNER JOIN color clr ON g.id_color = clr.id_color INNER JOIN brand b ON b.id_brand = g.id_brand INNER JOIN category c ON c.id_cat = g.id_cat INNER JOIN glove_size gs ON g.id_glove = gs.id_glove INNER JOIN size s ON s.id_size = gs.id_size INNER JOIN image i ON i.id_glove = g.id_glove INNER JOIN measure m ON m.id_measure = s.id_measure WHERE g.date_removed IS NULL AND gs.date_remove IS NULL GROUP BY g.id_glove";
    return $conn->query($query)->fetchAll();
}

// Filter Active Gloves
function filterActiveGlovesQuery($categories, $brands, $colors, $search, $sort, $range, $perPage, $page)
{
    global $conn;
    // $num_rows = countAllActiveGloves($search);
    // display($num_rows);

    // // If number of rows divided by number of pages is greater than 0
    // // We define LIMIT $start $offset -
> LIMIT 0(start from) 6(take next 6 rows)
    // $numOfPages = $num_rows / $perPage;
    // if ($num_rows % $perPage > 0) {
    //     // display("I GO IF");
    //     $start = ($page - 1) * $perPage;
    // } else {
    //     // display("I GO ELSE");
    //     $start = ($page - 1) * $perPage;
    // }
    // // Important to use ceil to get exact number of pages
    // // 2 per page, 7 total -> 7 % 2 -> 2 + 2 + 2 + 1 -> 4 pages
    // // 6 per page, 7 total -> 7 % 6 -> 6 + 1 -> 2 pages
    // $numOfPages = ceil($numOfPages);
    // // display($numOfPages);
}

```

```

$where .= " WHERE g.date_removed IS NULL AND gs.date_remove IS NULL ";
// Every variable is a string so if it not empty string, we concatenate result
$t
if (!empty($categories)) {
    $where .= " AND c.id_cat IN ($categories)";
}
if (!empty($brands)) {
    $where .= " AND b.id_brand IN ($brands)";
}
if (!empty($colors)) {
    $where .= " AND clr.id_color IN ($colors)";
}
if (!empty($range)) {
    $where .= " AND p.price < $range";
}
if (!empty($search)) {
    $where .= " AND LOWER(g.name_glove) LIKE '%$search%'";
}
$where .= " GROUP BY g.id_glove ORDER BY ";

switch ($sort) {
    case "latest":
        $where .= "g.date_added DESC";
        break;
    case "nameAsc":
        $where .= "g.name_glove ASC";
        break;
    case "nameDesc":
        $where .= "g.name_glove DESC";
        break;
    case "priceDesc":
        $where .= "p.price DESC";
        break;
    case "priceAsc":
        $where .= "p.price ASC";
        break;
    default:
        $where .= "g.id_glove ASC";
}
// Get total amount

```

```

$query = "SELECT g.id_glove as id, g.name_glove as name, g.desc_glove as description, g.date_added, clr.color_name as color, c.name_cat as cat, b.name_brand as brand, GROUP_CONCAT(s.value_size SEPARATOR ', ') as sizes, p.price, m.name_measure as measure, i.normal_img as n_img, i.thumb_img as t_img FROM glove as g INNER JOIN color clr ON g.id_color = clr.id_color INNER JOIN brand b ON b.id_brand = g.id_brand INNER JOIN category c ON c.id_cat = g.id_cat INNER JOIN glove_size gs ON g.id_glove = gs.id_glove INNER JOIN size s ON s.id_size = gs.id_size INNER JOIN image i ON i.id_glove = g.id_glove INNER JOIN measure m ON m.id_measure = s.id_measure INNER JOIN price p ON p.id_glove = g.id_glove ";
$queryCount = $query . "" . $where;
// $gloves = $conn->query($queryCount)->fetchAll();
// display($gloves);
$num_rows = count($conn->query($queryCount)->fetchAll());
$resp = returnNumberOfPages($num_rows, $perPage, $page);
$numOfPages = $resp[0];
$start = $resp[1];
$numOfPages = ceil($numOfPages);

// Get products for one page (LIMIT)
$where .= " LIMIT $start, $perPage";
$query = $query . '' . $where;
$gloves = $conn->query($query)->fetchAll();
// display($query);
// display($query);
$res = array([
    "gloves" => $gloves,
    "pages" => $numOfPages,
    "total" => $num_rows
]);
// display($res);
return $res;
}

// Filter Orders
function filterOrdersQuery($search = "", $sort, $perPage, $page, $id)
{
    global $conn;

    if ($id) {
        $num_rows = countAllFromTableWhere("order_detail", "id_user", $id);
        // display($num_rows);
    } else {
        $num_rows = countAllOrdersUsersMessages("order_detail", $search);
    }
    // $num_rows = countAllOrdersUsersMessages("order_detail", $search);
}

```

```

$resp = returnNumberOfPages($num_rows, $perPage, $page);
$numOfPages = $resp[0];
$start = $resp[1];
$numOfPages = ceil($numOfPages);
// display($numOfPages);

if ($id) {
    $where .= " WHERE u.id_user = ? ";
} else {
    $where .= " ";
}

if ($search) {
    $where .= " WHERE (LOWER(u.first_name) LIKE '%$search%' OR LOWER(u.last_name) LIKE '%$search%')";
}

// $resp = returnNumberOfPages($num_rows, $perPage, $page);
// $numOfPages = $resp[0];
// $start = $resp[1];

// display($where);

$query = "SELECT od.id_order_detail, u.first_name, u.last_name, u.id_user, SUM(ol.price_purchase * ol.quantity) as total, od.date_order as date, COUNT(ol.id_order_line) as count FROM user u INNER JOIN order_detail od ON u.id_user = od.id_user INNER JOIN order_line ol ON od.id_order_detail = ol.id_order_detail";
// display($query);
// Every query has this condition
$query .= $where;
// display($query);
// Copy of the query, assignment not concatenaenation
// We will use this query NOT to SELECT products, but to COUNT
if ($search) {
    $queryCount = "SELECT COUNT(*) FROM order_detail od INNER JOIN user u ON u.id_user = od.id_user" . $where;
} else {
    $queryCount = "SELECT COUNT(*) FROM order_detail od INNER JOIN user u ON u.id_user = od.id_user" . $where;
}
// display($queryCount);
// Every query needs a group by
$query .= " GROUP BY od.id_order_detail";

```

```

if ($sort) {
    $query .= " ORDER BY ";
}
// display($sort);
switch ($sort) {
    case "latest":
        // display("in here");
        $query .= " od.date_order DESC ";
        break;
    case "nameAsc":
        $query .= " u.first_name ASC ";
        break;
    case "nameDesc":
        $query .= " u.first_name DESC";
        break;
    case "priceDesc":
        $query .= "SUM(ol.price_purchase * ol.quantity) DESC";
        break;
    case "priceAsc":
        $query .= "SUM(ol.price_purchase * ol.quantity) ASC";
        break;
    case "numberOfItems":
        $query .= "COUNT(ol.id_order_line) DESC";
        break;
    // default:
    //     $where .= "od.id_order_detail ASC";
}
if ($id) {
    // Id was passed, getting all orders for specific user
    $prep = $conn->prepare($queryCount);
    if ($prep->execute([$id])) {
        $total = $prep->fetchColumn();
    }
} else {
    // No Id was passed, getting all orders for all users
    // display($queryCount);
    $total = $conn->query($queryCount)->fetchColumn();
}

// Sets the limit of fetched rows
$limit = " LIMIT $start, $perPage";
$query = $query . $limit;

// display($query);

```

```

if ($id) {
    // Preparing query because we passed the parameter
    $prep = $conn->prepare($query);
    if ($prep->execute([$id])) {
        $orders = $prep->fetchAll();
    }
} else {
    // Basic query, because there are no parameters
    // display($query);
    $orders = $conn->query($query)->fetchAll();
}

$res = array([
    "orders" => $orders,
    "pages" => $numOfPages,
    "total" => $total
]);
// display($res);
return $res;
}

// Filter Messages
function filterMessagesQuery($search, $sort, $perPage, $page, $id)
{
    global $conn;

    // if ($id) {
    //     $num_rows = countAllFromTableWhere("message", "msg_email", $id);
    // } else {
    //     $num_rows = countAllOrdersUsersMessages("message", $search);
    // }
    $num_rows = countAllOrdersUsersMessages("message", $search);
    // display($num_rows);

    $resp = returnNumberOfPages($num_rows, $perPage, $page);
    $numOfPages = $resp[0];
    $start = $resp[1];

    $where = " ";
    $order = " ";
    $limit = " ";

    if (!empty($search)) {

```

```

        $where .= " WHERE (LOWER(msg_first_name) LIKE '%$search%' OR LOWER(msg_la
st_name) LIKE '%$search%')";
    }

    if ($sort) {
        $order .= " ORDER BY ";
    }

    switch ($sort) {
        case "latest":
            // display("in here");
            $order .= " msg_date DESC ";
            break;
        case "nameAsc":
            $order .= " msg_first_name ASC ";
            break;
        case "nameDesc":
            $order .= "msg_first_name DESC";
            break;
        case "subject":
            $order .= "msg_subject ASC";
            // default:
            //     $where .= "od.id_order_detail ASC";
    }

    $query = "SELECT * FROM message ";
    $query .= $where . $order;

    $limit = " LIMIT $start, $perPage";
    $query .= $query . $limit;

    $messages = $conn->query($query)->fetchAll();

    $res = array([
        "messages" => $messages,
        "pages" => $numOfPages,
        "total" => $num_rows
    ]);
    // display($res);
    return $res;
}

// Filter Users
function filterUsersQuery($sort, $search = "", $perPage, $page)

```

```

{

global $conn;
// $num_rows = countAllFromTable("user");
// $num_rows = countAllOrdersUsersMessages("users", $search);
// // display($num_rows);

// $resp = returnNumberOfPages($num_rows, $perPage, $page);
// $numOfPages = $resp[0];
// $start = $resp[1];

$where = " ";
$order = " ";

if (!empty($search)) {
    $where .= " AND (LOWER(u.first_name) LIKE '%$search%' OR LOWER(u.last_name) LIKE '%$search%')";
}

if ($sort) {
    $order .= " ORDER BY ";
}

switch ($sort) {
    case "latest":
        $order .= "u.date_register DESC";
        break;
    case "nameAsc":
        $order .= "u.first_name ASC";
        break;
    case "nameDesc":
        $order .= "u.first_name DESC";
        break;
    case "priceDesc":
        $order .= "p.price DESC";
        break;
    case "priceAsc":
        $order .= "p.price ASC";
        break;
    case "numberOfItems":
        $order .= "COUNT(ol.id_order_line) DESC";
        break;
    // default:
        // $where .= "od.id_order_detail ASC";
}

```

```

$query = "SELECT u.id_user, u.isLock, u.first_name, u.email_user, u.last_name
, SUM(ol.price_purchase * ol.quantity) as total, COUNT(DISTINCT od.id_order_detail)
as count FROM user u LEFT JOIN order_detail od ON u.id_user = od.id_user LEFT
JOIN order_line ol ON od.id_order_detail = ol.id_order_detail WHERE u.id_user <>
1";
// $query = "SELECT u.first_name, u.last_name, SUM(ol.price_purchase * ol.quantity)
as total, od.date_order as date, COUNT(od.id_order_line) as count FROM user u
INNER JOIN order_detail od ON u.id_user = od.id_user INNER JOIN order_line ol
ON od.id_order_detail = ol.id_order_detail GROUP BY od.id_order_detail";
$group = " GROUP BY u.id_user";
$queryCount = $query . $where . $group . $order;

$users = $conn->query($query)->fetchAll();
$total = count($users);

$num_rows = count($conn->query($queryCount)->fetchAll());
$resp = returnNumberOfPages($num_rows, $perPage, $page);
$numOfPages = $resp[0];
$start = $resp[1];
$numOfPages = ceil($numOfPages);

// display($queryCount);
$total = count($conn->query($queryCount)->fetchAll());
// display($total);
$limit = " LIMIT $start, $perPage";
$query = $query . $where . $group . $order . $limit;
// display($query);
$users = $conn->query($query)->fetchAll();
// $total = count($users);

$res = array([
    "users" => $users,
    "pages" => $numOfPages,
    // "total" => $total
    "total" => $total
    // "num" =>
]);
// display($query);
// display($res);
return $res;

```

```

}

// Returns number of pages
function returnNumberOfPages($num_rows, $perPage, $page)
{
    $numOfPages = $num_rows / $perPage;
    if ($num_rows % $perPage > 0) {
        // display("I GO IF");
        $start = ($page - 1) * $perPage;
    } else {
        // display("I GO ELSE");
        $start = ($page - 1) * $perPage;
    }
    $numOfPages = ceil($numOfPages);
    // display($numOfPages);
    return [$numOfPages, $start];
}

// Returns order detail info
function returnOrderDetailInfo($id)
{
    global $conn;

    // Begin Transaction
    // $conn->beginTransaction();

    $errs = [];

    $query = "SELECT COUNT(*) as count FROM order_detail WHERE id_order_detail = ?";
    $prep = $conn->prepare($query);
    if ($prep->execute([$id])) {
        $isOrder = true;
    }
    // Checks if there is such order
    $isOrder = countAllFromTable("order_detail");
    if (!$isOrder) {
        $errs[] = "There is no such order";
    }

    $query = "SELECT od.date_order as date, g.name_glove, g.id_glove, b.name_brand, cat.name_cat, ol.price_purchase as price, clr.color_name, i.thumb_img as img, s.value_size as size, m.name_measure as measure, ol.quantity FROM order_detail od INNER JOIN order_line ol ON od.id_order_detail = ol.id_order_detail INNER JOIN g

```

```

love_size gs ON ol.id_glove_size = gs.id_glove_size INNER JOIN size s ON s.id_size
e = gs.id_size INNER JOIN measure m ON m.id_measure = s.id_measure INNER JOIN glo
ve g ON gs.id_glove = g.id_glove INNER JOIN category cat ON cat.id_cat = g.id_cat
INNER JOIN image i ON i.id_glove = g.id_glove INNER JOIN brand b ON b.id_brand =
g.id_brand INNER JOIN color clr ON clr.id_color = g.id_color ";
$query .= "WHERE od.id_order_detail = ?";

$prep = $conn->prepare($query);
if ($prep->execute([$id])) {
    $orders = $prep->fetchAll();
    $count = count($orders);
} else {
    $errs[] = "Failed to retreive items";
}

$query = "SELECT u.first_name, u.last_name, u.email_user, u.id_user FROM user
u INNER JOIN order_detail od ON od.id_user = u.id_user";
$query .= " WHERE od.id_order_detail = ?";

$prep = $conn->prepare($query);
if ($prep->execute([$id])) {
    $user = $prep->fetch();
} else {
    $errs[] = "Failed to retreive user";
}

// End transaction with commit or rollback
if (!empty($errs)) {
    // Rollback
    // $conn->rollBack();
    return false;
} else {
    // Commit
    // $conn->commit();
    // display($orders);
    // display($user);
    return [$orders, $user, $count];
}
}

// Insert message
function insertMessage($msg_first_name, $msg_last_name, $msg_email, $msg_subject,
$msg_body)
{
    global $conn;

```

```

    $query = "INSERT INTO message(msg_first_name, msg_last_name, msg_email, msg_subject, msg_body) VALUES(?, ?, ?, ?, ?)";
    $prep = $conn->prepare($query);
    if ($prep)
>execute([$msg_first_name, $msg_last_name, $msg_email, $msg_subject, $msg_body]))
{
    return true;
}
}

// Returns message details
function returnMessageInfo($id)
{
    global $conn;
    try {
        // Begin Transaction
        $conn->beginTransaction();

        $query = "SELECT * from message ";
        $query .= "WHERE id_msg = ?";

        $prep = $conn->prepare($query);
        if ($prep->execute([$id])) {
            $message = $prep->fetch();
            if (!$message->id_msg) {
                header("Location: index.php?page=index");
            }
        } else {
            // $errs[] = "Failed to retreive messages";
            header("Location: index.php?page=index");
        }

        // End transaction with commit or rollback
        if (!empty($errs)) {
            // Rollback
            $conn->rollBack();
            return false;
        } else {
            // Commit
            $conn->commit();
            return [$message];
        }
    } catch (PDOException $ex) {
        // Try block failed
    }
}

```

```

        $errs[] = $ex->getMessage();
    }
}

// Log page visits
function pageVisitsWrite($page, $user = "Unknonwn user")
{
    $open = fopen(DATA . "/page-visits.txt", "a+");
    $file = file(DATA . "/page-visits.txt");
    $ip = $_SERVER['REMOTE_ADDR'];
    $date = date('Y-m-d H:i:s');

    $content = $user . "__" . $ip . "__" . $date . "__" . $page . "\n";
    fwrite($open, $content);
    fclose($open);
    pageVisitsOverwrite();
}

// Rewrite page visits
function pageVisitsOverwrite()
{
    $file = file(DATA . "/page-visits.txt");
    $log = [];
    foreach ($file as $i => $r) {
        list($email, $ip, $date, $page) = explode("__", $r);
        $difference = time() - strtotime($date);
        if ($difference < 60 * 60 * 24) {
            $log[] = $r;
        }
    }
    // display($log);
    $log = implode("", $log);
    $open = fopen(DATA . "/page-visits.txt", "w");
    fwrite($open, $log);
    fclose($open);
}

// Return page visits as array
function returnPageVisits()
{
    $file = file(DATA . "/page-visits.txt");
    $count = count($file);
}

```

```

foreach ($file as $i => $r) {
    list($email, $ip, $date, $page) = explode("__", $r);
    $pages[] = $page;
}

$total = count($pages);
$pages = array_count_values($pages);
// display($res);
return array($total, $pages);
}

```

4.1.23 models/Patterns.php

