

DOKUMENTACIJA

PRAKTIKUM IZ WEB
PROGRAMIRANJA PHP

Stefan Bogdanovski 85/18

SADRZAJ

1. Korisceni programski jezici.
2. Opis funkcionalnosti
3. Template
5. Slike stranica I opis
6. Dizajn baze
7. Struktura projekta
8. Kodovi

KORISCENI PROGRAMSKI JEZICI

I TEHNOLOGIJE

- HTML
- CSS
- JAVASCRIPT
- PHP
- SQL
- JQUERY BIBLIOTEKA

OPIS FUNKCIONALNOSTI

Pri dolasku na pocetnu stranu, korisniku je omoguceno da vidi proizvode, da ih filtrira ili sortira po ceni, korisnik moze da se registruje ili uloguje.

Klikom na proizvod otvara se proizvod sa vise informacija I omoguceno je korisniku da ga ubaci u korpu ukoliko je ulogovan.

Klikom na link korpe korisnik odlazi na pregled svoje korpe gde moze da dodaje kolicinu proizvoda ili da izbrise proizvod iz svoje korpe.

Klikom na FINALIZE PURCHASE korisniku se otvara forma za popunjavanje kako bi potvrdio kupovinu proizvoda.

Filtriranje se vrši po brendovima, dok se sortiranje vrši po ceni, silazno ili uzlazno.

TEMPLATE

Template nije koriscen.

Nema kopiranih gotovih kodova.

SLIKE STRANICA I OPIS

Index.php

Dolores perfumes

localhost/praktikumPHP/index.php

DOLORES









Home Sign out

Our products!

Sort by

Choose..

- ARMANY BEAUTY
- BOBBI BROWN
- CHANEL
- Dior
- GUCCI
- MARC JACOBS FRAGRANCES
- VERSACE

 <p>Acqua Di Gio Profumo Eau de Parfum \$74 40ml</p> <p>SHOW MORE</p>	 <p>Acqua di Gioia Eau de Parfum \$78 50ml</p> <p>SHOW MORE</p>	 <p>Beach Fragrance Eau de Parfum \$70 50ml</p> <p>SHOW MORE</p>	 <p>BLEU DE CHANEL Eau de Toilette \$77 50ml</p> <p>SHOW MORE</p>
 <p>Daisy Eau de Toilette \$106 100ml</p> <p>SHOW MORE</p>	 <p>Eros Flame Eau de Parfum \$80 50ml</p> <p>SHOW MORE</p>	 <p>Flora By Gucci - Gorgeous Gardenia Eau de Toilette \$80 50ml</p> <p>SHOW MORE</p>	 <p>Sauvage Eau de Parfum \$90 60ml</p> <p>SHOW MORE</p>

Dolores perfumes
Working hours: 9am to 5pm Monday -> Friday
We are delivering during whole week.
For any questions, please contact us at dolores@support.com

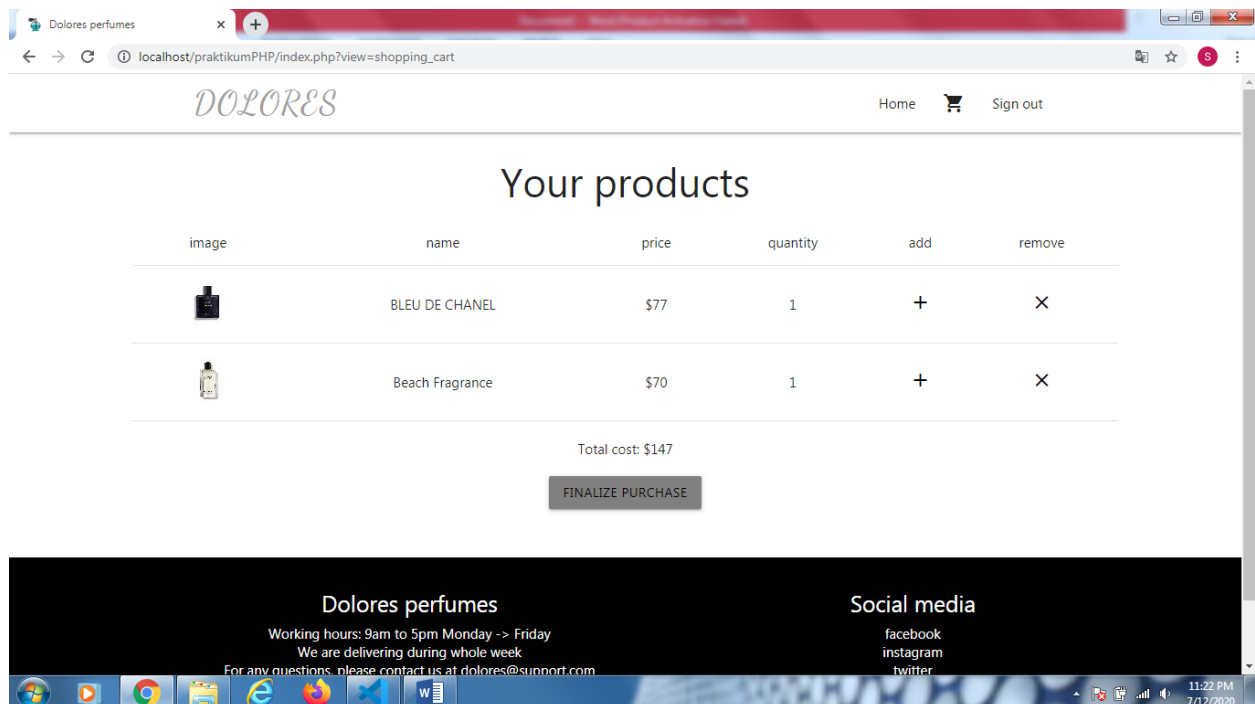
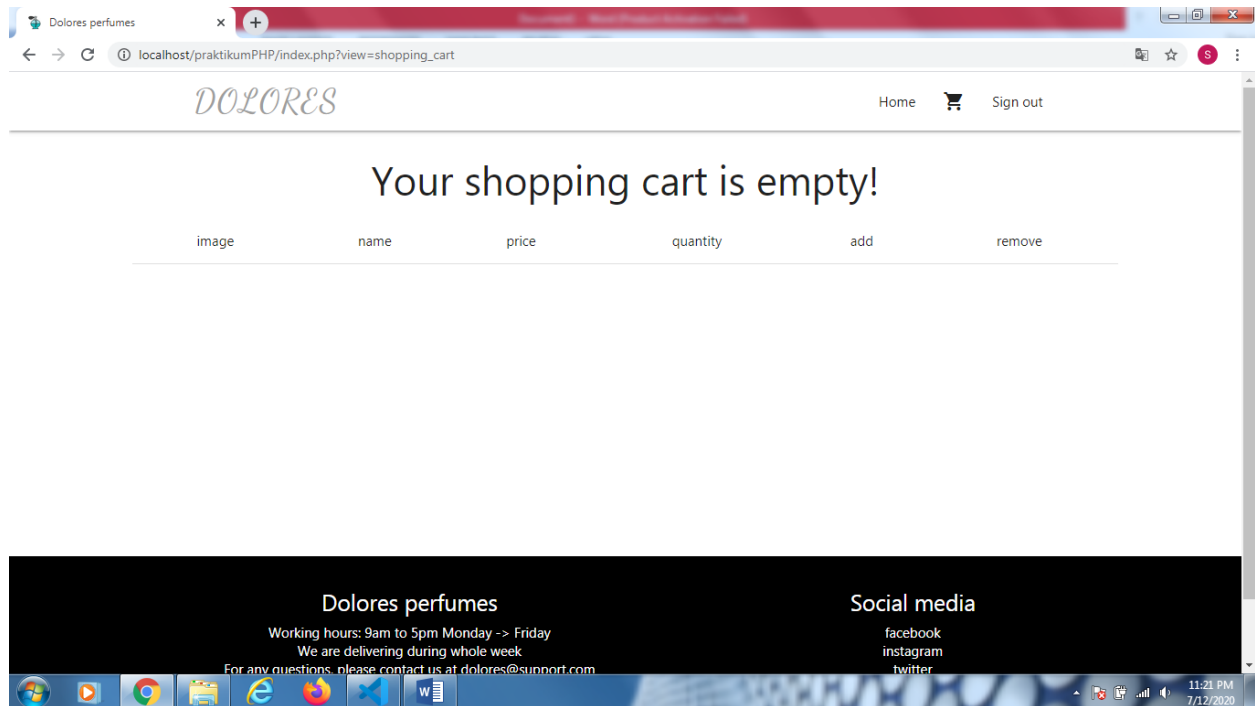
© 2020 Stefan Bogdanovski 85/18

Social media
facebook
instagram
twitter

Documentation Author

Index.php stranica služi za registraciju i prijavu korisnika, filtriranje sortiranje i pregled proizvoda, na dnu je footer sa informacijama i vezama ka socijalnim mrežama.

