

Web programiranje PHP 2

DOKUMENTACIJA

Mihajlović Ognjena

40 / 17

FEBRUAR, 2020

SADRŽAJ

1. UVOD

- a) Korišćeni programski jezici
- b) Opis funkcionalnosti
- c) Korišćeni CSS

2. ORGANIZACIJA

- a) Organizaciona šema
- b) Slike stranica sa opisom funkcionalnosti

3. KODOVI

UVOD

1. Korišćeni programski jezici i tehnologije

Mark-up jezici: HTML, CSS

JavaScript, JQuery, Bootstrap4

Od okruženja je korišćen Atom.

2. Opis funkcionalnosti

REGISTRACIJA I LOGOVANJE



Na početnoj stranici sajta klikom na ikonicu je omogućena registracija korisnika i logovanje.

Prilikom registracije korisnika svim korisnicima je inicijalno dodeljena uloga korisnika (user).

Nakon registracije, korisnik biva redirektovan na početnu stranicu i biva mu omogućeno da se loguje na isti način. Greške se ispisuju posebno za svako polje u okviru ovih formi, i realizovane su kako na klijentskoj tako i na serverskoj strani.

PAGINACIJA

Na stranici admina (panel) se nalazi paginacija proizvoda (Products sekcija) u vidu tabele.

Unutar svake stranice se nalazi po 5 proizvoda.

Ispis je odrađen dinamički.

GENERISANJE SADRŽAJA

Pored ikonice za pristup korisničkom nalogu, nalazi se ikonica za pretragu.

Pretraga se obavlja na osnovu imena proizvoda.

Više informacija o svakom proizvodu se dobija klikom na njega. U sklopu informacija su ime proizvoda, cena, opis, kategorija kojoj pripada.

Samo registrovan korisnik ima mogućnost da doda proizvod u korpu.

Svaka dodata slika unutar proizvoda je resize-ovana na 728px širine i 947px visine kako bi se postigla ista veličina blokova na stranici gde se nalaze proizvodi. Ovo je rađeno putem koda. Takođe, u Admin panel-u svaka novododata slika se resize-uje, a prethodna briše.

Svaka greška koja se desi u catch bloku se upisuje u fajl errors.txt.

Svaki pristup stranici se ispisuje u fajl log.txt. Tu se nalazi username korisnika (ili Guest ako korisnik nije ulogovan), datum i vreme pristupa, IP adresa pristupa i stranica u okviru sajta kojoj se pristupilo.

Za proizvod koji je dostupan se prikazuje opcija dodavanja u korpu. To može zavisiti od njega samog, ili od kategorije kojoj pripada, što u prevodu znači da ako se disable-uje kategorija u okviru admin

panela, za sve proizvode koji pripadaju istoj se gubi opcija dodavanja u korpu. Ceo CRUD je izgenerisan putem AJAX-a.

3. Korišćeni CSS

CSS je minifikovan.

ORGANIZACIJA

Organizaciona šema

<https://obakeryshop.000webhostapp.com/>



home

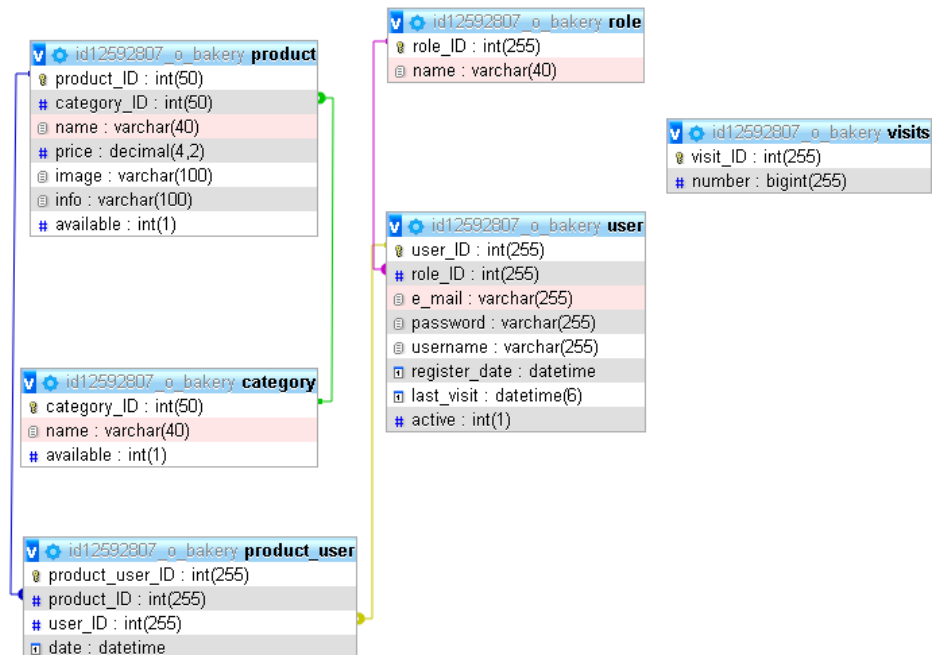


products



contact

Šema baze podataka



Slike stranica sa opisom funkcionalnosti

→ <https://obakeryshop.000webhostapp.com/products>

Dinamički ispis kategorija i proizvoda iz baze.

Klikom na proizvod, prikazuju se detaljnije informacije o njemu. Dodavanjem u korpu korisnik se

OUR PRODUCTS

All Products

Cupcakes

Pies

Donuts

Cookies



Bluebird



Birthday Blast



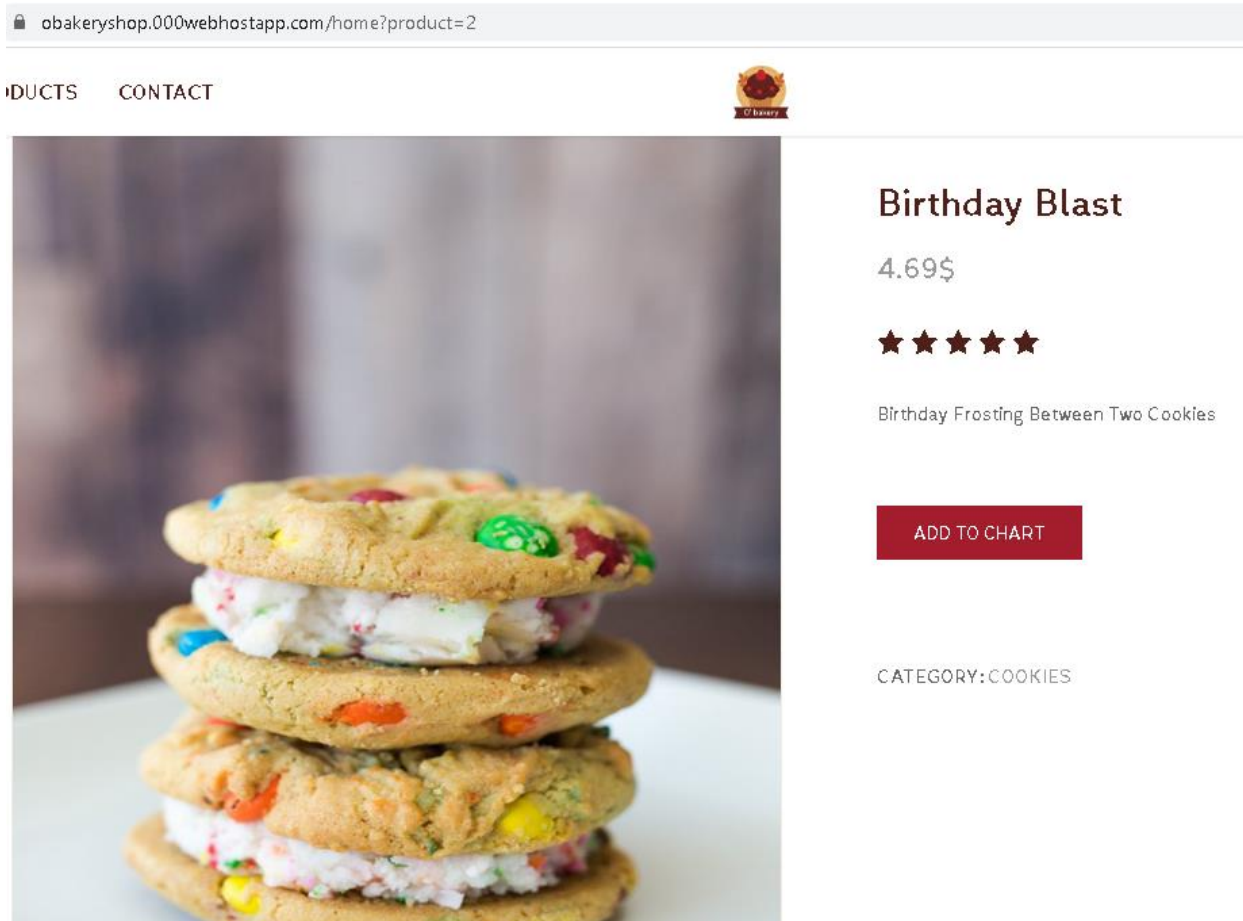
Cookie Dough Delight



White Out

preusmerava na stranicu gde je

prikazana njegova korpa sa
proizvodima.



Samo ulogovan korisnik može dodati
proizvod u korpu, i brisati ih iz korpe.




Home



Dashboard

Chart


Product name	Price	User	Date	
Cookie Dough Delight	4.69		2020-02-17 17:48:59	Remove

➔ <https://obakeryshop.000webhostapp.com/contact>

Unutar kontakt stranice sva polja su obavezna.

U slučaju da podaci nisu pravilno uneseni, ispod forme se ispisuje obaveštenje.

Ukoliko su podaci ispravno uneseni, korisniku se ispisuje obaveštenje o tome da je poruka uspešno poslata i u tom slučaju e-mail stiže administratoru na obakeryshop@gmail.com.

 Stay in loop with us

Name

E-mail

Message

Message has been sent!

→ [pretraga](#)

Na svakoj stranici se može izvršiti filtriranje proizvoda po nazivu klikom na ikonicu

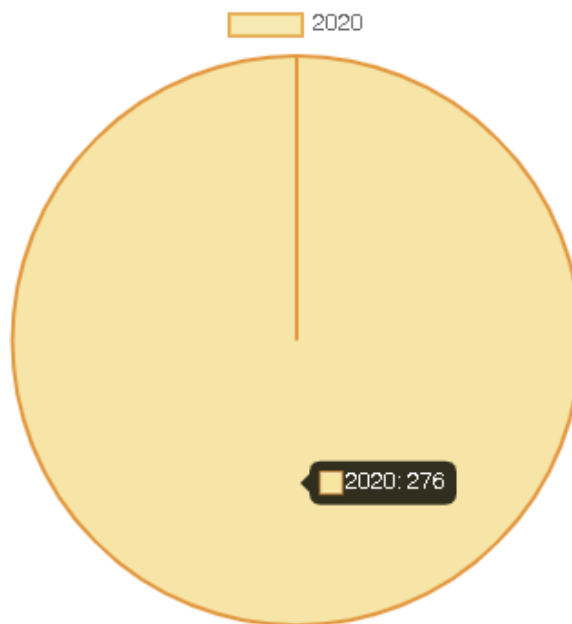


Ukoliko proizvod ne postoji prikazuju se svi proizvodi.

→ <https://obakeryshop.000webhostapp.com/admin>

Admin panelu **panel** može pristupiti samo admin klikom na .

Visits 2020



Visits predstavlja posete ka sajtu isključujući admin panel. Svake nove godine, posete se resetuju na 0.

Activities predstavlja korisnika koji je pristupio sajtu (tj. stranici kojoj pristupa, vreme pristupa i njegovu IP adresu.). Sa leve strane admin panela su prikazane kategorije, korisnici i proizvodi. Takođe se nalazi opcija dodavanja novih kategorija i novih proizvoda. Samo se za admina prikazuje slika, dok je za sve ostale korisnike univerzalna.



itsogii
admin



test
user



Activities


User	Date	IP address	URL address
itsogii	2020-02-16 21:03:24	77.243.22.118	obakeryshop.000webhostapp.com/admin
itsogii	2020-02-16 21:03:25	77.243.22.118	obakeryshop.000webhostapp.com/visits
itsogii	2020-02-16 21:03:33	77.243.22.118	obakeryshop.000webhostapp.com/admin
itsogii	2020-02-16 21:03:35	77.243.22.118	obakeryshop.000webhostapp.com/visits


KODOVI


STRUKTURA PROJEKTA


obakeryshop > public_html


Go Premium











































<input type="checkbox"/>	Name ▼	Size	Date
<input type="checkbox"/>	 app		2020-02-15 07:00:00
<input type="checkbox"/>	 assets		2020-02-15 07:01:00
<input type="checkbox"/>	 .htaccess	0.6 kB	2020-02-16 20:41:00
<input type="checkbox"/>	 index.php	21.5 kB	2020-02-17 00:05:00
<input type="checkbox"/>	 robots.txt	0.1 kB	2020-02-17 18:50:00
<input type="checkbox"/>	 sitemap.xml	4.3 kB	2020-02-17 18:03:00










obakeryshop > public_h... > app









Go Premium









     

	<input type="checkbox"/>	Name ▼	Size	Date
	<input type="checkbox"/>	 config		2020-02-15 07:00:00
	<input type="checkbox"/>	 controllers		2020-02-15 07:00:00
	<input type="checkbox"/>	 data		2020-02-15 07:00:00
	<input type="checkbox"/>	 models		2020-02-16 20:32:00
	<input type="checkbox"/>	 views		2020-02-16 17:15:00

obakeryshop > public_h... > app > views				Go Premium						
<input type="checkbox"/>	Name ▼	Size	Date							
<input type="checkbox"/>	 admin		2020-02-15 07:01:00							
<input type="checkbox"/>	 chart		2020-02-15 07:01:00							
<input type="checkbox"/>	 fixed		2020-02-15 07:00:00							
<input type="checkbox"/>	 pages		2020-02-15 07:00:00							
<input type="checkbox"/>	 403.php	0.3 kB	2020-02-16 17:42:00							
<input type="checkbox"/>	 404.php	0.3 kB	2020-02-16 17:42:00							

obakeryshop > public_h... > app > config				Go Premium						
<input type="checkbox"/>	Name ▼	Size	Date							
<input type="checkbox"/>	 database.php	0.2 kB	2020-02-15 07:07:00							
<input type="checkbox"/>	 DB.php	1.3 kB	2020-02-15 07:00:00							
<input type="checkbox"/>	 functions.php	1.9 kB	2020-02-15 07:00:00							

obakeryshop > public_h... > app > data				Go Premium						
<input type="checkbox"/>	Name ▼	Size	Date							
<input type="checkbox"/>	 errors.txt	0.2 kB	2020-02-15 07:00:00							
<input type="checkbox"/>	 log.txt	40.2 kB	2020-02-17 18:03:00							