```

<?php
// Users
$regUsername = "^\w{2,25}$";
$regName = "^[A-ZČĆŽĐ][a-zććžđ]{2,15}$";
$regEmail = "^\w+([-.' ]\w+)*@\w+([-.\.]\w+)*\.\w+([-.\.]\w+)*$";
$regPass = "^(?=.*[A-Za-z])(?=.*\d)(?=.*[$!%*#?&])[A-Za-z\d@$!%*#?&]{8,30}$";
$regAddress = "^\#[.0-9a-zA-Z\s,-]{2,50}$";
$regPhone = "^[+]*[(]{0,1}[0-9]{1,4}[)]{0,1}[-\s./0-9]*$";
// Gloves
$regGloveName = "^\[\w\ ]{2,50}$";
$regPrice = "^\d{1,8}(?:\.\d{1,4})?$$";
$regSubject = "(^\[\w[((\w+)|(\w*))*$]|(^$\w$))";

```

4.1.24 models/Word.php

```

<?php session_start();
$word = new COM("word.application") or die("Unable to instantiate Word");
$word->Visible = 0;
$word->Documents->Add();
$txt = "Pavle Djurdjic";
$txt1 = "This is pretty much the last time I'll ever write any code without the use of Framework/Library(expecially Front-end). This bloody JS probably killed around 100 000 of my nerve cells, and holmes , I ain't going through that hell again. Next time we meet, you'll be seeing a lot of node modules (Angular app). Oh yeah my name is Pavle...blah...web...blah...dev...and that kind of stuff. Much word, many {{ string interpolation }}";
$txt2 = "Hasta luego, gracias por nada. Kibonga 7/19";
$result = $txt . "\n\n\n" . $txt1 . "\n\n\n" . $txt2;
$word->Selection->TypeText($result);

```

```

$filename = tempnam(sys_get_temp_dir(), "word");
$word->Documents[1]->SaveAs($filename);
$word->Quit();
$word = null;
header("Content-type: application/vnd.ms-word");
header("Content-Disposition: attachment;Filename=Kibonga.doc");
readfile($filename);
unlink($filename);

```

4.1.25 views/fixed/admin-table.php

```

<div <class="container">
    <?php if ($type != "order lines") : ?>
        <div <class="mt-5">
            <article <class="filter-group">
                <header <class="card-header bg-c-primary">
                    <h6 <class="lead text-white">Search for order</h6>
                </header>
                <div <class="filter-content collapse show" id="collapse_1">
                    <div <class="card-body">
                        <form <class="pb-3">
                            <div <class="input-group d-flex align-items-center">
                                <span <class="material-icons align-bottom ">
                                    search
                                </span>
                                <input type="text" <class="form-control" id="search" name="search" placeholder="Search">
                            </div>
                        </form>
                    </div> <!-- card-body.-->
                </div>
            </article> <!-- filter-group .-->
            <div>
                <p <class="p-0">
                    Number of <?= $type ?: <span id="<?= $totalCount ?>"></span></p>
                </div>
                <div <class="form-inline">
                    <div <class="mb-2">
                        <div>
                            <label <class="lead"><?= ucfirst($type) ?> per page: </label>
                        </div>
                        <select id="perPage" <class="form-select" name="perPage" id="perPage">

```

```

        <option value="1">1</option>
        <option value="2">2</option>
        <option value="4" selected>4</option>
        <option value="6">6</option>
        <option value="8">8</option>
        <option value="12">12</option>
        <option value="24">24</option>
    </select>
</div>
</div>
<div class="form-inline">
    <div class="mb-2">
        <label class="lead">Sort: </label>
        <select class="mr-2 form-select" id="sort" name="sort">
            <option value="">Select</option>
            <option value="nameAsc">Name A-Z</option>
            <option value="nameDesc">Name Z-A</option>
            <option value="latest">Latest</option>
            <?php if ($type == "orders") : ?>
                <option value="priceDesc">Most Expensive</option>
                <option value="priceAsc">Least Expensive</option>
            <?php elseif ($type == "users") : ?>
                <option value="numberOfItems">Number of orders</option>
            <?php elseif ($type == "messages") : ?>
                <option value="subject">Subject</option>
            <?php endif; ?>
        </select>
    </div>
</div>
<?php endif; ?>
<div class="my-5">
    <h3 class="lead" style="font-size: 1.8rem" id="tableName">All <?= $type ?></h3>
    <table class="table table-striped table-bordered text-center">
        <thead>
            <tr>
                <?php foreach ($tableColumns as $i => $c) : ?>
                    <th scope="col"><?= $c ?></th>
                <?php endforeach; ?>
            </tr>
        </thead>
        <tbody id="<?= $tableId ?>" class="table-striped align-center">
    
```

```

        <!-- Insert orders -->
        <?php if (isset($order)) :
            // If order is set call function for interpolations
            include_once FIXED . "/components/order-
detail.component.php";
            foreach ($order_lines as $i => $ol) :
        ?>
            <?= returnAdminTableOrderHTML($ol, $i) ?>
        <?php
            endforeach;
        endif; ?>
        </tbody>
    </table>
</div>
<nav aria-label="...">
    <ul class="pages pagination" id="pages">

    </ul>
</nav>
</div>

```

4.1.26 views/fixed/admin.php

```

<?php
if (!isset($_SESSION['loggedUser'])) {
    header("Location: index.php");
} else {
    $user = $_SESSION['loggedUser'];
    if ($user->name_role != "admin") {
        // User is unauthorized
        header("Location: index.php");
    }
}
?>
<div class="container">
    <nav class="navbar navbar-expand-lg bg-c-secondary box-shadow">
        <div class="container-fluid">
            <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNavDropdown" aria-
controls="navbarNavDropdown" aria-expanded="false" aria-
Label="Toggle navigation">
                <span class="material-icons navbar-toggler-icon text-white d-
flex align-items-center">
                    menu

```

```

        </span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavDropdown">
        <ul class="navbar-nav mx-auto mb-2 mb-lg-0">
            <?php foreach (ADMIN_NAV as $i => $n) : ?>
                <li class="nav-item">
                    <a class="nav-link text-
white acc <?= (!isset($_GET['page'])) ? "" : (( $_GET['page'] == $n["name_view"] ) ? "active" : "") ?>" href="index.php?page=<?= $n['name_view'] ?>">
                        <span class="material-icons align-bottom">
                            <?= $n['icon_name'] ?>
                        </span>
                        <span><?= $n['title_view'] ?></span>
                    </a>
                </li>
            <?php endforeach; ?>
        </ul>
    </div>
</nav>
</div>

```

4.1.27 views/fixed/footer.php

```

<footer class="text-white text-center text-lg-start mt-5 bg-c-primary">
    <!-- Grid container -->
    <div class="container p-4">
        <!--Grid row-->
        <div class="row">
            <!--Grid column-->
            <div class="col-lg-6 col-md-12 mb-4 mb-md-0">
                <h5 class="text-uppercase">Footer Content</h5>
                <p class="lead">
                    <?= FOOTER ?>
                </p>
            </div>
            <!--Grid column-->
            <!--Grid column-->
            <div class="col-lg-3 col-md-6 mb-4 mb-md-0">
                <h5 class="text-uppercase">Navigation</h5>
                <ul class="navbar-nav mx-auto mb-2 mb-lg-0 ">
                    <?php foreach (GENERAL_NAV as $i => $n) : ?>
                        <li class="nav-item">

```

```

        <a class="nav-link text-
white acc <?= (!isset($_GET['page'])) ? "" : (($_GET['page'] == $n["name_view"])
? "active" : "") ?>" href="index.php?page=<?= $n['name_view'] ?>">
            <span class="material-icons align-bottom">
                <?= $n['icon_name'] ?>
            </span>
            <span><?= $n['title_view'] ?></span>
        </a>
    </li>
<?php endforeach; ?>
</ul>
</div>
<!--Grid column-->
<!--Grid column-->
<div class="col-lg-3 col-md-6 mb-4 mb-md-0">
    <h5 class="text-uppercase mb-0">Useful</h5>

    <ul class="list-unstyled">
        <li>
            <a href="docs.pdf" target="_blank" class="text-
white">Docs</a>
        </li>
        <li>
            <a href="models/Word.php" target="_blank" class="text-
white">Word</a>
        </li>
        <li>
            <a href="assets/js/main.js" target="_blank" class="text-
white">JS</a>
        </li>
        <li>
            <a href="https://www.linkedin.com/" target="_blank" class
="text-white">LinkedIn</a>
        </li>
        <li>
            <a href="https://github.com/" target="_blank" class="text-
white">Github</a>
        </li>
        <li>
            <a href="https://pavle-say-what-one-mo-
time.lzivadinovic.com/" target="_blank" class="text-white">Portfolio</a>
        </li>
    </ul>
</div>
<!--Grid column-->

```

```

        </div>
        <!--Grid row-->
    </div>
    <!-- Grid container -->

    <!-- Copyright -->
    <div class="text-center p-3 bg-c-secondary">
        <p id="footer-copyright" class="lead">Made by Kibonga</p>
    </div>
    <!-- Copyright -->
</footer>
</body>

</html>

```

4.1.28 views/fixed/head.php

```

<!DOCTYPE html>
<html Lang="sr">

    <head>
        <!-- Metadata -->
        <meta name="description" contentname="keywords" content="<?= META_KEYWORDS ?>" />
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title><?= WEBSITE ?> | <?= TITLE ?></title>

        <!-- JQuery -->
        <script defer src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.min.js" integrity="sha256-xUj+3OJU5yExlq6GSYGSKh7tPXikynS7ogEvDej/m4=" crossorigin="anonymous"></script>
        <script defer src="https://cdn.jsdelivr.net/npm/jquery-validation@1.19.3/dist/jquery.validate.min.js"></script>
        <script defer src="http://cdn.jsdelivr.net/npm/jquery-validation@1.19.3/dist/additional-methods.min.js"></script>
        <!-- Bootstrap JS -->
        <script defer src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js" integrity="sha384-b5kHyXgcpbZJO/tY9U17kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOnJO" crossorigin="anonymous"></script>
        <!-- Bootstrap CSS -->

```

```

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-+0n0xVW2eSR50omGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxYbOO17+AMvyTG2x" crossorigin="anonymous">
    <!-- FontAwesome 5 -->
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
        <!-- <link href="https://cdn.jsdelivr.net/npm/jquery-validation@1.19.3/dist/additional-methods.min.js">
            <link href="https://cdn.jsdelivr.net/npm/jquery-validation@1.19.3/dist/jquery.validate.min.js"> -->
                <!-- Template CSS -->
                <link rel="stylesheet" href="assets/css/style.css" type="text/css" />
                <link rel="shortcut icon" href="favicon.png" type="image/x-icon">
                    <!-- Template JS -->
                    <script defer src="assets/js/main.js"></script>
</head>

<body>

```

4.1.29 views/fixed/nav.php

```

<!DOCTYPE html>
<html lang="sr">

<head>
    <!-- Metadata -->
    <meta name="description" content=<?= META_DESC ?>" />
    <meta name="keywords" content=<?= META_KEYWORDS ?>" />
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title><?= WEBSITE ?> | <?= TITLE ?></title>

    <!-- JQuery -->
    <script defer src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.min.js" integrity="sha256-xUj+3OJU5yExlq6GSYGSk7tPXikynS7ogEvDej/m4=" crossorigin="anonymous"></script>
        <script defer src="https://cdn.jsdelivr.net/npm/jquery-validation@1.19.3/dist/jquery.validate.min.js"></script>
            <script defer src="http://cdn.jsdelivr.net/npm/jquery-validation@1.19.3/dist/additional-methods.min.js"></script>
                <!-- Bootstrap JS -->

```

```

<script defer src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js" integrity="sha384-b5kHyXgcpbZJ0/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOnJ0" crossorigin="anonymous"></script>
    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-+0n0xVW2eSR50omGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxYbOO17+AMvyTG2x" crossorigin="anonymous">
        <!-- FontAwesome 5 -->
        <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
            <!-- <link href="https://cdn.jsdelivr.net/npm/jquery-validation@1.19.3/dist/additional-methods.min.js">
                <link href="https://cdn.jsdelivr.net/npm/jquery-validation@1.19.3/dist/jquery.validate.min.js"> -->
                    <!-- Template CSS -->
                    <link rel="stylesheet" href="assets/css/style.css" type="text/css" />
                    <link rel="shortcut icon" href="favicon.png" type="image/x-icon">
            <!-- Template JS -->
            <script defer src="assets/js/main.js"></script>
</head>

<body>
```

4.1.30 views/fixed/components/order-detail.component.php

```

<?php
function returnAdminTableOrderHTML($ol, $i)
{
    echo ""
?>
    <tr>
        <td class="align-middle"><?= $i + 1 ?></td>
        <td class="align-middle"><a href="index.php?page=glove&id=<?= $ol->id_glove ?>">name_glove ?>"></a></td>
        <td class="align-middle"><a class="text-secondary" href="index.php?page=glove&id=<?= $ol->id_glove ?>"><?= $ol->name_glove ?></a></td>
        <td class="align-middle"><?= $ol->name_cat ?></td>
        <td class="align-middle"><?= $ol->name_brand ?></td>
        <td class="align-middle">
```

```

        <div class="d-flex align-items-center justify-content-center">
            <span class="palette border align-bottom" style="background-
color:<?= $ol->color_name ?>!important;"></span>
            <span class="capitalize ms-2 lead align-bottom"><?= $ol-
>color_name ?></span>
        </div>
    </td>
    <td class="align-middle"><?= $ol->size ?> <?= $ol->measure == "OZ" ? $ol-
>measure : "" ?></td>
    <td class="align-middle">$<?= $ol->price ?></td>
    <td class="align-middle"><?= $ol->quantity ?></td>
    <td class="align-middle">$<?= ($+ $ol->quantity) * ($+ $ol->price) ?></td>
</tr>
<?php } ?>

```

4.1.31 views/pages/auth/login.php

```

<?php
if (isset($_SESSION['loggedUser'])) {
    header("Location: index.php");
}
?>
<div class="container mt-5">
    <div class="row">
        <div class="col-5 py-5">
            <form action="models/auth/Login.php" id="loginForm" name="loginForm"
method="post">
                <div class="form-group">
                    <div class="d-flex mb-2">
                        <span class="material-icons text-ternary ms-2 d-inline-
block">
                            alternate_email
                        </span>
                        <label for="" class="ms-2">Email</label>
                    </div>
                    <input type="email" name="inEmail" id="inEmail" class="form-
control" minlength="2" size="25" />
                </div>
                <div class="form-group mt-3">
                    <div class="d-flex mb-2">
                        <span class="material-icons text-ternary ms-2 d-inline-
block">
                            lock
                        </span>
                    </div>
                </div>
            </form>
        </div>
    </div>
</div>

```

```

                </span>
                <label for="" class="ms-2">Password</label>
            </div>
            <input type="password" name="inPass" id="inPass" class="form-control" minLength="2" size="25" />
        </div>
        <div class="my-4">
            <hr>
        </div>
        <div class="form-group mt-3">
            <button type="submit" name="btnLogin" id="btnLogin" class="text-white submit" value="btnLogin">
                <div class="btn bg-c-ternary d-inline-flex align-items-center text-white">
                    <span class="material-icons">
                        login
                    </span>
                    <span class="ms-2">Login</span>
                </div>
            </button>
        </div>
    </form>
</div>
</div>
</div>

```

4.1.32 views/pages/auth/register.php

```

<?php
if (isset($_SESSION['loggedUser'])) {
    header("Location: index.php");
}
?>
<div class="container mt-5">
    <div class="row">
        <div class="col-5 py-5">
            <form id="registerForm" name="registerForm" action="models/auth/Register.php" method="post">
                <div class="form-group">
                    <div class="d-flex mb-2">
                        <span class="material-icons text-primary mr-2 d-inline-block">

```

```
        <span class="material-icons text-ternary ms-2 d-
inline-block">
            person_outline
        </span>
    </span>
    <label for="" class="d-flex mb-2">Full name</label>
</div>
<div class="d-flex mb-2">
    <input type="text" name="inFname" id="inFname" placeholder="First name" class="form-control me-3">
        <input type="text" name="inLname" id="inLname" placeholder="Last name" class="form-control">
    </div>
    <div>
        <span id="inFnameErr" name="inFnameErr"></span><br>
        <span id="inLnameErr" name="inLnameErr"></span>
    </div>
</div>
<div class="form-group mt-3">
    <div class="d-flex mb-2">
        <span class="material-icons text-ternary ms-2 d-inline-
block">
            alternate_email
        </span>
        <label for="" class="d-flex ms-2 ">Email</label>
    </div>
    <input type="email" name="inEmail" id="inEmail" class="form-
control" />
    </div>
    <div class="form-group mt-3">
        <div class="d-flex mb-2">
            <span class="material-icons text-ternary ms-2 d-inline-
block">
                lock
            </span>
            <label for="" class="d-flex ms-2">Password</label>
        </div>
        <input type="password" name="inPass" id="inPass" class="form-
control" />
    </div>
    <div class="form-group mt-3">
        <div class="d-flex mb-2">
            <span class="material-icons text-ternary ms-2 d-inline-
block">
                lock_open
            </span>
        </div>
    </div>

```

```
        </span>
        <label for="" class="d-flex ms-
2">Confirm password</label>
    </div>
    <input type="password" name="inPassConf" id="inPassConf" clas-
s="form-control" />
    </div>
    <div class="form-group mt-3">
        <div class="d-flex mb-2">
            <span class="material-icons text-ternary ms-2 d-inlin-
block">
                call
            </span>
            <label for="" class="d-flex ms-2">Contact phone</label>
        </div>
        <input type="text" name="inPhone" id="inPhone" class="form-
control" />
    </div>
    <div class="form-group mt-3">
        <div class="d-flex mb-2">
            <span class="material-icons text-ternary ms-2 d-inlin-
block">
                home
            </span>
            <label for="" class="d-flex ms-2">Address</label>
        </div>
        <input type="text" name="inAddr" id="inAddr" class="form-
control" />
    </div>
    <div class="my-4">
        <hr>
    </div>
    <div class="form-group">
        <button type="submit" name="btnRegister" id="btnRegister" cla-
ss="text-white" value="btnRegister">
            <div class="btn bg-c-ternary d-inline-flex align-items-
center text-white">
                <span class="material-icons">
                    app_registration
                </span>
                <span class="ms-2">Register</span>
            </div>
        </button>
    </div>
    <?php
```