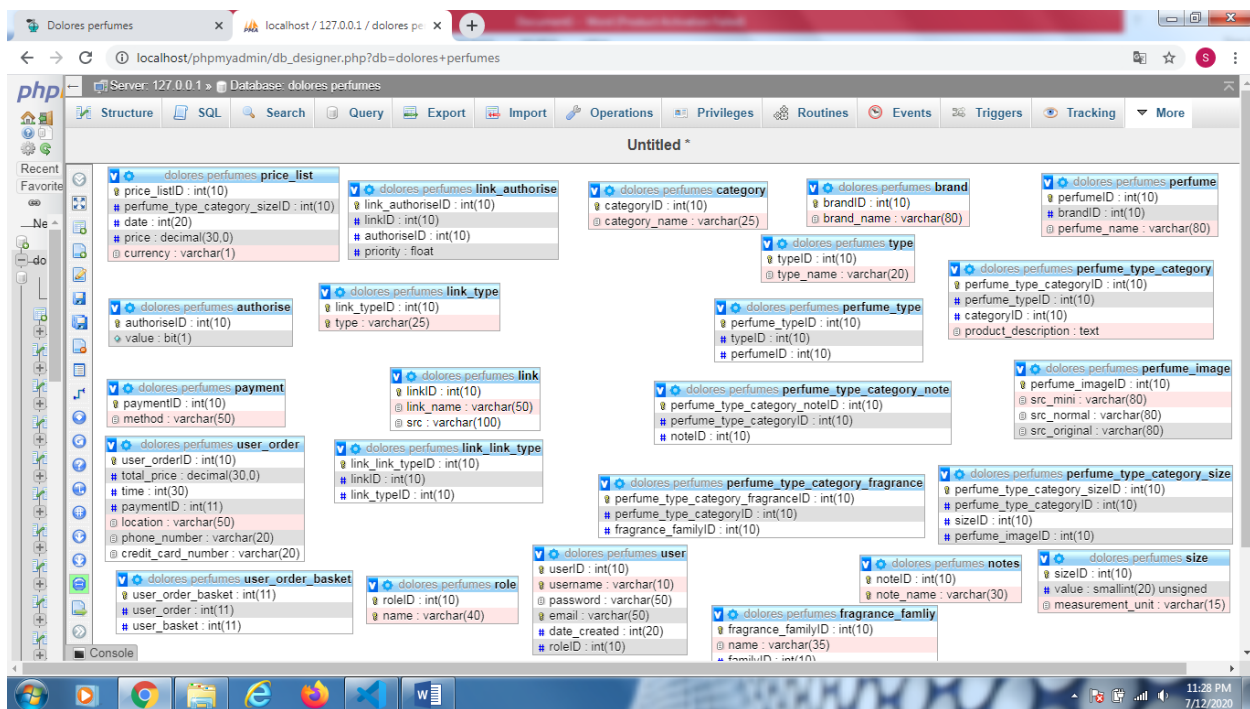
Index.php?view=shopping_cart



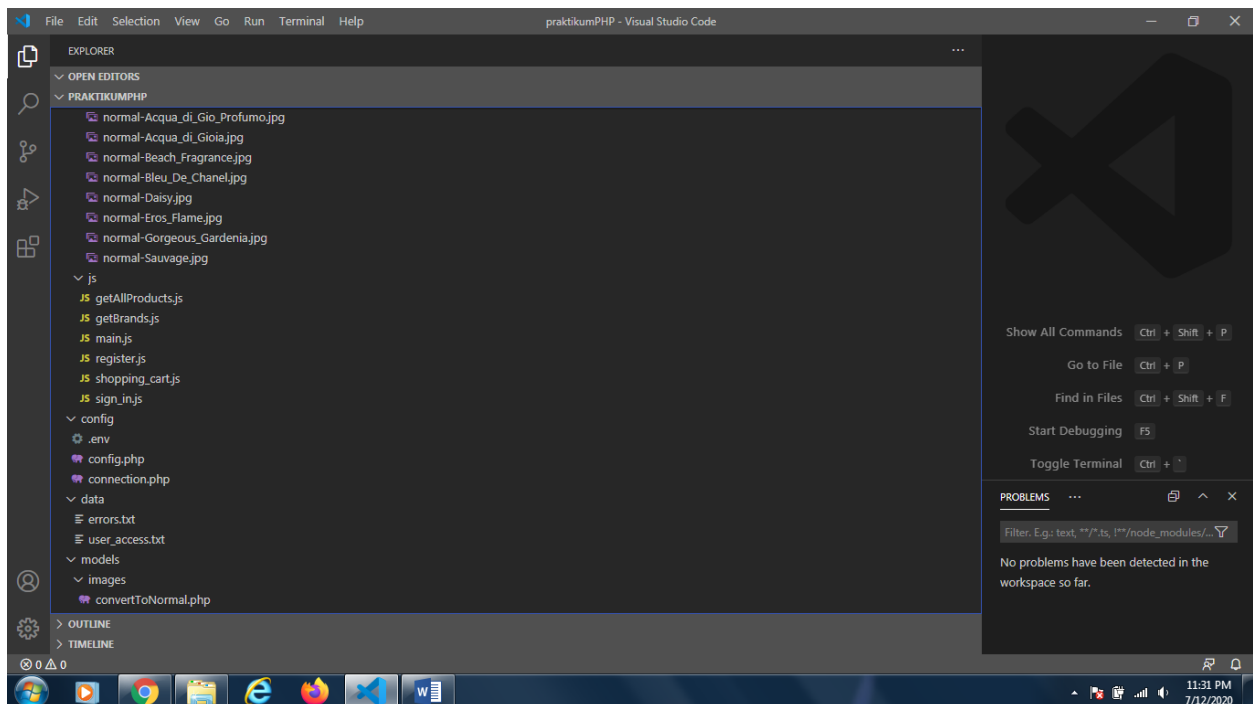
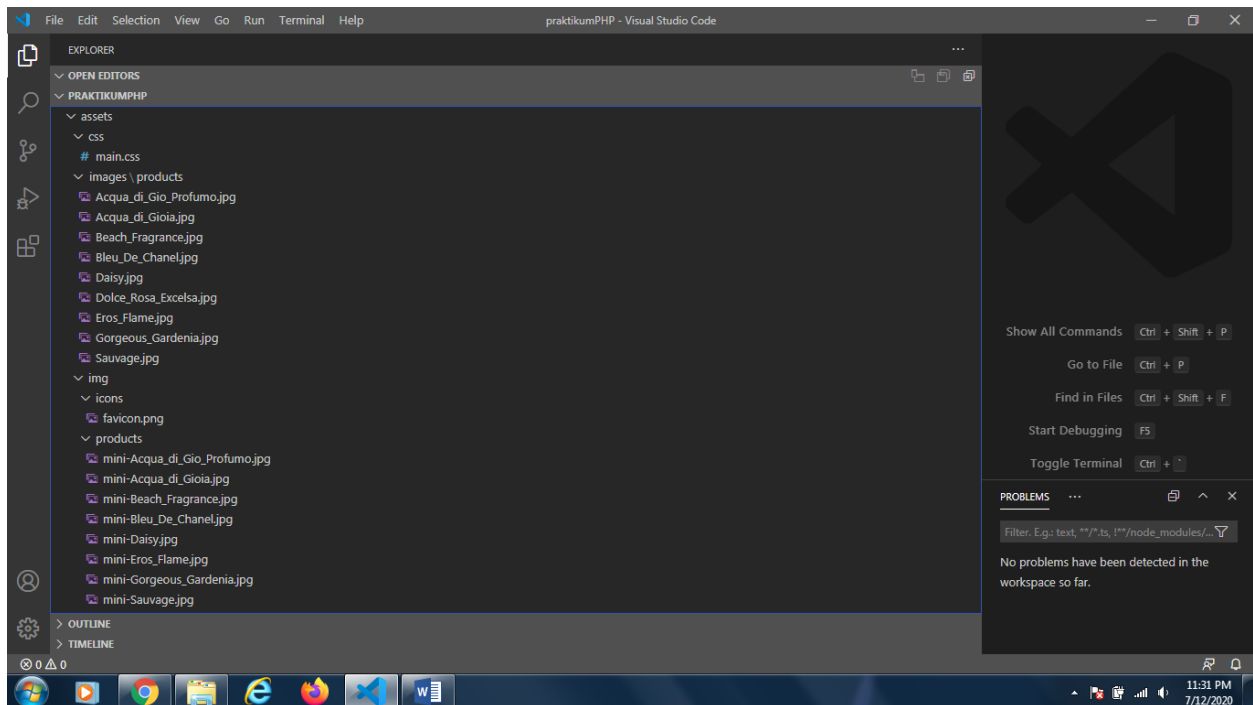
Prikaz korpe korisniku kome je omoguceno da dodaje i
brise proizvode, prikazana mu je ukupna cena za sve
proizvode, klikom na finalize purchase otvara se modal
koji se popunjava informacijama korisnika.

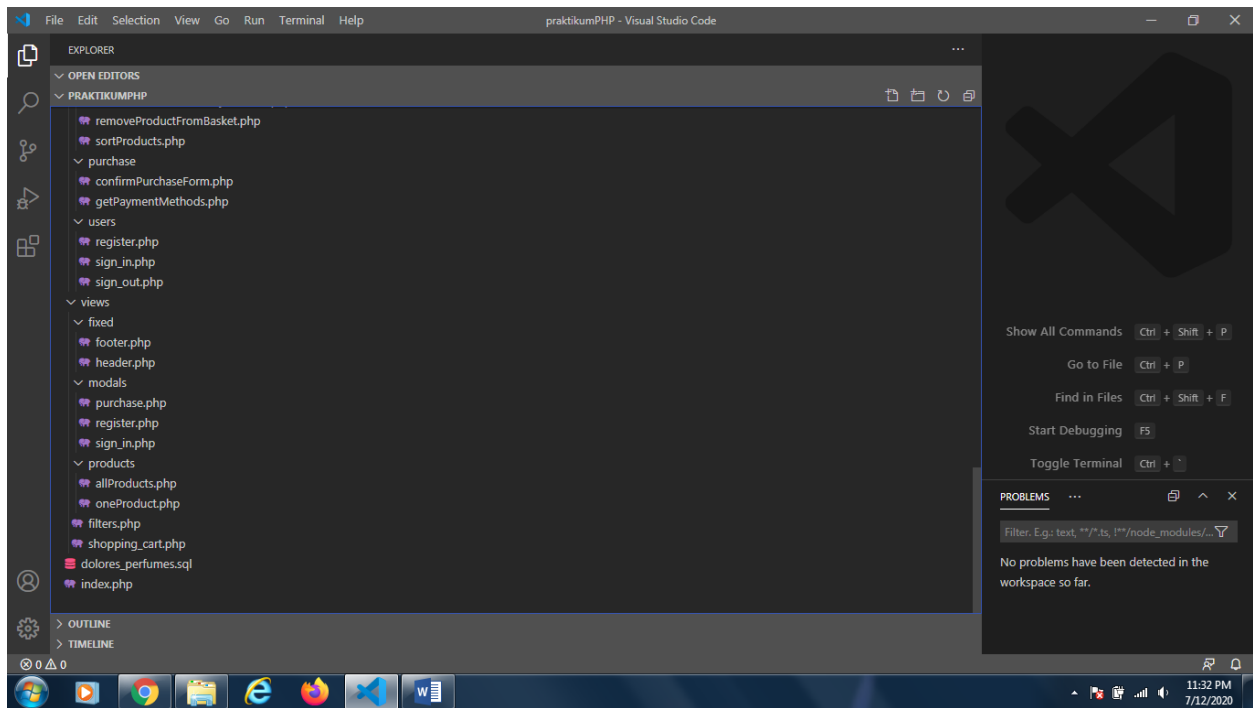
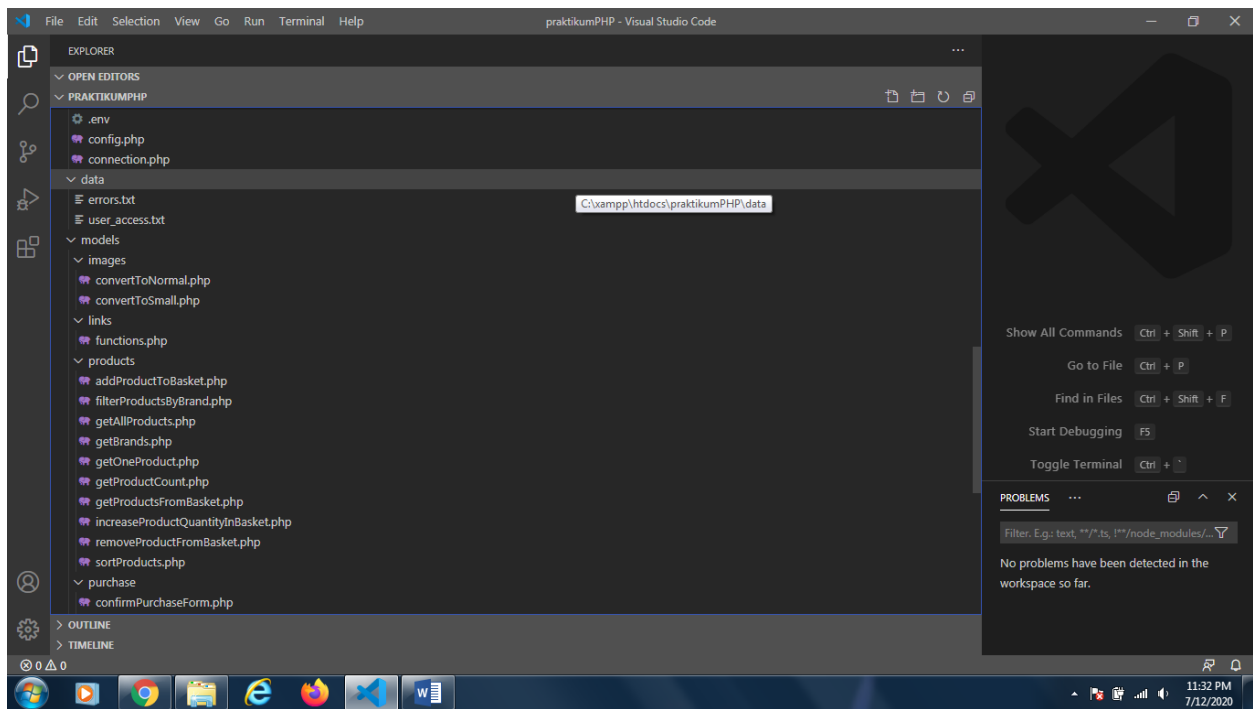
DIZAJN BAZE

* (Unapred se izvinjavam jer ne postoje linije koje oznacavaju veze, pokušavao sam na razne nacine da popravim ali nisam uspeo.)



