

Documentazione FREDWebTechnology

Progetto di tecnologia del web

Nella seguente documentazione sono riportati:

- 1) Operazione per installazione
- 2) Lo schema logico del database utilizzato
- 3) Scheletro di ogni cartella con i relativi file presenti e il codice di file più significativo.

1) Operazione di Installazione:

1.1 Requisiti

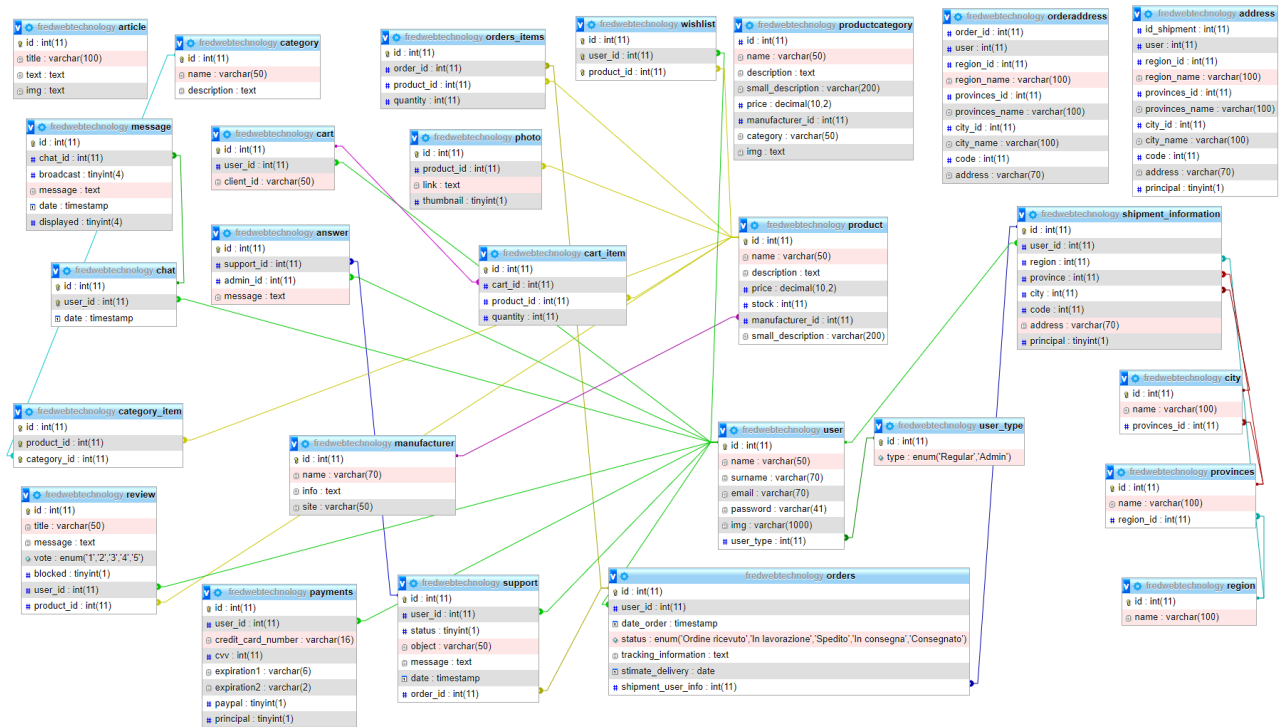
- Aver installato e configurato git
- Server web compatibile con php versione 7.*
- Aver installato composer

1.2 Operazioni

1. Clonare il repository nella directory desiderata
git clone https://github.com/3daniele/FREDWebTechnology.git
2. Spostarsi all'interno della directory del progetto
cd FREDWebTechnology
3. Installare tutte le dipendenze necessarie al funzionamento del progetto
composer install
4. Configurare i dati nel file config-example.php e copiare le informazioni in un nuovo file config.php nella cartella inc
cp config-example.php config.php
5. Importare il database presente nella cartella DB

Motore di templating utilizzato twig: <https://twig.symfony.com/>

2) Schema logico



3) Codice più significativo

Nei file dove il percorso è evidenziato in **blu** viene preso il codice e riportato nella documentazione, nei file dove i percorsi sono evidenziati in **verde** non viene preso il codice perché considerato poco importante (per prendere visione del codice di questi file si rimanda al repository github sopra menzionato nel punto 1.2)

3.1) Cartella Admin/pages:

- Article
- Message
- Order
- Product
- Review
- Support
- Template-parts
- Client.php
- Index.php

Cartella *Article*:

[Admin/pages/article/delete.php](#)

[Admin/pages/article/edit-article.php](#)

[Admin/pages/article/index.php](#)

[Admin/pages/article/save.php](#)

[Admin/pages/article/articles.php](#)

```
<?php include "../../inc/init.php"; ?>
```

```
<?php if (!isset($_SESSION["email"]) || $_SESSION["admin"] == 1) {  
    header("Location: " . ROOT_URL);  
}  
?>
```

```
<?php include ROOT_PATH . "public/template-parts/title.php"; ?>
```

```
<title>Articoli</title>
```

```
<?php include ROOT_PATH . "public/template-parts/header.php"; ?>
```

```
<?php include ROOT_PATH . "admin/pages/template-parts/adminSidebar.php"; ?>
```

```
<?php  
    $articleMgr = new ArticleManager();  
  
    $articles = $articleMgr->getAll();  
  
    $articletxt = array();  
  
    foreach($articles as $a){  
        array_push($articletxt,substr($a->text,0,220));  
    }  
  
    $loader = new \Twig\Loader\FilesystemLoader(ROOT_PATH . '/templates/admin/  
article/');  
    $twig = new \Twig\Environment($loader, []);  
  
    echo $twig->render('articles.html', [  
        'ROOT_URL' => ROOT_URL,  
        'articles' => $articles,  
        'articletxt' => $articletxt  
    ]);  
?>
```

```
<?php include ROOT_PATH . "public/template-parts/footer.php"; ?>
```