```

        if (isset($_SESSION['errors'])) {
            foreach ($_SESSION['errors'] as $i => $err) :
        ?>
            <div class="form-group">
                <span class="text-danger"><?= $err ?></span>
            </div>
        <?php endforeach;
    } ?
</form>
</div>
</div>
</div>

```

4.1.33 views/pages/public/about.php

```

<div class=" py-5">
    <div class="container">

        <h3 class="lead display-5 mb-4">Who are we</h3>

        <div class="row row-cols-1 row-cols-sm-2 row-cols-md-3 g-3">
            <div class="col">
                <div class="card shadow-sm">
                    <figure>
                        
                    </figure>
                    <div class="card-body">
                        <p class="card-text">We are a shop specialized in selling gloves for combat sports. Our products have been praised by many around the world.</p>
                        <div class="d-flex justify-content-between align-items-center">
                            </div>
                        </div>
                    </div>
                </div>
            <div class="col">
                <div class="card shadow-sm">
                    <figure>
                        
                    </figure>
                </div>
            </div>
        </div>
    </div>

```

```

        <div class="card-body">
            <p class="card-
text">Our top notch materials make for the best quality by which we are recognize
d. Our gloves have gathered many awards for their pristine design.</p>
            <div class="d-flex justify-content-between align-items-
center">
                </div>
            </div>
        </div>
        <div class="col">
            <div class="card shadow-sm">
                <figure>
                    
                </figure>

                <div class="card-body">
                    <p class="card-
text">Combat zone has a stong base of satisfied customers, and we tend to keep it
that way. Our quality of service is our number one moto.</p>
                    <div class="d-flex justify-content-between align-items-
center">
                        </div>
                    </div>
                </div>
            </div>
        </div>

        </div>
    </div>
</div>

<div class="container">
    <hr>
</div>

<div class="container my-5">
    <h3 class="lead display-5 mb-4">What others think about us</h3>
    <!-- Three columns of text below the carousel -->
    <div class="row my-5">
        <div class="col-lg-4 text-center">

```

```



    <h4 class="lead my-3">Ben Askren</h4>
    <p>I've been a fan of Combat Zone from day one. I'm really impressed with the quality and desing, not to mention the prices.</p>
    </div><!-- /.col-lg-4 -->
    <div class="col-lg-4 text-center">
        

            <h4 class="lead my-3">Jorge Masvidal</h4>
            <p>If you are like me and like to change gloves frequently I would suggest looking into their newest Venum collection, its awesome.</p>
            </div><!-- /.col-lg-4 -->
            <div class="col-lg-4 text-center">
                

                    <h4 class="lead my-3">Kamaru Ousman</h4>
                    <p>They have gloves for all the seasons, doesn't matter if its boxing of MMA or even karate. They rock with everything.</p>
                    </div><!-- /.col-lg-4 -->
            </div><!-- /.row -->
        </div>

<div class="container">
    <hr>
</div>
```

4.1.34 views/pages/public/author.php

```

<div class="container">

    <div class="row">
        <hr class="featurette-divider">

        <div class="row featurette my-5 d-flex align-items-center">
            <div class="col-md-7">
                <h2 class="lead display-6">Author</h2>
                <p class="lead">My name is Pavle and as you can see... I'm doing just fine. My main goal is to become a professional web developer. I like problem
```

```

solving and building things, that's probably the reason i enjoy web development
so much.</p>
        <hr>
        <h3 class="lead mb-3 display-6">Personal info</h3>
        <p class="lead">Making this got me like:</p>
        <p class="ps-
4">"Holmes, ain't no way in hell I'm using cloth PHP no more..."</p>
        <p class="lead">Favourite book:</p>
        <p>Illiad by Homer, <small>the chaddest book</small></p>
        <p class="lead">Miscellaneous:</p>
        <p>I really hate jQuery</p>
        <hr>
    </div>
    <div class="col-md-5">
        <figure>
            
        </figure>
    </div>
</div>
</div>
</div>

```

4.1.35 views/pages/public/contact.php

```

<?php ?>
<div class="container">
    <div class="row">
        <hr class="featurette-divider">
        <div class="container my-5">
            <div class="row">
                <div class="col-md-6">
                    <figure>
                        
                    </figure>
                </div>
                <div class="col-md-6">
                    <div>
                        <h4 class="text-
primary">Have any questions? <span class="text-
secondary">Send us a message.</span></h4>

```

```

        <p class="lead">Don't hesitate to contact us if you have
any questions, we are here to help.</p>
    </div>
    <form class="" id="messageForm" name="messageForm" action="mo
dels/public/SendMessage.php" method="POST">
        <div class="row g-3">
            <div class="col-sm-6">
                <label for="firstName" class="form-label text-
secondary">First name</label>
                <input type="text" class="form-
control" id="inFname" name="inFname" placeholder="Mark" value=<?= iss
et($_SESSION['loggedUser']) ? ($_SESSION['loggedUser']-
>name_role == 'regular_user' ? $_SESSION['loggedUser']-
>first_name : "") : "" ?>">
                <span clas='inFnameError'> </span>
            </div>
            <div class="col-sm-6">
                <label for="lastName" class="form-label text-
secondary">Last name</label>
                <input type="text" class="form-
control" id="inLname" name="inLname" placeholder="Cuban" value=<?= iss
et($_SESSION['loggedUser']) ? ($_SESSION['loggedUser']-
>name_role == 'regular_user' ? $_SESSION['loggedUser']->last_name : "") : "" ?>">
                <span clas='inLnameError'> </span>
            </div>
            <div class="col-12">
                <label for="email" class="form-label text-
secondary">Email <span class="text-muted"></span></label>
                <input type="email" class="form-
control" id="inEmail" name="inEmail" placeholder="you@example.com" value=<?= iss
et($_SESSION['loggedUser']) ? ($_SESSION['loggedUser']-
>name_role == 'regular_user' ? $_SESSION['loggedUser']-
>email_user : "") : "" ?>">
                <span clas='inEmailError'> </span>
            </div>
            <div class="col-12">
                <label for="subject" class="form-label text-
secondary">Subject <span class="text-muted"></span></label>
                <input type="text" class="form-
control" id="inSubject" name="inSubject" placeholder="Subject" value="">
                <span clas='text-danger'>
            </div>
            <div class="col-12">
                <label for="email" class="form-label text-
secondary">Message <span class="text-muted"></span></label>

```

```

        <textarea class="form-
control" id="taMessage" name="taMessage" cols="30" rows="3"></textarea>
            <span clas='text-danger'>
            </span>
        </div>
        <hr class="my-4">
        <div>
            <?php if (isset($_SESSION['loggedUser'])) ? ($_SES-
SION['loggedUser']->name_role == 'admin' ? false : true) : ?>
                <button class="w-35 btn bg-c-secondary text-
white border btn col-md-5 mb-
3" id="btnSend" name="btnSend" value="btnSend" type="submit">Send</button>
                <?php endif; ?>
            </div>
            <div>
                <!-- <span id="message-sent" class="bg-c-
primary text-white px-3 py-2 rounded">Message successfully sent</span> -->
            </div>
        </form>
    </div>
</div>
</div>
</div>
</div>

```

4.1.36 views/pages/public/contact.php

```

<?php ?>
<div class="container">
    <div class="row">
        <hr class="featurette-divider">
        <div class="container my-5">
            <div class="row">
                <div class="col-md-6">
                    <figure>
                        
                    </figure>
                </div>
                <div class="col-md-6">
                    <div>

```

```

        <h4 class="text-primary">Have any questions? <span class="text-secondary">Send us a message.</span></h4>
            <p class="lead">Don't hesitate to contact us if you have any questions, we are here to help.</p>
        </div>
        <form class="" id="messageForm" name="messageForm" action="models/public/SendMessage.php" method="POST">
            <div class="row g-3">
                <div class="col-sm-6">
                    <label for="firstName" class="form-label text-secondary">First name</label>
                        <input type="text" class="form-control" id="inFname" name="inFname" placeholder="Mark" value=<?= isset($_SESSION['loggedUser']) ? ($_SESSION['loggedUser']->name_role == 'regular_user' ? $_SESSION['loggedUser']->first_name : "") : "" ?>">
                            <span clas='inFnameError'> </span>
                </div>
                <div class="col-sm-6">
                    <label for="lastName" class="form-label text-secondary">Last name</label>
                        <input type="text" class="form-control" id="inLname" name="inLname" placeholder="Cuban" value=<?= isset($_SESSION['loggedUser']) ? ($_SESSION['loggedUser']->name_role == 'regular_user' ? $_SESSION['loggedUser']->last_name : "") : "" ?>">
                            <span clas='inLnameError'> </span>
                </div>
                <div class="col-12">
                    <label for="email" class="form-label text-secondary">Email <span class="text-muted"></span></label>
                        <input type="email" class="form-control" id="inEmail" name="inEmail" placeholder="you@example.com" value=<?= isset($_SESSION['loggedUser']) ? ($_SESSION['loggedUser']->name_role == 'regular_user' ? $_SESSION['loggedUser']->email_user : "") : "" ?>">
                            <span clas='inEmailError'> </span>
                </div>
                <div class="col-12">
                    <label for="subject" class="form-label text-secondary">Subject <span class="text-muted"></span></label>
                        <input type="text" class="form-control" id="inSubject" name="inSubject" placeholder="Subject" value="">
                            <span clas='text-danger'>
                </div>

```

```

<div class="col-12">
    <label for="email" class="form-label text-secondary">Message <span class="text-muted"></span></label>
    <textarea class="form-control" id="taMessage" name="taMessage" cols="30" rows="3"></textarea>
    <span clas='text-danger'>
        </span>
    </div>
    <hr class="my-4">
    <div>
        <?php if (isset($_SESSION['loggedUser'])) ? ($_SESSION['loggedUser']->name_role == 'admin' ? false : true) : true ?>
            <button class="w-35 btn bg-c-secondary text-white border btn col-md-5 mb-3" id="btnSend" name="btnSend" value="btnSend" type="submit">Send</button>
        <?php endif; ?>
    </div>
    <div>
        <!-- <span id="message-sent" class="bg-c-primary text-white px-3 py-2 rounded">Message successfully sent</span> -->
    </div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>

```

4.1.37 views/pages/public/index.php

```

<div class="container mt-5">
    <div class="row">
        <hr class="featurette-divider">

        <div class="row featurette my-5 d-flex align-items-center">
            <div class="col-md-7">
                <h4 class="display-6 lead">Need a glove for a fight? <span class="text-muted">It'll blow your mind.</span></h4>
                <p class="lead">Match any combination and rock every style with our gloves. Don't believe us?<a href="shop.php" class="text-muted bg-light"> Check them out</a></p>

```

```

        </div>
        <div class="col-md-5">
            <figure>
                
            </figure>
        </div>
    </div>
</div>
<section class="py-5 text-center bg-light">
    <div class="container">
        <div class="row py-lg-5">
            <div class="col-lg-6 col-md-8 mx-auto">
                <h1 class="fw-light">Be the change you desire</h1>
                <p class="lead text-muted">All of our gloves are certified and proven to be the best by many awards we've gotten over the past 15 years.</p>
                <p>
                    <a href='index.php?page=shop' class="btn bg-c-secondary text-white my-2">Start shopping</a>
                </p>
            </div>
        </div>
    </div>
</section>
<div class="container">
    <div class="row">
        <div class="row featurette my-5 d-flex align-items-center">
            <div class="col-md-7 order-md-2">
                <h4 class="display-6 lead">Oh yeah, it's that good. <span class="text-muted">See for yourself.</span></h4>
                <p class="lead">Quality and design are our base principles and only the finest is the best. Our products are made entirely of eco-friendly materials.</p>
            </div>
            <div class="col-md-5 order-md-1">
                <figure>
                    
                </figure>
            </div>
        </div>
    </div>
</div>

```

```
</div>
```

4.1.38 views/pages/public/shop.php

```
<?php
$brands = selectAllFromTable("brand");
$categories = selectAllFromTable("category");
$colors = selectAllFromTable("color");
?>
<div id="shop-add-div"></div>

<section class="mt-5">
    <div class="p-5">

        <div class="row">
            <aside class="col-md-3">

                <div class="card">
                    <article class="filter-group">
                        <header class="card-header">
                            <a class="text-primary" href="#" data-
toggle="collapse" data-target="#collapse_1" aria-expanded="true" class="">
                                <i class="icon-control fa fa-chevron-down"></i>
                                <h6 class="title">Search for glove</h6>
                            </a>
                        </header>
                        <div class="filter-
content collapse show" id="collapse_1">
                            <div class="card-body">
                                <form class="pb-3">
                                    <div class="input-group">
                                        <input type="text" class="form-
control" id="search" name="search" placeholder="Search">
                                            <div class="input-group-append">
                                                <button class="btn btn-
light" type="button"><i class="fa fa-search"></i></button>
                                            </div>
                                        </div>
                                    </form>
                                </div> <!-- card-body.-->
                            </div>
                        </article> <!-- filter-group .-->
                    </div>
                </div>
            </aside>
        </div>
    </div>
</section>
```

```

<!-- #region -->
<article class="filter-group">
    <header class="card-header">
        <a href="#" class="text-primary" data-
toggle="collapse" data-target="#collapse_2" aria-expanded="true" class="">
            <i class="icon-control fa fa-chevron-down"></i>
            <h6 class="title">Glove type</h6>
        </a>
    </header>
    <div class="p-3" id="collapse_2">
        <div id="categories">
            <?php foreach ($categories as $i => $c) : ?>
                <label class="custom-control custom-
checkbox d-flex justify-content-between mb-2">
                    <div>
                        <input type="checkbox" value='<?= $c-
>id_cat ?>' name="categories" id="cat-<?= $c->id_cat ?>" class="custom-control-
input"> <?= $c->name_cat ?>
                    </div>
                    <span class="badge badge-pill badge-
danger bg-primary"></span>
                </label>
            <?php endforeach ?>
        </div>
    </div>
</article> <!-- filter-group .// -->
<!-- #endregion -->

<!-- #region -->
<article class="filter-group">
    <header class="card-header">
        <a href="#" class="text-primary" data-
toggle="collapse" data-target="#collapse_2" aria-expanded="true" class="">
            <i class="icon-control fa fa-chevron-down"></i>
            <h6 class="title">Brand</h6>
        </a>
    </header>
    <div class="p-3" id="collapse_2">
        <div id="brands">
            <?php foreach ($brands as $i => $b) : ?>
                <label class="custom-control custom-
checkbox d-flex justify-content-between mb-2">
                    <div>

```

```

                <input type="checkbox" value='<?= $b->
>id_brand ?>' name="brands" id="cat-<?= $b->id_brand ?>" class="custom-control-
input"> <?= $b->name_brand ?>
            </div>
            <span class="badge badge-pill badge-
danger bg-primary"></span>
        </label>
    <?php endforeach ?>
</div> <!-- card-body.-->
</div>
</article> <!-- filter-group .-->
<!-- #endregion -->

<article class="filter-group">
    <header class="card-header">
        <a href="#" class="text-primary" data-
toggle="collapse" data-target="#collapse_2" aria-expanded="true" class="">
            <i class="icon-control fa fa-chevron-down"></i>
            <h6 class="title">Color</h6>
        </a>
    </header>
    <div class="p-3" id="collapse_2">
        <div id="colors">
            <?php foreach ($colors as $i => $c) : ?>
            <label class="custom-control custom-
checkbox d-flex justify-content-between mb-2">
                <div>
                    <input type="checkbox" value='<?= $c->
>id_color ?>' name="colors" id="cat-<?= $c->id_color ?>" class="custom-control-
input"> <?= ucfirst($c->color_name) ?>
                </div>
                <span class="badge badge-pill badge-
danger bg-primary"></span>
            </label>
        <?php endforeach ?>
    </div> <!-- card-body.-->
</div>
</article> <!-- filter-group .-->

<article class="filter-group">
    <header class="card-header">

```

```

        <a href="#" class="text-primary" data-
        toggle="collapse" data-target="#collapse_3" aria-expanded="true" class="">
            <i class="icon-control fa fa-chevron-down"></i>
            <h6 class="title">Price range</h6>
        </a>
    </header>
    <div class="filter-
content collapse show" id="collapse_3">
        <div class="card-body">
            <input type="range" class="custom-
range" min="0" step="25" max="100" value="100" name="range" id="range">
            <span id='range-value'>$100</span>
        </div><!-- card-body.-->
    </div>
</article> <!-- filter-group .-->
</div> <!-- card.-->

</aside> <!-- col.-->
<main class="col-md-9">

    <header class="border-bottom mb-4 pb-3">
        <div>
            <p class="p-
0">Available gloves: <span id="gloveCount"></span></p>
        </div>
        <div class="form-inline">
            <div class="mb-2">
                <div>
                    <label class="lead">Gloves per page: </label>
                    <select id="perPage" class="form-
select" name="perPage" id="perPage">
                        <option value="4">4</option>
                        <option value="6">6</option>
                        <option value="8" selected>8</option>
                        <option value="12">12</option>
                        <option value="24">24</option>
                    </select>
                </div>
            </div>
            <div class="mb-2">
                <label class="lead">Sort: </label>
                <select class="mr-2 form-
select" id="sort" name="sort">
                    <option value="">Select</option>
                    <option value="nameAsc">Name A-Z</option>

```