	<input type="checkbox"/>	Name ▼	Size	Date
	<input type="checkbox"/>	 phpmailer		2020-02-15 07:01:00
	<input type="checkbox"/>	 Category.php	1.8 kB	2020-02-15 07:00:00
	<input type="checkbox"/>	 Chart.php	0.8 kB	2020-02-15 07:00:00
	<input type="checkbox"/>	 Login.php	1.2 kB	2020-02-15 07:00:00
	<input type="checkbox"/>	 Product.php	3.2 kB	2020-02-15 17:26:00
	<input type="checkbox"/>	 Register.php	0.9 kB	2020-02-15 07:00:00
	<input type="checkbox"/>	 User.php	0.3 kB	2020-02-15 07:00:00
	<input type="checkbox"/>	 Visit.php	0.4 kB	2020-02-15 07:00:00

Index.php

```
<?php
```

```
ob_start();
```

```
session_start();
```

```
require_once 'app/config/database.php';
```

```
require_once 'app/config/DB.php';
```

```
require_once 'app/config/functions.php';
```

```
require_once
```

```
'app/controllers/ProductController.php';
```

```
require_once
```

```
'app/controllers/CategoryController.php';
```

```
require_once 'app/controllers/ChartController.php';
```

```
require_once 'app/controllers/UserController.php';
```

```
require_once
```

```
'app/controllers/RegisterController.php';
```

```
require_once 'app/controllers/LoginController.php';
```

```
require_once 'app/controllers/VisitController.php';
```

```
// $root = $_SERVER['DOCUMENT_ROOT'];  
// echo $root;
```

```
$database = new DB(SERVER, DATABASE, USERNAME,  
PASS);
```

```
//
```

```
$productController = new ProductController();
```

```
$categoryController = new CategoryController();
```

```
$chartController = new ChartController();
```

```
$UserController = new UserController();
```

```
$registerController = new RegisterController();
```

```
$loginController = new LoginController();
```

```
$visitController = new VisitController();
```

```
$url = $_SERVER['REQUEST_URI'];
```

```
// var_dump($url);
```

```
//
```

```
$url =
```

```
str_replace('https://obakeryshop.000webhostapp.co  
m/', '', $url);
```

```
//  
// echo "</br>";  
//  
  
// var_dump($url);  
  
$guest = isset($_SESSION['username']) ?  
$_SESSION['username'] : 'Guest';  
logActivity($guest);  
  
  
if(isset($_SESSION['user'])) {  
    if($_SESSION['user_role'] == 1)  
    {  
  
        if($url == '/admin')  
        {
```



```
$data = @ fopen(__DIR__ . '/app/data/log.txt', 'r');
if($data)
{

    $file_data = file (__DIR__ . '/app/data/log.txt');
    fclose($data);

}
includeAdminFiles('/home', ['data' =>
$file_data]);exit;
}

else if($url == '/admin/categories')
{
    $categoryController -> displayAllCategories();

}
```

```
else if($url == '/admin/products')
{

    $productController ->
allProductsAndCategories();

}

else if(isset($_GET['page']))
{

    global $counter;
    $page = $_GET['page'];

    $limit = ($page - 1) * 5;

    $productController -> displayPagination($limit);
```

```
}
```

```
else if($url == '/admin/users')
```

```
{
```

```
    $UserController -> displayAllUsers();
```

```
}
```

```
else if($url == '/admin/add/category')
```

```
{
```

```
    includeAdminFiles('/add_category');
```

```
}
```

```
else if($url == '/admin/add/product')  
{  
    $categoryController ->  
displayCategoryNamesDropdown();  
  
}
```

```
else if(isset($_GET['product_id']))  
{  
  
    $prod_id = $_GET['product_id'];  
    $productController ->  
displayProductByIdAdminPanel($prod_id);  
  
}
```

```
else if(isset($_GET['category_id']))  
{
```

```
    $category_id = $_GET['category_id'];  
    $categoryController ->  
    displayCategoryById($category_id);  
  
    }  
  
    }  
  
    }  
  
    else {  
        if(($url == '/admin') || ($url ==  
'/admin/categories') || ($url == '/admin/products')  
        || ($url == '/admin/users') || ($url ==  
'/admin/add/product') || ($url ==  
'/admin/add/category')) {  
  
            http_response_code(400);
```

```
    header('Location:
https://obakeryshop.000webhostapp.com/app/view
s/404.php');exit();
```

```
    }
}
```

```
if(isset($_GET['product']))
{
    view('fixed/head');
    view('fixed/header');
    view('fixed/slider');
    $product_id = $_GET['product'];
    $productController -> displayProduct($product_id);
    view('fixed/footer');
}
```

```
switch($url) {  
  
    case "":  
    case '/index.php':  
    case '/home':  
    case '/':  
        view('fixed/head');  
        view('fixed/header');  
        view('fixed/slider');  
        $database -> visits(); // When goes to website  
        $categoryController ->  
        countAllProductsByCategory();  
  
        view('fixed/footer');  
        break;  
  
    case '/user_chart':
```

```
$user = $_SESSION['user'];
```

```
$chartController ->  
displayAllOrdersByUserId($_SESSION['user'] ->  
user_ID);break;
```

```
case '/products':  
view('fixed/head');  
view('fixed/header');  
view('fixed/slider');  
$productController ->  
displayAllProductsAndAllCategories();
```

```
view('fixed/footer');  
break;
```

```
case '/contact':
```



```
view('fixed/head');
view('fixed/header');
view('fixed/slider');
view('pages/contact');
view('fixed/footer');
break;
```

```
case '/product_search':
```

```
if(isset($_POST['_product_name'])){
    try {
        $product_name= '%' . $_POST['_product_name'] . '%';
        $productController ->
        displayProductTroughSearch(strtolower($product_name));
    }
}
```

```
catch(PDOException $e)
{
```

```
        catchErrors($e -> getMessage());
        http_response_code(500);
    }

}

else {
    http_response_code(400);
    header('Location:
https://obakeryshop.000webhostapp.com/app/view
s/404.php');exit();
}

break;

case '/chart':
if(isset($_POST['_product_id']))
{

    try {
        $product_id = $_POST['_product_id'];
```

```
$productController ->  
displayProductById($product_id);  
}
```

```
catch(PDOException $e)  
{  
    catchErrors($e -> getMessage());  
    http_response_code(500);  
}  
}  
else {  
    http_response_code(400);  
    header('Location:  
https://obakeryshop.000webhostapp.com/app/view  
s/404.php');exit();  
}  
break;
```

```
case '/remove_from_chart':  
if(isset($_POST['_product_id']))  
{  
  
    try {  
        $product_id = $_POST['_product_id'];  
        $user_id = $_SESSION['user'] -> user_ID;  
  
        $chartController -> deletedFromChart($product_id,  
$user_id);  
    }  
  
    catch(PDOException $e)  
    {  
        catchErrors($e -> getMessage());  
        http_response_code(500);  
    }  
}
```

```
}  
}  
else {  
    http_response_code(400);  
    header('Location:  
https://obakeryshop.000webhostapp.com/app/view  
s/404.php');exit();  
}
```

break;