STRUKTURA PROJEKTA





KODOVI

Assets/css/Main.css

```
@import url('https://fonts.googleapis.com/css2?family=Dancing+Script&display=swap');
*
{
    margin: 0px;
    padding: 0px;
}
html{
    line-height: normal;
}
a {
    text-decoration: none;
}
/* OVERRIDE MATERIALIZE */

#menuBars
{
    font-size: 35px !important;
}
/* label color */
.input-field label {
    color: #000 !important;
}
/* label focus color */
.input-field input[type=text]:focus + label, .input-field input[type=password]:focus + label {
    color: #000 !important;
}
/* label underline focus color */
.input-field input[type=text]:focus, .input-field input[type=password]:focus {
    border-bottom: 1px solid #000 !important;
    box-shadow: 0 1px 0 0 #000 !important;
}
/* valid color */
.input-field input[type=text].valid, .input-field input[type=password].valid {
    border-bottom: 1px solid #000 !important;
    box-shadow: 0 1px 0 0 #000 !important;
}
```

```

/* invalid color */
.input-field input[type=text].invalid, .input-
field input[type=password].invalid {
  border-bottom: 1px solid #000 !important;
  box-shadow: 0 1px 0 0 #000 !important;
}
/* icon prefix focus color */
.input-field .prefix.active {
  color: #000 !important;
}

/* OVERRIDE MATERIALIZE - END */

.center
{
  margin: 0px auto !important;
}
#Logo
{
  letter-spacing: 2px;
  text-transform: uppercase;
  font-size: 2.4rem;
  font-family: 'Dancing Script', cursive;
}
#ModelSignInConfirm, #ModelSignInCancel
{
  font-size: 16px;
  padding: 10px;
}
.card .card-action a:not(.btn):not(.btn-large):not(.btn-small):not(.btn-
large):not(.btn-floating)
{
  margin-right: 0px;
  font-weight: bold;
  color: #000;
}
.btn, .btn-large, .btn-small
{
  background-color: grey !important;
}
.btn:hover, .btn-large:hover, .btn-small:hover{
  background-color: honeydew !important;
  color: #000 !important;
}
.text-bold{

```

```

    font-weight: bold;
}
#sortPriceDdl
{
    display: block;
    width: 70%;
    margin: 0px auto;
    border-radius: 10px;
}
#filter{
    border-radius: 10px;
    -webkit-box-shadow: 0 2px 2px 0 rgba(0,0,0,0.14), 0 3px 1px -
2px rgba(0,0,0,0.12), 0 1px 5px 0 rgba(0,0,0,0.2);
    box-shadow: 0 2px 2px 0 rgba(0,0,0,0.14), 0 3px 1px -
2px rgba(0,0,0,0.12), 0 1px 5px 0 rgba(0,0,0,0.2);
}

#brands
{
    margin: 0px auto;
    margin-top: 10px;
    margin-bottom: 10px;
    border-radius: 10px;
    border: 1px solid #000;
    width: 70%;
    padding: 10px;
}
[type="checkbox"]:not(:checked), [type="checkbox"]:checked {
    position: static;
    opacity: 1;
    pointer-events: all;
}
.checkboxWrapper
{
    display: flex;
    justify-content: space-between;
}
.fluid-padding{
    padding: 0% 3%;
}
.card .card-content
{
    padding: 0px;
}
.card

```

```
{
  margin: 0 0 1rem 0;
  border-radius:5px;
}
#productsHeaderDown{
  display: none;
}
.pt-2
{
  padding-top: 20px;
}
.mt-2{
  margin-top:15px;
}
.mt-3{
  margin-top:30px;
}
#product_details_card
{
  width: 100%;
  margin: 0 auto;
  margin-top: 5%;
}
#productInBasket{
  margin: 0 auto;
}
#shopping_wrapper{
  width:80%;
  min-height: 350px;
  margin: 0px auto;
}
#total_cost
{
  padding: 20px 0px;
}
#paymentDdl
{
  display : block !important;
  width: 85% !important;
  margin: 0 auto;
}
.footer_bottom_links
{
  margin-right: 10px;
}
```



```
#footer_logo
{
    text-align: center;
}
.fontFooterInfo
{
    font-size:20px !important;
}
/* MEDIA QUERY*/
@media screen and (max-width : 600px)
{
}
@media screen and (max-width : 992px)
{
    .card
    {
        margin: .5rem 0 0 0;
    }
    #sortPriceDdl
    {
        display: block;
        width: 50%;
        margin: 0px auto;
        border-radius: 10px;
    }
    #productsHeaderUp{
        display: none;
    }
    #productsHeaderDown{
        display: block;
        padding: 10px 0px 10px 0px;
    }
    #brands
    {
        margin: 0px auto;
        margin-top: 10px;
        margin-bottom: 10px;
        border-radius: 10px;
        border: 1px solid #000;
        width:50%;
        padding: 10px;
    }
}
```

Assets/js/getAllProducts.js

```
$(document).ready(function()
{
    getAllProducts();
    $('#sortPriceDdl').change(sortProducts);
    $('body').on('change', 'input[name="brand[]"]', filterProductsByBrand);
});

function getAllProducts(brands, page)
{
    $.ajax(
    {
        method : "post",
        url : "models/products/getAllProducts.php",
        dataType : "JSON",
        contentType : "application/json",
        data : {
            request : true,
            brand : brands,
            page : page
        },
        success : function(items)
        {
            populateProducts(items);
        },
        error : function(xhr, error, status)
        {
            console.log(xhr, error, status);
        }
    }
    );
}

function populateProducts(items)
{
    html = ``;
    items.forEach(element => {
        html += `
        <div class="col s6 m4 l3">
        <div class="card medium">
        <div class="card-image">
        <a href="index.php?id=${element.perfume_type_category_sizeID}">
        
    `;
    });
}
```

```

        </a>
    </div>
    <div class="card-content center-align card_padding">
        <h6 > <a href="index.php?id=${element.perfume_type_category_sizeID}" class="black-text text-bold"> ${element.perfume_name} </a> </h6>
        <h6>${element.type_name}</h6>
        <span class="productPrice"> ${element.currency}${element.price} </span>
        <br/>
        <span class="productVolume"> ${element.value}${element.measurement_unit} </span>
    </div>
    <div class="card-action center-align">
        <a href="index.php?id=${element.perfume_type_category_sizeID}" class="grey-text darken-2">Show more</a>
    </div>
</div>`;
});
document.querySelector("#products").innerHTML = html;
}

```

```

function sortProducts()
{
    let sort = $(this).val();
    let checkboxes = $('input[name="brand[]"]');
    let checked = [];
    for(let i = 0; i < checkboxes.length; i++)
    {
        if(checkboxes[i].checked)
        {
            checked.push(checkboxes[i].value);
        }
    }
    console.log(checked);
    if(checked.length)
    {
        $.ajax(
        {
            method : "post",
            url : "models/products/sortProducts.php",
            dataType : "json",
            data : {
                request : true,
                sort : sort,
                brand : checked,
            }
        })
    }
}

```

```

        },
        success : function(items)
        {
            populateProducts(items);
        },
        error : function(xhr, error, status)
        {
            console.log(xhr, error, status);
        }
    }
    );
}
else
{
    $.ajax(
    {
        method : "post",
        url : "models/products/sortProducts.php",
        dataType : "json",
        data : {
            request : true,
            sort : sort
        },
        success : function(items)
        {
            populateProducts(items);
        },
        error : function(xhr, error, status)
        {
            console.log(xhr, error, status);
        }
    }
    );
}
}

function filterProductsByBrand()
{
    let checkboxes = $('input[name="brand[]"]');
    //console.log(checkboxes);
    let checked = [];
    //console.log(checkboxes.length);
    for(let i = 0; i < checkboxes.length; i++)
    {
        if(checkboxes[i].checked)
    
```

```

        {
            checked.push(checkboxes[i].value);
        }
    }
    console.log(checked);
    let sort = $('#sortPriceDdl').val();
    if(checked.length && sort == 0)
    {
        $.ajax(
            {
                method : "post",
                url : "models/products/filterProductsByBrand.php",
                dataType : "json",
                data : {
                    request : true,
                    brand : checked
                },
                success : function(items)
                {
                    populateProducts(items);
                },
                error : function(xhr, error, status)
                {
                    console.log(xhr, error, status);
                }
            }
        );
    }
    else if(sort != 0 && checked.length)
    {
        $.ajax(
            {
                method : "post",
                url : "models/products/filterProductsByBrand.php",
                dataType : "json",
                data : {
                    request : true,
                    sort : sort,
                    brand : checked
                },
                success : function(items)
                {
                    populateProducts(items);
                },
                error : function(xhr, error, status)

```

```
        {
            console.log(xhr, error, status);
        }
    }
    );
}
else
{
    $.ajax(
    {
        method : "post",
        url : "models/products/sortProducts.php",
        dataType : "json",
        data : {
            request : true,
            sort : sort
        },
        success : function(items)
        {
            populateProducts(items);
        },
        error : function(xhr, error, status)
        {
            console.log(xhr, error, status);
        }
    }
    );
}
}
```

Assets/js/getBrands.js

```
$(document).ready(function()
{
    getBrands();
});
function getBrands()
{
    $.ajax(
    {
        method : "POST",
        url : "models/products/getBrands.php",
        dataType: "json",
        contentType: "application/json",
        data : {
            "request" : true
        },
        success : function(brands)
        {
            populateBrands(brands);
        },
        error : function(xhr, status, error)
        {
            console.log(xhr);
        }
    }
    );
}

function populateBrands(items)
{
    let html = ``;
    items.forEach(element => {
        html+= `<div class="checkboxWrapper">
            <label for="${element.brand_name}"> ${element.brand_name} </label><input type
            ="checkbox" value="${element.brandID}" id="${element.brand_name}" name="brand["/
            >
            </div>`;
    });
    document.querySelector("#brands").innerHTML += html;
}
```

Assets/js/Main.js

```
$(document).ready(function()  
{  
    $('.sidenav').sidenav();  
    $('.modal').modal();  
});
```