```

        <option value="nameDesc">Name Z-A</option>
        <option value="newest">Latest</option>
        <option value="priceDesc">Most Expensive</option>
        <option value="priceAsc">Least Expensive</option>
    </select>
</div>
</div>
</header><!-- sect-heading -->
<div class="row" id="gloves">

</div> <!-- row end.// -->

<nav aria-label="...">
    <ul class="pages pagination" id="pages">

    </ul>
</nav>

</main> <!-- col.// -->

</div>

</div> <!-- container .// -->
</section>
<!-- ===== SECTION CONTENT END// ===== -->

```

4.1.39 views/pages/users/cart.php

```

<div class="container">
    <div>
        <p class="lead">Items in cart: <span id="numItemsCart"></span></p>
    </div>
    <table class="table table-striped table-bordered text-center">
        <thead>
            <tr>
                <th scope="col">No.</th>
                <th scope="col">Image</th>
                <th scope="col">Name</th>
                <th scope="col">Category</th>
                <th scope="col">Brand</th>
                <th scope="col">Color</th>
            </tr>
        <tbody>
            <tr>
                <td>1</td>
                <td><img alt="Product Image" style="width: 50px;"></td>
                <td>Leather Gloves</td>
                <td>Accessories</td>
                <td>Gucci</td>
                <td>Black</td>
            </tr>
            <tr>
                <td>2</td>
                <td><img alt="Product Image" style="width: 50px;"></td>
                <td>Silk Scarf</td>
                <td>Accessories</td>
                <td>Prada</td>
                <td>Red</td>
            </tr>
            <tr>
                <td>3</td>
                <td><img alt="Product Image" style="width: 50px;"></td>
                <td>Silk Tie</td>
                <td>Accessories</td>
                <td>Tom Ford</td>
                <td>Blue</td>
            </tr>
        </tbody>
    </table>
</div>

```

```

        <th scope="col">Size</th>
        <th scope="col">Price</th>
        <th scope="col">Quantity</th>
        <th scope="col">Total</th>
        <th scope="col">Remove</th>
    </tr>
</thead>
<tbody id="cart" class="table-striped">

    </tbody>
</table>
</div>
<section class="section-content padding-y">
    <div class="container">

        <div class="row">
            <main class="col-md-9">
                <div class="card">
                    <div>
                        <!-- <input type="hidden" name="cart-user-id" id="cart-user-id" value="php echo $_SESSION['loggedUser']-user_id ?>" -->
                    </div>
                    <div class="card-body border-top">
                        <a href="#" class="btn bg-c-ternary text-white box-shadow float-md-right" id="purchase"> Make Purchase </a>
                        <a href="index.php?page=shop" class="btn box-shadow"> <i class="fa fa-chevron-left"></i> Continue shopping </a>
                        <br>
                        <span class="lead bg-c-primary text-white mt-3 box-shadow rounded mt-2 d-inline-block" id="purchased"></span>
                    </div>
                    <p class="lead text-success ms-3" id="cart-success"></p>
                </div> <!-- card.-->

                <div class="alert text-white bg-c-secondary mt-3 d-flex">
                    <span class="material-icons">
                        local_shipping
                    </span>
                    <p class="icontext ms-2">Free Delivery within 1-2 weeks</p>
                </div>

            </main> <!-- col.-->
            <aside class="col-lg-3">
                <div class="card">
                    <div class="card-body">

```

```

        <div class="card-title">
            <h3 class="lead">Order</h3>
        </div>
        <dl class="dlist-align">
            <dt>Price:</dt>
            <dd class="text-
right">&dollar;<span id='price'></span></dd>
        </dl>
        <hr>
        <dl class="dlist-align">
            <dt>Total:</dt>
            <dd class="text-
right h5">&dollar;<strong id="totalPrice"></strong></dd>
        </dl>
    </div> <!-- card .// -->
</aside> <!-- col./. -->
</div>

</div> <!-- container .// -->
</section>

```

4.1.40 views/pages/users/admin/admin-dashboard.php

```

<?php
if (!isset($_SESSION['loggedUser'])) {
    header("Location: index.php");
} else {
    $user = $_SESSION['loggedUser'];
    if ($user->name_role != "admin") {
        // User is unauthorized
        header("Location: index.php");
    }
}
// equalizer
$visits = returnPageVisits();
list($total, $pages) = $visits;
?>
<div class="container mt-5">
    <div class="lead text-white bg-c-primary px-5 py-3 d-flex justify-content-
between align-items-center" style="font-size: 1.8rem!important;">
        <h3 class="lead">Total visits: <?= $total ?></h3>
    </div>
    <div class="mb-5 px-5 py-3">

```

```

<?php foreach ($pages as $i => $p) : ?>
    <div class="box-shadow my-4">
        <div class="lead" style="font-size: 1.5rem!important;">
            <div class="d-inline-block px-2 py-1">
                <span class="material-icons text-secondary">
                    description
                </span>
                <span>
                    Page: <?= $i ?>
                </span>
            </div>
        </div>
        <div class="lead" style=" font-size: 1.5rem!important;">
            <div class="d-inline-block px-2 py-1">
                <span class="material-icons text-secondary">
                    equalizer
                </span>
                <span>
                    Visits:(<?= $p ?>) <?= round((+$p / +$total) * 100, 2
                ) ?>%
                </span>
            </div>
        </div>
    </div>
    <?php endforeach; ?>
</div>
</div>

```

4.1.41 views/pages/users/admin/admin-gloves.php

```

<?php
if (!isset($_SESSION['loggedUser'])) {
    header("Location: index.php");
} else {
    $user = $_SESSION['loggedUser'];
    if ($user->name_role != "admin") {
        // User is unauthorized
        header("Location: index.php");
    }
}
// Variables
$brands = selectAllFromTable("brand");
$categories = selectAllFromTable("category");

```

```
$colors = selectAllFromTable("color");

// Checks if action paramter is set
if (isset($_GET['mode'])) {
    $mode = $_GET['mode'];
    include V_ADMIN_MODE . "/gloves.php";
} else {
?>
    <div class="container">
        <div class="col-12 my-5">
            <div class="form-group">
                <a href="index.php?page=admin-gloves&mode=insert" class="box-
shadow btn bg-c-ternary d-inline-flex align-items-center text-white">
                    <span class="material-icons">
                        add_circle
                    </span>
                    <span class="ms-2">Add new glove</span>
                </a>
            </div>
        </div>
        <hr>
        <div>
            <article class="filter-group">
                <header class="card-header bg-c-primary">
                    <h6 class="lead text-white">Search for glove</h6>
                </header>
                <div class="filter-content collapse show" id="collapse_1">
                    <div class="card-body">
                        <form class="pb-3">
                            <div class="input-group d-flex align-items-center">
                                <span class="material-icons align-bottom ">
                                    search
                                </span>
                                <input type="text" class="form-
control" id="search" name="search" placeholder="Search">
                            </div>
                        </form>
                    </div> <!-- card-body.-->
                </div>
            </article> <!-- filter-group .-->
            <div>
                <p class="p-
0">Available gloves: <span id="gloveCount"></span></p>
            </div>
            <div class="form-inline">
```

```

<div class="mb-2">
    <div>
        <label class="lead">Gloves per page: </label>
        <select id="perPage" class="form-select" name="perPage" id="perPage">
            <option value="4" selected>4</option>
            <option value="6">6</option>
            <option value="8">8</option>
            <option value="12">12</option>
            <option value="24">24</option>
        </select>
    </div>
</div>
<div class="mb-2">
    <div>
        <label class="lead">Categories: </label>
        <select class="form-select" name="categories" id="categories">
            <option value="">Select</option>
            <?php foreach ($categories as $i => $c) : ?>
                <option value="<?= $c->id_cat ?>"><?= $c->name_cat ?></option>
            <?php endforeach ?>
        </select>
    </div>
</div>
<div class="mb-2">
    <div>
        <label class="lead">Brands: </label>
        <select class="form-select" name="brands" id="brands">
            <option value="">Select</option>
            <?php foreach ($brands as $i => $b) : ?>
                <option value="<?= $b->id_brand ?>"><?= $b->name_brand ?></option>
            <?php endforeach ?>
        </select>
    </div>
</div>
<div class="mb-2">
    <div>
        <label class="lead">Colors: </label>
        <select class="form-select" name="colors" id="colors">
            <option value="">Select</option>
            <?php foreach ($colors as $i => $c) : ?>

```

```

                <option value="<?= $c-
>id_color ?>"><?= ucfirst($c->color_name) ?></option>
                    <?php endforeach ?>
                </select>
            </div>
        </div>
        <div class="mb-2">
            <label class="lead">Sort: </label>
            <select class="mr-2 form-select" id="sort" name="sort">
                <option value="">Select</option>
                <option value="nameAsc">Name A-Z</option>
                <option value="nameDesc">Name Z-A</option>
                <option value="newest">Latest</option>
                <option value="priceDesc">Most Expensive</option>
                <option value="priceAsc">Least Expensive</option>
            </select>
        </div>
    </div>
    <article class="filter-group">
        <header class="card-header bg-c-primary">
            <h6 class="lead text-white">Price range</h6>
        </header>
        <div class="filter-content collapse show" id="collapse_3">
            <div class="card-body">
                <input type="range" class="custom-
range" min="0" step="25" max="100" value="100" name="range" id="range">
                <span id='range-value'>$100</span>
            </div><!-- card-body.-->
        </div>
    </article> <!-- filter-group .-->
</div>
<div class="my-5">
    <table class="table table-striped table-bordered text-center">
        <thead>
            <tr>
                <th scope="col">No.</th>
                <th scope="col">Image</th>
                <th scope="col">Name</th>
                <th scope="col">Category</th>
                <th scope="col">Brand</th>
                <th scope="col">Color</th>
                <th scope="col">Sizes</th>
                <th scope="col">Measure</th>
                <th scope="col">Date</th>
                <th scope="col">Edit</th>
            
```

```

                <th scope="col">Delete</th>
            </tr>
        </thead>
        <tbody id="glovesTableBody" class="table-striped align-center">
            <!-- Insert gloves -->
        </tbody>
    </table>
</div>
<nav aria-label="...">
    <ul class="pages pagination" id="pages">
        </ul>
</nav>
</div>
<?php
}
?>
```

4.1.42 views/pages/users/admin/admin-message.php

```

<?php
if (!isset($_GET['id'])) {
    // Order detail id was not set
    header("Location: index.php");
}
if (!isset($_SESSION['loggedUser'])) {
    // User tried to access admin only page
    header("Location: index.php");
} else {
    $user = $_SESSION['loggedUser'];
    if ($user->name_role != "admin") {
        // Customer tried to access admin only page
        header("Location: index.php");
    }
}
$id = $_GET['id'];
$m = returnMessageInfo($id)[0];
?>
<div class="container mt-5">
    <div class="mb-5">
        <a href="index.php?page=admin-messages" class="box-shadow btn bg-c-
secondary d-inline-flex align-items-center text-white">
            <span class="material-icons me-2">
                arrow_back
            </span>
```

```
        <span>All Messages</span>
    </a>
</div>
<div>
    <h3 class="lead text-white bg-c-primary px-5 py-3" style="font-size: 1.8rem!important;">Message</h3>
    <div class="mb-5 box-shadow px-5 py-3">
        <div class="lead my-2" style="font-size: 1.5rem!important;">
            <div class="box-shadow d-inline-block px-2 py-1">
                <span class="material-icons text-secondary align-center">
                    person
                </span>
                <span>
                    Sender
                </span>
            </div>
            <p class="lead mt-3"><?= $m->msg_first_name ?> <?= $m->msg_last_name ?></p>
        </div>
        <div class="lead my-2" style="font-size: 1.5rem!important;">
            <div class="box-shadow d-inline-block px-2 py-1">
                <span class="material-icons text-secondary align-bottom">
                    event
                </span>
                <span>
                    Date received
                </span>
            </div>
            <p class="lead mt-3"><?= $m->msg_date ?></p>
        </div>
        <div class="lead my-2" style="font-size: 1.5rem!important;">
            <div class="box-shadow d-inline-block px-2 py-1">
                <span class="material-icons text-secondary align-bottom">
                    short_text
                </span>
                <span>
                    Subject
                </span>
            </div>
            <p class="lead mt-3"><?= $m->msg_subject ?></p>
        </div>
        <div class="lead my-2" style="font-size: 1.5rem!important;">
            <div class="box-shadow d-inline-block px-2 py-1">
                <span class="material-icons text-secondary align-bottom">
                    article
                </span>
            </div>
        </div>
    </div>
</div>
```

```

        </span>
        <span>
            Message body
        </span>
    </div>
    <p class="lead mt-3"><?= $m->msg_body ?></p>
</div>
</div>
</div>

```

4.1.43 views/pages/users/admin/admin-messages.php

```

<?php
if (!isset($_SESSION['loggedUser'])) {
    header("Location: index.php");
} else {
    $user = $_SESSION['loggedUser'];
    if ($user->name_role != "admin") {
        // User is unauthorized
        header("Location: index.php");
    }
}

$tableId = "messagesTableBody";
$tableCount= "messagesCount";
$type = "messages";
$tableColumns = ["No.", "Full name", "Email", "Subject", "Date", "More details"];
$tableIcon = "drafts";
include FIXED . "./admin-table.php";

```

4.1.44 views/pages/users/admin/order.php

```

<?php
if (!isset($_GET['id'])) {
    // Order detail id was not set
    header("Location: index.php");
}
if (!isset($_SESSION['loggedUser'])) {
    // User tried to access admin only page
    header("Location: index.php");
}

```

```

} else {
    $user = $_SESSION['loggedUser'];
    if ($user->name_role != "admin") {
        // Customer tried to access admin only page
        header("Location: index.php");
    }
}
$id = $_GET['id'];
$order = returnOrderDetailInfo($id);
list($order_lines, $u, $count) = $order;
$total = 0;
if (!isset($order) || empty($order[0])) {
    header("Location: index.php");
}


```

```
        <a href="models/users/admin/orders/Excel.php?id=<?= $_GET['id'] ?>" id="download-order-detail-excel" data-orderid='<?= $_GET["id"] ?>' class="d-inline-block d-flex align-items-center text-white justify-content-between">
            <span class="material-icons ">
                file_download
            </span>
            <span>
                Excel
            </span>
        </a>
    </div>
</h3>
<div class="mb-5 box-shadow px-5 py-3">
    <div class="lead my-2" style="font-size: 1.5rem!important;">
        <div class="box-shadow d-inline-block px-2 py-1">
            <span class="material-icons text-secondary">
                pin
            </span>
            <span>
                Order ID
            </span>
        </div>
        <p class="lead mt-3"><a class="text-secondary" href="index.php?page=admin-user&id=<?= $u->id_user ?>"><?= isset($_GET['id']) ? $_GET["id"] : "" ?></a></p>
    </div>
    <div class="lead my-2" style="font-size: 1.5rem!important;">
        <div class="box-shadow d-inline-block px-2 py-1">
            <span class="material-icons text-secondary">
                person
            </span>
            <span>
                Customer
            </span>
        </div>
        <p class="lead mt-3"><a class="text-secondary" href="index.php?page=admin-user&id=<?= $u->id_user ?>"><?= $u->first_name ?> <?= $u->last_name ?></a></p>
    </div>
    <div class="lead my-2" style="font-size: 1.5rem!important;">
        <div class="box-shadow d-inline-block px-2 py-1">
            <span class="material-icons text-secondary">
                event
            </span>
            <span>
```

```

        Date of purchase
    </span>
</div>
<p class="lead mt-3"><?= $order_lines[0]->date ?></p>
</div>
<div class="lead my-2" style="font-size: 1.5rem!important;">
    <div class="box-shadow d-inline-block px-2 py-1">
        <span class="material-icons text-secondary">
            shopping_cart
        </span>
        <span>
            Number of items
        </span>
    </div>
    <p class="lead mt-3"><?= $num ?></p>
</div>
<div class="lead my-2" style="font-size: 1.5rem!important;">
    <div class="box-shadow d-inline-block px-2 py-1">
        <span class="material-icons text-secondary">
            request_quote
        </span>
        <span>
            Total
        </span>
    </div>
    <p class="lead mt-3">$ <?= $total ?></p>
</div>
</div>
</div>
<?php
include FIXED . "./admin-table.php";
?>
```

4.1.45 views/pages/users/admin/admin-orders.php

```

<?php
if (!isset($_GET['id'])) {
    // Order detail id was not set
    header("Location: index.php");
}
if (!isset($_SESSION['loggedUser'])) {
    // User tried to access admin only page
```

```

        header("Location: index.php");
    } else {
        $user = $_SESSION['loggedUser'];
        if ($user->name_role != "admin") {
            // Customer tried to access admin only page
            header("Location: index.php");
        }
    }
$id = $_GET['id'];
$order = returnOrderDetailInfo($id);
list($order_lines, $u, $count) = $order;
$total = 0;
if (!isset($order) || empty($order[0])) {
    header("Location: index.php");
}


```

```
        <a href="models/users/admin/orders/Excel.php?id=<?= $_GET['id'] ?>" id="download-order-detail-excel" data-orderid='<?= $_GET["id"] ?>' class="d-inline-block d-flex align-items-center text-white justify-content-between">
            <span class="material-icons ">
                file_download
            </span>
            <span>
                Excel
            </span>
        </a>
    </div>
</h3>
<div class="mb-5 box-shadow px-5 py-3">
    <div class="lead my-2" style="font-size: 1.5rem!important;">
        <div class="box-shadow d-inline-block px-2 py-1">
            <span class="material-icons text-secondary">
                pin
            </span>
            <span>
                Order ID
            </span>
        </div>
        <p class="lead mt-3"><a class="text-secondary" href="index.php?page=admin-user&id=<?= $u->id_user ?>"><?= isset($_GET['id']) ? $_GET["id"] : "" ?></a></p>
    </div>
    <div class="lead my-2" style="font-size: 1.5rem!important;">
        <div class="box-shadow d-inline-block px-2 py-1">
            <span class="material-icons text-secondary">
                person
            </span>
            <span>
                Customer
            </span>
        </div>
        <p class="lead mt-3"><a class="text-secondary" href="index.php?page=admin-user&id=<?= $u->id_user ?>"><?= $u->first_name ?> <?= $u->last_name ?></a></p>
    </div>
    <div class="lead my-2" style="font-size: 1.5rem!important;">
        <div class="box-shadow d-inline-block px-2 py-1">
            <span class="material-icons text-secondary">
                event
            </span>
            <span>
```