case '/contact_form':

```
    if(isset($_POST['_name']) && isset($_POST['_email'])  
&& isset($_POST['_message']))  
    {
```

```
        $name = $_POST['_name'];
```

```
        $email = $_POST['_email'];
```

```
$message = $_POST['_message'];
```

```
$reName = '/^[a-zA-Z]{2,}(\s[a-zA-Z]{2,})*$/';
```

```
$reMessage = '/([a-zA-Z]){5,}(\s)*[0-9]*/';
```

```
$flag = 0;
```

```
$errorName = "";
```

```
$errorMessage = "";
```

```
$errorEmail = "";
```

```
if(!preg_match($reName, $name))
```

```
{
```

```
    $flag++;
```

```
    $errorName .= 'Incorrect name!';
```

```
}
```

```
if(!preg_match($reMessage, $message))
```

```
{
```

```
    $flag++;
```

```
    $errorMessage .= 'Incorrect message!';
```

```
}
```

```
if(!filter_var($email, FILTER_VALIDATE_EMAIL))
{
    $flag++;
    $errorEmail .= 'Incorrect email!';
}
```

```
if($flag > 0)
{
    echo json_encode(['errorName' => $errorName,
'errorEmail' => $errorEmail, 'errorMessage' =>
$errorMessage]);
}
```

```
else {
```

```
    require_once
__DIR__.'/app/models/phpmailer/contactMailer.php';
    $result = sendContactMail($email, $message,
$name);
```

```
    $result ? $response = 'Message has been sent!' :  
    $response = 'Message has not been sent!';  
    header('Content-type: application/json');  
    echo json_encode(['errorName' => $errorName,  
'errorEmail' => $errorEmail, 'errorMessage' =>  
$errorMessage, 'mailer' => $response]);exit;  
}
```

```
}
```

```
else {  
    http_response_code(400);  
    header('Location:  
https://obakeryshop.000webhostapp.com/app/view  
s/404.php');exit();  
}
```

```
break;
```



```
case '/edit_a_category':
```

```
if($_SERVER['REQUEST_METHOD'] === 'POST'){
```

```
    $flag = 0;
```

```
    $errorEditCategory = '';
```

```
    $reName = '/^[A-z]{2,}(\s([A-z]{2,}))*$/';
```

```
    $category_name = $_POST['_category_name'];
```

```
    $cat_id = $_POST['_category_id'];
```

```
    if(!preg_match($reName, $category_name))
```

```
    {
```

```
        $flag++;
```

```
        $errorEditCategory .= 'Inccorect name!';
```

```
    }
```

```
if($flag > 0)
{
    header('Content-type: application/json');
    echo json_encode(['error' =>
$errorEditCategory]);exit;
}

else {

    try {

        $categoryController ->
updateCategoryName($category_name, $cat_id);
        http_response_code(204);

    }
```

```
catch(PDOException $e)
{

    catchErrors($e -> getMessage());
    http_response_code(500);
}
}

}

else {
    http_response_code(400);
    header('Location:
https://obakeryshop.000webhostapp.com/app/view
s/404.php');exit();
}

break;

case '/edit_a_product':
```

```
if($_SERVER['REQUEST_METHOD'] === 'POST'){  
    $reName = '/^[A-z]{2,}((\s[A-z]{2,}))*$$/';  
    $reDescr = '/[A-z]{3,}\s*$/';  
    $rePrice = '/^\d?\d{1}\.\d{2}$/';  
  
    $product_id = $_POST['product_id'];  
  
    $product_details = $productController ->  
displayProductByIdEdit($product_id);  
  
    global $new_image_updated;  
    $product_name = $_POST['product_name'];  
  
    $product_price = $_POST['product_price'];  
    $product_descr = $_POST['product_descr'];  
  
    $image_allowed_extensions = ['image/jpg',  
'image/png', 'image/jpeg'];
```

```
$flag = 0;  
$errorEditProdName = "  
$errorEditProdDescr = "  
$errorEditProdPrice = "  
$errorEditProdCat = "  
$errorEditProdImg = "
```

```
$imageFromDatabase = $product_details ->  
image;
```

```
$new_image_updated = $imageFromDatabase;
```

```
if(isset($_FILES['image'])) // IMG
```

```
{  
    unlink( __DIR__ . '/assets/images/' .  
$product_details -> image);
```

```
$category_name = $product_details ->  
categoryName;
```

```
$image_type = $_FILES['image']['type'];
```

```
$img_name = $_FILES['image']['name'];
```

```
$img_size =  
getimagesize($_FILES['image']['tmp_name']); // width,  
height
```

```
$img_tmp = $_FILES['image']['tmp_name'];
```

```
list($old_w, $old_h) = getimagesize($img_tmp);
```

```
$width = 720;
```

```
$height = 947;
```

```
if($image_type == 'image/jpeg')
```

```
{
```

```
    $blank = imagecreatefromjpeg($img_tmp);
```

```
}
```

```
else if($image_type == 'image/png'){
```

```
$blank = imagecreatefrompng($img_tmp);  
}
```

```
else if($image_type == 'image/jpeg'){  
    $blank = imagecreatefromjpeg($img_tmp);  
}
```

```
$blank_img = imagecreatetruecolor($width,  
$height);  
imagecopyresampled($blank_img, $blank, 0, 0, 0,  
0, $width, $height, $old_w, $old_h);
```

```
$new_image = $blank_img;
```

```
$microtime_int = round(microtime(true) * 1000);  
$new_path_image = __DIR__ . '/assets/images/' .  
$category_name . '/' . $microtime_int . $img_name;
```

```
$new_image_updated_01 = $new_path_image;
```

```
if($image_type == 'image/jpg')  
{  
    imagejpeg($new_image,  
$new_image_updated_01, 100);  
}
```

```
else if($image_type == 'image/png'){  
    imagepng($new_image,  
$new_image_updated_01, 100);  
}
```

```
else if($image_type == 'image/jpeg'){  
    imagejpeg($new_image,  
$new_image_updated_01, 100);  
}
```

```
$new_image_updated =  
$new_image_updated_01;
```



```
$explode_image = explode('/',  
$new_path_image);  
$new_image_updated = $explode_image[8] . '/' .  
$explode_image[9];  
// header('Content-type: application/json');  
// echo json_encode($explode_image);exit;
```

```
}
```

```
try {  
    $productController ->  
updateProduct($product_name, $product_price,  
$new_image_updated, $product_descr,  
$product_id);  
    http_response_code(204);  
}
```

```
catch(PDOException $e)
```

```
{

    catchErrors($e -> getMessage());
    http_response_code(500);
}

}

else {
    http_response_code(400);
    header('Location:
https://obakeryshop.000webhostapp.com/app/view
s/404.php');exit();
}

break;

case '/get_back_a_product':

    if(isset($_POST['_product_id'])) {
```

```
$product_id = $_POST['_product_id'];

try {
    $productController ->
displayBackProduct($product_id);
}

catch(PDOException $e)
{

    catchErrors($e -> getMessage());
    http_response_code(500);
}
}

else {
    http_response_code(400);
    header('Location:
https://obakeryshop.000webhostapp.com/app/view
s/404.php');exit();
```

```
}
```

```
break;
```

```
case '/remove_a_product':
```

```
if(isset($_POST['_product_id'])){\
```

```
    $product_id = $_POST['_product_id'];
```

```
    try {\
```

```
        $productController ->  
removeProduct($product_id);
```

```
    }
```

```
    catch(PDOException $e)
```

```
{
```

```
        catchErrors($e -> getMessage());
```

```
        http_response_code(500);
```

```
}
```

```
}
```

```
else {
```

```
    http_response_code(400);
```

```
    header('Location:
https://obakeryshop.000webhostapp.com/app/view
s/404.php');exit();
```

```
}
```

```
break;
```

```
case '/remove_a_category':
```

```
if(isset($_POST['_category_id'])){
```

```
    $category_id = $_POST['_category_id'];
```

```
    try {
```

```
        $categoryController ->
removeCategory($category_id);
```

```
    }
```

```
catch(PDOException $e)
{

    catchErrors($e -> getMessage());
    http_response_code(500);
}
}

else {
    http_response_code(400);
    header('Location:
https://obakeryshop.000webhostapp.com/app/view
s/404.php');exit();
}

break;

case '/get_back_a_category':
```

```
if(isset($_POST['_category_id'])){  
    $category_id = $_POST['_category_id'];  
  
    try {  
        $categoryController ->  
displayBackCategory($category_id);  
    }  
  
    catch(PDOException $e)  
    {  
  
        catchErrors($e -> getMessage());  
        http_response_code(500);  
    }  
}  
  
else {  
    http_response_code(400);  
}
```

```
    header('Location:
https://obakeryshop.000webhostapp.com/app/view
s/404.php');exit();
}
```

```
break;
```

```
case '/register':
```

```
    if(isset($_POST['_username']) &&
isset($_POST['_email']) && isset($_POST['_pass']))
    {
```

```
        $username = $_POST['_username'];
```

```
        $email = $_POST['_email'];
```

```
        $pass = md5($_POST['_pass']);
```

```
        $errorUsername = "";
```

```
        $errorPassword = "";
```



```
$errorMessage = "";
```

```
$flag = 0;
```

```
$inserted = 0;
```

```
// if he already exists
```

```
$userExists = $registerController ->  
displayUserByUsername($username);
```

```
$emailExists = $registerController ->  
displayUserByEmail($email);
```

```
if($userExists)
```

```
{
```

```
    $errorMessage .= 'Username already exists!';
```

```
    $flag++;
```

```
}
```

```
if($emailExists)
```

```
{
```

```
$errorMessage .= 'E-mail already exists!';  
$flag++;  
}
```

```
if($flag > 0)  
{  
    header('Content-type: application/json');  
    echo json_encode(['errorUsername' =>  
$errorMessage, 'errorMessage' => $errorMessage]);exit;  
}
```

```
else if($flag == 0) {  
    $regUsername = "/^[a-  
z]{3,8}((\_{0,2})(\d{0,3}))$/";  
    $regPassword = "/([w\W\D\d]){7,}/";  
    $regEmail = "/^[A-z\d]{2,}(\.?(W\D)?[A-  
z\d]{2,})*\@\w{2,}(\.\w{2,})*$/";
```

```
if(!preg_match($regUsername, $username))  
{
```

```
$flag++;  
$errorUsername .= 'Not valid username!';  
}
```

```
if(!preg_match($regPassword, $pass))  
{  
    $flag++;  
    $errorPassword .= 'At least 7 characters!';  
}
```

```
if(!preg_match($regEmail, $email))  
{  
    $flag++;  
    $errorEmail .= 'Not valid e-mail!';  
}
```

```
if($flag > 0)  
{  
    header('Content-type: application/json');
```

```
        echo json_encode(['errorUsername' =>
$errorUsername, 'errorPassword' => $errorPassword,
'errorEmail' => $errorEmail]);exit;
    }
```

```
else if($flag == 0)
```

```
{
    $date = date('Y-m-d H:i:s');
    $last_visit = 0;
    $active = 0;
```

```
    try {
```

```
        $registerController -> insertUser($email, $pass,
$username, $date, $last_visit, $active);
```

```
        $inserted = 1;
```

```
        echo json_encode(['registered' =>
$inserted]);exit;
```

```
    }
```

```
    catch(PDOException $e)
```

```
{
```

```
        catchErrors($e -> getMessage());  
        http_response_code(500);  
    }
```

```
}
```

```
}
```

```
}
```

```
else {  
    http_response_code(400);  
    header('Location:  
https://obakeryshop.000webhostapp.com/app/view  
s/404.php');exit();  
}
```

```
break;
```

```
case '/login':
```

```
if(isset($_POST['_email']) && isset($_POST['_pass']))  
{
```

```
    $email = $_POST['_email'];
```

```
    $pass = md5($_POST['_pass']);
```

```
    $errorPassword = "";
```

```
    $errorEmail = "";
```

```
    $flag = 0;
```

```
    $regPassword = "/([\w\W\D\d]){7,}/";
```

```
    $regEmail = "/^[A-z\d]{2,}(\.?(\\W\\D)?[A-  
z\d]{2,})*\\@\\w{2,}(\\.\\w{2,})*$/";
```

```
    if(!preg_match($regEmail, $email))
```

```
{  
    $errorMessage .= 'Not valid e-mail!';  
    $flag++;  
}
```

```
if(!preg_match($regPassword, $pass))  
{  
    $errorPassword .= 'At least 7 characters!';  
    $flag++;  
}
```

```
if($flag > 0)  
{  
    header('Content-type: application/json');  
    echo json_encode(['errorPassword' =>  
$errorPassword, 'errorMessage' => $errorMessage]);exit;  
}  
else if($flag == 0)  
{
```

```
$emailExists = $loginController ->  
displayUserByEmail($email);
```

```
if(!$emailExists)  
{  
    $errorMessage .= 'E-mail doesn\'t exist!';  
    header('Content-type: application/json');  
    echo json_encode(['errorMessage' =>  
$errorMessage]);exit;  
}  
else if($emailExists)  
{  
    if($pass != $emailExists -> password)  
    {  
        $errorMessage .= 'Password doesn\'t match!';  
        header('Content-type: application/json');  
        echo json_encode(['errorMessage' =>  
$errorMessage]);exit;  
    }  
}
```



```
else if($pass == $emailExists -> password){
```

```
    $result = $loginController ->
displayUserByEmailAndPassword($email, $pass);
    if($result)
    {
        $user = $result;
        $_SESSION['user'] = $user;
        $_SESSION['user_ID'] = $user -> user_ID;
        $_SESSION['username'] = $user -> username;
        $_SESSION['user_role'] = $user -> role_ID;
```

```
$last_visit = date("Y-m-d H:i:s");
```

```
$loginController ->
displayLastVisitByUser($last_visit, $_SESSION['user_ID']);
```

```
$loginController ->  
displayOnlineActiveStatus($_SESSION['user_ID']);
```

```
if($_SESSION['user_role'] == 1)  
{  
    echo json_encode(['role' => 1]);exit;  
}  
else if($_SESSION['user_role'] == 2)  
{  
    echo json_encode(['role' => 2]);exit;  
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
else {
```

```
    http_response_code(400);
```

```
    header('Location:
https://obakeryshop.000webhostapp.com/app/views/404.php');exit();
```

```
}
```

```
break;
```

```
case '/logout':
```

```
if($_SERVER['REQUEST_METHOD'] === 'POST'){
```

```
if(isset($_SESSION['user']))
```

```
{
```

```
    $current_user_ID = $_SESSION['user_ID'];
```

```
try {
```

```
        $loginController ->
displayOfflineActiveStatus($current_user_ID);
    }

    catch(PDOException $e)
    {

        catchErrors($e -> getMessage());
        http_response_code(500);
    }

}
}

else {
    http_response_code(400);
    header('Location:
https://obakeryshop.000webhostapp.com/app/view
s/404.php');exit();
}
```

```
break;
```

```
case '/products_by_category':
```

```
if(isset($_POST['_category_ID']))
```

```
{
```

```
    $category = $_POST['_category_ID'];
```

```
    if($category == 0)
```

```
{
```

```
    $productController -> displayAllProducts();
```

```
}
```

```
else {
```

```
    $productController ->
```

```
displayAllProductsByCategory($category);
```

```
}
```

```
}
```

```
else {  
    http_response_code(400);  
    header('Location:  
https://obakeryshop.000webhostapp.com/app/view  
s/404.php');exit();  
}  
  
break;  
  
case '/add_a_category':  
  
if ($_SERVER['REQUEST_METHOD'] === 'POST') {  
    if(isset($_POST['_category_name']))  
    {  
        $category_value = $_POST['_category_name'];  
  
        $flag = 0;  
        $errorAddCategory = "";
```

```
$reCategoryName = '/^[A-z]{2,}(\s([A-z]{2,}))*$/';
```

```
if(!preg_match($reCategoryName,  
$category_value))
```

```
{  
    $flag++;  
    $errorAddCategory .= 'Incorrect value!';  
}  
if($flag > 0)  
{  
    header('Content-type: application/json');  
    echo json_encode(['errAddCategory' =>  
$errorAddCategory]);exit;  
}  
else {  
    try {  
        $categoryController ->  
newCategory($category_value);
```

```
    $inserted = $categoryController ->  
newCategory($category_value);
```

```
    header('Content-type: application/json');  
    echo json_encode(['errAddCategory' =>  
$errorAddCategory]);exit;
```

```
}
```

```
catch(PDOException $e)  
{
```

```
    catchErrors($e -> getMessage());  
    http_response_code(500);  
}
```

```
}
```



```
}  
}  
else {  
    http_response_code(400);  
    header('Location:  
https://obakeryshop.000webhostapp.com/app/view  
s/404.php');exit();  
}
```

```
break;
```

```
case '/add_a_product':
```

```
if($_SERVER['REQUEST_METHOD'] === 'POST'){
```

```
    $reName = '/^[A-z]{2,}((\s[A-z]{2,}))*$/';
```

```
    $reDescr = '/[A-z]{3,}\s*/';
```

```
    $rePrice = '/^\d?\d{1}\.\d{2}$/';
```

```
    $product_name = $_POST['product_name'];
```

```
$product_price = $_POST['product_price'];
$product_descr = $_POST['product_descr'];

$image_allowed_extensions = ['image/jpg',
'image/png', 'image/jpeg'];
$flag = 0;
$errorAddProdName = "";
$errorAddProdDescr = "";
$errorAddProdPrice = "";
$errorAddProdCat = "";
$errorAddProdImg = "";

if(isset($_POST['product_category']))
{
    $category = $_POST['product_category']; //
CATEGORY

}

$category_id = $category;
```

```
$category_name = $categoryController ->  
displayCategoryNameById($category_id);
```

```
$dir_name = __DIR__ . '/assets/images/' .  
$category_name -> name . '/';
```

```
if(!file_exists($dir_name))  
{  
    mkdir($dir_name);  
}  
if(isset($_FILES['image'])) // IMG
```

```
{  
    $image_type = $_FILES['image']['type'];
```

```
// resize
```

```
$img_name = $_FILES['image']['name'];
```

```
$img_size =  
getimagesize($_FILES['image']['tmp_name']); // width,  
height
```

```
$img_tmp = $_FILES['image']['tmp_name'];
```

```
list($old_w, $old_h) = getimagesize($img_tmp);
```

```
$width = 720;
```

```
$height = 947;
```

```
if($image_type == 'image/jpeg')
```

```
{
```

```
    $blank = imagecreatefromjpeg($img_tmp);
```

```
}
```

```
else if($image_type == 'image/png'){
```

```
    $blank = imagecreatefrompng($img_tmp);
```

```
}
```

```
else if($image_type == 'image/jpeg'){
```

```
    $blank = imagecreatefromjpeg($img_tmp);
```

```
}
```

```
    $blank_img = imagecreatetruecolor($width,  
$height);
```

```
    imagecopyresampled($blank_img, $blank, 0, 0, 0,  
0, $width, $height, $old_w, $old_h);
```

```
$new_image = $blank_img;
```

```
$microtime_int = round(microtime(true) * 1000);
```

```
$new_path_image = $dir_name . $microtime_int .  
$img_name;
```

```
if($image_type == 'image/jpg')  
{  
    imagejpeg($new_image, $new_path_image,  
100);  
}
```

```
else if($image_type == 'image/png'){
```

```
    imagepng($new_image, $new_path_image,  
100);  
    }
```

```
    else if($image_type == 'image/jpeg'){  
        imagejpeg($new_image, $new_path_image,  
100);  
    }
```

```
    $explode_image = explode('/',  
$new_path_image);
```

```
    $db_img_path = $explode_image[8] . '/' .  
$explode_image[9];
```

```
try {
```

```
$productController -> newProduct($category_id,  
$product_name, $product_price, $db_img_path,  
$product_descr, 1);
```

```
}
```

```
catch(PDOException $e)
```

```
{
```

```
    catchErrors($e -> getMessage());
```

```
    http_response_code(500);
```

```
}
```

```
}
```

```
}
```

```
else {
```

```
    http_response_code(400);
```

```
    header('Location:
https://obakeryshop.000webhostapp.com/app/view
s/404.php');exit();
```

```
}
```

```
break;
```

```
case '/visits':
```

```
    if($_SERVER['REQUEST_METHOD'] === 'POST') {
```

```
        try {
```

```
            $visitController -> displayAllVisits();
```

```
        }
```

```
        catch(PDOException $e)
```

```
{
```



```
catchErrors($e -> getMessage());  
http_response_code(500);  
}
```

```
}
```

```
else {  
    http_response_code(400);  
    header('Location:  
https://obakeryshop.000webhostapp.com/app/view  
s/404.php');exit();  
}
```

```
break;
```

```
}
```

app/config/DB.php

```
<?php
```

```
class DB {
```

```
    private $database;
```

```
    private $server;
```

```
    private $username;
```

```
    private $password;
```

```
    public $conn;
```

```
    public function __construct($server, $db,  
    $username, $pass)
```

```
{
```

```
    $this->server = $server;
```

```
    $this->database = $db;
```

```
    $this->username = $username;
```

```
    $this->password = $pass;
```

```
$this->conn = new PDO("mysql:host={$this->server};dbname={$this->database};charset=utf8",  
$this->username, $this->password);
```

```
$this->conn->  
setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE,  
PDO::FETCH_OBJ);
```

```
$this->conn->setAttribute(PDO::ATTR_ERRMODE,  
PDO::ERRMODE_EXCEPTION);
```

```
}
```

```
public function getMyDB()
```

```
{
```

```
if($this->conn instanceof PDO)
```

```
{
```

```
    return $this->conn;
```

```
}
```

```
}
```

```
public function executeQuery($query)
```

```
{  
    return $this -> conn -> query($query) -> fetchAll();  
}
```

```
public function executeOneRow($query)  
{  
    return $this -> conn -> query($query) -> fetch();  
}
```

```
// public function userByUsername($username)  
// {  
//     $query = $this -> conn -> prepare("SELECT * from  
// user where username = ?");  
//     $query -> execute([$username]);  
//     return $query -> fetch();  
// }
```

```
public function visits()  
{
```

```
$query = $this -> conn -> prepare("UPDATE visits SET  
number = number + 1");
```

```
$query -> execute();
```

```
}
```

```
}
```