Assets/js/Register.js

```
$(document).ready(function()
{
    //alert("Cao");
    $("#ModelRegisterConfirm").click(CheckUserRegistration);
});

function CheckUserRegistration()
{
    let success = true;
    let usernameTb = $("#tbUsernameReg");
    let passwordTb = $("#tbPasswordReg");
    let emailTb = $("#tbEmailReg");
    let username = $("#tbUsernameReg").val();
    let password = $("#tbPasswordReg").val();
    let email = $("#tbEmailReg").val();
    //console.log(username, password, email);
    let usernameRegEx = /^[A-Z]+[A-z0-9]{5,9}$/;
    let passwordRegEx = /^[^.{6,}$/;
    let emailRegEx = /^[a-z]+[0-9a-z\.*@]{2,10}(\.[a-z]{2,10})+$/;
    if(!usernameRegEx.test(username))
    {
        removeHelperClass(usernameTb, "teal-text");
        addHelperClass(usernameTb, "red-text");
        success = false;
    }
    else
    {
        addHelperClass(usernameTb, "teal-text");
        removeHelperClass(usernameTb, "red-text");
    }
    if(!passwordRegEx.test(password))
    {
        removeHelperClass(passwordTb, "teal-text");
        addHelperClass(passwordTb, "red-text");
        success = false;
    }
    else
    {
        addHelperClass(passwordTb, "teal-text");
        removeHelperClass(passwordTb, "red-text");
    }
    if(!emailRegEx.test(email))
```

```
{
    removeHelperClass(emailTb, "teal-text");
    addHelperClass(emailTb, "red-text");
    success = false;
}
else
{
    addHelperClass(emailTb, "teal-text");
    removeHelperClass(emailTb, "red-text");
}
if(success)
{
    $.ajax(
    {
        method : "POST",
        dataType : "json",
        url : "models/users/register.php",
        data : {
            username : username,
            password : password,
            email : email,
            request : success
        },
        success : function(code)
        {
            if(code == 200)
            {
                location.reload();
                alert("Account successfully created.");
            }
            if (code.includes("Invalid username"))
            {
                removeHelperClass(usernameTb, "teal-text");
                addHelperClass(usernameTb, "red-text");
            }
            else
            {
                addHelperClass(usernameTb, "teal-text");
                removeHelperClass(usernameTb, "red-text");
            }
            if(code.includes("Already member"))
            {
                alert("Account with same username already exists.");
                addHelperClass(usernameTb, "red-text");
                removeHelperClass(usernameTb, "teal-text");
            }
        }
    })
}
```

```

        }
        if (code.includes("Invalid password"))
        {
            removeHelperClass(passwordTb, "teal-text");
            addHelperClass(passwordTb, "red-text");
        }
        else
        {
            addHelperClass(passwordTb, "teal-text");
            removeHelperClass(passwordTb, "red-text");
        }
        if (code.includes("Invalid email"))
        {
            removeHelperClass(emailTb, "teal-text");
            addHelperClass(emailTb, "red-text");
        }
        else
        {
            removeHelperClass(emailTb, "red-text");
            addHelperClass(emailTb, "teal-text");
        }
    },
    error : function(xhr, status, error)
    {
        console.log(xhr);
    }
}
)
}
}

function removeHelperClass(input, className)
{
    input.next().next().removeClass(className);
}
function addHelperClass(input, className)
{
    input.next().next().addClass(className);
}

```

Assets/js/Shopping_cart.js

```
$(document).ready(function()
{
    $("#add_to_basket").click(addToBasket);
    const urlParams = new URLSearchParams(window.location.search);
    const myParam = urlParams.get('view');
    if(myParam=="shopping_cart")
    {
        getProductsFromBasket();
        getPaymentMethods();
        $('body').on('click', 'a.increaseQuantity', increaseQuantity);
        $('body').on('click', 'a.removeItem', removeFromBasket);
        $('body').on('click', 'a#ModelPurchaseConfirm', checkPurchaseForm);
        $('body').on('change', 'select#paymentDdl', displayCardInput);
    }
});

function addToBasket()
{
    let idProduct = ($(this).data("idproduct"));
    $.ajax(
    {
        method : "POST",
        url : "models/products/addProductToBasket.php",
        dataType : "json",
        data : {
            "id" : idProduct,
            "request" : true
        },
        success : function(message)
        {
            if(message == 200)
            {
                alert("Successfully added!");
                getProductsFromBasket();
            }
            else if(message == 400)
            {
                alert("Increasing product quantity unavailable.");
                alert(message);
            }
            else
            {
            }
```

```

        alert(message.message);
    }
},
error : function(xhr)
{

}

}

);
}
function displayCardInput()
{
    let value = $(this).val();
    console.log(value);
    if(value == 2)
    {
        $("#credit_card_number").html(` <div class="input-field col s12">
            <i class="material-icons prefix">credit_card</i>
            <input type="text" id="ccNumber" name="ccNumber"/>
            <label for="tbAddress">Your credit card number</label>
            <span class="helper-text"> E.g. 1111-1111-1111-1111</span>
        </div>`);
    }
    else
    {
        $("#credit_card_number").html("");
    }
}
function getProductsFromBasket()
{
    $.ajax(
    {
        method : "POST",
        url : "models/products/getProductsFromBasket.php",
        dataType : "json",
        data : {
            "request" : true
        },
        success : function(items)
        {
            showProductsFromBasket(items);
        },
        error : function(xhr)
        {

```

```

    }
  }
);
}
function showProductsFromBasket(items)
{
  if(items.length)
  {
    $("#shopping_title").html("Your products");
    let html='';
    let sum=0;
    items.forEach(element => {
      sum+=element.price*element.quantity;
      html += `
        <tr>
          <td class="center-align">
            <span class="center-
align">  </span>
            </td>
            <td>
              <p class="center-align"> ${element.perfume_name} </p>
            </td>
            <td>
              <p class="center-
align"> ${element.currency}${element.price}</p>
            </td>
            <td>
              <p class="center-align"> ${element.quantity} </p>
            </td>
            <td>
              <p class="center-
align"> <a href="#" id="${element.price_listID}" class="increaseQuantity"><i clas
s="material-icons black-text">add</i></a>
            </td>
            <td>
              <p class="center-
align"> <a href="#" id="${element.price_listID}" class="removeItem"><i class="mat
erial-icons black-text">clear</i></a>
            </td>
          </tr>
        `;
    });
    $("#products").html(html);
    $("#total_cost").html("Total cost: $" + sum);
  }
}

```

```

        $("#total_cost").next().html(`<a href="#modalPurchase" class="black-
text btn modal-trigger">Finalize purchase</a>`);
    }
    else
    {
        $("#products").html("");
        $("#total_cost").html("");
        $("#shopping_title").html("Your shopping cart is empty!");
    }
}
function increaseQuantity()
{
    let id = $(this).attr("id");
    $.ajax(
        {
            method : "POST",
            url : "models/products/increaseProductQuantityInBasket.php",
            dataType : "text",
            data : {
                "request" : true,
                "id" : id
            },
            success: function(response) {
                return getProductsFromBasket();
            },
            error : function(xhr)
            {
            },
        }
    );
}
function removeFromBasket()
{
    let id = $(this).attr("id");
    $.ajax(
        {
            method : "POST",
            url : "models/products/removeProductFromBasket.php",
            dataType : "text",
            data : {
                "request" : true,
                "id" : id
            },
            success: function(response) {
                alert("Succesfully deleted.");
            }
        }
    );
}

```

```

        return getProductsFromBasket();
    },
    error : function(xhr)
    {

    },
    },
    }
);
}
function getPaymentMethods()
{
    $.ajax(
    {
        method : "POST",
        url : "models/purchase/getPaymentMethods.php",
        dataType : "json",
        data : {
            "request" : true,
        },
        success: function(items) {
            console.log(items);
            populatePaymentMethods(items);
        },
        error : function(xhr)
        {

        },
        },
    }
    );
}
function populatePaymentMethods(items)
{
    html = `<option value="0"> Choose.. </option>`;
    console.log(items);
    items.forEach(element => {
        console.log(element);
        html += `<option value="${element.paymentID}"> ${element.method} </option>`;
    });
    console.log(html);
    $("#paymentDdl").html(html);
}
function checkPurchaseForm()
{
    let success = true;

```



```

let inputPhone = $("#tbPhoneNumber");
let inputAddress = $("#tbAddress");
let inputPayment = $("#paymentDdl");
let phoneRegex = /^[0-9]{9,15}$/;
let addressRegex = /^[A-zČĆĐŠŽčćđšž\.\s+](\s[A-zČĆĐŠŽčćđšž][a-zčćđšž]+)+\s?((([0-9\w]{2,4})|([0-9]+)))/;
if(!phoneRegex.test(inputPhone.val()))
{
    removeHelperClass(inputPhone, "teal-text");
    addHelperClass(inputPhone, "red-text");
    success = false;
}
else
{
    addHelperClass(inputPhone, "teal-text");
    removeHelperClass(inputPhone, "red-text");
}
if(!addressRegex.test(inputAddress.val()))
{
    removeHelperClass(inputAddress, "teal-text");
    addHelperClass(inputAddress, "red-text");
    success = false;
}
else
{
    addHelperClass(inputAddress, "teal-text");
    removeHelperClass(inputAddress, "red-text");
}
if(inputPayment.val() == 0)
{
    removeHelperClass(inputPayment, "teal-text");
    addHelperClass(inputPayment, "red-text");
    success = false;
}
else
{
    addHelperClass(inputPayment, "teal-text");
    removeHelperClass(inputPayment, "red-text");
}

if(success)
{
    let phoneNumber = inputPhone.val();
    let address = inputAddress.val();
    let paymentMethod = inputPayment.val();

```

```
if(paymentMethod == 2)
{
    $.ajax(
    {
        method : "POST",
        url : "models/purchase/confirmPurchaseForm.php",
        dataType : "text",
        data : {
            "request" : true,
            "address" : address,
            "phoneNumber" : phoneNumber,
            "paymentMethod" : paymentMethod,
            "ccNumber" : ccNumber
        },
        success: function(message) {
            alert(message);
        },
        error : function(xhr)
        {

        },

    });
}
else
{
    $.ajax(
    {
        method : "POST",
        url : "models/purchase/confirmPurchaseForm.php",
        dataType : "text",
        data : {
            "request" : true,
            "address" : address,
            "phoneNumber" : phoneNumber,
            "paymentMethod" : paymentMethod
        },
        success: function(message) {
            alert(message);
        },
        error : function(xhr)
        {

        },

    });
}
```

```
    }  
}  
function removeHelperClass(input, className)  
{  
    input.next().next().removeClass(className);  
}  
function addHelperClass(input, className)  
{  
    input.next().next().addClass(className);  
}
```

Assets/js/Sign_in.js

```
$(document).ready(function() {
    $("#ModelSignInConfirm").click(validateUser);
});

function validateUser()
{
    let success = true;
    let usernameTb = $("#tbUsernameSignIn");
    let passwordTb = $("#tbPasswordSignIn");
    let username = usernameTb.val();
    let password = passwordTb.val();
    let usernameRegex = /^[A-Z]+[A-z0-9]{5,9}$/;
    let passwordRegex = /^.{6,}$/;
    if(!usernameRegex.test(username))
    {
        success = false;
        printMessage(usernameTb.next().next(), "Username is 6 to 10 character
s long.");
    }
    else
    {
        printMessage(usernameTb.next().next(), "");
    }
    if(!passwordRegex.test(password))
    {
        success = false;
        printMessage(passwordTb.next().next(), "Password is at least 6 charac
ters long.");
    }
    else
    {
        printMessage(passwordTb.next().next(), "");
    }
    if(success)
    {
        $.ajax({
            method : "POST",
            dataType : "json",
            url : "models/users/sign_in.php",
            data : {
                username : username,
                pw : password,
```

```

        request : true
    },
    success : function(code)
    {
        if(code == 200)
        {
            location.reload();
        }
        if(code.includes("Invalid username"))
        {
            printMessage(usernameTb.next().next(), "Username doesn't
meet our requirements.");
        }
        if(code.includes("Invalid password"))
        {
            printMessage(passwordTb.next().next(), "Password doesn't
meet our requirements");
        }
        if(code.includes("Incorrect username"))
        {
            printMessage(usernameTb.next().next(), "Username does not
exist.");
        }
        if(code.includes("Incorrect password"))
        {
            printMessage(passwordTb.next().next(), "Password is incor
rect.");
        }
    },
    error : function(xhr, status, error)
    {
        console.log(xhr);
    }
});
}

function printMessage(input, message)
{
    input.html(message);
}

```

Config/.env

database=dolores perfumes!

server=localhost!

username=root!

password=!