```

        Date of purchase
    </span>
</div>
<p class="lead mt-3"><?= $order_lines[0]->date ?></p>
</div>
<div class="lead my-2" style="font-size: 1.5rem!important;">
    <div class="box-shadow d-inline-block px-2 py-1">
        <span class="material-icons text-secondary">
            shopping_cart
        </span>
        <span>
            Number of items
        </span>
    </div>
    <p class="lead mt-3"><?= $num ?></p>
</div>
<div class="lead my-2" style="font-size: 1.5rem!important;">
    <div class="box-shadow d-inline-block px-2 py-1">
        <span class="material-icons text-secondary">
            request_quote
        </span>
        <span>
            Total
        </span>
    </div>
    <p class="lead mt-3">$ <?= $total ?></p>
</div>
</div>
</div>
<?php
include FIXED . "./admin-table.php";
?>
```

4.1.46 views/pages/users/admin/user.php

```

<?php
if (!isset($_GET['id'])) {
    // Order detail id was not set
    header("Location: index.php");
}
if (!isset($_SESSION['loggedUser'])) {
    // User tried to access admin only page
```

```

        header("Location: index.php");
    } else {
        $user = $_SESSION['loggedUser'];
        if ($user->name_role != "admin") {
            // Customer tried to access admin only page
            header("Location: index.php");
        }
    }
$id = $_GET['id'];
$u = selectAllFromTableWhere("user", "id_user", $id);
if (!$u) {
    header("Location: index.php");
}
?>
<div class="container">
    <div class="my-5">
        <a href="index.php?page=admin-users" classclass="material-icons me-2">
                arrow_back
            </span>
            <span>All Users</span>
        </a>
    </div>
    <div>
        <h3 class="lead text-white bg-c-primary px-5 py-3" style="font-size: 1.8rem!important;">User</h3>
        <div class="mb-5 box-shadow px-5 py-3">
            <div class="lead my-2" style="font-size: 1.5rem!important;">
                <div class="box-shadow d-inline-block px-2 py-1">
                    <span class="material-icons text-secondary align-center">
                        person
                    </span>
                    <span>
                        Full name
                    </span>
                </div>
                <p class="lead mt-3"><?= $u->first_name ?> <?= $u->last_name ?></p>
            </div>
            <div class="lead my-2" style="font-size: 1.5rem!important;">
                <div class="box-shadow d-inline-block px-2 py-1">
                    <span class="material-icons text-secondary align-center">
                        email
                    </span>
                </div>
            </div>
        </div>
    </div>

```

```
        <span>
            Email
        </span>
    </div>
    <p class="lead mt-3"><?= $u->email_user ?></p>
</div>
<div class="lead my-2" style="font-size: 1.5rem!important;">
    <div class="box-shadow d-inline-block px-2 py-1">
        <span class="material-icons text-secondary align-center">
            home
        </span>
        <span>
            Address
        </span>
    </div>
    <p class="lead mt-3"><?= $u->address ?></p>
</div>
<div class="lead my-2" style="font-size: 1.5rem!important;">
    <div class="box-shadow d-inline-block px-2 py-1">
        <span class="material-icons text-secondary align-center">
            phone_iphone
        </span>
        <span>
            Phone number
        </span>
    </div>
    <p class="lead mt-3"><?= $u->phone_num ?></p>
</div>
</div>
<div class="mt-5">
    <div>
        <p class="p-0">Number of orders: <span id="ordersCount"></span></p>
    </div>
    <div class="form-inline">
        <div class="mb-2">
            <div>
                <label class="lead">Orders per page: </label>
                <select id="perPage" class="form-select" name="perPage" id="perPage">
                    <option value="1">1</option>
                    <option value="2">2</option>
                    <option value="4" selected>4</option>
                    <option value="6">6</option>
                    <option value="8">8</option>
                </select>
            </div>
        </div>
    </div>
</div>
```

```

        <option value="12">12</option>
        <option value="24">24</option>
    </select>
</div>
</div>
<div class="form-inline">
    <div class="mb-2">
        <label class="lead">Sort: </label>
        <select class="mr-2 form-select" id="sort" name="sort">
            <option value="">Select</option>
            <option value="latest">Latest</option>
            <option value="numberOfItems">Number of items</option>
            <option value="priceDesc">Most Expensive</option>
            <option value="priceAsc">Least Expensive</option>
        </select>
    </div>
</div>
<div class="my-5">
    <h3 class="lead" style="font-
size: 1.8rem" id="tableName">Customer's orders</h3>
    <table class="table table-striped table-bordered text-center">
        <thead>
            <tr>
                <th scope="col">No.</th>
                <th scope="col">Customer</th>
                <th scope="col">Items</th>
                <th scope="col">Total</th>
                <th scope="col">Date</th>
                <th scope="col">More details</th>
            </tr>
        </thead>
        <tbody id="ordersTableBody" class="table-striped align-center">
            <!-- Insert orders -->
        </tbody>
    </table>
</div>
<nav aria-label="...">
    <ul class="pages pagination" id="pages">

    </ul>
</nav>
</div>

```

4.1.47 views/pages/users/admin/admin-users.php

```
<?php
if (!isset($_SESSION['loggedUser'])) {
    header("Location: index.php");
} else {
    $user = $_SESSION['loggedUser'];
    if ($user->name_role != "admin") {
        // User is unauthorized
        header("Location: index.php");
    }
}
$tableId = "usersTableBody";
$tableCount = "usersCount";
$type = "users";
$tableColumns = ["No.", "Customer", "Email", "Orders", "Total spent", "More details", "Account status"];
$tableIcon = "person";
include FIXED . "./admin-table.php";
```

4.1.48 views/pages/users/admin/mode/gloves.php

```
<?php
include_once M_ADMIN . "/Admin.php";
if (!isset($_SESSION['loggedUser'])) {
    header("Location: index.php");
} else {
    $user = $_SESSION['loggedUser'];
    if ($user->name_role != "admin") {
        // User is unauthorized
        header("Location: index.php");
    }
}
if (isset($_GET["mode"])) {
    // Determine which mode is it
    $brands = selectAllFromTable("brand");
    $categories = selectAllFromTable("category");
    $colors = selectAllFromTable("color");

    $mode = $_GET['mode'];
    if ($mode == 'edit') {
        // Edit mode
```

```

if (isset($_GET['id'])) {
    $id = $_GET['id'];
    // Get the values for glove with ID
    // An object that contains all info for selected glove
    // FK are represented as id's
    // Image names(paths)
    // Stringified array of sizes represented as id's
    $filters = selectCurrentGloveValues($id);
    if (!isset($filters) || empty($filters)) {
        header("Location: index.php");
    }
    // display($filters);
    $color = $filters->colorID;
    $cat = $filters->catID;
    $brand = $filters->brandID;
    $name = $filters->name;
    $desc = $filters->description;
    $price = $filters->price;
    $n_img = $filters->n_img;
    $t_img = $filters->t_img;

    // Array of sizes that need to be checked, converted from string
    $sizesGloveID = explode(",", $filters->sizesID);
    // display($sizesGloveID);

    // Available sizes is an array of objects that contain measure[OZ, Ci
    rcumference], size[m, l, 8, xl, 16], id[1, 2, 3...]
    // Used to extract info for Helpers arrays
    // Used for looping through and checking if ID's match
    // NOT all sizes are present in every glove (8, 10, m, l) thats why w
    e use available sizes
    $availableSizesObj = selectAvailableSizes($cat);
    // display($availableSizesObj);

    foreach ($availableSizesObj as $i => $as) {
        $as = array($as);
        foreach ($as as $i => $s) {
            // Helpers arrays
            $sizesGloveValues[] = $s->size;
            $sizesGloveMeasures[] = $s->measure;
        }
    }
} else {
    // JUST FOR INSERT PAGE
}

```

```

        }
    }
?>
<div class="container">
    <div class="row">
        <div class="container">
            <div class="col-12 mt-5">
                <div class="form-group">
                    <a href="index.php?page=admin-gloves" class="box-
shadow btn bg-c-primary d-inline-flex align-items-center text-white">
                        <span class="material-icons">
                            arrow_back
                        </span>
                        <span class="ms-2">Back to gloves</span>
                    </a>
                </div>
            </div>
        <div class="col-12 my-5">
            <form id="glovesModeForm<?= isset($_GET['id']) ? "Edit" : "Insert" ?>
" action="models/users/admin/gloves/<?= isset($_GET['id']) ? "Edit.php" : "Insert
.php" ?>" method="POST" enctype="multipart/form-data">
                <?php if (isset($_GET['id'])) : ?>
                    <input type="hidden" value=<?= $_GET['id'] ?>" name="inHidde
n" id="inHidden">
                <?php endif; ?>
                <div class="form-row">
                    <!-- BRANDS -->
                    <div class="form-group col-md-7 my-3">
                        <div class="box-shadow btn bg-c-ternary d-inline-
flex align-items-center text-white mb-3">
                            <span class="material-icons">
                                sell
                            </span>
                            <span class="ms-2">Brand</span>
                        </div>
                        <select name="ddlBrand" id="ddlBrand" class="form-
control">
                            <option value="">Select brand</option>
                            <!-- FOREACH -->
                            <?php if (!isset($_GET['id'])) {
                                foreach ($brands as $i => $b) : ?>
                                    <option value=<?= $b->id_brand ?>"><?= $b-
name_brand ?></option>
                            <?php } ?>
                        </select>
                    </div>
                </div>
            </form>
        </div>
    </div>
</div>

```

```

        <?php endforeach;
    } else {
        foreach ($brands as $i => $b) : ?>
            <option value="<?= $b->id_brand ?>" <?= $b-
>id_brand == $brand ? "selected" : "" ?>><?= $b->name_brand ?></option>
            <?php endforeach;
        } ?>
        <!-- ENDFOREACH -->
    </select>
</div>
<!-- CATEGORIES -->
<div class="form-group col-md-7 my-3">
    <div class="box-shadow btn bg-c-ternary d-inline-
flex align-items-center text-white mb-3">
        <span class="material-icons">
            category
        </span>
        <span class="ms-2">Category</span>
    </div>
    <!--
    - Selected category determines what type of sizes will be available
      eg. boxing -> oz; karate/mma -> s/m/l/xl -->
    <select name="ddlCat" id="ddlCat" class="form-control">
        <option value="">Select category</option>
        <!-- FOREACH -->
        <?php if (!isset($_GET['id'])) {
            foreach ($categories as $i => $c) : ?>
                <option value="<?= $c->id_cat ?>" <?= $c-
>name_cat ?></option>
            <?php endforeach;
        } else {
            foreach ($categories as $i => $c) :
            ?>
                <option value="<?= $c->id_cat ?>" <?= $c-
>id_cat == $cat ? "selected" : "" ?>><?= $c->name_cat ?></option>
            <?php endforeach;
        } ?>
        <!-- ENDFOREACH -->
    </select>
</div>
<!-- SIZES -->
<div class="form-group col-md-7 my-3">
    <div class="box-shadow btn bg-c-ternary d-inline-
flex align-items-center text-white mb-3">
        <span class="material-icons">

```

```

                straighten
            </span>
            <span class="ms-2">Sizes</span><br>
        </div>
        <!--
    - Selected category determines what type of sizes will be available
        eg. boxing -> oz; karate/mma -> s/m/l/xl -->
        <div id="cbSizes">
            <?php if (isset($_GET['id'])) {
                foreach ($availableSizesObj as $i => $as) :
            ?>
                <div class="form-check">
                    <input class="form-check-
input" type="checkbox" name="cbSize[]" value="<?= $as->id ?>" <?= in_array($as-
>id, $sizesGloveID) ? "checked" : "" ?> id=""
                    <label class="form-check-
label" for="flexCheckDefault"><?= $sizesGloveValues[$i] ?> <?= $as-
>measure == "OZ" ? $sizesGloveMeasures[$i] : "" ?></label>
                </div>
            <?php endforeach;
            } ?>
        </div>
        </div>
        <!-- COLORS -->
        <div class="form-group col-md-7 my-3">
            <div class="box-shadow btn bg-c-ternary d-inline-
flex align-items-center text-white mb-3">
                <span class="material-icons">
                    palette
                </span>
                <span class="ms-2">Colors</span>
            </div>
            <!--
    - Selected category determines what type of sizes will be available
        eg. boxing -> oz; karate/mma -> s/m/l/xl -->
            <select name="ddlColor" id="ddlColor" class="form-
control">
                <option value="">Select color</option>
                <!-- FOREACH -->
                <?php if (!isset($_GET['id'])) {
                    foreach ($colors as $i => $c) :
                ?>
                    <option value="<?= $c-
>id_color ?>"><?= ucfirst($c->color_name) ?></option>
                <?php endforeach;

```

```

        } else {
            foreach ($colors as $i => $c) :
        ?>
            <option value=<?= $c->id_color ?> <?= $c-
>id_color == $color ? "selected" : "" ?>><?= ucfirst($c->color_name) ?></option>
            <?php endforeach;
        } ?>
        <!-- ENDFOREACH -->
    </select>
</div>
</div>

<div class="form-row">
    <div class="form-group col-md-7 my-3">
        <div class="box-shadow btn bg-c-ternary d-inline-
flex align-items-center text-white mb-3">
            <span class="material-icons">
                description
            </span>
            <span class="ms-2">Product name</span>
        </div>
        <input type="text" placeholder="Leone Pink Fight" value=""
<?= isset($_GET['id']) ? $name : "" ?>" name="inName" id="inName" class="form-
control" />
    </div>
    <div class="form-group col-md-7 my-3">
        <div class="box-shadow btn bg-c-ternary d-inline-
flex align-items-center text-white mb-3">
            <span class="material-icons">
                image
            </span>
            <span class="ms-2">Image</span>
        </div>
        <br>
        <input type="file" class="" id="fileImg" name="fileImg" />
<br>
        <span class="lead d-block my-3">" /></span>
    </div>
    <div class="form-group col-md-7 my-3">
        <div class="box-shadow btn bg-c-ternary d-inline-
flex align-items-center text-white mb-3">

```

```

        <span class="material-icons">
            paid
        </span>
        <span class="ms-2">Price</span>
    </div>
    <input type="text" placeholder="59.99" value=<?= isset($_GET['id']) ? $price : "" ?>" class="form-control" id="inPrice" name="inPrice" />
    </div>
</div>
<div class="form-group col-md-7 my-3 p-0">
    <div class="box-shadow btn bg-c-tertiary d-inline-flex align-items-center text-white mb-3">
        <span class="material-icons">
            article
        </span>
        <span class="ms-2">Description</span>
    </div>
    <textarea name="taDesc" id="taDesc" class="form-control" placeholder="Enter description"><?= isset($_GET['id']) ? $desc : "" ?></textarea>
    </div>
    <div class="my-5 col-md-7">
        <hr>
    </div>
    <div class="form-group">
        <button type="submit" name="btnSubmit" id="btnSubmit" class="text-white" value="btnSubmit">
            <div class="box-shadow btn bg-c-secondary d-inline-flex align-items-center text-white">
                <span class="material-icons">
                    add_circle
                </span>
                <span class="ms-2"><?= !isset($_GET['id']) ? "Insert glove" : "Update glove" ?></span>
            </div>
        </button>
    </div>
    </div>
</div>

```

4.2 CSS

4.2.1 style.css

```
.bg-c-primary {  
    background-color: #0d2137 !important;  
    /* background-color: #1565c0 !important; */  
}  
.bg-c-secondary {  
    background-color: #2e77ae !important;  
    /* background-color: #1e88e5 !important; */  
}  
.bg-c-ternary {  
    background-color: #ff8e2b !important;  
    /* background-color: #1e88e5 !important; */  
}  
  
.text-primary {  
    color: #0d2137 !important;  
    /* color: #1565c0 !important; */  
}  
.text-secondary {  
    color: #2e77ae !important;  
    /* color: #1e88e5 !important; */  
}  
.text-ternary {  
    color: #ff8e2b !important;  
    /* color: #1e88e5 !important; */  
}  
.active {  
    color: #ff8e2b !important;  
}  
a {  
    text-decoration: none;  
    cursor: pointer;  
}  
button {  
    border: none !important;  
    background-color: transparent;  
}  
input[type="file"]::focus {  
    cursor: pointer !important;  
}
```

```
.btn:focus {
  box-shadow: none !important;
  outline: none !important;
}
select:focus,
input:focus,
textarea:focus {
  box-shadow: 0 0.5rem 1rem rgba(0, 0, 0, 0.3) !important;
  border-color: #ff8e2b !important;
}
.box-shadow {
  box-shadow: 0 0.5rem 1rem rgba(0, 0, 0, 0.3) !important;
  /* box-shadow: 5px 15px 5px 0px #000; */
}
.box-shadow-hover:hover {
  box-shadow: 0 0.5rem 1rem rgba(0, 0, 0, 0.3) !important;
  /* box-shadow: 5px 15px 5px 0px #000; */
}
html {
  overflow-x: hidden;
}

.btn,
.btn:link,
.btn:visited {
  text-decoration: none;
  display: inline-block;
  transition: all 0.2s;
  position: relative;
  cursor: pointer;
  outline: none !important;
}
.btn:hover {
  transform: translateY(-0.1rem);
  box-shadow: 0 0.5rem 1rem rgba(0, 0, 0, 0.2);
}
.btn:active,
.btn:focus {
  outline: none !important;
  transform: translateY(0.1rem);
  box-shadow: 0 0.5rem 1rem rgba(0, 0, 0, 0.2) !important;
  border: none !important;
}
.navbar-toggler {
  box-shadow: none !important;
```