Admin/pages/article/edit.php

```
<?php include "../././inc/init.php"; ?>
<?php if (!isset($_SESSION["email"]) || $_SESSION["admin"] == 1) {
    header("Location: " . ROOT_URL);
}
?>
<?php
$title = $_POST["title"];

$text = (string)$_POST["text"];

$img = (string)$_POST["img"];

$id = (int)$_POST["id"];

$articleMgr = new ArticleManager();
$articleMgr->updateArticle($id, $title, $text, $img);

// upload immagine
if (isset($_FILES)) {

    $tmp = $_FILES['image']['tmp_name'];
    $folder = ROOT_PATH . "public/img/slider/slide"; //cartella di destinazione dell'immagine
    $name = $id;
    $ext = ".png";

    if (is_uploaded_file($tmp)) {
        $uploadDir = $folder . $name . $ext; //aggiungo il folder di destinazione
        if (move_uploaded_file($tmp, $uploadDir)) {
            $articleMgr->updateImg($id);
        } else {
            echo "Non è stato possibile caricare l'immagine<br/>";
        }
    }
}

header("Location: ".ROOT_URL."admin/pages/article/edit-article.php?article=".$id);

?>
```

[Admin/pages/article/new-article.php](#)

```
<?php include "../../inc/init.php"; ?>
<?php
if (!defined('ROOT_URL')) {
    die;
}

if ($_SESSION["admin"] == 1) {
    header("Location: " . ROOT_URL);
}

?>
<?php include ROOT_PATH . "public/template-parts/title.php" ?>

<title>Nuovo articolo</title>

<?php include ROOT_PATH . 'public/template-parts/header.php'; ?>

<?php include ROOT_PATH . "admin/pages/template-parts/adminSidebar.php"; ?>

<?php

$manufacturerMgr = new ManufacturerManager();
$manufacturers = $manufacturerMgr->getAll();

$loader = new \Twig\Loader\FilesystemLoader(ROOT_PATH . '/templates/admin/article/');
$twig = new \Twig\Environment($loader, []);

echo $twig->render('new-article.html', [
    'ROOT_URL' => ROOT_URL,
    'manufacturers' => $manufacturers
]);

?>

<?php include ROOT_PATH . 'public/template-parts/footer.php'; ?>
```

Cartella *messege*:

[Admin/pages/messege/send.php](#)

[Admin/pages/massege/message.php](#)

```
<?php include "../../inc/init.php"; ?>
```

```

<?php if (!isset($_SESSION["email"]) || $_SESSION["admin"] == 1) {
    header("Location: " . ROOT_URL);
}
?>

<?php include ROOT_PATH . "public/template-parts/title.php"; ?>

<title>Messaggi</title>

<?php include ROOT_PATH . "public/template-parts/header.php"; ?>

<?php include ROOT_PATH . "admin/pages/template-parts/adminSidebar.php"; ?>

<?php
$chatMgr = new ChatManager();
$userMgr = new UserManager();
$messageMgr = new MessageManager();

$chat = $chatMgr->getAllChat();
$users = array();
foreach ($chat as $c){
    $user = $userMgr->get($c['user_id']);
    array_push($users, $user);
}

if(isset($_GET['chat'])){
    $chat_id = $_GET['chat'];
    $chatuser = $userMgr->get($chatMgr->get($chat_id)->user_id);

    $messages = $messageMgr->getMessage($chat_id);
}
else{
    $chat_id = null;
}

$loader = new \Twig\Loader\FilesystemLoader(ROOT_PATH . '/templates/admin/
message/');
$twig = new \Twig\Environment($loader, []);

echo $twig->render('message.html', [
    'ROOT_URL' => ROOT_URL,
    'users' => $users,
    'chat' => $chat,
    'chat_id' => $chat_id,
    'chatuser' => $chatuser,

```

```
'messages' => $messages  
  
]);  
?>
```

```
<?php include ROOT_PATH . "public/template-parts/footer.php"; ?>
```

Cartella *Order*:

[Admin/pages/order/order.php](#)

[Admin/pages/order/update-order.php](#)

[Admin/pages/order/single-order.php](#)

```
<?php include "../././inc/init.php"; ?>
```

```
<?php if (!isset($_SESSION["email"]) || $_SESSION["admin"] == 1) {  
    header("Location: " . ROOT_URL);  
}  
?>
```

```
<?php
```

```
include ROOT_PATH . "public/template-parts/title.php";
```

```
include ROOT_PATH . "public/template-parts/header.php";
```

```
include ROOT_PATH . "admin/pages/template-parts/adminSidebar.php";
```

```
if (!defined('ROOT_URL')) {  
    die;  
}
```

```
if ($orderId == null || !isset($_SESSION["email"]) || $_SESSION["userid"] !=  
$_ordine['user_id']) {  
    header("Location: " . ROOT_URL);  
}
```

```
$orderMgr = new OrdersManager();  
$orderAMgr = new OrderAddressManager();  
$orderItemMgr = new OrdersItemsManager();  
$productMgr = new ProductManager();  
$imgMgr = new ImgManager();
```

```

$orderID = $_GET["order"];

$order = $orderMgr->getOrderById($orderID);
$address = $orderAMgr->getAddressByOrder($orderID);

$date = substr($order['date_order'], 0, 10);
$yy = substr($date, 0, 4);
$mm = substr($date, 5, 2);
$dd = substr($date, 8, 10);
$date = $dd . "-" . $mm . "-" . $yy;
$sum = $orderMgr->getsum($order["id"]);

$date_ = substr($order['estimate_delivery'], 0, 10);
$yy_ = substr($date_, 0, 4);
$mm_ = substr($date_, 5, 2);
$dd_ = substr($date_, 8, 10);
$estimateDate = $dd_ . "/" . $mm_ . "/" . $yy_;

$listProduct = $orderItemMgr->getItems($orderID);

$products = array();
$imgs = array();

foreach($listProduct as $product){
    $p = $prodcutMgr->get($product['product_id']);
    $img = $imgMgr->get_thumbnail($product['product_id']);
    array_push($products, $p);
    array_push($imgs, $img);
}

$loader = new \Twig\Loader\FilesystemLoader(ROOT_PATH . 'templates/admin/order/');
$twig = new \Twig\Environment($loader, []);

echo $twig->render('single-order.html', [
    'ROOT_URL' => ROOT_URL,
    'order' => $order,
    'date' => $date,
    'sum' => $sum,
    'estimateDate' => $estimateDate,
    'address' => $address,
    'products' => $products,
    'quantity' => $listProduct,
    'imgs' => $imgs
]);

include ROOT_PATH . "public/template-parts/footer.php";