app/config/functions.php

```
<?php
```

```
function view($filename, $parametersX = [],  
$parametersY = [])  
{  
    extract($parametersX);  
    extract($parametersY);  
    require_once 'app/views/' . $filename . '.php';
```

```
}
```

```
function includeAdminFiles($filename, $params = [],  
$countProducts = [])  
{  
    extract($countProducts);  
    extract($params);  
    require_once 'app/views/admin/fixed/head.php';
```

```
require_once 'app/views/admin/fixed/nav.php';
require_once 'app/views/admin/pages/' . $filename
.'.php';
require_once 'app/views/admin/fixed/footer.php';
}
```

```
function includeChart($filename, $username,
$user_role, $parameterX = [])
{
    extract($parameterX);
    require_once 'app/views/admin/fixed/head.php';
    require_once 'app/views/chart/fixed/nav.php';
    require_once 'app/views/chart/pages/' . $filename .
    '.php';
    require_once 'app/views/admin/fixed/footer.php';
}
```

```
function catchErrors($error)
{
    $file = @ fopen(__DIR__ . '/../data/errors.txt', 'a');
```



```
if($file)
{
    $date = date('Y-m-d H:i:s');
    $data = $error . "\t" . $date . "\t\n";
    fwrite($file, $data);
    fclose($file);
}
}
```

```
function logActivity($guest)
{
    $file = @ fopen(__DIR__ . '/../data/log.txt', 'a');
```

```
if(isset($_SERVER['HTTP_CLIENT_IP'])){
    $ip_address = $_SERVER['HTTP_CLIENT_IP'];
}
```

```
else if(isset($_SERVER['HTTP_X_FORWARDED_FOR']))
```

```
{  
    $ip_address =  
$_SERVER['HTTP_X_FORWARDED_FOR'];  
}
```

```
else if(isset($_SERVER['HTTP_FORWARDED']))  
{  
    $ip_address = $_SERVER['HTTP_FORWARDED'];  
}
```

```
else if(isset($_SERVER['HTTP_X_CLUSTER_CLIENT_IP']))  
{  
    $ip_address =  
$_SERVER['HTTP_X_CLUSTER_CLIENT_IP'];  
}
```

```
else {  
    $ip_address = $_SERVER['REMOTE_ADDR'];  
}
```

```
if($file)
{
    $date = date("Y-m-d H:i:s");
    $data = $guest . "\t" . $date . "\t" . $ip_address . "\t"
. $_SERVER['HTTP_HOST'] . $_SERVER['REQUEST_URI'] .
"\n";
    fwrite($file, $data);
    fclose($file);
}
}
```

app/controllers/CategoryController.php

```
<?php
```

```
require_once __DIR__.'../models/Category.php';
```

```
use app\models\Category;
```

```
class CategoryController {
```

```
    public function countAllProductsByCategory()  
{
```

```
        global $database;
```

```
        $categoryModel = new Category($database);
```

```
        $countAllProductsByCategory = $categoryModel -  
> countAllProductsByCategory();
```

```
        view('pages/home', ['productsByCategory' =>  
$countAllProductsByCategory]);
```

```
    }
```

```
public function displayAllCategories()  
{  
    global $database;  
  
    $categoryModel = new Category($database);  
    $allCategories = $categoryModel ->  
countAllProductsByCategory();  
  
    includeAdminFiles('/categories', ['allCategories' =>  
$allCategories]);  
  
}  
  
public function displayCategoryNamesDropdown()  
{  
    global $database;  
    $categoryModel = new Category($database);
```

```
$allCategories = $categoryModel ->  
getAllCategories();
```

```
includeAdminFiles('/add_product', ['allCategories'  
=> $allCategories]);
```

```
}
```

```
public function newCategory($val)
```

```
{
```

```
    global $database;
```

```
    $categoryModel = new Category($database);
```

```
    $categoryModel -> addCategory($val);
```

```
}
```

```
public function displayCategoryNameById($id)
```

```
{
```

```
global $database;
$categoryModel = new Category($database);
$cat_name = $categoryModel ->
getCategoryName($id);
return $cat_name;

}
```

```
public function displayCategoryById($id)
{
    global $database;
    $categoryModel = new Category($database);
    $category = $categoryModel -> getCategory($id);

    includeAdminFiles('/edit_category', ['category' =>
$category]);

}
```

```
public function updateCategoryName($value, $id)
{
    global $database;
    $categoryModel = new Category($database);
    $categoryModel -> editCategoryName($value,
    $id);

}
```

```
public function removeCategory($id)
{
    global $database;
    $categoryModel = new Category($database);
    $categoryModel -> unavailableCategory($id);

    header('Content-type: application/json');
    echo json_encode(['removed' => true]);
}
```



```
public function displayBackCategory($id)
{
    global $database;
    $categoryModel = new Category($database);
    $categoryModel -> availableCategory($id);

    header('Content-type: application/json');
    echo json_encode(['backed' => true]);
}

}
```

app/controllers/ChartController.php

```
<?php
```

```
require_once __DIR__ . '/../models/Chart.php';
```

```
use app\models\Chart;
```

```
class ChartController
```

```
{
```

```
    public function displayAllOrdersByUserId($user)
```

```
    {
```

```
        global $database;
```

```
        $chartModel = new Chart($database);
```

```
        $chart = $chartModel ->  
            getAllOrdersByUserId($user);
```

```
$username = $_SESSION['user'] -> username;
$user_role = $_SESSION['user'] -> role_ID;
includeChart('chart', $username, $user_role,
['chart' => $chart]);
}

public function deletedFromChart($id, $user_id)
{
    global $database;
    $chartModel = new Chart($database);

    $chartModel -> deleteProductFromChart($id);

    $chart = $chartModel ->
getAllOrdersByUserId($user_id);

    header('Content-type: application/json');
    echo json_encode(['chart' => $chart]);exit;
```

```
}
```

```
public function displayAllOrdersByUserIdAjax($user)
```

```
{
```

```
    global $database;
```

```
    $chartModel = new Chart($database);
```

```
    $chart = $chartModel ->
```

```
    getAllOrdersByUserId($user);
```

```
    return $chart;
```

```
}
```

```
}
```

app/controllers/LoginController.php

```
<?php
// require_once 'app/models/Login.php';
require_once 'app/models/Login.php';

use app\models\Login;

class LoginController
{

    public function displayUserByEmail($email)
    {
        global $database;
        $loginModel = new Login($database);
        $emailExists = $loginModel ->
        getUserByEmail($email);

        return $emailExists;
    }
}
```

```
public function
displayUserByEmailAndPassword($email, $pass)
{
    global $database;
    $loginModel = new Login($database);
    $user = $loginModel ->
getUserByEmailAndPassword($email, $pass);
    return $user;
}
```

```
public function displayLastVisitByUser($date,
$user_ID)
{
    global $database;
    $loginModel = new Login($database);
    $loginModel -> updateLastVisitByUser($date,
$user_ID);
}
```

```
public function displayOnlineActiveStatus($user_id)
```

```
{  
    global $database;  
    $loginModel = new Login($database);  
    $loginModel ->  
updateActiveStatusToOnline($user_id);  
}
```

```
public function displayOfflineActiveStatus($user_id)  
{  
    global $database;  
    $loginModel = new Login($database);  
    $loginModel ->  
updateActiveStatusToOffline($user_id);
```

```
unset($_SESSION['user']);  
session_destroy();  
header('Content-type: application/json');  
echo json_encode(['logout' => true]);exit;
```

}

}

app/controllers/ProductController.php

```
<?php
```

```
require_once __DIR__.'../models/Product.php';
```

```
require_once __DIR__.'../models/Category.php';
```

```
use app\models\Category;
```

```
use app\models\Product;
```

```
class ProductController {
```

```
    public function  
    displayAllProductsAndAllCategories()
```

```
{
```

```
    global $database;
```

```
    $productModel = new Product($database);
```

```
    $categoryModel = new Category($database);
```

```
    $products = $productModel -> getAllProducts();
```

```
$categories = $categoryModel ->  
getAllCategories();
```

```
view('pages/products', ['allProducts' => $products],  
['allCategories' => $categories]);  
}
```

```
public function  
displayAllProductsByCategory($category)  
{  
    global $database;  
    $productModel= new Product($database);
```

```
$products = $productModel ->  
getProductsByCategory($category);
```

```
header('Content-type: application/json');  
echo json_encode(['products' => $products]);exit;
```

```
// return $products;
```

```
}
```

```
public function displayProductTroughSearch($val)
{
    global $database;
    $productModel = new Product($database);
    $product = $productModel ->
getProductTroughName($val);
    header('Content-type: application/json');

    if($product)
    {
        $product_id = $product -> product_ID;

        $product_final = $productModel ->
getProduct($product_id);
```

```
    echo json_encode(['product' =>
$product_final]);exit;
}
```

```
else {
```

```
    echo json_encode(['product' => false]);exit;
}
```

```
// return $product;
}
```

```
public function displayProductById($id)
```

```
{  
    global $database;  
    $productModel = new Product($database);  
    // $product = $productModel -> getProduct($id);  
  
    $user = $_SESSION['user'] -> user_ID;  
    $date = date("Y-m-d H:i:s");  
  
    $inserted = $productModel -> addToChart($id,  
    $user, $date);  
  
    header('Content-type: application/json');  
    echo json_encode(['inserted' => true]);exit;  
  
}
```

```
// public function displayToChart($product, $user,
$date)
// {
//   global $database;
//   $productModel = new Product($database);
//   $chart = $productModel ->
addToChart($product, $user, $date);
//
//
// }
```

```
public function displayProductByIdAdminPanel($id)
{
    global $database;
    $productModel = new Product($database);
    $product = $productModel -> getProduct($id);

    // return $product;
```

```
includeAdminFiles('/edit_product', ['product' =>
$product]);
```

```
}
```

```
public function displayProductByIdEdit($id)
```

```
{
```

```
    global $database;
```

```
    $productModel = new Product($database);
```

```
    $product = $productModel -> getProduct($id);
```

```
    return $product;
```

```
}
```

```
public function displayAllProducts()
{
    global $database;
    $productModel = new Product($database);
    $products = $productModel -> getAllProducts();

    header('Content-type: application/json');
    echo json_encode(['products' => $products]);exit;

    // return $products;

}

public function displayProduct($id)
{
    global $database;
```



```
$productModel = new Product($database);  
$result = $productModel -> getProduct($id);  
view('pages/details', ['product' => $result]);  
  
}
```

```
public function newProduct($cat, $name, $price,  
$img, $info)  
{  
    global $database;  
    $productModel = new Product($database);  
    $productModel -> addNewProduct($cat, $name,  
$price, $img, $info);  
}
```

```
public function removeProduct($id)  
{  
    global $database;  
    $productModel = new Product($database);  
    $productModel -> unavailableProduct($id);  
}
```

```
header('Content-type: application/json');  
echo json_encode(['removed' => true]);  
}
```

```
public function displayBackProduct($id)  
{  
    global $database;  
    $productModel = new Product($database);  
    $productModel -> availableProduct($id);
```

```
header('Content-type: application/json');  
echo json_encode(['backed' => true]);  
}
```

```
public function allProductsAndCategories()  
{  
    global $database;
```

```
$productModel = new Product($database);
```

```
$products = $productModel ->  
paginationLimitProducts();
```

```
$number = $productModel -> countAllProducts();
```

```
includeAdminFiles('/products', ['products' =>  
$products], ['numberOfProducts' => $number]);
```

```
}
```

```
public function displayPagination($limit)
```

```
{
```

```
    global $database;
```

```
    $productModel = new Product($database);
```

```
    $products = $productModel ->  
getPagination($limit);
```

```
    $number = $productModel -> countAllProducts();
```

```
    includeAdminFiles('/products', ['products' =>
$products], ['numberOfProducts' => $number]);
}
```

```
// public function totalProducts()
// {
//   global $database;
//   $productModel= new Product($database);
//   $number = $productModel ->
countAllProducts();
//   return $number;
//
//
//
//
// }
```

```
public function updateProduct($product_name,
$product_price, $new_image_updated, $info,
$prod_id)
{
```

```
global $database;

$productModel = new Product($database);

$productModel -> editProduct($product_name,
$product_price, $new_image_updated, $info,
$prod_id);

}

}
```

app/controllers/RegisterController.php

```
<?php
```

```
require_once 'app/models/Register.php';
```

```
use app\models\Register;
```

```
class RegisterController
```

```
{
```

```
    public function displayUserByUsername($username)
```

```
{
```

```
    global $database;
```

```
    $registerModel = new Register($database);
```

```
    $userExists = $registerModel ->  
    getUserByUsername($username);
```

```
    return $userExists;
```

```
}
```

```
public function displayUserByEmail($email)
{
    global $database;
    $registerModel = new Register($database);
    $emailExists = $registerModel ->
getUserByEmail($email);
    return $emailExists;
}
```

```
public function insertUser($email, $pass, $username,
$register_date, $last_visit, $active)
{
    global $database;
    $registerModel = new Register($database);
    $registerModel -> insertUserModel($email, $pass,
$username, $register_date, $last_visit, $active);
}
```