```
}

.acc {
  transition: all 0.2s;
}
.acc:hover {
  color: #ff8e2b !important;
}
#cartNav {
  font-size: 0.8rem;
  background: #2e77ae !important;
  color: #fff;
  padding: 2px 6px;
  vertical-align: top;
  margin-left: -1.5rem;
  top: 7%;
}
.badge {
  padding-left: 9px;
  padding-right: 9px;
  -webkit-border-radius: 9px;
  -moz-border-radius: 9px;
  border-radius: 9px;
}

.fit-cover {
  float: left;
  width: 25rem;
  height: 10rem;
  object-fit: cover;
}

.palette {
  float: left;
  height: 20px;
  width: 20px;
  margin-bottom: 15px;
  margin-top: 15px;
  border: 1px solid black;
  clear: both;
}
.capitalize {
  text-transform: capitalize;
}
.active-page {
```

```

background-color: #ff8e2b !important;
color: #fff;
}
.active-page:hover {
color: #ff8e2b !important;
background-color: #fff !important;
}
input[type="number"] {
width: 3.2rem !important;
}

#hero-slider {
background-image: url("../img/hero/hero1.jpg");
background-size: cover;
background-position: center;
height: 50vh;
}

```

4.3 JS

4.3.1 main.js

```

$(document).ready(() => {
// Products class
class Data {
// Returns a Order Line object whic will be inserted into order
static returnOrderLine() {
const raSizes = document.querySelectorAll("input[name=raSizes]");
let raSize = null;
raSizes.forEach((rs) => {
console.log(rs);
if (rs.checked) {
raSize = rs.value;
}
});
// Prepare data object
const sizesNames = document.querySelector("#sizes").value.split(",");
const sizesID = document.querySelector("#sizesID").value.split(",");
console.log(sizesNames);
console.log(sizesID);
}
}

```

```
let sizeName = null;
sizesID.forEach((id, i) => {
  if (id == raSize) {
    sizeName = sizesNames[i];
  }
});

const data = {
  inGloveName: document.querySelector("#name").value,
  inGloveID: document.querySelector("#id").value,
  inCatName: document.querySelector("#cat").value,
  inCatID: document.querySelector("#catID").value,
  inBrandName: document.querySelector("#brand").value,
  inBrandID: document.querySelector("#brandID").value,
  inColor: document.querySelector("#color").value,
  inColorID: document.querySelector("#colorID").value,
  // inSizesNames: document.querySelector("#sizes").value,
  // inSizesID: document.querySelector("#sizesID").value,
  inMeasure: document.querySelector("#measure").value,
  inPrice: document.querySelector("#price").value,
  inImg: document.querySelector("#t_img").value,
  inQuantity: document.querySelector("#inQuantity").value,
  raSize: raSize,
  raSizeName: sizeName,
  inUserID: document.querySelector("#inUserID").value,
};

return data;
}

// Return
static returnOrderDetail() {
  const order = Storage.returnOrder();
  const data = new FormData();
  let arr = [];
  order.forEach((o, i) => {
    arr.push({
      id_user: o.inUserID,
      id_size: o.raSize,
      id_glove: o.inGloveID,
      price_purchase: o.inPrice,
      quantity: o.inQuantity,
      name_glove: o.inGloveName,
      name_size: o.raSizeName,
    });
  });
  data.append("arr", arr);
  return data;
}
```

```

        measure: o.inMeasure,
    });
});
console.log(JSON.stringify(arr));
arr = JSON.stringify(arr);
data.append("order", arr);
// data.append("order", JSON.stringify(Storage.returnOrder));
return data;
}
}

class Storage {
// Get Gloves order
static returnOrder() {
let order = null;
if (localStorage.getItem("order") === null) {
    order = [];
} else {
    order = JSON.parse(localStorage.getItem("order"));
}
return order;
}

// Add new order line to order
static addOrderLine(orderLine, type = "add") {
// Gets the current order
console.log("How many times");
const order = this.returnOrder();
const ol = this.inStorage(orderLine, type);

// Checks if order line (ol) exists in order
if (ol) {
    // Order line already exists
    const i = ol.index;
    order[i] = ol;
} else {
    // Order line does not exist
    // We used orderLine instead of ol because inStorage() would return either
    // ol object or false
    order.push(orderLine);
}
localStorage.setItem("order", JSON.stringify(order));
// Display number of gloves in cart
// Helper.cartItemsNumber();
}
}

```

```
// Checks if order line is already in storage
static inStorage(ol, type) {
    // Will return array so it's never null, but can be empty
    const order = this.returnOrder();

    for (let i in order) {
        if (
            order[i].inUserID == ol.inUserID &&
            order[i].raSize == ol.raSize &&
            order[i].inGloveID == ol.inGloveID
        ) {
            // User id, size id and glove id are all equal
            if (type == "add") {
                console.log("inStorage");
                console.log(ol);
                console.log(ol.inQuantity);
                order[i].inQuantity =
                    Number(order[i].inQuantity) + Number(ol.inQuantity);
            } else if (type == "replace") {
                order[i].inQuantity = Number(ol.inQuantity);
            }
            order[i].index = i;

            return order[i];
        }
    }
    return false;
}

// Remove order line from order
static removeOrderLine(userID, gloveID, sizeID) {
    const order = this.returnOrder();

    for (let i in order) {
        if (
            Number(order[i].inUserID) == Number(userID) &&
            Number(order[i].inGloveID) == Number(gloveID) &&
            Number(order[i].raSize) == Number(sizeID)
        ) {
            // All 3 conditions are checked userID, gloveID, sizeID
            // Creates shallow copy of the array
            order.splice(i, 1);
        }
    }
}
```

```

// Create new localStorage array
localStorage.setItem("order", JSON.stringify(order));
// Initialize cart
Helper.cartItemsNumber();
Display.glovesCart();
}

// Remove order from localStorage
static removeOrder() {
    localStorage.removeItem("order");
    // Initialize cart
}

// Removes user from localStorage
static removeUser() {
    localStorage.removeItem("user");
}

// Creates order from cart
static createOrderFromCart(cart) {
    console.log(cart);
    for (let i in cart) {
        this.addOrderLine(cart[i]);
    }
}
}

class Helper {
    // Error Modal
    static errModal(status, arrMsg) {
        const errModal = new bootstrap.Modal(document.querySelector("#errModal"));
        document.querySelector("#errModalStatus").innerHTML = status;
        let content = ``;
        for (let i in arrMsg) {
            if (typeof arrMsg[i] !== "object") {
                content += `
                    <p class="lead">${Number(i) + 1}. ${arrMsg[i]}</p>
                `;
            } else {
                for (let j in arrMsg[i]) {
                    content += `
                        <p class="lead">${Number(i) + 1}.${Number(j) + 1} ${arrMsg[i][j]}
                    `;
                }
            }
        }
    }
}

```

```

        }
    }
    document.querySelector("#errModalMessage").innerHTML = content;
    return errModal;
}
// Success Modal
//("cart", "Order", "You are about to make purchase. Are you sure?", "continue")
static defaultModal(type, title, msg, btn = "Close") {
    console.log(type);
    console.log(title);
    console.log(msg);
    console.log(btn);
    const defModal = new bootstrap.Modal(
        document.querySelector("#defaultModal")
    );
    const arr = [title, msg, btn];
    const div = [
        "defaultModalTitle",
        "defaultModalMessage",
        "defaultModalBtn",
    ];
    arr.forEach((x, i) => {
        document.querySelector(`#${div[i]}`).innerHTML = x;
    });
    // document.querySelector("#defaultModalTitle").innerHTML = status;
    // document.querySelector("#defaultModalMessage").innerHTML = msg;
    return defModal;
}
// Returns array of checked checkboxes
static getCheckboxes(name) {
    let array = [];
    `#${name} input:checked`.each(function () {
        array.push($(this).val());
        console.log(this);
    });
    return array.join(",");
}
// Display number of items in cart in navigation
static cartItemsNumber() {
    // if (localStorage.getItem("user") === "regular_user") {
    const user = this.isCustomer();
    if (user) {
        const cart = document.querySelector("#cartNav");
        const order = Storage.returnOrder();

```

```

        cart.innerHTML = order.length;
        if (document.querySelector("#numItemsCart")) {
            document.querySelector("#numItemsCart").innerHTML = order.length;
        }
    }
}

// Checks if user has logged out, and if so, destroy localStorage user
static isLogout() {
    if (document.querySelector("#btnLogout")) {
        document.querySelector("#btnLogout").addEventListener("click", (e) => {
            // localStorage.removeItem("user");
            e.preventDefault();
            if (this.isOrder()) {
                console.log("Yeah this is order");
                const order = Storage.returnOrder();
                // Listen up Motherfucker, we sending array of obejcts
                // Form Data only accepts, you guessed it, form elements
                // This won't work FormData(order)
                // We need to append it to form data shit
                // But that form data ain't no dumbass, it knows we not passing no fo
rm element
                // So we need a workaround, Stringify that bitch nigga
                const data = new FormData();
                data.append("order", JSON.stringify(order));
            }
        });
    }
}

Ajax.isFormData(
    "models/auth/Logout.php",
    "POST",
    data,
    (data) => {
        console.log(data);
        Storage.removeUser();
        Storage.removeOrder();
        Callback.redirect();
    },
    (err) => {
        console.log(err);
        const msg = err.responseJSON;
        const status = `Oops...${err.status} error occured`;
        // Helper.defaultModal("error-login", status, msg).show();
        Helper.errModal(status, msg).show();
    }
);
} else {
    // There is no order
}

```

```

        console.log("No it's not");
        Ajax.isFormData(
            "models/auth/Logout.php",
            "POST",
            {},
            (data) => {
                console.log(data);
                Storage.removeUser();
                Storage.removeOrder();
                Callback.redirect();
            },
            (err) => {
                // Helper.defaultModal();
                console.log(err);
                const msg = err.responseJSON;
                const status = `Oops...${err.status} error occurred`;
                // Helper.defaultModal("error-login", status, msg).show();
                Helper.errModal(status, msg).show();
            }
        );
    });
}

// Checks if customer is logged in
static isCustomer() {
    if (localStorage.getItem("user") === "regular_user") {
        return true;
    } else {
        return false;
    }
}

// Checks if user is logged in
static isUser() {
    if (localStorage.getItem("user")) {
        return true;
    } else {
        return false;
    }
}

// Checks if there are Order lines in Order
static isOrder() {

```

```
const order = Storage.returnOrder();
console.log(order);
if (order.length > 0) {
    return true;
} else {
    return false;
}

// Check if current user is locked
static isLock(isLock, id) {
    let content = ``;
    if (Number(isLock)) {
        content += `
            <div id="unlock-${id}">
                <a href="#" data-id="${id}" class="text-ternary remove-link locked-
account">
                    <span class="material-icons">
                        lock
                    </span>
                </a>
            </div>
        `;
    } else {
        content += `
            <span class="material-icons">
                check
            </span>
        `;
    }
    return content;
}
}
```

```
class Callback {
    // Authorization/Redirect callbacks
    static redirect() {
        window.location = "index.php";
    }
}

class RegExp {
```

```

static regPass =
  /^(?=.*[A-Za-z])(?=.*\d)(?=.*[$!%*#?&])[A-Za-z\d@$!%*#?&]{8,30}$/;
static regEmail = /^[\w+([-+.'])\w+]@[\w+([-.\'])\w+]*\.[\w+([-.\'])\w+]*$/;
static regPhone = /^[+]*([0,1][0-9]{1,4}[]){0,1}[-\s./0-9]*$/;
static regAddress = /^[#.0-9a-zA-Z\s,-]{2,50}$/;
static regName = /^[A-ZČĆŽĐ][a-zććžđ]{2,15}$/;
static regGloveName = /^[\w ]{2,50}$/;
static regPrice = /^[\d]{1,8}(?:\.\d{1,4})?$/;
static regSubject = /(^\w[((\w+)|(\w*))*)$)|(^\w$/);
}

class Ajax {
  static callback(url, method, data, callback, err) {
    $.ajax({
      url: url,
      method: method,
      data: data,
      dataType: "json",
      success: callback,
      error: err,
    });
  }
  // Callback for handling Form data
  // Form data only accepts form elements, so it needs to be stringified
  static isFormData(url, method, data, callback, err) {
    $.ajax({
      url: url,
      method: method,
      data: data,
      // Important we define these
      contentType: false,
      processData: false,
      cache: false,

      success: callback,
      error: err,
    });
  }
}

// Validator
class Validator {
  // Login/Register user
  static userValidation() {
    // Errors

```

```
$.validator.setDefaults({
    errorClass: "text-danger",
    highlight: function (element) {
        $(element).closest(".form-group").addClass("text-tertiary");
    },
    unhighlight: function (element) {
        $(element).closest(".form-group").removeClass("text-tertiary");
    },
    errorPlacement: function (error, element) {
        if (element.prop("type") === "checkbox") {
            error.insertAfter(element.parent());
        } else {
            error.insertAfter(element);
        }
    },
});
// Password Check
$.validator.addMethod(
    "passCheck",
    (val, el) => {
        return val.length > 7 && RegExp.regPass.test(val) && val.length <= 25;
    },
    "Password must contain at least one uppercase, number and symbol (length 8-25)"
);
// Email Check
$.validator.addMethod(
    "emailCheck",
    (val, el) => {
        return RegExp.regEmail.test(val);
    },
    "Password must contain at least one uppercase, number and symbol (length 8-25)"
);
// Phone Check
$.validator.addMethod(
    "phoneCheck",
    (val, el) => {
        return RegExp.regPhone.test(val);
    },
    "Invalid phone number format. Must only contain numbers and/or (+, -, /))"
);
// Address Check
$.validator.addMethod(
```

```

"addrCheck",
(val, el) => {
    return RegExp.regAddress.test(val);
},
"Invalid address format. eg. Mean Street No.44"
);
// Name Check
$.validator.addMethod(
    "nameCheck",
    (val, el) => {
        return RegExp.regName.test(val);
},
""
);
// Subject Check
$.validator.addMethod("subjectCheck", (val, el) => {
    return RegExp.regSubject.test(val);
});

// LOGIN VALIDATION
$("#loginForm").validate({
    rules: {
        inEmail: {
            required: true,
            emailCheck: true,
        },
        inPass: {
            required: true,
            passCheck: true,
        },
    },
    messages: {
        inEmail: {
            required: "Please enter a valid email",
            email: "Email is not valid",
        },
    },
    // SUBMIT HANDLER
    submitHandler: function (form) {
        Ajax.callback(
            form.action,
            form.method,
            $(form).serialize(),
            (data) => {
                console.log(data);
            }
        );
    }
});

```

```

        Callback.redirect();
        localStorage.setItem("user", data.user);
        if (data.hasOwnProperty("cart")) {
            Storage.createOrderFromCart(data.cart);
        }
    },
    (err) => {
        console.log(err);
        const msg = err.responseJSON;
        const status = `Oops...${err.status} error occurred`;
        // Helper.defaultModal("error-login", status, msg).show();
        Helper.errorModal(status, msg).show();
    }
);
},
// END SUBMIT HANDLER
});
// END LOGIN VALIDATION

// REGISTER VALIDATION
$("#registerForm").validate({
    rules: {
        inEmail: {
            required: true,
            emailCheck: true,
        },
        inFname: {
            required: true,
            nameCheck: true,
            // errorPlacement: (err, el) => {
            //     if (err) {
            //         document.querySelector("#inFnameErr").innerHTML =
            //             "First name must be between 2-15 letters";
            //     }
            // },
        },
        inLname: {
            required: true,
            nameCheck: true,
            // errorPlacement: (err, el) => {
            //     document.querySelector("#inLnameErr").innerHTML =
            //         "Last name must be between 2-15 letters";
            // },
        },
        inPass: {

```

```

        required: true,
        passCheck: true,
    },
    inPassConf: {
        required: true,
        passCheck: true,
        equalTo: "#inPass",
    },
    inPhone: {
        required: true,
        phoneCheck: true,
    },
    inAddr: {
        required: true,
        addrCheck: true,
    },
},
// SUBMIT HANDLER
submitHandler: function (form) {
    Ajax.callback(
        form.action,
        form.method,
        $(form).serialize(),
        (data) => {
            console.log(data);
            localStorage.setItem("user", data.user);
            Callback.redirect();
        },
        (err) => {
            console.log("I go in here");
            // const msg = err.responseJSON["msg"];
            // const status = `Oops...${err.status} error occurred`;
            // console.log(status);
            // Helper.defaultModal("error-register", status, msg).show();
            console.log(err);
            const msg = err.responseJSON;
            const status = `Oops...${err.status} error occurred`;
            // Helper.defaultModal("error-login", status, msg).show();
            Helper.errModal(status, msg).show();
        }
    );
},
// END SUBMIT HANDLER
});
// END REGISTER VALIDATION

```

```
// SEND MESSAGE
$("#messageForm").validate({
    rules: {
        inFname: {
            required: true,
            nameCheck: true,
        },
        inLname: {
            required: true,
            nameCheck: true,
        },
        inEmail: {
            required: true,
            emailCheck: true,
        },
        inSubject: {
            required: true,
            subjectCheck: true,
            minlength: 2,
        },
        taMessage: {
            required: true,
            minlength: 2,
        },
    },
    messages: {
        inFname: {
            required: "First name is required",
            nameCheck: "First name must be between 2-15 characters",
        },
        inLname: {
            required: "Last name is required",
            nameCheck: "Last name must be between 2-15 characters",
        },
        inEmail: {
            required: "Email is required",
            emailCheck: "Must enter a valid email",
        },
        inSubject: {
            required: "Message subject is required",
            subjectCheck:
                "Subject name must only contain alphanumeric characters",
            minlength: "Message subject must contain at least 2 characters",
        },
        taMessage: {
    }
});
```