```


Cartella Product:

[Admin/pages/product/new-product.php](#)

[Admin/pages/product/save-new-product.php](#)

[Admin/pages/product/ save -product.php](#)

[Admin/pages/product/edit-product.php](#)

```
<?php include "../../inc/init.php"; ?>
<?php
if (!defined('ROOT_URL')) {
    die;
}

if ($_GET["product"] == null || $_SESSION["admin"] == 1 || !
isset($_SESSION["email"])) {
    header("Location: " . ROOT_URL . "admin/index.php");
}

$productid = $_GET["product"];
?>
<?php include ROOT_PATH . "public/template-parts/title.php" ?>

<?php $productmgr = new ProductManager();
$product = $productmgr->get($productid);

if ($product->name == "") {
    header("Location: " . ROOT_URL);
}

?>

<title>Modifica:<?php echo " " . $product->name; ?></title>

<?php include ROOT_PATH . 'public/template-parts/header.php'; ?>

<?php include ROOT_PATH . "admin/pages/template-parts/adminSidebar.php"; ?>

<?php

$imgMgr = new ImgManager();
$categoryItemMgr = new CategoryItemManager();
$categoryMgr = new CategoryManager();

$category = $categoryMgr->get($product->id);

$categories = $categoryItemMgr->get_category_id_by_product($product->id);
```

```

$categoriesS = array();

foreach ($categories as $category) {
    $results = $categoryMgr->getCategory($category['category_id']);
    foreach ($results as $result) {
        $string = $result['id'];
        array_push($categoriesS, $string);
    }
}

$imgs = $imgMgr->getAllImgForAdmin($product->id);

$loader = new \Twig\Loader\FilesystemLoader(ROOT_PATH . '/templates/admin/
product/');
$twig = new \Twig\Environment($loader, []);

echo $twig->render('edit-product.html', [
    'ROOT_URL' => ROOT_URL,
    'product' => $product,
    'imgs' => $imgs,
    'category' => $categoriesS,
    'allCategory' => $allCategory
]);
?>

<?php include ROOT_PATH . 'public/template-parts/footer.php'; ?>

```

Cartella *review*:

[Admin/pages/review/block.php](#)
[Admin/pages/review/unlock.php](#)
[Admin/pages/review/review.php](#)

```

<?php include "../../inc/init.php"; ?>

<?php if (!isset($_SESSION["email"]) || $_SESSION["admin"] == 1) {
    header("Location: " . ROOT_URL);
}
?>

<?php include ROOT_PATH . "public/template-parts/title.php"; ?>

<title>Recensioni</title>

```

```
<?php include ROOT_PATH . "public/template-parts/header.php"; ?>
```

```
<?php include ROOT_PATH . "admin/pages/template-parts/adminSidebar.php"; ?>
```

```
<?php
```

```
    $reviewMgr = new ReviewManager();  
    $productMgr = new ProductManager();  
    $userMgr = new UserManager();
```

```
    $reviews = $reviewMgr->getAllOrder();
```

```
    $products = array();
```

```
    $utenti = array();
```

```
    foreach($reviews as $review){  
        $product = $productMgr->getName($review['product_id']);  
        array_push($products, $product[0]['name']);  
  
        $users = $userMgr->getName($review['user_id']);  
        $user = $userMgr->getSurname($review['user_id']);  
        $stampa = $users[0]['name'].", ".$user[0]['surname'];  
        array_push($utenti, $stampa);  
    }
```

```
$loader = new \Twig\Loader\FilesystemLoader(ROOT_PATH . '/templates/admin/review/');
```

```
$twig = new \Twig\Environment($loader, []);
```

```
echo $twig->render('review.html', [  
    'ROOT_URL' => ROOT_URL,  
    'reviews' => $reviews,  
    'products' => $products,  
    'users' => $utenti
```

```
]);
```

```
?>
```

```
<?php include ROOT_PATH . "public/template-parts/footer.php"; ?>
```

Cartella *support*:

[Admin/pages/support/close.php](#)

[Admin/pages/support/open.php](#)

[Admin/pages/support/support.php](#)