}

app/controllers/UserController.php

```
<?php
```

```
// require_once '../User.php';
```

```
require_once 'app/models/User.php';
```

```
use app\models\User;
```

```
class UserController {
```

```
    public function displayAllUsers()
```

```
    {
```

```
        global $database;
```

```
        $userModel = new User($database);
```

```
        $users = $userModel -> getAllUsers();
```

```
includeAdminFiles('/users', ['users' => $users]);
```

```
}
```

```
}
```

app/controllers/VisitController.php

```
<?php
```

```
require_once 'app/models/Visit.php';
```

```
use app\models\Visit;
```

```
class VisitController {
```

```
    public function displayAllVisits()
```

```
    {
```

```
        global $database;
```

```
        $visitModel = new Visit($database);
```

```
        $visits = $visitModel -> countAllVisits();
```

```
        $date = date("m/d");
```

```
$january = "01/01";
```

```
if($date == $january)
```

```
{
```

```
    $visitModel -> restartVisits();
```

```
}
```

```
header("Content-type: application/json");
```

```
echo json_encode(['visits' => $visits]);exit;
```

```
}
```

```
// public function newYearVisits()
```

```
// {
```

```
//     global $database;
```

```
//     $visitModel = new Visit($database);
```

```
//     $updateVisits = $visitModel -> restartVisits();
```

```
// }
```

```
}
```

app/models/Category.php

```
<?php
```

```
namespace app\models;
```

```
class Category {
```

```
    private $db;
```

```
    public function __construct($database)
```

```
{
```

```
        $this->db = $database;
```

```
}
```

```
    public function getAllCategories()
```

```
{
```

```
        return $this->db->executeQuery('SELECT * FROM  
category');
```

```
}
```

```
public function countAllProductsByCategory()
{
    return $this->db->executeQuery('SELECT c.name
as categoryName, c.available, c.category_ID as
category_ID, COUNT(p.category_ID) as
productsNumber FROM product p RIGHT JOIN
category c ON p.category_ID = c.category_ID
GROUP BY c.name ORDER BY c.category_ID ASC');
}
```

```
public function addCategory($value)
{
    $query = $this->db->getMyDB()->
prepare('INSERT INTO category VALUES(NULL, ?, 1)');
    $query->execute([$value]);
}
```

```
public function editCategoryName($value, $id)
{
```

```
$query = $this->db->getMyDB()->
prepare('UPDATE category SET name = ? WHERE
category_ID = ?');
```

```
$query->execute([$value, $id]);
```

```
}
```

```
public function getCategoryName($id)
```

```
{
```

```
$query = $this->db->getMyDB()->
prepare('SELECT name FROM category WHERE
category_ID = ?');
```

```
$query->execute([$id]);
```

```
return $query->fetch();
```

```
}
```

```
public function getCategory($id)
```

```
{
```

```
$query = $this->db->getMyDB()->
prepare('SELECT * FROM category WHERE
category_ID = ?');
```

```
$query -> execute([$id]);  
return $query -> fetch();  
}
```

```
public function unavailableCategory($id)  
{  
    $query = $this -> db -> getMyDB() ->  
prepare('UPDATE category SET available = 0 WHERE  
category_ID = ?');  
    $query -> execute([$id]);  
}  
  
public function availableCategory($id)  
{  
    $query = $this -> db -> getMyDB() ->  
prepare('UPDATE category SET available = 1 WHERE  
category_ID = ?');  
    $query -> execute([$id]);  
}  
  
}
```


app/models/Chart.php

```
<?php
```

```
namespace app\models;
```

```
class Chart
```

```
{
```

```
    private $db;
```

```
    public function __construct($database)
```

```
    {
```

```
        $this->db = $database;
```

```
    }
```

```
public function getAllOrdersByUserId($user_id)
{
    $query = $this->db->getMyDB()->
prepare('SELECT pu.*, p.name AS productName,
p.price as productPrice, p.image as productImage,
u.username FROM product_user pu INNER JOIN
product p ON pu.product_ID = p.product_ID INNER
JOIN user u on pu.user_ID = u.user_ID WHERE u.user_ID
= ?');

    $query->execute([$user_id]);
    return $query->fetchAll();
}
```

```
public function deleteProductFromChart($id)
{
    $query = $this->db->getMyDB()->
prepare('DELETE from product_user WHERE
product_user_ID = ?');

    $query->execute([$id]);
}
```

}

app/models/Login.php

```
<?php
```

```
namespace app\models;
```

```
class Login {
```

```
    private $db;
```

```
    public function __construct($database)
    {
        $this->db = $database;
    }
```

```
    public function getUserByEmail($email)
    {
```

```
$query = $this -> db -> getMyDB() ->
prepare("SELECT * FROM user WHERE e_mail = ?");
$query -> execute([$email]);
return $query -> fetch();
}
```

```
public function getUserByEmailAndPassword($email,
$pass)
{
    $query = $this -> db -> getMyDB() ->
prepare("SELECT * FROM user WHERE e_mail = ? AND
password = ?");
    $query -> execute([$email, $pass]);
    return $query -> fetch();

}
```

```
public function updateLastVisitByUser($date,
$user_ID)
{
```

```
$query = $this -> db -> getMyDB() ->
prepare("UPDATE user SET last_visit = ? WHERE user_ID
= ?");

$query -> execute([$date, $user_ID]);
}
```

```
public function
updateActiveStatusToOnline($user_ID)
{
    $query = $this -> db -> getMyDB() ->
prepare("UPDATE user SET active = 1 WHERE user_ID =
?");
    $query -> execute([$user_ID]);
}
```

```
public function
updateActiveStatusToOffline($user_ID)
{
    $query = $this -> db -> getMyDB() ->
prepare("UPDATE user SET active = 0 WHERE user_ID =
?");
    $query -> execute([$user_ID]);
}
```

}

}

app/models/Product.php

```
<?php
```

```
namespace app\models;
```

```
class Product {
```

```
    private $db;
```

```
    public function __construct($database)
```

```
{
```

```
        $this->db = $database;
```

```
}
```

```
    public function getAllProducts()
```

```
{
```



```

    return $this -> db -> executeQuery('SELECT * FROM
product');
}

public function getAllProductsAndCategories()
{
    return $this -> db -> executeQuery('SELECT p.*,
c.available as categoryAvailable FROM product p
INNER JOIN category c ON p.category_ID =
c.category_ID');
}

public function paginationLimitProducts()
{
    return $this -> db -> executeQuery('SELECT p.*,
c.available as categoryAvailable FROM product p
INNER JOIN category c ON p.category_ID =
c.category_ID ORDER BY p.product_ID ASC LIMIT 0,5');
}

public function getPagination($limit)
{
    $query = $this -> db -> getMyDB() ->
prepare("SELECT p.*, c.available as

```

```
categoryAvailable FROM product p INNER JOIN  
category c ON p.category_ID = c.category_ID  
ORDER BY p.product_ID ASC LIMIT $limit,5 ");
```

```
$query -> execute([$limit]);
```

```
return $query -> fetchAll();
```

```
}
```

```
public function addToChart($product, $user, $date)
```

```
{
```

```
$query = $this -> db -> getMyDB() ->
```

```
prepare('INSERT INTO product_user VALUES(NULL, ?, ?,  
?);');
```

```
$query -> execute([$product, $user, $date]);
```

```
}
```

```
public function countAllProducts()
```

```
{
```

```
    return $this -> db -> executeOneRow('SELECT  
COUNT(*) as countAll FROM product');  
}
```

```
public function  
getProductsByCategory($category_id)  
{  
    $query = $this -> db -> getMyDB() ->  
prepare('SELECT p.* FROM product p INNER JOIN  
category c ON p.category_ID = c.category_ID  
WHERE c.category_ID = ?');  
    $query -> execute([$category_id]);  
    return $query -> fetchAll();  
  
}
```

```
public function getProductTroughName($val)  
{  
    $query = $this -> db -> getMyDB() ->  
prepare('SELECT p.*, c.name as categoryName  
FROM product p INNER JOIN category c ON
```

```
p.category_ID = c.category_ID WHERE p.name LIKE  
?');
```

```
$query -> execute([$val]);
```

```
return $query -> fetch();
```

```
}
```

```
public function getProduct($id)
```

```
{
```

```
$query = $this -> db -> getMyDB() ->
```

```
prepare('SELECT p.*, c.name AS categoryName,  
c.available as categoryAvailable FROM product p  
INNER JOIN category c ON p.category_ID =  
c.category_ID WHERE p.product_ID = ?');
```

```
$query -> execute([$id]);
```

```
return $query -> fetch();
```

```
}
```

```
public function addNewProduct($cat, $name,  
$price, $img, $info)
```

```
{
```

```
$query = $this -> db -> getMyDB() ->  
prepare('INSERT INTO product VALUES(NULL, ?, ?, ?, ?,  
?, 1)');
```

```
$query -> execute([$cat, $name, $price, $img,  
$info]);
```

```
}
```

```
public function unavailableProduct($id)
```

```
{
```

```
$query = $this -> db -> getMyDB() ->  
prepare('UPDATE product SET available = 0 WHERE  
product_ID = ?');
```

```
$query -> execute([$id]);
```

```
}
```

```
public function availableProduct($id)
```

```
{
```

```
$query = $this -> db -> getMyDB() ->  
prepare('UPDATE product SET available = 1 WHERE  
product_ID = ?');
```

```
$query -> execute([$id]);
```

```
}
```

```
public function editProduct($name, $price, $img,  
$info, $prod_id)
```

```
{
```

```
    $query = $this -> db -> getMyDB() ->  
prepare('UPDATE product SET name = ?, price = ?,  
image = ?, info = ? WHERE product_id = ?');
```

```
    $query -> execute([$name, $price, $img, $info,  
$prod_id]);
```

```
}
```

```
}
```

app/models/Register.php

```
<?php
```

```
namespace app\models;
```

```
class Register {
```

```
    private $db;
```

```
    public function __construct($database)
```

```
    {
```

```
        $this->db = $database;
```

```
    }
```

```
    public function getUserByUsername($username)
```

```
    {
```

```

    $query = $this -> db -> getMyDB() ->
prepare("SELECT * FROM user WHERE username = ?");
    $query -> execute([$username]);
    return $query -> fetch();
}

```

```

public function getUserByEmail($email)
{
    $query = $this -> db -> getMyDB() ->
prepare("SELECT * FROM user WHERE e_mail = ?");
    $query -> execute([$email]);
    return $query -> fetch();
}

```

```

public function insertUserModel($email, $pass,
$username, $register_date, $last_visit, $active)
{

```

```

    $query = $this -> db -> getMyDB() ->
prepare("INSERT INTO user VALUES(NULL, 2, ?, ?, ?, ?,
0, 0)");

```



```
$query -> execute([$email, $pass, $username,  
$register_date]);
```

```
}
```

```
}
```

app/models/User.php

```
<?php
```

```
namespace app\models;
```

```
class User {
```

```
    private $db;
```

```
    public function __construct($database)
```

```
{
```

```
        $this->db = $database;
```

```
}
```

```
    public function getAllUsers()
```

```
{
```

```
        return $this->db->executeQuery('SELECT u.*,  
r.name as role FROM user u INNER JOIN role r WHERE  
u.role_ID = r.role_ID');
```

}

}

App/models/Visit.php

```
<?php
```

```
namespace app\models;
```

```
class Visit {
```

```
    private $db;
```

```
    public function __construct($database)
```

```
{
```

```
        $this->db = $database;
```

```
}
```

```
    public function countAllVisits()
```

```
{
```

```
        return $this->db->executeOneRow("SELECT  
        number FROM visits");
```

```
}
```

```
public function restartVisits()  
{  
    $query = $this->db->getMyDb()->  
prepare("UPDATE visits SET number = 0");  
    $query->execute();  
}  
}
```