Config/Config.php

```
<?php

#Settings
define("base_url", "http://127.0.0.1/praktikumPHP/");

define("absolute_path", $_SERVER['DOCUMENT_ROOT'] . "/praktikumPHP" . "/");

define("env_file" , absolute_path . "config/.env");

define("log_file", absolute_path . "data/user_access.txt");

define("err_file", absolute_path . "data/errors.txt");
#Database parameters

define("database" , read_environment_variable("database"));
define("server", read_environment_variable("server"));
define("username", read_environment_variable("username"));
define("password", read_environment_variable("password"));

#Getting database parameters
function read_environment_variable($variable_name)
{
    $file = file(env_file);
    $variable_value = "";
    foreach ($file as $key => $value) {
        $line = explode("=", $value);
        if($line[0]==$variable_name)
        {
            $variable_value = explode("!", $line[1])[0];
        }
    }
    return $variable_value;
}

?>
```

Config/Connection.php

```
<?php

session_start();
require_once "config.php";
note_user_access();
    try {
        $base = new PDO("mysql:host=".server.";dbname=".database,username,password);
    } catch (Throwable $ex) {
        exit("Connection with database is currently unavailable, please try later.");
    }
    function select_query($query)
    {
        global $base;
        try {
            return $base->query($query)->fetchAll();
        } catch (Throwable $ex) {
            note_error("Select query error:" . $ex->getMessage());
        }
    }
    function note_user_access()
    {
        $open_log_file = fopen(log_file, "a");
        if($open_log_file)
        {
            fwrite($open_log_file, "{$_SERVER['PHP_SELF']}{$_SERVER['QUERY_STRING']} \t {$_SERVER['REMOTE_ADDR']}" . "\t\t" . date("d.m.Y \t H:i:s", time()) . "\r\n");
            fclose($open_log_file);
        }
    }
    function note_error($message)
    {
        $open_err_file = fopen(err_file, "a");
        if($open_err_file)
        {
```



```
        fwrite($open_err_file, "{$_SERVER['PHP_SELF']}{$_SERVER['QUERY_STRING']}\t {$_SERVER['REMOTE_ADDR']}" . "\t\t" . $message . "\t\t" . date("d.m.Y \t H:i:s", time())) . "\r\n");
        fclose($open_err_file);
    }
}
?>
```

Models/Images/ConvertToNormal.php

```
<?php
function imageConvertToNormal($path){
    $fixedWidth = 500;
    $image = imagecreatefromjpeg($path);
    list($width, $height) = getimagesize($path);
    $newHeight = $height * $fixedWidth / $width;
    $plate = imagecreatetruecolor($fixedWidth, $newHeight);
    imagecopyresized($plate,$image,0,0,0,0,$fixedWidth,$newHeight,$width,$height)
;
    $name = getNormalImageName($path);
    imagejpeg($plate, "assets/img/products/" . $name);
    imagedestroy($plate);
}
function getNormalImageName($path)
{
    $array = explode('/', $path);
    $full_name = end($array);
    return "normal-" . $full_name;
}

?>
```

Models/images/ConvertToSmall.php

```
<?php
function imageConvertToSmall($path){
    $fixedWidth = 50;
    $image = imagecreatefromjpeg($path);
    list($width, $height) = getimagesize($path);
    $newHeight = $height * $fixedWidth / $width;
    $plate = imagecreatetruecolor($fixedWidth, $newHeight);
    imagecopyresized($plate,$image,0,0,0,0,$fixedWidth,$newHeight,$width,$height)
;
    $name = getMiniImageName($path);
    imagejpeg($plate, "assets/img/products/" . $name);
    imagedestroy($plate);
}
function getMiniImageName($path)
{
    $array = explode('/', $path);
    $full_name = end($array);
    return "mini-" . $full_name;
}

?>
```

Models/links/functions.php

```
<?php

function get_nav_links($authorised)
{
    $query_nav_links = "SELECT l.link_name, l.src from link l
    INNER JOIN link_link_type llt
    ON l.linkID = llt.linkID
    INNER JOIN link_type lt
    ON llt.link_typeID = lt.link_typeID
    INNER JOIN link_authorise la
    ON l.linkID = la.linkID
    INNER JOIN authorise a
    ON a.authoriseID = la.authoriseID
    WHERE lt.type = 'header' AND a.value = $authorised
    ORDER BY la.priority DESC";

    return select_query($query_nav_links);
}

function get_social_links()
{
    $query_social_links = "SELECT l.link_name, l.src from link l
    INNER JOIN link_link_type llt
    ON l.linkID = llt.linkID
    INNER JOIN link_type lt
    ON llt.link_typeID = lt.link_typeID
    WHERE lt.type = 'footer_social'";

    return select_query($query_social_links);
}

?>
```

Models/products/addProductToBasket.php

```
<?php
    if(isset($_POST['id']) && isset($_POST['request']) && is_numeric($_POST['id'])
))
    {
        session_start();
        error_reporting(E_ALL & ~E_NOTICE);
        if(!isset($_SESSION['user']))
        {
            http_response_code(404);
        }
        $id = $_POST['id'];
        $user = $_SESSION['user'];
        $userId = $user[0]->userID;
        $quantity = 1;
        require_once "../../config/connection.php";
        try {
            //CheckIfProductExist
            $check = "SELECT * FROM user_basket ub WHERE ub.price_listID = $id AND
D ub.userID = $userId";
            $ready = $base->prepare($check);
            $ready->execute();
            $count = $ready->rowCount();
            if($count)
            {
                $quantityArray = select_query("SELECT quantity FROM user_basket u
b WHERE ub.price_listID = $id AND ub.userID = $userId");
                //increase product quantity
                $quantity= $quantityArray[0]->quantity;
                $quantity++;
                $prepare_query = "UPDATE user_basket ub
SET ub.quantity = ?
WHERE ub.price_listID = ? AND ub.userID = ?";
                $ready = $base->prepare($prepare_query);
                $finish = $ready->execute(array($quantity, $id, $userId));
                if($finish)
                {
                    header("Content-Type: application/json");
                    echo json_encode(200);
                }
            }
            else
            {
                header("Content-Type: application/json");
```

```

        echo json_encode(400);
    }
}
else
{
    //add product
    try
    {
        $insertBasketQuery = "INSERT INTO user_basket VALUES (null, ?, ?, ?)";

        $insertBasket = $base->prepare($insertBasketQuery);
        $success = $insertBasket->execute(array($id, $userId, $quantity));
        if($success)
        {
            header("Content-Type: application/json");
            $message = 'Product added to basket!';
            $toSend = [ "message" => $message ];
            echo json_encode($toSend);
        }
        else
        {
            header("Content-Type: application/json");
            $message = 'Problem occurred while adding product to basket. Please contact administrator.';
            $toSend = [ "message" => $message ];
            echo json_encode($toSend);
        }
    }
    catch(Throwable $ex)
    {
        note_error("Insert product to basket: " . $ex->getMessage());
    }
}
} catch (\Throwable $th) {
    note_error("Increase product quantity: " . $th->getMessage());
}
// header("Content-Type: application/json");
// $message = 'Product added to basket!';
// $toSend = [ "message" => $message ];
// echo json_encode($toSend);
}
else
{
    http_response_code(400);
}

```

```
}
```

```
?>
```

Models/products/filterProductsByBrand.php

```
<?php

if(isset($_POST['request']) && !isset($_POST['sort']))
{
    $brands = $_POST['brand'];
    require_once "../config/connection.php";
    $query = "SELECT * FROM perfume_type_category_size ptcs
    INNER JOIN perfume_type_category ptc
    ON ptcs.perfume_type_categoryID = ptc.perfume_type_categoryID
    INNER JOIN perfume_type pt
    ON ptc.perfume_typeID = pt.perfume_typeID
    INNER JOIN perfume p
    ON pt.perfumeID = p.perfumeID
    INNER JOIN brand b
    ON p.brandID = b.brandID
    INNER JOIN type t
    ON pt.typeID = t.typeID
    INNER JOIN perfume_image p_i
    ON ptcs.perfume_imageID = p_i.perfume_imageID
    INNER JOIN price_list pl
    ON ptcs.perfume_type_category_sizeID = pl.perfume_type_category_sizeID
    INNER JOIN size s
    ON ptcs.sizeID = s.sizeID
    WHERE ";
    foreach($brands as $key => $value)
    {
        #not last brand
        if($key < count($brands) - 1)
        {
            $query .= "b.brandID = " . $value. " OR ";
        }
        #last brand
        else
        {
            $query .= "b.brandID = " . $value;
        }
    }
    $products = select_query($query);
    header("Content-Type: application/json");
    echo json_encode($products);
}
else if(isset($_POST['request']) && isset($_POST['sort']))
```



```

{
    $brands = $_POST['brand'];
    $order;
    $sort = $_POST['sort'];
    $order;
    switch ($sort) {
        case '1':
            $order = " ORDER BY pl.price DESC";
            break;

        case '2':
            $order = " ORDER BY pl.price ASC";
            break;
        default:
            $order = " ORDER BY p.perfume_name ASC";
            break;
    }
    require_once "../../config/connection.php";
    $brands = $_POST['brand'];
    $query = "SELECT * FROM perfume_type_category_size ptcs
    INNER JOIN perfume_type_category ptc
    ON ptcs.perfume_type_categoryID = ptc.perfume_type_categoryID
    INNER JOIN perfume_type pt
    ON ptc.perfume_typeID = pt.perfume_typeID
    INNER JOIN perfume p
    ON pt.perfumeID = p.perfumeID
    INNER JOIN brand b
    ON p.brandID = b.brandID
    INNER JOIN type t
    ON pt.typeID = t.typeID
    INNER JOIN perfume_image p_i
    ON ptcs.perfume_imageID = p_i.perfume_imageID
    INNER JOIN price_list pl
    ON ptcs.perfume_type_category_sizeID = pl.perfume_type_category_sizeID
    INNER JOIN size s
    ON ptcs.sizeID = s.sizeID
    WHERE ";
    foreach($brands as $key => $value)
    {
        #not last brand
        if($key < count($brands) - 1)
        {
            $query .= "b.brandID = " . $value. " OR ";
        }
        #last brand

```

```
        else
        {
            $query .= "b.brandID = " . $value;
        }
    }
    $query .= $order;
    $products = select_query($query);
    header("Content-Type: application/json");
    echo json_encode($products);
}
else
{
    http_response_code(404);
}

?>
```