[Admin/pages/support/ticket.php](#)

```
<?php include "../././inc/init.php"; ?>
```

```

<?php
if (!defined('ROOT_URL')) {
    die;
}

if ($_SESSION["admin"] == 1 || !isset($_SESSION["email"])) {
    header("Location: " . ROOT_URL . "admin/index.php");
}

$ticketid = $_GET["ticket"];

$supportMgr = new SupportManager();
$ticket = $supportMgr->get($ticketid);

?>

<?php include ROOT_PATH . "public/template-parts/title.php" ?>

<title>Ticket numero<?php echo " " . $ticket->id; ?></title>

<?php include ROOT_PATH . 'public/template-parts/header.php'; ?>

<?php include ROOT_PATH . "admin/pages/template-parts/adminSidebar.php"; ?>

<?php
$orderMgr = new OrdersManager();
$order = $orderMgr->get($ticket->order_id);

$answerMgr = new AnswerManager();
$answer = $answerMgr->getBySupport($ticket->id);
$answer = $answer[0]['message'];

$userMgr = new UserManager();
$user = $userMgr->get($ticket->user_id);

$imgpath = ROOT_URL . $user->img;

$counttickets = $supportMgr->countTicket($user->id);
$countticket = $counttickets[0]['nticket'];

$orderUser = $orderMgr->getOrder($user->id);
$countOrder = count($orderUser);

$loader = new \Twig\Loader\FilesystemLoader(ROOT_PATH . 'templates/admin/support');
$twig = new \Twig\Environment($loader, []);

echo $twig->render('ticket.html', [

```

```

        'ROOT_URL' => ROOT_URL,
        'ticket' => $ticket,
        'order' => $order,
        'answer' => $answer,
        'user' => $user,
        'countticket' => $countticket,
        'imgpath' => $imgpath,
        'countorder' => $countOrder

    ));

?>

<?php include ROOT_PATH . 'public/template-parts/footer.php'; ?>

```

Cartella *Template-parts*:

[Admin/pages/ template-parts /adminSidebar.php](#)

```

<?php include "../../inc/init.php"; ?>

<?php if (!isset($_SESSION["email"]) || $_SESSION["admin"] == 1) {
    header("Location: " . ROOT_URL);
}
?>
<?php
$request = $_SERVER['REQUEST_URI'];

if (substr($_SERVER['REQUEST_URI'], 0, 37) == "/FREDWebTechnology/admin/pages/
order/"){
    $request = substr($_SERVER['REQUEST_URI'], 0, 37);
}

if (substr($_SERVER['REQUEST_URI'], 0, 39) == "/FREDWebTechnology/admin/pages/
product/"){
    $request = substr($_SERVER['REQUEST_URI'], 0, 39);
}

if (substr($_SERVER['REQUEST_URI'], 0, 39) == "/FREDWebTechnology/admin/pages/
support/"){
    $request = substr($_SERVER['REQUEST_URI'], 0, 39);
}

if (substr($_SERVER['REQUEST_URI'], 0, 39) == "/FREDWebTechnology/admin/pages/
article/"){
    $request = substr($_SERVER['REQUEST_URI'], 0, 39);
}

```

```

}

if (substr($_SERVER['REQUEST_URI'], 0, 39) == "/FREDWebTechnology/admin/pages/
message/") {
    $request = substr($_SERVER['REQUEST_URI'], 0, 39);
}

$loader = new \Twig\Loader\FilesystemLoader(ROOT_PATH . '/templates/public/template-
parts/');
$twig = new \Twig\Environment($loader, []);

echo $twig->render('sidebarAdmin.html', [
    'ROOT_URL' => ROOT_URL,
    'request' => $request

]);
?>
Admin/pages/client.php
Admin/pages/index.php

```

3.2) Cartella *asest*:

- css
- js

Cartella *css*:

[Assest/css/bootstrap.css](#)

[Assest/css/style.css](#)

Cartella *js*:

[Assest/js/address.js](#)

[Assest/js/main.js](#)

[Assest/js/payment..js](#)

[Assest/js/review.js](#)

```

$(document).ready(function () {

    var addresses = jQuery.parseJSON($('#addresses').text());
    var regions = jQuery.parseJSON($('#regions').text());
    var provinces = jQuery.parseJSON($('#provinces').text());
    var cities = jQuery.parseJSON($('#cities').text());

    $('#modal').on('show.bs.modal', function (event) {

        var type = event.relatedTarget.getAttribute('data-whatever');
    });

```

```

if (type == "aggiungi") {
    $('#modalTitle').text('Inserisci un nuovo indirizzo');

    $("#regione").empty();
    $('#regione').append('<option value="" + 0 + "">' + 'Seleziona la regione' + '</option>');
    for (let i in regions) {
        $('#regione').append('<option value="" + regions[i].id + "">' + regions[i].name + '</option>');
    }

    $("#provincia").empty();
    $('#provincia').append('<option value="" + 0 + "">' + 'Seleziona la provincia' + '</option>');

    $("#comune").empty();
    $('#comune').append('<option value="" + 0 + "">' + 'Seleziona il comune' + '</option>');

    $('#cap').val("");
    $('#address').val("");

    $('#provincia').attr('disabled', true);
    $('#comune').attr('disabled', true);
    $('#cap').attr('disabled', true);
    $('#address').attr('disabled', true);

    $('#modalBtn').attr('name', 'add');
    console.log($('#modalBtn').attr('name'));
} else {
    var shipmentID = event.relatedTarget.getAttribute('data-bs-whatever');

    $('#modalTitle').text('Modifica il tuo indirizzo');

    $("#regione").empty();
    $('#regione').append('<option value="" + 0 + "">' + 'Seleziona la regione' + '</option>');
    for (let i in regions) {
        $('#regione').append('<option value="" + regions[i].id + "">' + regions[i].name + '</option>');
    }

    $('#provincia').attr('disabled', true);
    $('#comune').attr('disabled', true);
    $('#cap').attr('disabled', true);
    $('#address').attr('disabled', true);