assets/js/main.js

```
window.onload = function() {  
  
    window.name_add_img = "";  
    window.full_path = "";  
    let button_all_products = $("#products_wrapper  
button");  
  
    for(let i = 0; i < button_all_products.length; i++)  
    {  
        if(button_all_products[i].dataset.id == 0)  
        {  
            button_all_products[i].click();  
        }  
    }  
}
```

```
}
```

```
};
```

```
// CHART
```

```
$(document).ready(function(){
```

```
    $('img[alt="www.000webhost.com"]').hide();
```

```
    $("div:last").hide();
```

```
$(document).on("click", "#add_to_chart_btn"  
,function(){
```

```
    let product_id = $(this).data('id');
```

```
    console.log(product_id);
```

```
$.ajax({

    url : "/chart",

    dataType : 'json',
    method : 'post',
    data : {
        _product_id : product_id
    },
    success : function(data)
    {

        if(data.inserted == true)
        {
            window.location.href = '/user_chart';
        }

    },
    error : function(xhr, status, responseText)
```



```
{  
  console.log(status);  
}
```

```
});
```

```
});
```

```
// CHART REMOVE
```

```
$(document).on("click", "#remove_from_chart_btn"  
,function(){
```

```
let product_user_id = $(this).data('id');
```

```
$.ajax({  
  url : "/remove_from_chart",  
  
  dataType : 'json',  
  method : 'post',  
  data : {  
    _product_id : product_user_id  
  },  
  success : function(data)  
  {
```

```
if(data.chart.length != 0)
{
    displayChart(data.chart);
}

else {
    displayChart();
}

},
error : function(xhr, status, responseText)
{
    console.log(status)
}
```

```
});
```

```
});
```

```
function displayChart(data){
```

```
  let x = ``;
```

```
  for(let d in data)
```

```
  {
```

```
    x += `
```

```
    <tr>
```

```
      <td class="text-capitalize  
">${data[d].productName}</td>
```

```
      <td>${data[d].productPrice}</td>
```

```
<td></td>
```

```
<td>${data[d].date}</td>
```

```
<td>
```

```
<button id='remove_from_chart_btn'  
type='button' data-id="${data[d].product_user_ID}"  
class=" btn btn-block btn-lg btn-gradient-primary mt-  
4"><a> Remove </a> </button>
```

```
</td>
```

```
</tr>
```

```
`;  
`;
```

```
}
```

```
$("#chart_user").html(x);
```

```
}
```

```
});
```

```
$(document).ready(function(){
```

```
// CONTACT
```

```
$("#btn_contact").click(function(){
```

```
    let name = $("#contact_name_surname").val();
```

```
    let e_mail = $("#contact_email").val();
```

```
    let message = $("#contact_message").val();
```

```
    let reName = /^[a-zA-Z]{2,}(\s[a-zA-Z]{2,})*$$/;
```

```
    let reEmail = /^[A-z\d]{2,}(\.?(\\W\\D)?[A-z\d]{2,})*\@\\w{2,}(\\.\\w{2,})(\\.\\w{2,})*$$/;
```

```
    let reMessage = /^[a-zA-Z]{5,}(\s)*[0-9]*$/;
```

```
    let flag = true;
```

```
if(!reName.test(name))  
{  
    flag = false;  
    $("#errorContactName").text('Incorrect name!');  
}
```

```
else {  
    $("#errorContactName").text("");  
}
```

```
if(!reEmail.test(e_mail))  
{  
    flag = false;  
    $("#errorContactEmail").text('Incorrect e-mail!');  
}
```

```
else {  
    $("#errorContactEmail").text("");  
}
```



```
if(!reMessage.test(message))  
{  
    flag = false;  
    $("#errorContactMessage").text('Incorrect  
message!');  
}
```

```
else {  
    $("#errorContactMessage").text("");  
}
```

```
if(flag )  
{
```

```
    $.ajax({
```

```
        url : "/contact_form",
```

```
        dataType : 'json',
```

```
        method : 'post',
```

```
data : {  
  _name : name,  
  _email : e_mail,  
  _message : message  
  
},  
success : function(data)  
{
```

```
  // if(data.mailer)  
  // {
```

```
    $("#responseMailer").text(data.mailer).fadeOut(2000)  
    ;
```

```
    $("#contact_name_surname").val("");  
    $("#contact_email").val("");
```

```
$("#contact_message").val("");  
// }
```

```
if(data.errorName != "")  
{  
    $("#errorContactName").text(data.errorName);  
}
```

```
else {  
    $("#errorContactName").text("");  
}
```

```
if(data.errorEmail != "")  
{  
    $("#errorContactEmail").text(data.errorEmail);  
}
```

```
else {  
    $("#errorContactEmail").text("");  
}
```

```
        if(data.errorMessage != "")
        {

$("#errorContactMessage").text(data.errorMessage);

        }

        else {
            $("#errorContactMessage").text("");
        }

    },
    error : function(xhr, status, responseText)
    {
        console.log(status);
    }

});
}
```

```
});
```

```
// SEARCH A PRODUCT
```

```
$("#search_products").click(function(){
```

```
let value = $("#products_input").val();
```

```
$.ajax({
```

```
url : "/product_search",
```

```
dataType : 'json',
```

```
method : 'post',
```

```
data : {
```

```
    _product_name : value
```

```
},  
success : function(data)  
{  
  
    if(data.product)  
    {  
        window.location.href='/home?product=' +  
data.product.product_ID;  
    }  
  
    else if(data.product == false){  
        window.location.href='/products';  
    }  
  
},  
error : function(xhr, status, responseText)  
{
```

```
    console.log(status);  
  }
```

```
});
```

```
});
```

```
});
```

```
// EDIT CATEGORY
```

```
$(document).on("click", "#btn_edit_a_category",  
function(){
```

```
let name = $("#edit_name_cat").val();
```

```
let cat_id = $("#hidden_category_id").val();
```

```
let reCategoryName = /^[A-z]{2,}(\s([A-z]{2,}))*$$/;
```

```
let flag = true;
```

```
if(!reCategoryName.test(name))
```

```
{
```

```
    flag = false;
```

```
    $("#regErrorEditCategoryName").html("Incorrect  
name!")
```

```
}
```

```
else {
```

```
    $("#regErrorEditCategoryName").html("")
```

```
}
```

```
if(flag)
```



```
{
$.ajax({

    url : "/edit_a_category",
    dataType : 'json',
    method : 'post',
    data : {
        _category_name : name,
        _category_id : cat_id
    },
    success : function(data)
    {

        if(data != undefined)
        {
            $("#regErrorEditCategory").html(data);
        }

        else {
            window.location.href = '/admin';
        }
    }
});
}
```

```
}
```

```
},
```

```
error : function(xhr, status, responseText)
```

```
{
```

```
    console.log(status);
```

```
}
```

```
});
```

```
}
```

```
})
```

```
// EDIT PRODUCT
```

```
$(document).on("click", "#btn_edit_a_product",  
function(){
```

```
let name = $("#edit_name_prod").val();
```

```
let price = $("#edit_price_prod").val();
```

```
let description = $("#edit_descr_prod").val();
```

```
let image = $("#image_edit_product").val();
```

```
let product_id = $("#hidden_product_id").val();
```

```
let reProdName = /^[A-z]{2,}((\s[A-z]{2,}))*$;/;
```

```
let reProdDescr = /[A-z]{3,}\s*/;
```

```
let rePrice = /^\\d?\\d{1}\\.\\d{2}$;/;
```

```
let flag = true;
```

```
let new_image = $("#img_path_edit").val();
```

```
if(!reProdName.test(name))
{
    flag = false;
    $("#regErrorEditProductName").html("Incorrect
name!")

}

else {
    $("#regErrorEditProductName").html("")
}

if(!reProdDescr.test(description))
{
    flag = false;
    $("#regErrorEditProductDescr").html("Incorrect
description!")
}
```

```
}
```

```
else {
```

```
    $("#regErrorEditProductDescr").html("")
```

```
}
```

```
if(!rePrice.test(price))
```

```
{
```

```
    flag = false;
```

```
    $("#regErrorEditProductPrice").html("Incorrect  
value!")
```

```
}
```

```
else {
```

```
    $("#regErrorEditProductPrice").html("")
```

```
}
```

```
if(new_image != "")
```

```
{  
  let img = $("#image_edit_product");  
  let image_size =  
    $("#image_edit_product")[0].files[0].size;  
  
  let file_size_mb = Math.round((image_size / 1024));  
  
  let extension = name_add_img.split('.');  
  
  let allowed_ext = extension[1].toLowerCase();  
  
  let errors = "";  
  
  if(((allowed_ext == 'jpg') || (allowed_ext == 'jpeg')  
    || (allowed_ext == 'png')) && file_size_mb < 2048)  
  {
```

```
$("#regErrorEditProductImg").html("");

}

else {
    flag = false;
    $("#regErrorEditProductImg").html('jpeg / jpg / png
    & not above 2MB');
}
}

if(flag)
{

    let request = new XMLHttpRequest();
    let formData = new FormData();
    let img = $("#image_edit_product");

    formData.append('image', img[0].files[0]);
```

```
formData.append('product_name', name);
formData.append('product_id', product_id);
formData.append('product_price', price);
formData.append('product_descr', description);

// request.open('post',
'/o_bakery/app/models/products/edit_a_product.php');
request.open('post', '/edit_a_product');
request.responseType = 'json';
request.send(formData);

request.onload = function(e)
{
    // console.log(request.response);
    window.location.href = '/admin';
}
});
```



```
$(document).ready(function(){
```

```
$(document).on("click", "#products_admin  
.delete_product", function(){
```

```
let product_id = $(this).data("id");
```

```
if($(this).hasClass('btn-gradient-red'))
```

```
{
```

```
$.ajax({
```

```
url : "/get_back_a_product",
```

```
dataType : 'json',
```

```
method : 'post',
```

```
data : {
```

```
_product_id : product_id
```

```
    },  
    success : function(data)  
    {  
        if(data.backed)  
        {  
            window.location.href = '/admin/products';  
        }  
  
    },  
    error : function(xhr, status, responseText)  
    {  
        console.log(status);  
    }  
  
    });  
}  
  
else {  
    $.ajax({
```

```
url : "/remove_a_product",
dataType : 'json',
method : 'post',
data : {
    _product_id : product_id
},
success : function(data)
{

    if(data.removed)
    {
        window.location.href = '/admin/products';
    }

},
error : function(xhr, status, responseText)
{
    console.log(status);
}
```

```
});  
}
```

```
// updateAndDisplayProducts();  
});
```

```
// AVAILABLE / UNAVAILABLE A CATEGORY
```

```
$(document).on("click", "#categories_admin  
.delete_category", function(){
```

```
console.log($(this).data('id'))
```

```
let category_id = $(this).data("id");
```

```
if($(this).hasClass('btn-gradient-red'))
{
$.ajax({
    url : "/get_back_a_category",

    dataType : 'json',
    method : 'post',
    data : {
        _category_id : category_id
    },
    success: function (data)
    {
        if(data.backed)
        {
            window.location.href = '/admin/categories';
        }
    },
    error : function(xhr, status, responseText)
    {
        console.log(status);
```

```
}
```

```
});
```

```
}
```

```
else {
```

```
$.ajax({
```

```
    url : "/remove_a_category",
```

```
    dataType : 'json',
```

```
    method : 'post',
```

```
    data : {
```

```
        _category_id : category_id
```

```
    },
```

```
    success: function(data)
```

```
    {
```

```
        if(data.removed)
```

```
        {
```

```
            window.location.href = '/admin/categories';
```

```
    }  
    },  
    error : function(xhr, status, responseText)  
    {  
        console.log(status);  
    }  
  
    });  
}  
  
// updateAndDisplayCategories();
```

```
});
```

```
// EDIT CATEGORY
```

```
$(document).on("click", "#categories_admin  
.edit_category", function(){
```

```
let category_id = $(this).data('id');
```

```
});
```

```
var all_products = $("#products_wrapper button");
```



```
for(let i of all_products)
{
  if($(i).data("id") !== 0)
  {
    $(i).removeClass("how-active1");
  }
}
```

```
});
```

```
$(document).on("click", "#btn_add_image",
function(){
```

```
$("#image_add_product").click();
```

```
})
```

```
$(document).on("click", "#btn_edit_image",  
function(){
```

```
    $("#image_edit_product").click();
```

```
})
```

```
$(document).on("change", "#image_add_product",  
function(){
```

```
    full_path = $("input[type=file]").val().split("\\\\");
```

```
    name_add_img = full_path[2];
```

```
    $("#img_path").val(full_path[2]);
```

```
})
```

```
$(document).on("change", "#image_edit_product",  
function(){
```

```
    full_path = $("input[type=file]").val().split("\\\\");
```

```
    name_add_img = full_path[2];
```

```
    $("#img_path_edit").val(full_path[2]);
```

```
})
```

```
/*[ Smooth scroll ]
```

```
=====
```

```
=====*/
```

```

$(function() {

    $('a[href*=\#\]:not([href=\#\]))'.on('click',
function() {

    var target = $(this.hash);

    target = target.length ? target : $('[name=' +
this.hash.substr(1) +']');
    if (target.length) {
        $('html,body').animate({
            scrollTop: target.offset().top
        }, 1000);
        return false;
    }
    });
});

```

```

(function ($) {
    "use strict";

    /*[ Load page ]

```

```
=====
=====*/
```

```
try {
    $(".animsition").animsition({
        inClass: 'fade-in',
        outClass: 'fade-out',
        inDuration: 1500,
        outDuration: 800,
        linkElement: '.animsition-link',
        loading: true,
        loadingParentElement: 'html',
        loadingClass: 'animsition-loading-1',
        loadingInner: '<div class="loader05"></div>',
        timeout: false,
        timeoutCountdown: 5000,
        onLoadEvent: true,
        browser: [ 'animation-duration', '-webkit-
animation-duration'],
        overlay : false,
```

```
        overlayClass : 'animations-overlay-slide',
        overlayParentElement : 'html',
        transition: function(url){ window.location.href =
url; }
    });
} catch(er) {console.log(er);}
```

```
/*[ Back to top ]
```

```
=====
=====*/
```

```
try {
    var windowH = $(window).height()/2;

    $(window).on('scroll',function(){
        if ($(this).scrollTop() > windowH) {
            $("#myBtn").css('display','flex');
        } else {
            $("#myBtn").css('display','none');
        }
    });
}
```

```
    }  
});
```

```
$('#myBtn').on("click", function(){  
    $('html, body').animate({scrollTop: 0}, 300);  
});  
} catch(er) {console.log(er);}
```

```
/*=====
```

```
[ Isotope ]*/
```

```
try {  
    var $topeContainer = $('.isotope-grid');  
    var $filter = $('.filter-tope-group');
```

```
// filter items on button click
$filter.each(function () {
    $filter.on('click', 'button', function () {
        var filterValue = $(this).attr('data-filter');
        $topeContainer.isotope({filter: filterValue});
    });
});
```

```
// init Isotope
$(window).on('load', function () {
    var $grid = $topeContainer.each(function () {
        $(this).isotope({
            itemSelector: '.isotope-item',
            layoutMode: 'masonry',
            percentPosition: true,
            animationEngine : 'best-available',
            masonry: {
                columnWidth: '.isotope-item'
            }
        });
    });
});
```



```
    }  
  });  
});  
});
```

```
var isotopeButton = $('.filter-tope-group button');
```

```
$(isotopeButton).each(function(){  
  $(this).on('click', function(){  
    for(var i=0; i<isotopeButton.length; i++) {  
      $(isotopeButton[i]).removeClass('how-  
active1');  
    }  
  })  
});
```

```
$(this).addClass('how-active1');
```

```
if($(this).data('filter') === "") {  
  $('.isotope-grid-gallery .isotope-item .js-  
gallery').addClass('js-show-gallery');
```

```
        $(' .gallery-lb.isotope-grid-  
gallery').each(function() {  
            $(this).find(' .js-show-  
gallery').magnificPopup({  
                type: 'image',  
                gallery: {  
                    enabled:true  
                },  
                mainClass: 'mfp-fade'  
            });  
        });  
    }  
    else {  
        $(' .isotope-grid-gallery .isotope-item .js-  
gallery').removeClass('js-show-gallery');  
        $(' .isotope-grid-gallery ' +  
$(this).data('filter') + ' .js-gallery').addClass('js-show-  
gallery');
```

```

        $('.gallery-lb.isotope-grid-
gallery').each(function() {
            $(this).find('.js-show-
gallery').magnificPopup({
                type: 'image',
                gallery: {
                    enabled:true
                },
                mainClass: 'mfp-fade'
            });
        });
    }
});
});
} catch(er) {console.log(er);}

```

```

/*=====
=====

```

```
[ Fixed Header ]*/  
try {  
    var headerDesktop = $('.container-menu-  
desktop');  
    var wrapMenu = $('.wrap-menu-desktop');  
  
    if($('.top-bar').length > 0) {  
        var posWrapHeader = $('.top-bar').height();  
    }  
    else {  
        var posWrapHeader = 0;  
    }  
  
    if($(window).scrollTop() > posWrapHeader) {  
        $(headerDesktop).addClass('fix-menu-  
desktop');  
        $(wrapMenu).css('top',0);  
    }  
    else {
```

```
        $(headerDesktop).removeClass('fix-menu-  
desktop');
```

```
        $(wrapMenu).css('top',posWrapHeader -  
$(this).scrollTop());
```

```
    }
```

```
$(window).on('scroll',function(){
```

```
    if($(this).scrollTop() > posWrapHeader) {
```

```
        $(headerDesktop).addClass('fix-menu-  
desktop');
```

```
        $(wrapMenu).css('top',0);
```

```
    }
```

```
    else {
```

```
        $(headerDesktop).removeClass('fix-menu-  
desktop');
```

```
        $(wrapMenu).css('top',posWrapHeader -  
$(this).scrollTop());
```

```
    }
```

```
});
```

```
} catch(er) {console.log(er);}
```

```

/*=====
=====

[ Menu mobile ]*/
try {
    $('.btn-show-menu-mobile').on('click', function(){
        $(this).toggleClass('is-active');
        $('.menu-mobile').slideToggle();
    });

    var arrowMainMenu = $('.arrow-main-menu-m');

    for(var i=0; i<arrowMainMenu.length; i++){
        $(arrowMainMenu[i]).on('click', function(){
            $(this).parent().find('.sub-menu-
m').slideToggle();
            $(this).toggleClass('turn-arrow-main-menu-
m');
        })
    }
}

```

```
}
```

```
$(window).on('resize',function(){  
    if($(window).width() >= 992){  
        if($('.menu-mobile').css('display') === 'block')  
{  
            $('.menu-mobile').css('display','none');  
            $('.btn-show-menu-  
mobile').toggleClass('is-active');  
        }  
  
        $('.sub-menu-m').each(function(){  
            if($(this).css('display') === 'block') {  
                $(this).css('display','none');  
                $(arrowMainMenu).removeClass('turn-  
arrow-main-menu-m');  
            }  
        });  
    }  
})
```

```
});  
} catch(er) {console.log(er);}

```

```
/*=====
=====
[ Show / hide modal search ]*/
try {
    $('.js-show-modal-search').on('click', function(){
        $('.modal-search-header').addClass('show-
modal-search');
        // $(this).css('opacity','0');
    });

    $('.js-hide-modal-search').on('click', function(){
        $('.modal-search-header').removeClass('show-
modal-search');
        // $('.js-show-modal-search').css('opacity','1');
    });