Models/products/getAllProducts.php

```
<?php

if(!isset($_POST['request']))
{
    require_once "../../config/connection.php";
    $query = "SELECT * FROM perfume_type_category_size ptcs
INNER JOIN perfume_type_category ptc
ON ptcs.perfume_type_categoryID = ptc.perfume_type_categoryID
INNER JOIN perfume_type pt
ON ptc.perfume_typeID = pt.perfume_typeID
INNER JOIN perfume p
ON pt.perfumeID = p.perfumeID
INNER JOIN brand b
ON p.brandID = b.brandID
INNER JOIN type t
ON pt.typeID = t.typeID
INNER JOIN perfume_image p_i
ON ptcs.perfume_imageID = p_i.perfume_imageID
INNER JOIN price_list pl
ON ptcs.perfume_type_category_sizeID = pl.perfume_type_category_sizeID
INNER JOIN size s
ON ptcs.sizeID = s.sizeID
ORDER BY p.perfumeID";
    $products = select_query($query);
    if(!empty($products))
    {
        header("Content-Type: application/json");
        echo json_encode($products);
    }
    else
    {
        header("Content-Type: application/json");
        echo json_encode("Getting products unavailable, contact administrator.");
    }
}
else
{
    http_response_code(404);
}

?>
```

Models/products/getBrands.php

```
<?php
if(!isset($_POST['request']))
{
    require_once "../../config/connection.php";
    $query = "SELECT * FROM brand
ORDER BY brand_name";
    $brands = select_query($query);
    header("Content-Type: application/json");
    echo json_encode($brands);
}
else
{
    http_response_code(404);
}

?>
```

getOneProduct.php

```
<?php

function getProduct($id)
{
    $query = "SELECT perfume_name,pl.price_listID, brand_name, category_name,
src_normal, src_mini,note_name, price, currency, value, measurement_unit, type_n
ame, fff.name as Family, ff.name as Scent FROM perfume_type_category_size ptcs
    INNER JOIN perfume_type_category ptc
    ON ptcs.perfume_type_categoryID = ptc.perfume_type_categoryID
    INNER JOIN perfume_type pt
    ON ptc.perfume_typeID = pt.perfume_typeID
    INNER JOIN perfume p
    ON pt.perfumeID = p.perfumeID
    INNER JOIN brand b
    ON p.brandID = b.brandID
    INNER JOIN type t
    ON pt.typeID = t.typeID
    INNER JOIN perfume_image p_i
    ON ptcs.perfume_imageID = p_i.perfume_imageID
    INNER JOIN price_list pl
    ON ptcs.perfume_type_category_sizeID = pl.perfume_type_category_sizeID
    INNER JOIN size s
    ON ptcs.sizeID = s.sizeID
    INNER JOIN perfume_type_category_fragrance ptcf
    ON ptcs.perfume_type_categoryID = ptcf.perfume_type_categoryID
    INNER JOIN fragrance_famliy ff
    ON ptcf.fragrance_familyID = ff.fragrance_familyID
    INNER JOIN fragrance_famliy fff
    ON ff.familyID = fff.fragrance_familyID
    INNER JOIN perfume_type_category_note ptcn
    ON ptcs.perfume_type_categoryID = ptcn.perfume_type_categoryID
    INNER JOIN notes n
    ON ptcn.noteID = n.noteID
    INNER JOIN category c
    ON ptc.categoryID = c.categoryID
    WHERE p.perfumeID = $id";
    return select_query($query);
}

?>
```

Models/products/getProductCount.php

(Namenjena za paginaciju, nije iskorisceno)

```
<?php

function countProducts($brand)
{
    if(!$brand)
    {
        $query = "SELECT COUNT(*) as broj
        FROM perfume";
    }
    else
    {
        $query = "SELECT COUNT(*) as broj
        FROM perfume
        WHERE ";
        foreach($brand as $key => $value)
        {
            #Ako nije poslednji brend
            if($key < count($brand) - 1)
            {
                $query .= "brandID = " . $value. " OR ";
            }
            #Ako je poslednji brend
            else
            {
                $query .= "brandID = " . $value;
            }
        }
    }
    return select_query($query);
}

?>
```

Models/products/getProductsFromBasket.php

```
<?php
    session_start();
    error_reporting(E_ALL & ~E_NOTICE);
    if(isset($_POST['request']) && isset($_SESSION['user']))
    {
        require_once "../../config/connection.php";
        $userID = $_SESSION['user'][0]->userID;
        try {
            $query = "SELECT * FROM user_basket ub
            INNER JOIN price_list pl
                ON ub.price_listID = pl.price_listID
            INNER JOIN perfume_type_category_size ptcs
                ON pl.perfume_type_category_sizeID = ptcs.perfume_type_category_s
            INNER JOIN perfume_image per_img
                ON ptcs.perfume_imageID = per_img.perfume_imageID
            INNER JOIN perfume_type_category ptc
                ON ptcs.perfume_type_categoryID = ptc.perfume_type_categoryID
            INNER JOIN perfume_type pt
                ON ptc.perfume_typeID = pt.perfume_typeID
            INNER JOIN perfume p
                ON pt.perfumeID = p.perfumeID
            WHERE userID = $userID";
            $items = select_query($query);
            header("Content-Type: application/json");
            echo json_encode($items);
        } catch (\Throwable $th) {
            note_error("Getting products from basket: ".$th->getMessage());
        }
    }
    else
    {
        http_response_code(404);
    }
?>
```

Models/products/IncreaseProductQuantityInBasket.php

```
<?php

session_start();
error_reporting(E_ALL & ~E_NOTICE);
if(isset($_POST['request']) && isset($_POST['id']))
{
    $id = $_POST['id'];
    $userID = $_SESSION['user'][0]->userID;
    try {
        require_once "../../config/connection.php";
        //get quantity
        $GetQuery = "SELECT ub.quantity FROM user_basket ub WHERE ub.price_listID = $id AND ub.userID = $userID";
        $quantity = select_query($GetQuery)[0]->quantity;
        var_dump($quantity);
        try {
            //set quantity
            $SetQuery = "UPDATE user_basket SET quantity = ? WHERE price_listID = ? AND userID = ?";
            $prepared = $base->prepare($SetQuery);
            $success = $prepared->execute(array(++$quantity, $id, $userID));
            if($success)
            {
                header("Content-Type: application/json");
                $message = array("Successful");
                echo json_encode($message);
            }
            else
            {
                header("Content-Type: application/json");
                $message = array("Unsuccessful");
                echo json_encode($message);
            }
        } catch (\Throwable $th) {
            note_error("Setting product quantity " . $th->getMessage());
        }
    } catch (\Throwable $th) {
        note_error("Getting product quantity " . $th->getMessage());
    }
}
```



```
else
{
    http_response_code(404);
}

?>
```

Models/products/removeProductFromBasket.php

```
<?php
    session_start();
    error_reporting(E_ALL & ~E_NOTICE);
    if(isset($_POST['request']) && isset($_POST['id']))
    {
        $id = $_POST['id'];
        $userID = $_SESSION['user'][0]->userID;
        try {
            require_once "../config/connection.php";
            $removeQuery = "DELETE FROM user_basket WHERE price_listID = ? AND userID = ?";
            $prepared = $base->prepare($removeQuery);
            $success = $prepared->execute(array($id, $userID));
            var_dump($success);
            if($success)
            {
                header("Content-Type: application/json");
                echo json_encode("Success");
            }
            else
            {
                header("Content-Type: application/json");
                echo json_encode("Unsuccessful");
            }
        } catch (\Throwable $th) {
            note_error("Deleting product from basket: " . $th->getMessage());
        }
    }

?>
```

Models/products/sortProducts.php

```
<?php
if(isset($_POST['request']) && isset($_POST['brand']))
{
    $brands = $_POST['brand'];
    require_once "../../config/connection.php";
    $sort = $_POST['sort'];
    if($sort == 1)
    {
        $query = "SELECT * FROM perfume_type_category_size ptcs
INNER JOIN perfume_type_category ptc
ON ptcs.perfume_type_categoryID = ptc.perfume_type_categoryID
INNER JOIN perfume_type pt
ON ptc.perfume_typeID = pt.perfume_typeID
INNER JOIN perfume p
ON pt.perfumeID = p.perfumeID
INNER JOIN brand b
ON p.brandID = b.brandID
INNER JOIN type t
ON pt.typeID = t.typeID
INNER JOIN perfume_image p_i
ON ptcs.perfume_imageID = p_i.perfume_imageID
INNER JOIN price_list pl
ON ptcs.perfume_type_category_sizeID = pl.perfume_type_category_sizeI

D
INNER JOIN size s
ON ptcs.sizeID = s.sizeID WHERE ";
        foreach($brands as $key => $value)
        {
            #not last brand
            if($key < count($brands) - 1)
            {
                $query .= "b.brandID = " . $value. " OR ";
            }
            #last brand
            else
            {
                $query .= "b.brandID = " . $value;
            }
        }
        $order = " ORDER BY pl.price DESC";
        $query .= $order;
        $products = select_query($query);
    }
}
```

```

        if(!empty($products))
        {
            header("Content-Type: application/json");
            echo json_encode($products);
        }
        else
        {
            header("Content-Type: application/json");
            echo json_encode("Getting products unavailable, contact administrator.");
        }
    }
    else if ($sort == 2)
    {
        require_once "../../config/connection.php";
        $query = "SELECT * FROM perfume_type_category_size ptcs
INNER JOIN perfume_type_category ptc
ON ptcs.perfume_type_categoryID = ptc.perfume_type_categoryID
INNER JOIN perfume_type pt
ON ptc.perfume_typeID = pt.perfume_typeID
INNER JOIN perfume p
ON pt.perfumeID = p.perfumeID
INNER JOIN brand b
ON p.brandID = b.brandID
INNER JOIN type t
ON pt.typeID = t.typeID
INNER JOIN perfume_image p_i
ON ptcs.perfume_imageID = p_i.perfume_imageID
INNER JOIN price_list pl
ON ptcs.perfume_type_category_sizeID = pl.perfume_type_category_sizeID

INNER JOIN size s
ON ptcs.sizeID = s.sizeID WHERE ";
        foreach($brands as $key => $value)
        {
            #not last brand
            if($key < count($brands) - 1)
            {
                $query .= "b.brandID = " . $value. " OR ";
            }
            #last brand
            else
            {
                $query .= "b.brandID = " . $value;
            }
        }
    }

```

```

    }
    $order = " ORDER BY pl.price ASC";
    $query .= $order;
    $products = select_query($query);
    if(!empty($products))
    {
        header("Content-Type: application/json");
        echo json_encode($products);
    }
    else
    {
        header("Content-Type: application/json");
        echo json_encode("Getting products unavailable, contact administrator.");
    }
}
else
{
    require_once "../config/connection.php";
    $query = "SELECT * FROM perfume_type_category_size ptcs
    INNER JOIN perfume_type_category ptc
    ON ptcs.perfume_type_categoryID = ptc.perfume_type_categoryID
    INNER JOIN perfume_type pt
    ON ptc.perfume_typeID = pt.perfume_typeID
    INNER JOIN perfume p
    ON pt.perfumeID = p.perfumeID
    INNER JOIN brand b
    ON p.brandID = b.brandID
    INNER JOIN type t
    ON pt.typeID = t.typeID
    INNER JOIN perfume_image p_i
    ON ptcs.perfume_imageID = p_i.perfume_imageID
    INNER JOIN price_list pl
    ON ptcs.perfume_type_category_sizeID = pl.perfume_type_category_sizeID

    INNER JOIN size s
    ON ptcs.sizeID = s.sizeID WHERE ";
    foreach($brands as $key => $value)
    {
        #not last brand
        if($key < count($brands) - 1)
        {
            $query .= "b.brandID = " . $value. " OR ";
        }
        #last brand