```

```

    $('#flagRegion').text(1);
    $('#flagProvince').text(1);
    $('#flagCity').text(1);

    for (let i in addresses) {
        if (addresses[i].id_shipment == shipmentID) {
            $('#regione').append('<option selected value="" + addresses[i].region_id + "">' +
addresses[i].region_name + '</option>');
            $('#provincia').append('<option selected value="" + addresses[i].provinces_id +
"">' + addresses[i].provinces_name + '</option>');
            $('#comune').append('<option selected value="" + addresses[i].city + "">' +
addresses[i].city_name + '</option>');
            $('#cap').val(addresses[i].code);
            $('#address').val(addresses[i].address);
        }
    }
    $('#modalBtn').attr('name', 'update');
}
});

```

```

// popolamento della select delle province al settaggio della regione
$('#regione').on('change', function () {
    var value = $(this).val();
    $('#provincia').removeAttr('disabled');

    var flag = $('#flagRegion').text();
    console.log(flag);
    // verifica se è già stata settata una regione o meno
    if (flag == 0) {
        // aggiunta province
        for (let i in provinces) {
            if (provinces[i].region_id == value) {
                $('#provincia').append('<option value="" + provinces[i].id + "">' +
provinces[i].name + '</option>');
            }
        }
        $('#flagRegion').text(1);
    } else {
        // pulizia delle vecchie province
        $("#provincia").empty();
        $('#provincia').append('<option value="" + 0 + "">' + 'Seleziona la provincia' + '</
option>');
        // pulizia dei vecchi comuni
        $("#comune").empty();
        $('#comune').append('<option value="" + 0 + "">' + 'Seleziona il comune' + '</
option>');
    }
}

```



```

    $('#comune').attr('disabled', true);
    // pulizia del CAP
    $('#cap').val('');
    $('#cap').attr('disabled', true);
    // pulizia della via
    $('#address').val('');
    $('#address').attr('disabled', true);
    // aggiunta province
    for (let i in provinces) {
        if (provinces[i].region_id == value) {
            $('#provincia').append('<option value="' + provinces[i].id + '">' +
provinces[i].name + '</option>');
        }
    }
}
});

// popolamento della select dei comuni al settaggio della provincia
$('#provincia').on('change', function () {
    var value = $(this).val();
    $('#comune').removeAttr('disabled');

    var flag = $('#flagProvince').text();
    // verifica se è già stata settata una provincia o meno
    if (flag == 0) {
        // aggiunta comuni
        for (let i in cities) {
            if (cities[i].provinces_id == value) {
                $('#comune').append('<option value="' + cities[i].id + '">' + cities[i].name + '</
option>');
            }
        }
        $('#flagProvince').text(1);
    } else {
        // pulizia dei vecchi comuni
        $('#comune').empty();
        $('#comune').append('<option value="' + 0 + '">' + 'Seleziona il comune' + '</
option>');
        // pulizia del CAP
        $('#cap').val('');
        $('#cap').attr('disabled', true);
        // pulizia della via
        $('#address').val('');
        $('#address').attr('disabled', true);
        // aggiunta comuni
        for (let i in cities) {
            if (cities[i].provinces_id == value) {

```

```

        $('#comune').append('<option value=' + cities[i].id + '>' + cities[i].name + '</option>');
    }
}
});

$('#comune').on('change', function () {
    $('#cap').removeAttr('disabled');

    var flag = $('#flagCity').text();
    if (flag == 1) {
        // pulizia del CAP
        $('#cap').val("");
        // pulizia della via
        $('#address').val("");
        $('#address').attr('disabled', true);
        $('#flagCity').text(0);
    } else {
        $('#flagCity').text(1);
    }
});

$('#cap').on('change', function () {
    $('#address').removeAttr('disabled');
});

});

```

3.3) Cartella *classes*:

- Classes/Answer.php
- Classes/Article.php
- Classes/Cart.php
- Classes/Category.php
- Classes/CategoryItem.php
- Classes/Chat.php
- Classes/City.php
- Classes/Faq.php
- Classes/Img.php
- Classes/index.php
- Classes/Manufacturer.php
- Classes/Message.php
- Classes/Newsletter.php

- Classes/OrderAddress.php
- Classes/Orders.php
- Classes/OrdersItems.php
- Classes/Payment.php
- Classes/Product.php
- Classes/ProductCategory.php
- Classes/Provinces.php
- Classes/Region.php
- Classes/Review.php
- Classes/ShipmentInformation.php
- Classes/Support.php
- Classes/UserType.php
- Classes/Wishlist.php

-Classes/User.php

```
<?php
```

```
class User
```

```
{
```

```
    public $id;
```

```
    public $name;
```

```
    public $surname;
```

```
    public $email;
```

```
    public $password;
```

```
    public $img;
```

```
    public $user_type;
```

```
    public function __construct($id, $name, $surname, $email, $password, $img, $user_type)
```

```
    {
```

```
        $this->id = (int)$id;
```

```
        $this->name = $name;
```

```
        $this->surname = $surname;
```

```
        $this->email = $email;
```

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

```
        $this->img = $img;
```

```
        $this->user_type = $user_type;
```

```
    }
```

```
}
```

```
class UserManager extends DbManager
```

```
{
```

```
    public function __construct()
```

```

{
    parent::__construct();
    $this->columns = array('id', 'name', 'surname', 'email', 'password', 'img', 'user_type');
    $this->tableName = 'User';
}

public function login($email, $password)
{
    return $this->db->query("SELECT* FROM User WHERE email = '$email' AND password = '$password'");
}

public function register($name, $surname, $email, $password)
{
    $user = new User(0, $name, $surname, $email, $password, 'public/img/account.svg', 1);
    $userID = $this->create($user);

    return $userID;
}

//helper per verificare che un'email non sia già stata utilizzata
public function checkEmail($email)
{
    $result = $this->db->query("SELECT * FROM User WHERE email = '$email'");

    return count($result) > 0;
}

//helper per verificare che le password coincidono
public function checkPassword($password, $confPassword)
{
    return $password == $confPassword;
}

public function data($email)
{
    return $this->db->query("SELECT * FROM User WHERE email = '$email'");
}

public function getName($id)
{
    return $this->db->query("SELECT name, img FROM User WHERE id = $id");
}

public function getSurname($id)
{
    return $this->db->query("SELECT surname, img FROM User WHERE id = $id");
}