```



```
    $('.container-search-header').on('click',  
function(e){  
    e.stopPropagation();  
    });  
} catch(er) {console.log(er);}
```

```
/*=====
```

```
=====
```

```
[ Show / hide modal search ]*/
```

```
try {  
    $('.js-show-modal-user').on('click', function(){  
        $('.modal-user-header').addClass('show-  
modal-user');  
        // $(this).css('opacity','0');  
    });
```

```
    $('.js-hide-modal-user').on('click', function(){  
        $('.modal-user-header').removeClass('show-  
modal-user');
```

```
        // $('.js-show-modal-user').css('opacity','1');
    });

    $('.container-user-header').on('click', function(e){
        e.stopPropagation();
    });
} catch(er) {console.log(er);}
```

```
/*=====
```

```
=====
```

```
[ Cart header ]*/
```

```
try {
```

```
    $('.wrap-menu-click').each(function(){
```

```
        var wrapMenuClick = $(this);
```

```
        $(wrapMenuClick).find('.menu-click').on('click',
function(e){
```

```
e.stopPropagation();
```

```
if($(this).hasClass('showed')) {  
    $(wrapMenuClick).find('.menu-  
click').removeClass('show-menu-click showed');  
}  
else {  
    $(wrapMenuClick).find('.menu-  
click').removeClass('show-menu-click showed');  
    $(this).addClass('show-menu-click  
showed');  
}  
});
```

```
$(wrapMenuClick).find('.menu-click-  
child').on('click', function(e){  
    e.stopPropagation();  
});  
});
```

```
$(window).on('click', function(){
```

```
        $('.wrap-menu-click').find('.menu-  
click').removeClass('show-menu-click showed');  
    })  
} catch(er) {console.log(er);}

```

```
/*=====
=====

```

```
    [ Sweetalert ]*/  
    try {  
        $('.js-addwish-b1, .js-addwish1').on('click',  
function(e){  
            e.preventDefault();  
        });  
  
        $('.js-addwish-b1').each(function(){  
            var nameProduct =  
$(this).parent().parent().find('.js-name-b1').html();  
            $(this).on('click', function(){
```

```
        swal(nameProduct, "is added to wishlist !",  
"success");
```

```
        $(this).addClass('js-addedwish-b1');  
        $(this).removeClass('js-addwish-b1');  
        $(this).off('click');  
    });  
});
```

```
$('.js-addcart-b1').each(function(){  
    var nameProduct =  
$(this).parent().parent().find('js-name-b1').html();  
    $(this).on('click', function(e){  
        e.preventDefault();  
        swal(nameProduct, "is added to cart !",  
"success");  
    });  
});
```

```

/*-----*/
$('.js-addwish1').each(function(){
    var nameProduct = $(this).parent().find('.js-
name1').html();
    $(this).on('click', function(){
        swal(nameProduct, "is added to wishlist !",
"success");

        $(this).addClass('js-addedwish1');
        $(this).off('click');
    });
});

$('.js-addcart1').each(function(){
    var nameProduct =
$(this).parent().parent().find('.js-name1').html();
    $(this).on('click', function(e){
        e.preventDefault();
        swal(nameProduct, "is added to cart !",
"success");
    });
});

```

```
});
```

```
    } catch(er) {console.log(er);}


```

```
/*=====
=====

```

```
    [ Parallax100 ]*/
```

```
    try {
```

```
        $(''.parallax100').parallax100();
```

```
    } catch(er) {console.log(er);}


```

```
/*=====
=====

```

```
    [ Perfect scrollbar ]*/
```

```
    try {
```

```
        $(''.js-pscroll').each(function(){
```

```
            $(this).css('position','relative');
```

```

$(this).css('overflow','hidden');
var ps = new PerfectScrollbar(this, {
    wheelSpeed: 1,
    scrollingThreshold: 1000,
    wheelPropagation: false,
});

$(window).on('resize', function(){
    ps.update();
})
});
} catch(er) {console.log(er);}

```

```

/*=====
=====

```

```

[ Counter up ]*/
try {
    $('.counter').counterUp({

```



```
        delay: 10,  
        time: 1000  
    });  
} catch(er) {console.log(er);}
```

```
/*=====
```

```
=====
```

```
[ Countdown ]*/
```

```
try {
```

```
    $(' .coutdown100').each(function(){
```

```
        if($(this).data('year') != null){
```

```
            var year = Number($(this).data('year'));
```

```
        }
```

```
        else {var year = 0;}
```

```
        if($(this).data('month') != null){
```

```
            var month = Number($(this).data('month'));
```

```
        }
```

```
else {var month = 0;}
```

```
if($(this).data('date') != null){  
    var date = Number($(this).data('date'));  
}
```

```
else {var date = 0;}
```

```
if($(this).data('hours') != null){  
    var hours = Number($(this).data('hours'));  
}
```

```
else {var hours = 0}
```

```
if($(this).data('minutes') != null){  
    var minutes =  
Number($(this).data('minutes'));  
}
```

```
else {var minutes = 0;}
```

```
if($(this).data('seconds') != null){
```

```
        var seconds =  
Number($(this).data('seconds'));  
    }  
    else {var seconds = 0;}
```

```
        if($(this).data('timezone') != null &&  
$(this).data('timezone') != "auto"){  
            var timeZ = $(this).data('timezone');  
        }  
        else {var timeZ = "";}  
    }
```

```
$(this).countdown100({  
    /*Set Endtime here*/  
    /*Endtime must be > current time*/  
    endtimeYear: year,  
    endtimeMonth: month,  
    endtimeDate: date,  
    endtimeHours: hours,  
    endtimeMinutes: minutes,
```

```
        endTimeSeconds: seconds,  
        timeZone: timeZ  
        // ex: timeZone: "America/New_York"  
        //go to " http://momentjs.com/timezone/ "  
to get timezone  
    });  
});
```

```
} catch(er) {console.log(er);}
```

```
/*=====
```

```
=====
```

```
[ Video ]*/  
try {  
    $(' .btn-play').on('click', function(ev) {  
        $(' .wrap-iframe-video').children('iframe')[0].src  
+= "rel=0&autoplay=1";
```

```
        $('.wrap-iframe-video').addClass('show-  
video');
```

```
        $(this).fadeOut();  
        ev.preventDefault();  
    });  
} catch(er) {console.log(er);}
```

```
/*=====
```

```
[ Progress bar ]*/
```

```
try {  
    var progressItem = $('.progress-item');  
    var progressDone = false;  
  
    $(window).on('scroll',function(){  
        progressItem.each(function(){  
            var per = Number($(this).data('percent'));  

```

```

        var inner = $(this).children('.progress-inner');

        if($(window).scrollTop() + $(window).height()
> $(this).offset().top && per > 0) {
            $(this).data('percent','0');
            inner.html(per + "%");
            inner.animate({width: per + "%"},1500);
        }
    });
});
} catch(er) {console.log(er);}

```

```

/*=====
=====

[ JS height ]*/
try {
    $(window).on('resize', function(){
        $('.js-height').each(function(){

```

```
        $(this).css('height',$$(this).find('.js-height-  
child').height());  
    });  
});
```

```
$(window).on('load', function(){  
    $('.js-height').each(function(){  
        $(this).css('height',$$(this).find('.js-height-  
child').height());  
    });  
});  
} catch(er) {console.log(er);} 
```

```
/*=====
```

```
=====
```

```
[ Magnific-Popup ]*/
```

```
try {
```

```
    $('.gallery-lb').each(function() {
```

```
        $(this).find('.js-show-gallery').magnificPopup({
```

```

        type: 'image',
        gallery: {
            enabled: true
        },
        mainClass: 'mfp-fade'
    });
});
} catch(er) {console.log(er);}

```

```

/*=====
=====

```

```

    [ Noui ]*/
    try {
        var filterBar = document.getElementById('filter-
bar');
        var fromValue = Number($('#value-
lower').html());
        var toValue = Number($('#value-upper').html());

        nouiSlider.create(filterBar, {

```



```
start: [ fromValue, toValue ],
connect: true,
range: {
    'min': fromValue,
    'max': toValue
}
});
```

```
var skipValues = [
document.getElementById('value-lower'),
document.getElementById('value-upper')
];
```

```
filterBar.noUiSlider.on('update', function( values,
handle ) {
    skipValues[handle].innerHTML =
Math.round(values[handle]) ;
});
} catch(er) {console.log(er);}
```

```
/*=====
```

```
=====
```

```
[ Select2 ]*/
```

```
try {
```

```
    $(".js-select2").each(function(){
```

```
        $(this).select2({
```

```
            minimumResultsForSearch: 20,
```

```
            dropdownParent:
```

```
$(this).next('.dropDownSelect2')
```

```
        });
```

```
    });
```

```
} catch(er) {console.log(er);}
```

```
/*=====
```

```
=====
```

```
[ Show grid / list ]*/
```

```
try {
```

```
$('.js-show-grid').on('click', function(){  
    $(this).addClass('menu-active');  
    $('.js-show-list').removeClass('menu-active');  
  
    $('.shop-grid').fadeOut();  
    $('.shop-list').hide();  
});
```

```
$('.js-show-list').on('click', function(){  
    $(this).addClass('menu-active');  
    $('.js-show-grid').removeClass('menu-active');  
  
    $('.shop-list').fadeOut();  
    $('.shop-grid').hide();  
});
```

```
} catch(er) {console.log(er);}
```

```
/*=====
```

```
=====
```

```

[ +/- num product ]*/
try {
    $('.btn-num-product-down').on('click', function(){
        var numProduct = Number($(this).next().val());
        if(numProduct > 0)
            $(this).next().val(numProduct - 1);
    });

    $('.btn-num-product-up').on('click', function(){
        var numProduct = Number($(this).prev().val());
        $(this).prev().val(numProduct + 1);
    });
} catch(er) {console.log(er);}

```

```

/*=====
=====

```

```

[ Rating ]*/
try {
    $('.wrap-rating').each(function(){

```

```
var item = $(this).find('.item-rating');  
var rated = -1;  
var input = $(this).find('input');  
$(input).val(0);
```

```
$(item).on('mouseenter', function(){  
    var index = item.index(this);  
    var i = 0;  
    for(i=0; i<=index; i++) {  
        $(item[i]).removeClass('fa-star-o');  
        $(item[i]).addClass('fa-star');  
    }
```

```
    for(var j=i; j<item.length; j++) {  
        $(item[j]).addClass('fa-star-o');  
        $(item[j]).removeClass('fa-star');  
    }  
});
```

```
$(item).on('click', function(){
```

```
var index = item.index(this);
rated = index;
$(input).val(index+1);
});

$(this).on('mouseleave', function(){
    var i = 0;
    for(i=0; i<=rated; i++) {
        $(item[i]).removeClass('fa-star-o');
        $(item[i]).addClass('fa-star');
    }

    for(var j=i; j<item.length; j++) {
        $(item[j]).addClass('fa-star-o');
        $(item[j]).removeClass('fa-star');
    }
});

});
} catch(er) {console.log(er);}
```

```

/*=====
=====

[ Show/hide panel1 ]*/
try {
    $('.js-toggle-panel1').on('click', function(){
        $(this).parent().parent().find('.js-
panel1').slideToggle();
    });
} catch(er) {console.log(er);}

```

```

/*=====
=====

[ Chose pay ]*/
try {
    $("#radio1").on('change', function(){
        if ($(this).is(":checked")) {
            $('.content-payment').slideDown(300);
            $('.content-paypal').slideUp(300);
        }
    });
} catch(er) {console.log(er);}

```

```
    }  
});
```

```
$("#radio2").on('change', function(){  
    if ($(this).is(":checked")) {  
        $('.content-payment').slideUp(300);  
        $('.content-paypal').slideDown(300);  
    }  
});  
} catch(er) {console.log(er);}
```

```
/*=====
```

```
=====
```

```
[ Show/hide Reply cmt ]*/
```

```
try {  
    $('.js-show-reply-cmt').on('click', function(e){  
        e.preventDefault();  
        $(this).parent().parent().parent().find('.js-reply-  
cmt').show();
```



```
$(this).addClass('how-active2');  
});
```

```
$('.js-hide-reply-cmt').on('click', function(e){  
    e.preventDefault();  
    $(this).parent().parent().hide();  
    $(this).parent().parent().parent().find('.js-show-  
reply-cmt').removeClass('how-active2');  
    });  
} catch(er) {console.log(er);}
```

```
})(jQuery);
```

```
// REGISTRATION
```

```
$(document).ready(function(){
```

```
$("#btn_register").click(function(){

    let reg_username =
document.querySelector("#register_username").value;

    let reg_email =
document.querySelector("#register_email").value;

    let reg_password =
document.querySelector("#register_password").value
;

    let reUsername = /^[a-z]{3,8}(\_{0,2})(\d{0,3})$/;
    let reEmail = /^[A-z\d]{2,}(\.?(\\W\\D)?[A-
z\d]{2,})*\\@\\w{2,}(\\.\\w{2,})(\\.\\w{2,})*$/;
    let rePassword = /([\\w\\W\\D\\d]){7,}/;

    let flag = true;

    if(!reUsername.test(reg_username))
```

```
{  
    flag = false;  
    $("#regErrorUsername").html("Not valid  
username!");  
}
```

```
else {  
    $("#regErrorUsername").html("");  
  
}
```

```
if(!reEmail.test(reg_email))  
{  
    flag = false;  
    $("#regErrorEmail").html("Not valid e-mail!");  
}
```

```
else {  
    $("#regErrorEmail").html("");
```

```
}
```

```
if(!rePassword.test(reg_password))
```

```
{
```

```
    flag = false;
```

```
    $("#regErrorPassword").html("At least 7  
characters!");
```

```
}
```

```
else {
```

```
    $("#regErrorPassword").html("");
```

```
}
```

```
if(flag == true)
```

```
{
```

```
    $.ajax({
```

```
url : "/register",
```

```
dataType : 'json',
```

```
method : 'post',
```

```
data : {
```

```
    _username : reg_username,
```

```
    _email : reg_email,
```

```
    _pass : reg_password
```

```
},
```

```
success : function(data){
```

```
    if(data.registered == 1)
```

```
    {
```

```
        window.location.href = "home";
```

```
    }
```

```
if(data.errorUsername)
```

```
{
```

```
$("#regErrorUsername").html(data.errorUsername);  
    }  
    else {  
        $("#regErrorUsername").html("");  
    }  
  
    if(data.errorPassword)  
    {  
  
        $("#regErrorPassword").html(data.errorPassword);  
    }  
    else {  
        $("#regErrorPassword").html("");  
    }  
  
    if(data.errorEmail)  
    {  
        $("#regErrorEmail").html(data.errorEmail);  
    }
```

```
else {  
    $("#regErrorEmail").html("");  
}  
},
```

```
error : function(xhr, status, responseText){  
    console.log(status);  
}
```

```
});  
}
```

```
});
```

```
// LOGIN
```

```
$("#btn_sign_in").click(function(){
```

```
    let sign_in_email =  
    document.querySelector("#sign_in_email").value;
```

```
    let sign_in_password =  
    document.querySelector("#sign_in_password").value;
```

```
    let reEmail = /^[A-z\d]{2,}(\.?(\\W\\D)?[A-  
z\d]{2,})*\\@\\w{2,}(\\.\\w{2,})(\\.\\w{2,})*$/;
```

```
    let rePassword = /([\\w\\W\\D\\d]){7,}/;
```

```
    let flag = true;
```

```
    if(!rePassword.test(sign_in_password))
```

```
    {
```

```
        flag = false;
```

```
        $("#signInErrorPassword").html("Not valid  
password!");
```



```
}
```

```
else {
```

```
    $("#signInErrorPassword").html("");
```

```
}
```

```
if(!reEmail.test(sign_in_email))
```

```
{
```

```
    flag = false;
```

```
    $("#signInErrorEmail").html("Not valid e-mail!");
```

```
}
```

```
else {
```

```
    $("#signInErrorEmail").html("");
```

```
}
```

```
if(flag == true)
{
$.ajax({

    url : "/login",
    dataType : 'json',
    method : 'post',
    data : {
        _email : sign_in_email,
        _pass : sign_in_password

    },
    success : function(data)
    {

        if(data.role)
        {
            window.location.href = 'home';
        }
    }
}
```

```
    if(data.errorPassword)
    {

$("#signInErrorPassword").html(data.errorPassword);
    }
    else {
        $("#signInErrorPassword").html("");
    }

    if(data.errorEmail)
    {
        $("#signInErrorEmail").html(data.errorEmail);
    }
    else {
        $("#signInErrorEmail").html("");
    }
},
error : function(xhr, status, responseText)
{
```

```
    console.log(status);  
  }
```

```
  });  
}
```

```
});
```

```
// SIGN OUT
```

```
$(document).on("click", "#btn_sign_out", function(){  
  $.ajax({
```

```
    url : "/logout",  
    dataType : 'json',
```

```
success : function(data)
{

    if(data.logout)
    {
        window.location.href = "home";
    }

},
error : function(xhr, status, responseText)
{
    console.log(status);
}

});

});
```