```

```

        else
        {
            $query .= "b.brandID = " . $value;
        }
    }
    $order = " ORDER BY p.perfumeID";
    $query .= $order;
    $products = select_query($query);
    if(!empty($products))
    {
        header("Content-Type: application/json");
        echo json_encode($products);
    }
    else
    {
        header("Content-Type: application/json");
        echo json_encode("Getting products unavailable, contact administrator.");
    }
}
}
else if(isset($_POST['request']) && !isset($_POST['brand']))
{
    require_once "../config/connection.php";
    $sort = $_POST['sort'];
    $order;
    switch ($sort) {
        case '1':
            $order = "ORDER BY pl.price DESC";
            break;

        case '2':
            $order = "ORDER BY pl.price ASC";
            break;
        default:
            $order = "ORDER BY p.perfume_name ASC";
            break;
    }
    $query = "SELECT * FROM perfume_type_category_size ptcs
INNER JOIN perfume_type_category ptc
ON ptcs.perfume_type_categoryID = ptc.perfume_type_categoryID
INNER JOIN perfume_type pt
ON ptc.perfume_typeID = pt.perfume_typeID
INNER JOIN perfume p
ON pt.perfumeID = p.perfumeID";

```

```

INNER JOIN brand b
ON p.brandID = b.brandID
INNER JOIN type t
ON pt.typeID = t.typeID
INNER JOIN perfume_image p_i
ON ptcs.perfume_imageID = p_i.perfume_imageID
INNER JOIN price_list pl
ON ptcs.perfume_type_category_sizeID = pl.perfume_type_category_sizeID
INNER JOIN size s
ON ptcs.sizeID = s.sizeID " . $order;
$products = select_query($query);
if(!empty($products))
{
    header("Content-Type: application/json");
    echo json_encode($products);
}
else
{
    header("Content-Type: application/json");
    echo json_encode("Getting products unavailable, contact administrator.");
}
}
else
{
    http_response_code(404);
}
?>

```

Models/purchase/confirmPurchaseForm.php

```
<?php
    session_start();
    error_reporting(E_ALL & ~E_NOTICE);
    if(isset($_POST['request']))
    {
        require_once "../../config/connection.php";
        $userID = $_SESSION['user'][0]->userID;

        //get all products from basket
        $queryGetProducts = "SELECT * FROM user_basket ub
        INNER JOIN price_list pl
        ON ub.price_listID = pl.price_listID
        WHERE userID = $userID";
        $products = select_query($queryGetProducts);

        //make an order
        $address = $_POST['address'];
        $phone = $_POST['phoneNumber'];
        $payment = $_POST['paymentMethod'];
        if(isset($_POST['ccNumber']))
        {
            $ccNumber = $_POST[['ccNumber']];
        }

        try {
            if(isset($ccNumber))
            {
                $insertQuery = "INSERT INTO user_order
                VALUES (null, null, ?, ?, ?, ?, ?)";
                $ready = $base->prepare($insertQuery);
                $success = $ready->
>execute(array(time(), $payment, $address, $phone, $ccNumber));

                if($success)
                {
                    //insert into common table
                    $lastID = $base->
>query("SELECT MAX(user_orderID) as max FROM user_order")->fetch()->max;

                    try {
                        $sum = 0;
                        foreach ($products as $key)
```



```

        {
            $sum += $key->price*$key->quantity;
            $query = "INSERT INTO user_order_basket
VALUES (null, ?, ?)";
            $prepared = $base->prepare($query);
            $finished = $prepared->execute(array($lastID, $key-
>user_basketID));
        }
        $queryUpdateSum = "UPDATE user_order SET total_price
= ?";

        $prepareUpdateQuery = $base-
>prepare($queryUpdateSum);
        $finish = $prepareUpdateQuery->execute(array($sum));
        if($finish)
        {
            header("Content-Type: application/json");
            $message = "Order succesfully made!";
            echo json_encode($message);
        }
    }
    catch (\Throwable $th) {
        note_error("Insert into common: ". $th);
        header("Content-Type: application/json");
        echo json_encode(401);
    }
}
else
{
    header("Content-Type: application/json");
    echo json_encode(402);
}
}
else
{
    $insertQuery = "INSERT INTO user_order
VALUES (null, null, ?, ?, ?, ?, null)";
    $ready = $base->prepare($insertQuery);
    $success = $ready-
>execute(array(time(), $payment, $address, $phone));
    if($success)
    {
        //insert into common table
        $lastID = $base-
>query("SELECT MAX(user_orderID) as max FROM user_order")->fetch()->max;

```

```

        try {
            $sum = 0;
            foreach ($products as $key)
            {
                $sum += $key->price*$key->quantity;
                $query = "INSERT INTO user_order_basket
VALUES (null, ?, ?)";
                $prepared = $base->prepare($query);
                $finished = $prepared->execute(array($lastID, $key->user_basketID));

            }
            $queryUpdateSum = "UPDATE user_order SET total_price
= ?";

            $prepareUpdateQuery = $base->prepare($queryUpdateSum);
            $finish = $prepareUpdateQuery->execute(array($sum));
            if($finish)
            {
                header("Content-Type: application/json");
                $message = "Order succesfully made! Check your email.";

                echo json_encode($message);
            }
        }
        catch (\Throwable $th) {
            note_error("Insert into common: ". $th);
            header("Content-Type: application/json");
            echo json_encode(401);
        }
    }
    else
    {
        header("Content-Type: application/json");
        echo json_encode(402);
    }
}
} catch (\Throwable $th) {
    note_error("Making order: " . $th);
}
}
else
{
    http_response_code(400);
}

```

?>

Models/purchase/getPaymentMethods.php

```
<?php

if(isset($_POST['request']))
{
    require_once "../../config/connection.php";
    try {
        $query = "SELECT * FROM payment";
        $methods = select_query($query);
        header("Content-Type: application/json");
        echo json_encode($methods);
    } catch (\Throwable $th) {
        note_error("Getting payment methods: ". $th->getMessage());
    }
}
else
{
    http_response_code(404);
}

?>
```

Models/users/register.php

```
<?php

if(isset($_POST['request']) && $_POST['request'])
{
    $arrayErr = [];
    $username = trim($_POST['username']);
    $password = $_POST['password'];
    $email = $_POST['email'];
    if(!preg_match('/^[A-Z]+[A-z0-9]{5,9}$/ ', $username))
    {
        array_push($arrayErr, "Invalid username");
    }
    if(!preg_match('/^.{6,}$/ ', $password))
    {
        array_push($arrayErr, "Invalid password");
    }
    if(!preg_match('/^[a-z]+[0-9a-z\.\_]*@[a-z]{2,10}(\.[a-z]{2,10})+$/ ', $email))
    {
        array_push($arrayErr, "Invalid email");
    }

    if(!empty($arrayErr))
    {
        header("Content-Type: application/json");
        echo json_encode($arrayErr);
        exit();
    }
    else
    {
        require_once "../config/connection.php";
        $alreadyMember = $base->
>prepare("SELECT * FROM user WHERE username = '$username'");
        $alreadyMember->execute();
        $count = $alreadyMember->rowCount();
        if(!$count)
        {
            $password = md5($password);
            $role = 1;
            $insert_query = $base->prepare("INSERT INTO user
            VALUES (null, :username, :passw, :email, :register_time, :user_role)");
        }
    }
};

try {
```

```
        $success = $insert_query->
>execute(array(':username' => $username, ':passw' => $password, ':email' => $email, ':register_time' => time(), ':user_role' => $role));
        header("Content-Type: application/json");
        if($success)
        {
            echo json_encode(200);
        }
    } catch (\Throwable $th) {
        note_error("Insert query registration:" . $th->getMessage());
    }
}
else
{
    header("Content-Type: application/json");
    echo json_encode("Already member");
}
}
}
else
{
    http_response_code(404);
}
?>
```

Models/users/sign_in.php

```
<?php
if(isset($_POST['request']) && $_POST['request'])
{
    $username = trim($_POST['username']);
    $password = $_POST['pw'];
    $arrayErr = [];
    if(!preg_match('/^[A-Z]+[A-z0-9]{5,9}$/ ', $username))
    {
        array_push($arrayErr, "Invalid username");
    }
    if(!preg_match('/^.{6,}$/', $password))
    {
        array_push($arrayErr, "Invalid password");
    }
    if(!empty($arrayErr))
    {
        header("Content-Type: application/json");
        echo json_encode($arrayErr);
    }
    else
    {
        $password = md5($password);
        require_once "../config/connection.php";
        $check_username = $base-
>prepare("SELECT * FROM user WHERE username = :username");
        $check_username->execute(array(':username' => $username));
        if($check_username->rowCount() == 1)
        {
            $query_check = $base-
>prepare("SELECT * FROM user WHERE username = :username AND password = :pass");
            try {
                $query_check-
>execute(array(':username' => $username, ':pass' => $password));
                $count = $query_check->rowCount();
                if($count == 1)
                {
                    try {
                        $_SESSION['user'] = select_query("SELECT * FROM user
WHERE username = '$username' AND password = '$password'");
                        header("Content-Type: application/json");
                        echo json_encode(200);
                    } catch (\Throwable $th) {
```

```
                                note_error("Getting user after signing:" . $th->getMessage());
>getMessage());
    }
    }
    else
    {
        header("Content-Type: application/json");
        echo json_encode("Incorrect password");
    }
} catch (\Throwable $th) {
    note_error("Sign in check query:" . $th->getMessage());
}
}
else
{
    header("Content-Type: application/json");
    echo json_encode("Incorrect username");
}
}
}
else
{
    http_response_code(404);
}
?>
```


Models/users/sign_out.php

```
<?php
  session_start();
  unset($_SESSION['user']);
  header("Location:" . $_SERVER['HTTP_REFERER']);
?>
```

Views/fixed/footer.php

```
<?php

?>
<footer class="page-footer black">
    <div class="container">
        <div class="row">
            <div class="col l6 s12">
                <h5 class="white-text" id="footer_logo">Dolores perfumes</h5>
                <p class="white-text text-lighten-4 center-align">
                    <ul class="center-align">
                        <li>Working hours: 9am to 5pm Monday -> Friday</li>
                        <li>We are delivering during whole week</li>
                        <li>For any questions, please contact us at dolores@support.
com</li>

                    </ul>
                </p>
            </div>
            <div class="col l4 offset-l2 s12 center-align">
                <h5 class="white-text">Social media</h5>
                <ul class="center-align" id="social_media">
                    <?php
                        require_once "models/links/functions.php";
                        $links = get_social_links();
                    ?>
                    <?php foreach ($links as $link):?>
                        <li><a href="<?=$link->src?" target="_blank" class="white-
text"> <?= $link->link_name ?></a> </li>
                    <?php endforeach;?>
                </ul>
            </div>
        </div>
        <div class="footer-copyright">
            <div class="container white-text">
                <span class="footer_bottom_links"> &copy; 2020 Stefan Bogdanovski 85/
18 </span>
                <a class="white-
text right footer_bottom_links" target="_blank" href="https://technologyy.netlify
.app/about.html">Author</a>
                <a class="white-
text right footer_bottom_links" href="#!">Documentation</a>
```

```
        </div>
    </div>
</footer>
```