```

```

}

public function updateImg($userID, $email) {
    $url = "public/img/user/" . $email . ".png";
    return $this->db->execute("UPDATE User SET img = '$url' WHERE id = $userID");
}

/* DASHBOARD */

public function getClient()
{
    return $this->db->query("SELECT *FROM User ORDER BY name");
}

/* Aggiornamento */

public function updateUser($id,$name, $surname, $email, $password){
    $this->db->query("UPDATE User SET name = '$name', surname = '$surname', email
    ='$email', password = '$password' WHERE id = $id");
}
}

```

[-Classes/DB.php](#)

<?php

```

class DB {

    private $conn;
    public $pdo;

    public function __construct() {
        global $conn;
        $this->conn = $conn;
        if (mysqli_connect_errno()) {
            echo 'Failed to connect to MySQL ' . mysqli_connect_errno();
            die;
        }
        $this->pdo = new PDO('mysql:dbname='. DB_NAME .';host=' . DB_HOST, DB_USER,
        DB_PASS);
        $this->pdo->setAttribute( PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION );
    }

    public function query($sql) {
        try
        {
            $q = $this->pdo->query($sql);
            if (!$q){
                throw new Exception("Error executing query...");
            }
        }
    }
}

```

```

    }
    $data = $q->fetchAll();
    return $data;
}
catch(Exception $e) {
    throw $e;
}
}

```

```

public function execute($sql) {
    $stmt = $this->pdo->prepare($sql);
    $stmt->execute();
}

```

```

public function select_all($tableName, $columns = array()) {

```

```

    $query = 'SELECT ';
    $strCol = '';

```

```

    foreach($columns as $colName) {
        $strCol .= ' '. esc($colName) . ',';
    }
    $strCol = substr($strCol, 0, -1);

```

```

    $query .= $strCol . ' FROM ' . $tableName;

```

```

    $result = mysqli_query($this->conn, $query);
    $resultArray = mysqli_fetch_all($result, MYSQLI_ASSOC);

```

```

    mysqli_free_result($result);

```

```

    return $resultArray;
}

```

```

public function select_one($tableName, $columns = array(), $id) {

```

```

    $strCol = '';
    foreach($columns as $colName) {
        $colName = esc($colName);
        $strCol .= ' '. $colName . ',';
    }
    $strCol = substr($strCol, 0, -1);
    $id = esc($id);
    $query = "SELECT $strCol FROM $tableName WHERE id = $id";

```

```

    $result = mysqli_query($this->conn, $query);

```

```

$resultArray = mysqli_fetch_assoc($result);

mysqli_free_result($result);

return $resultArray;
}

public function delete_one($tableName, $id) {

    $id = esc($id);
    $query = "DELETE FROM $tableName WHERE id = $id";

    if (mysqli_query($this->conn, $query)) {
        $rowsAffected = mysqli_affected_rows($this->conn);

        return $rowsAffected;
    } else {

        return -1;
    }
}

public function update_one($tableName, $columns = array(), $id) {

    $id = esc($id);
    $strCol = "";
    foreach($columns as $colName => $colValue) {
        $colName = esc($colName);
        $strCol .= " " . $colName . " = '$colValue' ,";
    }
    $strCol = substr($strCol, 0, -1);

    $query = "UPDATE $tableName SET $strCol WHERE id = $id";
    $query = str_replace("'NULL'", "NULL", $query);

    if (mysqli_query($this->conn, $query)) {
        $rowsAffected = mysqli_affected_rows($this->conn);

        return $rowsAffected;
    } else {

        return -1;
    }
}

public function insert_one ($tableName, $columns = array()) {

```

```

$strCol = "";
foreach($columns as $colName => $colValue) {
    $colName = esc($colName);
    $strCol .= ' ' . $colName . ',';
}
$strCol = substr($strCol, 0, -1);

$strColValues = "";
foreach($columns as $colName => $colValue) {
    $colValue = esc($colValue);
    $strColValues .= " '" . $colValue . "' ,";
}
$strColValues = substr($strColValues, 0, -1);

$query = "INSERT INTO $tableName ($strCol) VALUES ($strColValues)";

if (mysqli_query($this->conn, $query)) {
    $lastId = mysqli_insert_id($this->conn);

    return $lastId;
} else {

    return -1;
}
}
}

```

```

class DBManager {

    protected $db;
    protected $columns;
    protected $tableName;

    public function __construct(){
        $this->db = new DB();
    }

    public function get($id) {
        $resultArr = $this->db->select_one($this->tableName, $this->columns, (int)$id);
        return (object) $resultArr;
    }

    public function getAll() {
        $results = $this->db->select_all($this->tableName, $this->columns);
        $objects = array();
    }
}

```



```

    foreach($results as $result) {
        array_push($objects, (object)$result);
    }
    return $objects;
}

public function create($obj) {
    $newId = $this->db->insert_one($this->tableName, (array) $obj);
    return $newId;
}

public function delete($id) {
    $rowsDeleted = $this->db->delete_one($this->tableName, (int)$id);
    return (int) $rowsDeleted;
}

public function update($obj, $id) {
    $rowsUpdated = $this->db->update_one($this->tableName, (array) $obj, (int)$id);
    return (int) $rowsUpdated;
}
}