// PRODUCTS BY CATEGORY

```
$(document).on("click", "#products_wrapper  
button", function(){
```

```
    let category = this.dataset.id;
```

```
    $.ajax({
```

```
        url : "/products_by_category",
```

```
        dataType : 'json',
```

```
        method : 'post',
```

```
        data : {
```

```
            _category_ID : category
```

```
        },
```

```
        success : function(data)
```

```
        {
```

```
            displayProductsByCategory(data.products);
```

```
        },
```

```
        error : function(xhr, status, responseText)
```

```
        {
```

```
        console.log(status);
    }

});

});

// ADD A CATEOGRY

$(document).on("click", "#btn_add_a_category",
function(){

    let category_name = $("#add_a_category").val();
    let reCategroryName = /^[A-z]{2,}(\s([A-z]{2,}))*/;

    if(reCategroryName.test(category_name))
    {
```

```
$("#regErrorAddCategory").html("");  
$.ajax({  
    url : "/add_a_category",  
    dataType : 'json',  
    method : 'post',  
    data : {  
        _category_name : category_name  
    },  
    success : function(data)  
    {  
  
        if(data.errAddCategory != "")  
        {  
  
            $("#regErrorAddCategory").html(data.errAddCategory);  
  
        }  
  
        else if(data.errAddCategory == "")  
        {
```



```
$("#add_a_category").val("");

// let added_cat =
$("#regErrorAddCategory").html("Category added!");
//
// $(added_cat).fadeOut(2000);
window.location.href = '/admin'

}

},
error : function(xhr, status, responseText)
{
```

```
    console.log(status);  
  }
```

```
  });  
}
```

```
else {  
    $("#regErrorAddCategory").html("Incorrect  
value!");  
}
```

```
});
```

```
// ADD A PRODUCT
```

```
$(document).on("click", "#btn_add_a_product",  
function(){
```

```
    let product_name = $("#add_name_prod").val();
```

```
    let product_price = $("#add_price_prod").val();
```

```
    let product_description =  
    $("#add_descr_prod").val();
```

```
    let product_category =  
    $("#dropdown_categories").val();
```

```
    let reProdName = /^[A-z]{2,}((\s[A-z]{2,}))*$$/;
```

```
    let reProdDescr = /[A-z]{3,}\s*/;
```

```
    let rePrice = /^\\d?\\d{1}\\.\\d{2}$/;
```

```
    let flag = true;
```

```
if(!reProdName.test(product_name))
{
    flag = false;
    $("#regErrorAddProductName").html("Incorrect
name!")

}

else {
    $("#regErrorAddProductName").html("")
}

if(!reProdDescr.test(product_description))
{
    flag = false;
    $("#regErrorAddProductDescr").html("Incorrect
description!")
```

```
}
```

```
else {
```

```
    $("#regErrorAddProductDescr").html("")
```

```
}
```

```
if(!rePrice.test(product_price))
```

```
{
```

```
    flag = false;
```

```
    $("#regErrorAddProductPrice").html("Incorrect  
value!")
```

```
}
```

```
else {
```

```
    $("#regErrorAddProductPrice").html("")
```

```
}
```

```
if(name_add_img == "")
```

```
{
```

```
flag = false;

$("#regErrorAddProductImg").html('Please supply
image!');

}

else {

    let image_size =
$("##image_add_product")[0].files[0].size;

    let file_size_mb = Math.round((image_size / 1024));

    let extension = name_add_img.split('.');

    let allowed_ext = extension[1].toLowerCase();

    let errors = "";
```

```
if(((allowed_ext == 'jpg') || (allowed_ext ==  
'jpeg') || (allowed_ext == 'png')) && file_size_mb <  
2048)
```

```
{
```

```
$("#regErrorAddProductImg").html("");
```

```
}
```

```
else {
```

```
flag = false;
```

```
$("#regErrorAddProductImg").html('jpeg / jpg /  
png & not above 2MB');
```

```
}
```

```
        // if((allowed_ext === 'jpg') || (allowed_ext ===
'jpeg') || (allowed_ext === 'png'))
        // {
        //   $("#regErrorAddProductImg").html("");
        //
        // }
        //
        // else if((allowed_ext !== 'jpg') || (allowed_ext !==
'jpeg') || (allowed_ext !== 'png')){
        //   flag = false;
        //   $("#regErrorAddProductImg").html('jpg / png /
jpeg');
        // }

    }
```



```
if(product_category == "Choose...")
{
    $("#regErrorAddProductCat").html('Please select
category');
}

else {
    $("#regErrorAddProductCat").html("");
}

if(flag)
{

    // $.ajax({
    //   url :
"/o_bakery/app/models/products/add_a_product.p
hp",
    //   dataType : 'json',
    //   method : 'post',
    //   success : function(data)
```

```
// {  
//  
//   console.log(data);  
//  
// },  
// error : function(xhr, status, responseText)  
// {  
//   console.log(status);  
// }  
//  
//  
// });
```

```
let request = new XMLHttpRequest();
```

```
let formData = new FormData();
```

```
let img = $("#image_add_product");
```

```
formData.append('image', img[0].files[0]);
```

```
    formData.append('product_name',  
product_name);  
    formData.append('product_price',  
product_price);  
    formData.append('product_descr',  
product_description);  
    formData.append('product_category',  
product_category);
```

```
    // request.open('post',  
'/o_bakery/app/models/products/add_a_product.ph  
p');
```

```
    request.open('post', '/add_a_product');  
    request.responseType = 'json';  
    request.send(formData);
```

```
    request.onload = function(e)  
{
```

```
        window.location.href='/admin';  
        // if(request.response != null)
```

```
// {  
//  
//  
// if(request.response.errorName != "")  
// {  
//  
$("#regErrorAddProductName").html("Incorrect  
name!")  
// }  
// else if(request.response.errorName == ""){  
//   $("#regErrorAddProductName").html("")  
// }  
//  
//  
// if(request.response.errorDescr != "")  
// {  
//  
$("#regErrorAddProductDescr").html("Incorrect  
description!")  
// }  
// else if(request.response.errorDescr == ""){
```

```
//    $("#regErrorAddProductDescr").html("")
// }
//
// if(request.response.errorPrice != "")
// {
//
//    $("#regErrorAddProductPrice").html("Incorrect
value!")
// }
// else if(request.response.errorPrice == ""){
//    $("#regErrorAddProductPrice").html("")
// }
//
// if(request.response.errorCat != "")
// {
//    $("#regErrorAddProductCat").html('Please
select category');
// }
// else if(request.response.errorCat == ""){
//    $("#regErrorAddProductCat").html("");
```

```
// }  
  
//  
// if(request.response.errorImg != "")  
// {  
//  
$("#regErrorAddProductImg").html(request.response.e  
rrorImg);  
// }  
// else if(request.response.errorImg == ""){  
//   $("#regErrorAddProductImg").html("");  
// }  
//}  
//  
// else {  
//   $("#regErrorAddProductName").html("");  
//   $("#regErrorAddProductDescr").html("");  
//   $("#regErrorAddProductPrice").html("");  
//   $("#regErrorAddProductCat").html("");  
//   $("#regErrorAddProductImg").html("");  
//
```

```
// }
```

```
//
```

```
//
```

```
//
```

```
}
```

```
}
```

```
});
```

```

function displayProductsByCategory(products)
{
    let x = ``;

    for(let p in products)
    {

        x += `

        <div class="col-sm-6 col-md-4 col-lg-3 p-b-75
isotope-item">

            <div class="block1">

                <div class="block1-bg wrap-pic-w bo-all-1
bocl12 hov3 trans-04">

```



```
<div class="block1-content flex-col-c-m p-b-46">
```

```
<a href="product"
class="capitalizeProductName txt-m-103 cl3 txt-
center hov-cl10 trans-04 js-name-b1">
```

```
    ${products[p].name}
```

```
</a>
```

```
<div class="block1-wrap-icon flex-c-m flex-w
trans-05">
```

```
<a target="_blank" data-id =
"${products[p].product_ID}"
href="/home?product=${products[p].product_ID}"
class="block1-icon flex-c-m wrap-pic-max-w">
```

```
<i class="fa fa-search-plus" aria-
hidden="true"></i>
```

```
</a>
```

</div>

</div>

</div>

</div>

</div>

`;
`;

}

\$("#products_grid").html(x);

```
}
```

```
// $(document).ready(function(){
```

```
    // USERS  
    $.ajax({
```

```
//  
//    url : "/o_bakery/index.php",  
//    dataType : 'json',  
//    success : function(data)  
//    {  
//  
//        console.log("hi");  
//        displayUsers(data);  
//  
//  
//  
//    },  
//    error : function(xhr, status, responseText)  
//    {  
//        console.log(status);  
//    }  
//  
//  
//    });  
//
```

```
//    function displayUsers(data)
//    {
//        let x = ``;
//        let counter = 1;
//        for(let i of data)
//        {
//            x += `
//            <tr>
//                <td data-id=${i.user_ID}>${counter++}</td>
//                <td>${i.role} </td>
//                <td>${i.username}</td>
//                <td>${i.e_mail}</td>
//                <td>`;
//
//            if(i.role == "admin")
//            {
//                x += ` - `;
//            }
//
//            else {
```

```
//          let register_date_split =  
i.register_date.split(" ");  
  
//  
  
//          let date_register_date_split =  
register_date_split[0].split("-");  
  
//          let time_register_date_split =  
register_date_split[1].split(":");  
  
//  
  
//          let time_register_date =  
time_register_date_split[0] + ":" +  
time_register_date_split[1];  
  
//          let date_register_date =  
date_register_date_split[2] + "/" +  
date_register_date_split[1] + "/" +  
date_register_date_split[0];  
  
//  
  
//          x += `${date_register_date}  
${time_register_date}`;  
  
//      }  
  
//  
  
//  
  
//
```

```
//  
//      let last_active_split = i.last_visit.split(" ");  
//  
//      let date_last_active_split =  
last_active_split[0].split("-");  
//      let time_last_active_split =  
last_active_split[1].split(":");  
//  
//      let date_last_active =  
date_last_active_split[2] + "/" +  
date_last_active_split[1] + "/" +  
date_last_active_split[0];  
//      let time_last_active =  
time_last_active_split[0] + ":" +  
time_last_active_split[1];  
//  
//  
//  
//      x += `  
//  
//
```

```
//  
//  
//  
//  
//      </td>  
//      <td>${date_last_active}  
${time_last_active}</td>  
//      <td>  
//  
//      `;  
//      if(i.active == 0)  
//      {  
//          x += `<i class="fa fa-red fa-circle"></i>`;   
//      }  
//  
//  
//      else if(i.active == 1){  
//          x += `<i class="fa fa-green fa-  
circle"></i>`;   
//      }
```



```
//  
//      x += `  
//  
//  
//      </td>  
//  
//      </tr>  
//  
//  
//  
//      `;  
//      }  
//  
//  
//      $("#users_admin").html(x);  
//  
//  
//      }  
//  
// });
```

```
});
```

```
assets/admin/assets/js/chart.js
```

```
$(document).ready(function(){
```

```
$.ajax({  
  url : "/visits",  
  method: 'post',  
  dataType : 'json',  
  success : function(data)  
  {  
  
    let visits = data.visits.number;  
    fillChart(visits);  
  
  },  
  error : function(xhr, status, responseText)  
  {  
    console.log(status);  
  }  
  
});
```

```
function fillChart(visits)
{

    let current_time = new Date();


    var doughnutPieData = {
        datasets: [{
            data: [visits],
            backgroundColor: [

                'rgba(255, 206, 86, 0.5)',

            ],
            borderColor: [
```

```
    'rgba(255, 159, 64, 1)'  
  ],  
}],
```

// These labels appear in the legend and in the
tooltips when hovering different arcs

```
  labels: [  
    `${current_time.getFullYear()}`  
  ]  
};  
  
var doughnutPieOptions = {  
  responsive: true,  
  animation: {  
    animateScale: true,  
    animateRotate: true  
  }  
};
```

// Get context with jQuery - using jQuery's .get() method.

```
if ($("#pieChart").length) {  
    var pieChartCanvas =  
    $("#pieChart").get(0).getContext("2d");  
    var pieChart = new Chart(pieChartCanvas, {  
        type: 'pie',  
        data: doughnutPieData,  
        options: doughnutPieOptions  
    });  
}  
  
}  
  
})
```