```

        required: "Message body is required",
        minlength: "Message body must contain at least 2 characters",
    },
},
// SUBMIT HANDLER
submitHandler: function (form) {
    Ajax.callback(
        form.action,
        form.method,
        $(form).serialize(),
        (data) => {
            Helper.defaultModal(
                "contact",
                "Message sent",
                "You have successfully sent a message"
            ).show();
        },
        (err) => {
            console.log("I go in here");
            console.log(err);
            const msg = err.responseJSON;
            const status = `Oops...${err.status} error occurred`;
            // Helper.defaultModal("error-login", status, msg).show();
            Helper.errModal(status, msg).show();
        }
    );
},
// END SUBMIT HANDLER
});
// END SEND MESSAGE
}
///////////////
// Insert/Edit product
static submitProduct() {
    // Errors
    $.validator.setDefaults({
        errorClass: "text-danger",
        highlight: function (element) {
            $(element).closest(".form-group").addClass("text-tertiary");
        },
        unhighlight: function (element) {
            $(element).closest(".form-group").removeClass("text-tertiary");
        },
        errorPlacement: function (error, element) {
            if (element.prop("type") === "checkbox") {

```

```

        error.insertAfter(element.parent().parent());
    } else {
        error.insertAfter(element);
    }
},
});

// Name Check
$.validator.addMethod(
    "nameCheck",
    (val, el) => {
        return RegExp.regGloveName.test(val);
    },
    "Invalid glove name"
);
// Price Check
$.validator.addMethod(
    "priceCheck",
    (val, el) => {
        return RegExp.regPrice.test(val);
    },
    "Invalid price format eg. 59.99"
);
// // Checkbox Check
// $.validator.addMethod(
//     "cb",
//     (val, el) => {
//         return RegExp.regPrice.test(val);
//     },
//     "Invalid price format eg. 59.99"
// );
const mode = new URLSearchParams(window.location.search).get("mode");
console.log(mode);
if (mode == "insert") {
    // GLOVE VALIDATION INSERT
    $("#glovesModeFormInsert").validate({
        rules: {
            inName: {
                required: true,
                nameCheck: true,
            },
            inPrice: {
                required: true,
                priceCheck: true,
            },
        },
    });
}

```

```
taDesc: {
    required: true,
},
ddlCat: {
    required: true,
},
ddlBrand: {
    required: true,
},
ddlColor: {
    required: true,
},
"cbSize[]": {
    required: true,
},
},
messages: {
    inName: {
        required: "Glove name cannot be left empty",
    },
    inPrice: {
        required: "Price cannot be left empty eg. 59.99",
    },
    taDesc: {
        required: "Glove must have some description",
    },
    ddlBrand: {
        required: "Brand must be checked",
    },
    ddlCat: {
        required: "Category must be checked",
    },
    ddlColor: {
        required: "Color must be checked",
    },
    cbSize: {
        required: "At least one size must be checked",
    },
// fileImg: {
//     required: "Image must be selected",
//     // accept: "Only image type jpg/png/jpeg are allowed",
//     filesize: "Image must be less than 3MB",
// },
},
// SUBMIT HANDLER
```

```

submitHandler: function (form) {
  const data = new FormData(form);
  $.ajax({
    url: form.action,
    method: form.method,
    data: data,
    // Must be set to false
    contentType: false,
    cache: false,
    // Must not be set
    // dataType: "json",
    processData: false,
    success: (data) => {
      console.log(data);
      console.log(data.msg);
      Helper.defaultModal(
        "glove-insert",
        "Glove inserted successfully",
        data.msg
      ).show();
      const mode = new URLSearchParams(window.location.search).get(
        "mode"
      );
      if (mode == "insert") {
        document.querySelector("#glovesModeFormInsert").reset();
        document.querySelector("#cbSizes").innerHTML = "";
      }
    },
    error: (err) => {
      console.log(err);
      // const msg = err.responseJSON["msg"];
      // console.log(msg);
      // const status = `Oops...${err.status} error occurred`;
      // Helper.defaultModal("error-glove-insert", status, msg).show();
      console.log(err);
      const msg = err.responseJSON;
      const status = `Oops...${err.status} error occurred`;
      // Helper.defaultModal("error-login", status, msg).show();
      Helper.errModal(status, msg).show();
    },
  });
},
// END SUBMIT HANDLER
});
// END GLOVE VALIDATION INSERT

```

```
    } else {
        console.log("I go in here");
        // GLOVE VALIDATION EDIT
        $("#glovesModeFormEdit").validate({
            rules: {
                inName: {
                    required: true,
                    nameCheck: true,
                },
                inPrice: {
                    required: true,
                    priceCheck: true,
                },
                taDesc: {
                    required: true,
                },
                ddlCat: {
                    required: true,
                },
                ddlBrand: {
                    required: true,
                },
                ddlColor: {
                    required: true,
                },
                "cbSize[]": {
                    required: true,
                },
                inHidden: {
                    required: true,
                },
            },
            messages: {
                inName: {
                    required: "Glove name cannot be left empty",
                },
                inPrice: {
                    required: "Price cannot be left empty eg. 59.99",
                },
                taDesc: {
                    required: "Glove must have some description",
                },
                ddlBrand: {
                    required: "Brand must be checked",
                },
            }
        });
    }
}
```

```
    ddlCat: {
      required: "Category must be checked",
    },
    ddlColor: {
      required: "Color must be checked",
    },
    cbSize: {
      required: "At least one size must be checked",
    },
    // fileImg: {
    //   // required: "Image must be selected",
    //   // accept: "Only image type jpg/png/jpeg are allowed",
    //   // filesize: "Image must be less than 3MB"
    // },
  },
  // SUBMIT HANDLER
  submitHandler: function (form) {
    const data = new FormData(form);
    $.ajax({
      url: form.action,
      method: form.method,
      data: data,
      // Must be set to false
      contentType: false,
      // Must not be set
      // dataType: "json",
      processData: false,
      success: (data) => {
        console.log(data);
        console.log(data.msg);
        const newID = data.id;
        const urlParams = new URLSearchParams(window.location.search);
        urlParams.set("id", newID);
        window.location.search = urlParams;
        Helper.defaultModal(
          "glove-update",
          "Glove updated successfully",
          data.msg
        ).show();
        if (data.hasOwnProperty("img")) {
          document.querySelector(
            "#editThumbImg"
          ).src = `assets/img/gloves/thumbnail/${data.img}`;
        }
      },
    },
  },
};
```

```

        error: (err) => {
            console.log(err);
            // const msg = err.responseJSON["msg"];
            // console.log(msg);
            // const status = `Oops...${err.status} error occurred`;
            // Helper.defaultModal("error-login", status, msg).show();
            console.log(err);
            const msg = err.responseJSON;
            const status = `Oops...${err.status} error occurred`;
            // Helper.defaultModal("error-login", status, msg).show();
            Helper.errorModal(status, msg).show();
        },
    });
},
// END SUBMIT HANDLER
});
// END GLOVE VALIDATION EDIT
}
}

// Send Message
static sendMessage() {}
}

```

```

// Listeners
class Listeners {
    // Load types of sizes for specific glove (based on category)
    static loadSizesForCategory() {
        // Selected category determines what type of sizes will be available
        // eg. boxing -> oz; karate/mma -> s/m/l/xl
        // Category dropdown
        const ddlCat = document.querySelector("#ddlCat");
        ddlCat.addEventListener("change", () => {
            const cat = Number(ddlCat.value);
            if (cat) {
                Ajax.callback(
                    "models/users/admin/gloves/Category.php",
                    "POST",
                    {
                        cat: cat,
                    },
                    (data) => {

```

```

        Display.sizeCheckboxes(data.data);
    },
    (err) => {
        // const msg = err.responseJSON["msg"];
        // const status = `Oops...${err.status} error occurred`;
        // Helper.defaultModal("error-login", status, msg).show();
        console.log(err);
        const msg = err.responseJSON;
        const status = `Oops...${err.status} error occurred`;
        // Helper.defaultModal("error-login", status, msg).show();
        Helper.errModal(status, msg).show();
    }
);
} else {
    document.querySelector("#cbSizes").innerHTML =
    "Category must be select first";
}
});
}

// Load all gloves that are not removed (date_removed)
static loadGlovesTable(page = 1) {
    console.log("Does it matter");
    const data = {
        type: "gloves",
        categories: document.querySelector("#categories").value,
        brands: document.querySelector("#brands").value,
        colors: document.querySelector("#colors").value,
        sort: document.querySelector("#sort").value,
        range: document.querySelector("#range").value,
        perPage: document.querySelector("#perPage").value,
        page: page,
        search: document.querySelector("#search").value,
    };
    console.log(data);
    // Ajax
    Ajax.callback(
        "models/Filter.php",
        // "models/users/admin/gloves/ActiveGloves.php",
        "POST",
        data,
        (data) => {
            const arrGlovesObj = data[0].gloves;
            const pages = data[0].pages;
            const total = data[0].total;

```

```

        console.log(arrGlovesObj);
        console.log(pages);
        console.log(total);
        Display.allGlovesTable(arrGlovesObj, total);
        Display.pagesNumber(pages, page, "admin-table");
    },
    (err) => {
        // const msg = err.responseJSON["msg"];
        // const status = `Oops...${err.status} error occurred`;
        // Helper.defaultModal("error-glove-table-admin", status, msg).show();
        console.log(err);
        const msg = err.responseJSON;
        const status = `Oops...${err.status} error occurred`;
        // Helper.defaultModal("error-login", status, msg).show();
        Helper.errModal(status, msg).show();
    }
);

}

// Remove glove from table
static removeGloveIdTable(id) {
    Ajax.callback(
        "models/users/admin/gloves/Remove.php",
        "POST",
        {
            id: id,
        },
        (data) => {
            console.log(data);
            Listeners.loadGlovesTable();
        },
        (err) => {
            console.log(err);
            const msg = err.responseJSON;
            const status = `Oops...${err.status} error occurred`;
            // Helper.defaultModal("error-login", status, msg).show();
            Helper.errModal(status, msg).show();
        }
    );
}

// SHOP - Attach Event Listeners to all filter/sort inputs
static shopFilterEvents() {
    const checks = ["categories", "brands", "colors"];
    checks.forEach((check, i) => {

```

```

const ch = document.querySelectorAll(`input[name='${check}']`);
ch.forEach((c) => {
  c.addEventListener("change", () => {
    this.loadGlovesShop();
  });
});
);
const change = ["range", "sort", "perPage"];
change.forEach((c) => {
  document.querySelector(`#${c}`).addEventListener("change", () => {
    this.loadGlovesShop();
  });
});
document.querySelector("#search").addEventListener("keyup", () => {
  this.loadGlovesShop();
});
}

// Load orders table (admin-orders)
static loadOrdersTable(page = 1) {
  const id = new URLSearchParams(window.location.search).get("id");
  console.log(id);
  // Filtering data
  const data = {
    type: "orders",
    sort: document.querySelector("#sort").value,
    perPage: document.querySelector("#perPage").value,
    page: page,
    id: id ? id : null,
  };
  if (document.querySelector("#search")) {
    data.search = document.querySelector("#search").value;
  }
  Ajax.callback(
    "models/Filter.php",
    "POST",
    data,
    (data) => {
      console.log(data);
      const arrOrdersObj = data[0].orders;
      const pages = data[0].pages;
      const total = data[0].total;
      // console.log(arrGlovesObj);
      // console.log(pages);
      // console.log(total);
    }
  );
}

```

```

        Display.allOrdersTable(arrOrdersObj, total, id ? id : null);
        Display.pagesNumber(
            pages,
            page,
            id ? "admin-table-orders-id" : "admin-table-orders"
        );
    },
    (err) => {
        // const msg = err.responseJSON["msg"];
        // const status = `Oops...${err.status} error occurred`;
        // Helper.defaultModal("error-glove-table-admin", status, msg).show();
        console.log(err);
        const msg = err.responseJSON;
        const status = `Oops...${err.status} error occurred`;
        // Helper.defaultModal("error-login", status, msg).show();
        Helper.errModal(status, msg).show();
    }
);

}

// Load users table (admin-users)
static loadUsersTable(page = 1) {
    // Filtering data
    const data = {
        type: "users",
        sort: document.querySelector("#sort").value,
        perPage: document.querySelector("#perPage").value,
        page: page,
        search: document.querySelector("#search").value,
    };
    Ajax.callback(
        "models/Filter.php",
        "POST",
        data,
        (data) => {
            console.log(data);
            const arrUsersObj = data[0].users;
            const pages = data[0].pages;
            const total = data[0].total;
            // console.log(arrGlovesObj);
            // console.log(pages);
            // console.log(total);
            Display.allUsersTable(arrUsersObj, total);
            Display.pagesNumber(pages, page, "admin-table-users");
        },
    );
}

```

```

(err) => {
    // const msg = err.responseJSON["msg"];
    // const status = `Oops...${err.status} error occurred`;
    // Helper.defaultModal("error-glove-table-admin", status, msg).show();
    console.log(err);
    const msg = err.responseJSON;
    const status = `Oops...${err.status} error occurred`;
    // Helper.defaultModal("error-login", status, msg).show();
    Helper.errModal(status, msg).show();
}
);

}

// Load messages table (admin-messages)
static loadMessagesTable(page = 1) {
    // Filtering data
    const data = {
        type: "messages",
        sort: document.querySelector("#sort").value,
        perPage: document.querySelector("#perPage").value,
        page: page,
        search: document.querySelector("#search").value,
    };
    console.log(data);
    Ajax.callback(
        "models/Filter.php",
        "POST",
        data,
        (data) => {
            console.log(data);
            const arrMessagesObj = data[0].messages;
            const pages = data[0].pages;
            const total = data[0].total;
            // console.log(arrGlovesObj);
            // console.log(pages);
            // console.log(total);
            Display.allMessagesTable(arrMessagesObj, total);
            Display.pagesNumber(pages, page, "admin-messages");
        },
        (err) => {
            // const msg = err.responseJSON["msg"];
            // const status = `Oops...${err.status} error occurred`;
            // Helper.defaultModal("error-glove-table-admin", status, msg).show();
            console.log(err);
            const msg = err.responseJSON;

```

```

        const status = `Oops...${err.status} error occured`;
        // Helper.defaultModal("error-login", status, msg).show();
        Helper.errModal(status, msg).show();
    }
);

}

// ADMIN-GLOVES - Attach Event Listeners to all filter/sort inputs
static adminGlovesTableFilterEvents() {
    const change = [
        "categories",
        "brands",
        "colors",
        "perPage",
        "sort",
        "range",
    ];
    change.forEach((c) => {
        document.querySelector(`#${c}`).addEventListener("change", () => {
            this.loadGlovesTable();
        });
    });
    document.querySelector("#search").addEventListener("keyup", () => {
        this.loadGlovesTable();
    });
}

static loadDefaultFilterEvents($type) {
    console.log("DEFAULT FILTER EVENTS");
    console.log($type);
    const change = ["perPage", "sort"];
    change.forEach((c) => {
        document.querySelector(`#${c}`).addEventListener("change", () => {
            console.log($type);
            switch ($type) {
                case "admin-orders":
                    this.loadOrdersTable();
                    break;
                case "admin-users":
                    this.loadUsersTable();
                    break;
                case "admin-user":
                    this.loadOrdersTable();
                    break;
                case "admin-messages":

```

```

        console.log("IN HERE");
        this.loadMessagesTable();
        break;
    }
}
});

if (document.querySelector("#search")) {
    document.querySelector("#search").addEventListener("keyup", () => {
        switch ($type) {
            case "admin-orders":
                this.loadOrdersTable();
                break;
            case "admin-users":
                this.loadUsersTable();
                break;
            case "admin-messages":
                this.loadMessagesTable();
                break;
        }
    });
}

// Load, in Shop, all gloves that are active
static loadGlovesShop(page = 1) {
    const data = {
        type: "gloves",
        categories: Helper.getCheckboxes("categories"),
        brands: Helper.getCheckboxes("brands"),
        colors: Helper.getCheckboxes("colors"),
        sort: document.querySelector("#sort").value,
        range: document.querySelector("#range").value,
        perPage: document.querySelector("#perPage").value,
        page: page,
        search: document.querySelector("#search").value,
    };

    Ajax.callback(
        "models/Filter.php",
        "POST",
        data,
        (data) => {
            const arrGlovesObj = data[0].gloves;
            const pages = data[0].pages;
            const total = data[0].total;

```

```

        Display.glovesShop(arrGlovesObj, pages, page, total);
    },
    (err) => {
        console.log(err);
        const msg = err.responseJSON;
        const status = `Oops...${err.status} error occurred`;
        // Helper.defaultModal("error-login", status, msg).show();
        Helper.errModal(status, msg).show();
    }
);

}

// Insert selected glove into cart
static insertIntoCart() {
    console.log("READY TO LOAD SINGLE GLOVE");
    document.querySelector("#btnSubmit").addEventListener("click", (e) => {
        e.preventDefault();
        // Get the data and call the function only when the Add to cart button is clicked

        // Get the Order Line object and pass it to Order in localStorage
        const data = Data.returnOrderLine();
        console.log(data);

        Helper.defaultModal(
            "add-glove-to-cart",
            "Glove added",
            `Successfully added ${data.inGloveName} to cart`
        ).show();
        Storage.addOrderLine(data);
        Helper.cartItemsNumber();

        // Initialize cart
    });
}

// Add event listeners to Quantity input as well as remove button
static cartEvents() {
    const remove = document.querySelectorAll(".remove-link");
    remove.forEach((r, i) => {
        r.addEventListener("click", (e) => {
            e.preventDefault();
            const userID = +r.dataset.userid;
            const sizeID = +r.dataset.sizeid;
            const gloveID = +r.dataset.gloveid;

```