```

3.4)Cartella public/auth:

-public/auth/pages

- public/auth/index.php

-login-action.php

<?php

include "../inc/init.php";

```

if (isset($_SESSION["email"])) {
    header("Location: " . ROOT_URL);
}

```

include ROOT_PATH . "public/template-parts/title.php";

include ROOT_PATH . "public/template-parts/header.php";

```

$email = $_POST["email"];
$password = $_POST["password"];

```

\$passHash = hash("sha1", \$password);

```

$userMgr = new UserManager();
$login = $userMgr->login($email, $passHash);

```

```

if ($login == null) {
    $error = true;

    $loader = new \Twig\Loader\FilesystemLoader(ROOT_PATH . 'templates/public/auth');
    $twig = new \Twig\Environment($loader, []);

    echo $twig->render('login.html', [
        'ROOT_URL' => ROOT_URL,
        'error' => $error,
    ]);
} else {
    $_SESSION["email"] = $email;
    foreach ($login as $log) {
        $_SESSION["img"] = $log["img"];
        $_SESSION["admin"] = $log["user_type"];
        $_SESSION["name"] = $log["name"];
        $_SESSION["userid"] = $log["id"];
    }
    header("Location: " . ROOT_URL . "shop");
}

include ROOT_PATH . "public/template-parts/footer.php";

```

-register-action.php

```

<?php
include "../inc/init.php";

if (isset($_SESSION["email"])) {
    header("Location: " . ROOT_URL);
}

$userMgr = new UserManager();

$name = $_POST["name"];
$surname = $_POST["surname"];
$email = $_POST["email"];
$password = $_POST["password"];
$confPassword = $_POST["confPassword"];

$passHash = hash("sha1", $password);
$confPassHash = hash("sha1", $confPassword);

// creazione del nuovo utente
$userID = $userMgr->register($name, $surname, $email, $passHash);

```

```

if ($userID > 0) {
    // login automatico dell'utente appena creato
    $login = $userMgr->login($email, $passHash);

    $_SESSION["email"] = $email;
    foreach ($login as $log) {
        $_SESSION["img"] = $log["img"];
        $_SESSION["admin"] = $log["user_type"];
        $_SESSION["name"] = $log["name"];
        $_SESSION["userid"] = $log["id"];
    }
    // creazione e sincronizzazione dei carrelli
    $cartMgr = new CartManager();

    $cartID = $cartMgr->createCart();
    $currentCartID = $cartMgr->getCurrentCartId();
    if ($cartID != $currentCartID) {
        $cartMgr->mergeCarts($cartID, $currentCartID);
    }
    // redirect alla home
    header("Location: " . ROOT_URL . "shop");
} else {
    /*
    $error = "Errore durante la registrazione, riprova più tardi!";
    */
}

```

3.5) [Cartella public/img](#)

3.6) [Cartella public/pages](#)

3.7) [Cartella public/template-parts](#)

-Footer.php

-Header.php

-Single-page.php

- Title.php

-Sidelbar.php

<?php

include "../inc/init.php";

```

if (isset($_SESSION['email'])) {
    $user1 = $_SESSION['name'];
}

```

```

}
$userMgr = new UserManager();
$user = $userMgr->data($_SESSION["email"]);

$addresMgr = new ShipmentInformationManager();
$regioneMgr = new RegionManager();
$provinciaMgr = new ProvincesManager();
$cittaMgr = new CityManager();
$cartMgr = new CartManager();

$indirizzi = $addresMgr->getFatturazionePrincipale($user[0]['id']);
$regione = $regioneMgr->getRegione($indirizzi[0]['region']);
$provincia = $provinciaMgr->getProvincia($indirizzi[0]['province']);
$citta = $cittaMgr->getCitta($indirizzi[0]['city']);
if (isset($_SESSION['userid'])) {
    $cartID = $cartMgr->findCart($_SESSION['userid']);
} else {
    $cartID = $cartMgr->getCart($_SESSION['client_id']);
}

if ($cartID) {
    $quantity = $cartItemMgr->countTotalItem($cartID);
}
$loader = new \Twig\Loader\FilesystemLoader(ROOT_PATH . 'templates/public/template-parts');
$twig = new \Twig\Environment($loader, []);

echo $twig->render('sidebar.html', [
    'ROOT_URL' => ROOT_URL,
    'user' => $user,
    'indirizzi' => $indirizzi,
    'citta' => $citta,
    'provincia' => $provincia,
    'regione' => $regione,
    'quantity' => $quantity,
]);

```

3.7) Cartella *shop*

- /pages
- article.php
- cart-action.php
- Checkout-action.php
- index.php
- review-action.php
- search.php

-single-product.php

-wishlist_action.php

```
<?php include "../inc/init.php";

if (!isset($_SESSION["email"])) {
    header("Location: " . ROOT_URL);
}

$wishlistMgr = new WishlistManager();

// AGGIUNTA ALLA LISTA
if (isset($_POST["wishlist"])) {

    $userid = $_SESSION["userid"];
    $productID = $_POST["wishlist"];

    $wishlistMgr->addProduct($userid, $productID);

    $productID = $_POST['productID'];
    header('Location: ' . ROOT_URL . 'shop/single-product.php?product=' . $productID);
}

// AGGIUNTA AL CARRELLO
if (isset($_POST['addToCart'])) {

    $productID = $_POST['addToCart'];

    // addToCart Logic
    $cartMgr = new CartManager();

    if (isset($_SESSION['userid'])) {
        $cartID = $cartMgr->findCart($_SESSION['userid']);
    } else {
        $cartID = $cartMgr->getCurrentCartId();
    }
    // aggiungi al carrello "cartID" il prodotto "productID"
    $cartMgr->addToCart($productID, $cartID);

    header("Location: " . ROOT_URL . "shop/pages/wishlist.php");
}

// RIMOZIONE DALLA LISTA
if (isset($_POST['rimuovi'])) {
    $productID = $_POST['rimuovi'];

    $wishlistMgr->deleteByProductId($productID, $_SESSION["userid"]);
}
```

```

    header("Location: " . ROOT_URL . "shop/pages/wishlist.php");
}

```

3.8) Cartella *templates*.