Views/fixed/header.php

```
<?php
    require_once "models/links/functions.php";
    $authorised = isset($_SESSION['user']) ? 1 : 0;
    $linkovi = get_nav_links($authorised);
?>
<header>
    <nav class="white">
        <div class="nav-wrapper">
            <div class="container">
                <a href="index.php" class="brand-logo grey-text darken-2" id="Logo">Dolores</a>
                <a href="#" data-target="mobile-demo" class="sidenav-trigger grey-text text-darken-4"><i class="material-icons small" id="menuBars">menu</i></a>
                <ul class="right hide-on-med-and-down grey-text text-darken-4">
                    <?php foreach ($linkovi as $key):?>

                        <li>
                            <a href="<?=$key->src?>" class="grey-text text-darken-4"
                                <?php if($key->link_name == "Sign in" || $key->link_name == "Register")
                                    {
                                        echo " modal-trigger";
                                    }
                                ?>"
                            <?=$key->link_name?>
                            </a>
                        </li>

                    <?php endforeach;?>
                </ul>
            </div>
        </nav>
        <ul class="sidenav grey-text text-darken-4" id="mobile-demo">
            <?php foreach ($linkovi as $key):?>

                <li>
                    <a href="<?=$key->src?>" class="grey-text text-darken-4"
                        <?php if($key->link_name == "Sign in" || $key->link_name == "Register")
                            {
```

```
        echo " modal-trigger";
    }
    ?>">
    <?=$key->link_name?>
</a>
</li>

    <?php endforeach;?>
</ul>
<?php require_once "views/modals/sign_in.php"; ?>
<?php require_once "views/modals/register.php"; ?>

</header>
```

Views/modals/purchase.php

```
<?php

?>

<div id="modalPurchase" class="modal">
  <div class="modal-content">
    <h4 class="center-align">Fill in the purchase form</h4>
    <div class="row">
      <form class="col s12">
        <div class="row">
          <div class="input-field col s12">
            <i class="material-icons prefix">contact_phone</i>
            <input id="tbPhoneNumber" type="text" class="validate">
            <label for="tbPhoneNumber">Phone number</label>
            <span class="helper-text">Numbers only.</span>
          </div>
        </div>
        <div class="row">
          <div class="input-field col s12">
            <i class="material-icons prefix">home</i>
            <input id="tbAddress" type="text">
            <label for="tbAddress">Your address</label>
            <span class="helper-text">City street and number</span>
          </div>
        </div>
        <div class="row">
          <div class="input-field col s12">
            <i class="material-icons prefix mt-2">credit_card</i>
            <p class="center-align">Payment method</p>
            <select name="payment" id="paymentDdl" class="center-align">
            </select>
            <span></span>
            <span class="helper-text">How would you like to pay?</span>
          </div>
        </div>
        <div class="row" id="credit_card_number">

        </div>
      </form>
    </div>
  </div>
</div>
```

```
<div class="modal-footer">
  <a href="#" id="ModelPurchaseConfirm" class="black-text">Confirm</a>
  <a href="#" id="ModelPurchaseCancel" class="modal-close black-
text">Cancel</a>
</div>
</div>
```

Views/modals/register.php

```
<div id="modal2" class="modal">
  <div class="modal-content center">
    <h4>Make new account within a minute!</h4>
    <div class="row">
      <form class="col s12">
        <div class="row">
          <div class="input-field col s12">
            <i class="material-icons prefix">account_circle</i>
            <input id="tbUsernameReg" type="text" class="validate">
            <label for="tbUsernameReg">Username</label>
            <span class="helper-
text" id="usernameErr">Username should start with capital letter. Numbers allowed
. Min:6, Max:10</span>
          </div>
        </div>
        <div class="row">
          <div class="input-field col s12">
            <i class="material-icons prefix">lock</i>
            <input id="tbPasswordReg" type="password">
            <label for="tbPasswordReg">Password</label>
            <span class="helper-
text" id="passwordErr">At least 6 characters.</span>
          </div>
        </div>
        <div class="row">
          <div class="input-field col s12">
            <i class="material-icons prefix">lock</i>
            <input id="tbEmailReg" type="text">
            <label for="tbEmailReg">E-mail</label>
            <span class="helper-text" id="emailErr">Input valid e-
mail address.</span>
          </div>
        </div>
      </form>
    </div>
    <div class="modal-footer">
      <a href="#" id="ModelRegisterConfirm" class="black-text">Confirm</a>
      <a href="#" id="ModelRegisterCancel" class="modal-close black-
text">Cancel</a>
    </div>
  </div>
</div>
```




Views/modals/sign_in.php

```
<div id="modal1" class="modal">
  <div class="modal-content center">
    <h4>Sign in with your account</h4>
    <div class="row">
      <div class="col s12">
        <div class="row">
          <div class="input-field col s12">
            <i class="material-icons prefix">account_circle</i>
            <input id="tbUsernameSignIn" type="text" class="validate">
            <label for="tbUsernameSignIn">Username</label>
            <span class="helper-text red-text"></span>
          </div>
        </div>
        <div class="row">
          <div class="input-field col s12">
            <i class="material-icons prefix">lock</i>
            <input id="tbPasswordSignIn" type="password">
            <label for="tbPasswordSignIn">Password</label>
            <span class="helper-text red-text"></span>
          </div>
        </div>
      </div>
    </div>
    <div class="modal-footer">
      <a href="#" id="ModelSignInConfirm" class="black-text">Confirm</a>
      <a href="#" id="ModelSignInCancel" class="modal-close black-
text">Cancel</a>
    </div>
  </div>
</div>
```

Views/products/allProducts.php

```
<?php
?>

<h4 class="center-align" id="productsHeaderDown"> Our products! </h4>

<div class="row" id="products">

</div>
<?php
    $limit = 6;
    # $linkNumber = ceil($count / $limit);

?>
```

Views/products/oneProduct.php

```
<?php
    require_once "models/products/getOneProduct.php";
    if(isset($_GET['id']) && is_numeric($_GET['id']))
    {
        $product = getProduct($_GET['id']);
        if(empty($product))
        {
            var_dump($product);
        }
    }
}

?>
<div class="container-fluid">
    <div class="row">
        <div class="col s12 m6 l5 center-align">
            <figure>
                perfume_name?>" class="responsive-img">
            </figure>
        </div>
        <div class="col s12 m6 center-align">
            <h5 class="pt-2 center-align">
                <b> <?=$product[0]->brand_name?> - <?=$product[0]->perfume_name?> </b>
            </h5>
            <p> <b> Price: </b> <?=$product[0]->currency?><?=$product[0]->price?></p>
            <p class="mt-2"> <b> Size: </b> <?=$product[0]->value?><?=$product[0]->measurement_unit?></p>
            <p class="mt-2"> <b> Category: </b> <?=$product[0]->category_name?></p>
            <p class="mt-2"> <b> Type: </b> <?=$product[0]->type_name?></p>
            <div class="card white" id="product_details_card">
                <div class="card-content black-text">
                    <span class="card-title black-text center-align">Details:</span>
                    <ul>
                        <li><b>Fragrance family: </b> <?=$product[0]->Family?></li>
                        <li class="mt-2"><b>Scent type: </b> <?=$product[0]->Scent?></li>
                        <li class="mt-2">
                            <b>Notes: </b>
                            <?php for($i = 0; $i<count($product); $i++)
```

```

        {
            if($i < count($product)-1)
            {
                echo $product[$i]->note_name . ", ";
            }
            else
            {
                echo $product[$i]->note_name;
            }
        }
    ?>
</li>
</ul>
</div>
<div class="card-action">
<div class="center-align">
    <?php if(!isset($_SESSION['user'])):?>
    <button class="btn-large disabled center-
align">Add to cart</button>
    <br/>
    <span class="center-
align"> <b>Only registered users</b></span>
    <?php else: ?>
    <button class="btn-large" id="add_to_basket" data-
idProduct="<?=$product[0]->price_listID?>"> Add to cart</button>
    <?php endif;?>
    </div>
</div>
</div>
</div>
</div>
</div>

```

Views/filters.php

```
<?php

?>

<div id="sort_by" class="center-align">
    <h5>Sort by</h5>
    <select id="sortPriceDdl">
        <option value="0">Choose..</option>
        <option value="1">Price descending</option>
        <option value="2">Price ascending</option>
    </select>
</div>

<div id="brands">

</div>
```

Views/shopping_cart.php

```
<?php

if(!isset($_SESSION['user']))
{
    header("Location: index.php");
}

?>

<div class="container-fluid">
<h3 class="header center-align" id="shopping_title"></h3>
    <div class="row" id="shopping_products">
        <div id="shopping_wrapper">
            <table>
                <thead>
                    <tr>
                        <td>
                            <p class="center-align"> image </p>
                        </td>
                        <td>
                            <p class="center-align"> name </p>
                        </td>
                        <td>
                            <p class="center-align"> price </p>
                        </td>
                        <td>
                            <p class="center-align"> quantity </p>
                        </td>
                        <td>
                            <p class="center-align"> add </p>
                        </td>
                        <td>
                            <p class="center-align"> remove </p>
                        </td>
                    </tr>
                </thead>
                <tbody id="products">

                </tbody>
            </table>
            <p class="center-align" id="total_cost"></p>
            <p class="center-align"></p>
        </div>
    </div>
</div>
```

```
        </div>
    </div>
</div>
<?php require_once "modals/purchase.php";?>
```


Index.php

```
<?php
    require "config/connection.php";

    ob_start();

?>

<?php if(isset($_SESSION['wrong_username'])): ?>
<script type="text/javascript"> alert(<?= $_SESSION['wrong_username']?>); </script>
<?php endif; unset($_SESSION['wrong_username']);?>

<?php if(isset($_SESSION['wrong_password'])): ?>
<script type="text/javascript"> alert(<?= $_SESSION['wrong_password']?>); </script>
<?php endif; unset($_SESSION['wrong_password']);?>

<?php if(isset($_SESSION['wrong_email'])): ?>
<script type="text/javascript"> alert(<?= $_SESSION['wrong_email']?>); </script>
<?php endif; unset($_SESSION['wrong_email']);?>

<?php if(isset($_SESSION['SameUsername'])): ?>
<script type="text/javascript"> alert(<?=$_SESSION['SameUsername']?>); </script>
<?php endif; unset($_SESSION['SameUsername']);?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8"/>
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet"/>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css"/>
    <link rel="stylesheet" type="text/css" href="assets/css/main.css"/>
    <link rel="shortcut icon" href="assets/img/icons/favicon.png"/>
    <!-- JQUERY -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <title >Dolores perfumes</title>
</head>
```

```

<body>
    <?php require_once "views/fixed/header.php"; ?>
    <?php if(isset($_GET['view']) && $_GET['view'] == 'shopping_cart'):?>
        <?php require_once "views/" . $_GET['view'] . ".php"?>
    <?php else:?>
        <?php if(!isset($_GET['id'])):?>
            <div class="container-fluid fluid-padding">
                <div class="row">
                    <h4 class="center-
align" id="productsHeaderUp"> Our products! </h4>

                    <div class="col s12 m12 l3" id="filter">
                        <?php require_once "views/filters.php"?>
                    </div>
                    <div class="col s12 m12 l9">
                        <?php require_once "views/products/allProducts.ph
p"?>
                    </div>
                </div>
                <script type="text/javascript" src="assets/js/getBrands.js"><
/script>

                <script type="text/javascript" src="assets/js/getAllProducts.
js"></script>
            <?php else:?>
                <?php
                    require_once "views/products/oneProduct.php";
                ?>

            <?php endif; ?>
        </div>
    <?php endif; ?>
    <?php require_once "views/fixed/footer.php"?>

    <!-- Compiled and minified JavaScript -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/material
ize.min.js"></script>
    <script type="text/javascript" src="assets/js/register.js"></script>
    <script type="text/javascript" src="assets/js/sign_in.js"></script>
    <script type="text/javascript" src="assets/js/shopping_cart.js"></script>
    <script src="assets/js/main.js"></script>
</body>
</html>

```