```

        console.log(userID, sizeID, gloveID);
        Storage.removeOrderLine(userID, gloveID, sizeID);
    });
});

const quantity = document.querySelectorAll(".quantity");
quantity.forEach((q, i) => {
    q.addEventListener("change", (e) => {
        e.stopPropagation();
        e.preventDefault();
        const val = +q.value;
        const userID = +q.dataset.userid;
        const sizeID = +q.dataset.sizeid;
        const gloveID = +q.dataset.gloveid;

        // Cannot be less than 1
        if (val > 0) {
            const order = Storage.returnOrder();
            order.forEach((o) => {
                if (
                    +o.inUserID === userID &&
                    +o.raSize === sizeID &&
                    +o.inGloveID === gloveID
                ) {
                    // Found the current order line
                    o.inQuantity = val;
                    Storage.addOrderLine(o, "replace");
                    Helper.cartItemsNumber();
                } else {
                    console.log("Not found");
                }
            });
            Display.glovesCart();
        } else {
            // Value is less than 1
        }
    });
});
});

// Add event listener to purchase button in Cart
static purchaseCart() {
    const purchase = document.querySelector("#purchase");
    purchase.addEventListener("click", (e) => {
        e.preventDefault();

```

```

const user = Helper.isCustomer();
if (user) {
    // Is customer
    const isOrder = Helper.isOrder();
    if (isOrder) {
        // Can make purchase
        console.log("I in");
        Helper.defaultModal(
            "customer-purchase",
            "Order",
            "Successfully purchased",
            "Close"
        ).show();
        document
            .querySelector("#defaultModal")
            .addEventListener("hidden.bs.modal", () => {
                const data = Data.returnOrderDetail();
                Ajax.isFormData(
                    "models/users/customer/Purchase.php",
                    "POST",
                    data,
                    (data) => {
                        console.log(data);
                        const div = document.querySelector("#purchased");
                        div.innerHTML =
                            `<span class="material-icons align-middle me-2">
                                check
                            </span>Successfully purchased`;
                        div.classList.add("py-2");
                        div.classList.add("px-2");
                        Storage.removeOrder();
                        Helper.cartItemsNumber();
                        Display.glovesCart();
                    },
                    (err) => {
                        console.log("I go into error");
                        // console.log(err);
                        const msg = err.responseJSON;
                        const status = `Oops...${err.status} error occurred`;
                        // Helper.defaultModal("error-login", status, msg).show();
                        Helper.errModal(status, msg).show();
                    }
                );
            });
    } else {
}
}

```

```

        // No items in cart, can't make purchase
        Helper.defaultModal(
            "admin-purchase",
            "Order failed",
            "Cannot make purchase, no items selected"
        ).show();
    }
} else {
    // Not customer
    Helper.defaultModal(
        "admin-purchase",
        "Order failed",
        "Cannot make purchase, you are not a customer"
    ).show();
}
});

}

// Download order detail as Excel
static downloadOrderDetailExcel() {
    const excel = document.querySelector("#download-order-detail-excel");
    console.log("Here's excel");
    console.log(excel);
    excel.addEventListener("click", (e) => {
        e.preventDefault();
        const orderID = excel.dataset.orderid;
        Ajax.callback(
            "models/users/admin/orders/Excel.php",
            "POST",
            { orderID: orderID },
            (data) => {
                console.log(data);
                console.log("we in ere");
                // const arrMessagesObj = data[0].messages;
                // const pages = data[0].pages;
                // const total = data[0].total;
                // console.log(arrGlovesObj);
                // console.log(pages);
                // console.log(total);
                // Display.allMessagesTable(arrMessagesObj, total);
                // Display.pagesNumber(pages, page, "admin-messages");
            },
            (err) => {
                // const msg = err.responseJSON["msg"];
                // const status = `Oops...${err.status} error occured`;

```

```

        // Helper.defaultModal("error-glove-table-
admin", status, msg).show();
        console.log(err);
        const msg = err.responseJSON;
        const status = `Oops...${err.status} error occurred`;
        // Helper.defaultModal("error-login", status, msg).show();
        Helper.errModal(status, msg).show();
    }
);

});

}

// // Unlock user's account
// static unlockUser() {

// }

// Display AJAX
class Display {
    // Checkboxes for size
    static sizeCheckboxes(sizes) {
        let content = ``;
        sizes = Object.values(sizes);
        const div = document.querySelector("#cbSizes");

        sizes.forEach((s, i) => {
            content += `
<div class="form-check">
    <input class="form-check-
input" type="checkbox" name="cbSize[]" value="${
                s.id
            }" id="cbSize-${s.id}">
    <label class="form-check-label" for="flexCheckDefault">
        ${s.size}
        ${s.measure == "OZ" ? s.measure : ""}
    </label>
</div>
`;
        });
        div.innerHTML = content;
    }

    // Display Gloves Table
    static allGlovesTable(gloves, total) {

```

```
let content = ``;
console.log(gloves);
document.querySelector("#gloveCount").innerHTML = total;
gloves = Object.values(gloves);
console.log(gloves);
gloves.forEach((g, i) => {
    content += `
        <tr>
            <td class="align-middle">${i + 1}</td>
            <td class="align-middle"></td>
            <td class="align-middle">${g.name}</td>
            <td class="align-middle">${g.cat}</td>
            <td class="align-middle">${g.brand}</td>
            <td class="align-middle">
                <div class="d-flex align-items-center">
                    <span class="palette border align-bottom" style="background-color:${g.color} !important;"></span>
                    <span class="capitalize ms-2 lead align-bottom">${g.color}</span>
                </div>
            </td>
            <td class="align-middle">${g.sizes}</td>
            <td class="align-middle">${g.measure}</td>
            <td class="align-middle">${g.date_added}</td>
            <td class="align-middle">
                <a href="index.php?page=admin-gloves&mode=edit&id=${g.id}" class="text-secondary">
                    <span class="material-icons">edit</span>
                </a>
            </td>
            <td class="align-middle">
                <a href="#" data-id="${g.id}" class="text-tertiary remove-link">
                    <span class="material-icons">delete</span>
                </a>
            </td>
        </tr>
    `;
});
```

```

        </td>
    </tr>
    `;
});
document.querySelector("#glovesTableBody").innerHTML = content;
// Listeners added after the content is rendered
const removeLinks = document.querySelectorAll(".remove-link");
removeLinks.forEach((l) => {
    l.addEventListener("click", (e) => {
        // Prevents the EVENT not the link itself
        e.preventDefault();
        Listeners.removeGloveIdTable(Number(l.dataset.id));
    });
});
}

// Display Orders Tabel
static allOrdersTable(orders, total, id) {
    console.log(orders);
    console.log(total);
    console.log("WE HERE");
    let content = ``;
    document.querySelector("#ordersCount").innerHTML = total;
    orders = Object.values(orders);
    orders.forEach((g, i) => {
        console.log(g);
        content += `
            <tr>
                <td class="align-middle">${i + 1}</td>
                <td class="align-middle"><a class="text-secondary" href="index.php?page=admin- ${
                    id ? "order&id=" + g.id_order_detail : "user&id=" + g.id_user
                }">${g.first_name} ${g.last_name}</a></td>
                <td class="align-middle">${g.count}</td>
                <td class="align-middle">${g.total}</td>
                <td class="align-middle">${g.date}</td>
                <td class="align-middle">
                    <a href="index.php?page=admin-order&id=${g.id_order_detail}" class="text-ternary more-details">
                        <span class="material-icons">
                            receipt_long
                        </span>
                    </a>
                </td>
        `;
    });
}

```

```

        </tr>
    `;
});
document.querySelector("#ordersTableBody").innerHTML = content;
// document.querySelector("#ordersTableBody").innerHTML = content;
// Listeners added after the content is rendered
}

// Display Users Table
static allUsersTable(users, total) {
    console.log(users);
    let content = ``;
    document.querySelector("#usersCount").innerHTML = total;
    users = Object.values(users);
    users.forEach((g, i) => {
        console.log(g);
        content += `
            <tr>
                <td class="align-middle">${i + 1}</td>
                <td class="align-middle"><a class="text-secondary" href="index.php?page=admin-user&id=${g.id_user}>${g.first_name} ${g.last_name}</a></td>
                <td class="align-middle">${g.email_user}</td>
                <td class="align-middle">${+g.count ? g.count : "/"}</td>
                <td class="align-middle">${g.total ? "$" + g.total : "/"}</td>
                <td class="align-middle">
                    <a href="index.php?page=admin-user&id=${g.id_user}" data-id="" class="text-ternary remove-link">
                        <span class="material-icons">
                            person
                        </span>
                    </a>
                </td>
                <td>
                    ${Helper.isLock(g.isLock, g.id_user)}
                </td>
            </tr>
        `;
    });
    document.querySelector("#usersTableBody").innerHTML = content;

    const unlock = document.querySelectorAll(".locked-account");
    unlock.forEach((u, i) => {

```

```

        u.addEventListener("click", (e) => {
            e.preventDefault();
            const userID = u.dataset.id;
            Ajax.callback(
                "models/users/admin/users/Unlock.php",
                "POST",
                { userID: userID },
                (data) => {
                    document.querySelector(`#unlock-${userID}`).innerHTML =
                        `<span class="material-icons">
                            check
                        </span>
                    `;
                },
                (err) => {
                    // const msg = err.responseJSON["msg"];
                    // const status = `Oops...${err.status} error occurred`;
                    // Helper.defaultModal("error-login", status, msg).show();
                    console.log(err);
                    const msg = err.responseJSON;
                    const status = `Oops...${err.status} error occurred`;
                    // Helper.defaultModal("error-login", status, msg).show();
                    Helper.errModal(status, msg).show();
                }
            );
        });
    });

    // Display.pagesNumber(pages, page, "shop");
}

// Display Messages Table
static allMessagesTable(messages, total) {
    console.log(messages);
    let content = ``;
    document.querySelector("#messagesCount").innerHTML = total;
    messages = Object.values(messages);
    messages.forEach((g, i) => {
        console.log(g);
        content += `
            <tr>
                <td class="align-middle">${i + 1}</td>
                <td class="align-middle"><a class="text-secondary" href="index.php?page=admin-message&id=${g.id_msg}">${g.msg_first_name} ${g.msg_last_name}</a></td>
        `;
    });
}

```



```
        <span class="material-icons align-bottom text-
secondary">
            attach_money
        </span>
        <label>
            ${arrGlovesObj[i].price}
        </label>
    </h5>
</div>
<div>
    <h5 class="d-block lead">
        <span class="material-icons align-bottom text-
secondary">
            sports_mma
        </span>
        <label>
            ${arrGlovesObj[i].name}
        </label>
    </h5>
</div>
<div>
    <h5 class="d-block lead">
        <span class="material-icons align-bottom text-
secondary">
            straighten
        </span>
        <label>
            ${arrGlovesObj[i].sizes} ${
arrGlovesObj[i].measure === "OZ"
? "(" + arrGlovesObj[i].measure + ")"
: ""
}
        </label>
    </h5>
</div>
<div class="mt-3 d-flex align-items-center">
    <span class="palette border align-
bottom" style="background-color:${
        arrGlovesObj[i].color
    }!important;"></span>
    <label class="capitalize ms-2 lead">${
        arrGlovesObj[i].color
    }</label>
</div>
</div>
```

```

        <div class="my-3">
            <a class="btn box-shadow-hover border mb-2" href="index.php?page=glove&id=${arrGlovesObj[i].id}" data-prodid="${arrGlovesObj[i].id}">View</a>
        </div>
    </figure>
</div>
`;
}
div.innerHTML = content;
Display.pagesNumber(pages, page, "shop");
}

// Display number of pages (pagination)
static pagesNumber(pages, page, type = "shop") {
    console.log(pages);
    let content = ``;
    for (let i = 1; i <= pages; i++) {
        content += `
            <li class="page-item"><a class="page-link ${page == i ? "active-page" : ""}" href="#" data-id="${i}">${i}</a></li>
        `;
    }
    // pages.forEach((p, i) => {
    //     content += `
    //         <li class="page-item"><a class="page-link ${page == (i+1) ? 'active' : ''}" href="#" dataid="${i+1}">1</a></li>
    //     `;
    // });
    document.querySelector("#pages").innerHTML = content;
    const links = document.querySelectorAll(".page-link");
    links.forEach((l) => {
        l.addEventListener("click", (e) => {
            e.preventDefault();
            const id = l.dataset.id;
            console.log(type);
            console.log(id);
            switch (type) {
                case "shop":
                    Listeners.loadGlovesShop(id);
                    break;
                case "admin-table":

```

```

        Listeners.loadGlovesTable(id);
        break;
    case "admin-table-orders":
        Listeners.loadOrdersTable(id);
        break;
    case "admin-table-orders-id":
        Listeners.loadOrdersTable(id);
        break;
    case "admin-table-users":
        Listeners.loadUsersTable(id);
        break;
    case "admin-messages":
        Listeners.loadMessagesTable(id);
        break;
    }
}
});

// Display, in cart, all gloves from localStorage
static glovesCart() {
    const cart = document.querySelector("#cart");
    const order = Storage.returnOrder();
    console.log(order);
    let content = ``;
    let total = 0;
    if (order.length >= 0) {
        order.forEach((g, i) => {
            total += Math.round(Number(g.inQuantity) * Number(g.inPrice) * 100) / 100;
            content += `
                <tr>
                    <td class="align-middle">${i + 1}</td>
                    <td class="align-middle"></td>
                    <td class="align-middle">${g.inGloveName}</td>
                    <td class="align-middle">${g.inCatName}</td>
                    <td class="align-middle">${g.inBrandName}</td>
                    <td class="align-middle">
                        <div class="d-flex align-items-center">
                            <span class="palette border align-bottom" style="background-color:${g.inColor}

```

```

}!important;"></span>
<span class="capitalize ms-2 lead align-bottom">${g.inColor}</span>
</div>
</td>
<td class="align-middle">${g.raSizeName}${g.inMeasure === "OZ" ? g.inMeasure : ""}</td>
<td class="align-middle">${g.inPrice}</td>
<td class="align-middle">
<input type="number" data-userid="${g.inUserID}" data-sizeid="${g.raSize}" data-gloveid="${g.inGloveID}" class="quantity" min="1" value="${g.inQuantity}">
</td>
<td class="align-middle">${Math.round(Number(g.inQuantity) * Number(g.inPrice) * 100) / 100}</td>
<td class="align-middle">
<a href="#" data-userid="${g.inUserID}" data-sizeid="${g.raSize}" data-gloveid="${g.inGloveID}" class="text-ternary remove-link">
<span class="material-icons">delete</a>
</td>
</tr>
`;;
});
// Append content to cart table
cart.innerHTML = content;

```

```

const prices = ["price", "totalPrice"];
total = Math.round(total * 100) / 100;
prices.forEach((p) => {
  document.querySelector(`#${p}`).innerHTML = total;
});
// console.log(Math.round(total * 100) / 100);
Listeners.cartEvents();
} else {
  // There are no items in cart
}
}

// IIFE
(() => {
  // Classes
  // 1. Listeners -> make request to backend and get the data
  // 2. Display -> get data from Listeners and render content
  // 3. Validator -> handle client side validation
  Helper.cartItemsNumber();
  Helper.isLogout();
  const urlParams = new URLSearchParams(window.location.search).get("page");
  if (urlParams) {
    switch (urlParams) {
      // GENERAL
      case "shop":
        Listeners.shopFilterEvents();
        Listeners.loadGlovesShop();
        break;
      case "cart":
        console.log("CART CLIENT SIDE");
        Display.glovesCart();
        Listeners.purchaseCart();
        break;
      case "glove":
        console.log("GLOVE CLIENT SIDE");
        Helper.isCustomer() ? Listeners.insertIntoCart() : "";
        // Listeners.insertIntoCart();
        break;
      case "contact":
        console.log("WE IN CONTACT");
        Validator.userValidation();
        break;
    }
  }
}

// AUTH

```

```
case "login":
    Validator.userValidation();
    break;
case "register":
    console.log("Register page");
    Validator.userValidation();
    break;

// CUSTOMER

// ADMIN
case "products-insert":
    Validator.insertProduct();
    console.log("PRODUCTS-INSERT");
    break;

// ADMIN GLOVES (DEFAULT, INSERT, UPDATE)
case "admin-gloves":
    const mode = new URLSearchParams(window.location.search).get("mode");
    if (mode) {
        switch (mode) {
            case "insert":
                Listeners.loadSizesForCategory();
                Validator.submitProduct();
                break;
            case "edit":
                Listeners.loadSizesForCategory();
                Validator.submitProduct();
                console.log("Edit existing glove");
                break;
            default:
                console.log("DEFAULT GLOVES");
        }
    } else {
        console.log("TABLE GLOVES");
        Listeners.adminGlovesTableFilterEvents();
        Listeners.loadGlovesTable();
    }
    break;
// END ADMIN GLOVES (DEFAULT, INSERT, UPDATE)

case "admin-orders":
    console.log("ADMIN ORDERS");
    Listeners.loadDefaultFilterEvents("admin-orders");
    Listeners.loadOrdersTable();
```

```
        break;
    case "admin-users":
        console.log("ADMIN USERS");
        Listeners.loadDefaultFilterEvents("admin-users");
        Listeners.loadUsersTable();
        break;
    case "admin-order":
        console.log("ADMIN ORDER DETAIL");
        // Listeners.downloadOrderDetailExcel();
        break;
    case "admin-user":
        console.log("ADMIN USER DETAIL");
        Listeners.loadDefaultFilterEvents("admin-user");
        Listeners.loadOrdersTable();
        break;
    case "admin-messages":
        Listeners.loadDefaultFilterEvents("admin-messages");
        Listeners.loadMessagesTable();
        console.log("ADMIN MESSAGES");
        break;
    // DEFAULT
    default:
        console.log("INDEX PAGE");
    }
} else {
    console.log("INDEX PAGE BUT ITS NOT SWITCH");
}
})();
});
```