-/admin

-/public

-/user

-index.html

-/shop

-/shop/search.html

```

<form action="{{ROOT_URL}}shop/search.php" method="POST">
  <div class="container-fluid">
    <div class="row">
      <nav id="sidebarMenu" class="col-md-3 col-lg-2 d-md-block sidebar collapse ">

        <div class="position-sticky pt-3" style="margin-top: 65px;">
          <ul class="nav flex-column">
            <fieldset class="form-group">
              <legend class="mt-4 text-primary">
                <h4><strong>Filtra per categoria</strong></h4>
              </legend>
              {% for row in category %}
                <li class="nav-item">
                  <div class="form-check">
                    <input type="hidden" value="{{search}}" id="search" value="search">
                    <input class="form-check-input" type="checkbox"
value="{{row.name}}"
                        id="selezionato[]" name="selezionato[]" {% if row.name in selected
%} checked {%
                        endif %}>
                    <label class="form-check-label" for="flexCheckDefault">
                      <h5>{{row.name}}</h5>
                    </label>
                  </div>
                </li>
              {% endfor %}
              <br>
              <button type="submit" class="btn btn-primary" style="width:
70%;">Filtra</button>
            </fieldset>
          </ul>
        </div>
      </form>
    </nav>
    <input type="hidden" id="search" name="search" value="{{search}}">

```

```

<main class="col-md-8 col-lg-8">

  <div
    class="d-flex justify-content-between flex-wrap flex-md-nowrap align-items-center pt-3
pb-2 mb-3 border-bottom">
    <h2><strong class="text-primary">"{{search}}"</strong></h2>
  </div>
  <div class="row">
    {% if product or provvisori %}
    {% for m in product %}
    {% set c = ROOT_URL ~ m.img %}
    {% set link = redirect ~ m.id %}
    <div class="col-md-3 clearfix d-none d-md-block" style="cursor:pointer;"
onclick="location.href = '{{link}}';">
      <div class="card mb-2">
        
        <div class="card-body">
          <h4 class="card-title">{{m.name}}</h4>
          <p class="card-text">{{m.small_description}}</p>
          <p class="card-text"><strong>{{m.price}}€</strong></p>
          <form method="POST" action="{{ROOT_URL}}shop/cart-action.php">
            <input type="hidden" value="{{REQUEST_URI}}" id="uri" name="uri">
            <input type="hidden" value="{{search}}" id="s" name="s">
            <button class="btn btn-primary btn-sm" name="product_id" type="submit"
value={{m.id}}><svg
xmlns="http://www.w3.org/2000/svg" width="16" height="16"
fill="currentColor"
class="bi bi-cart-plus-fill" viewBox="0 0 16 16">
              <path
d="M.5 1a.5.5 0 0 0 1.11l.401 1.607 1.498 7.985A.5.5 0 0 0 4
12h1a2 2 0 1 0 0 4 2 2 0 0 0 0-4h7a2 2 0 1 0 0 4 2 2 0 0 0 0-4h1a.5.5 0 0 0
.491-.408l1.5-8A.5.5 0 0 0 14.5 3H2.89l-.405-1.621A.5.5 0 0 0 2 1H.5zM6 14a1 1 0 1 1-2 0 1
1 0 0 1 2 0zm7 0a1 1 0 1 1-2 0 1 1 0 0 1 2 0zM9 5.5V7h1.5a.5.5 0 0 1 0 1H9v1.5a.5.5 0 0 1-1
0V8H6.5a.5.5 0 0 1 0-1H8V5.5a.5.5 0 0 1 1 0z" />
              </svg></button>
            </form>
          </div>
        </div>
      </div>
    {% endfor %}
    {% else %}
    <h5>Nessun prodotto trovato</h5>
    {% endif %}
  </div>
</main>
</form>

```

3.8)Cartella *user*:

- account.php
- accountUpdate.php
- address-action.php
- index.php
- invoice.php
- notification.php
- order-detalis.php
- payment-action.php
- payment.php
- support-action.php
- support.php
- ticket-support.php
- order.php

```
<?php
include "../inc/init.php";
if (!isset($_SESSION["email"])) {
    header("Location: " . ROOT_URL);
}

include ROOT_PATH . "public/template-parts/title.php";

include ROOT_PATH . "public/template-parts/header.php";

$orderMgr = new OrdersManager();
$orderAMgr = new OrderAddressManager();
$orderIMgr = new OrdersItemsManager();
$productMgr = new ProductManager();
$imgMgr = new ImgManager();

$orders = $orderMgr->getOrder($_SESSION['userid']);

for ($i = 0; $i < count($orders); $i++) {
    $items[$i] = $orderIMgr->getItems($orders[$i]['id']);
    $date = substr($orders[$i]['date_order'], 0, 10);
    $yy = substr($date, 0, 4);
    $mm = substr($date, 5, 2);
    $dd = substr($date, 8, 10);
    $dates[] = $dd . "-" . $mm . "-" . $yy;
    $sums[] = $orderMgr->getsum($orders[$i]['id']);
}

$size = count($orders);
```