

Customizarea unui magazin de comert electronic

folosind platforma Prestashop



Despre proiect

Proiectul se bazeaza pe configurarea unui site pentru comert electronic folosind platforma Prestashop. Prestashop pune la dispozitia utilizatorilor un template pe care acestia pot sa creeze, dezvolte si sa aduca modificari.

Site-ul va gazdui un magazin pentru vanzarea articolelor de bricolaj.



Prestashop

Prestashop este o platforma de comert electornic open-source bazata pe PHP. Este usor de utilizat, lucru care se datoreaza in primul rand faptului ca platforma dispune de diferite template-uri care pot fi descarcate. Pot fi adaugate module predefinite pentru a customiza site-ul. De asemenea, pot fi adaugate feature-uri noi sau modificate cele deja existente.

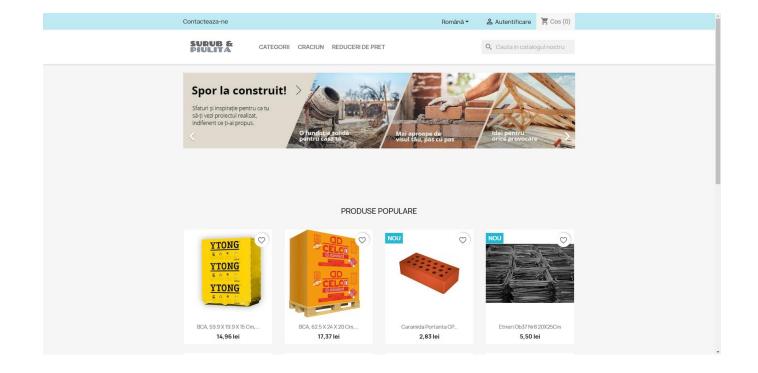
Prestashop este alcatuit din 2 parti principale:

- □ Front-office
- Back-office

Front-office

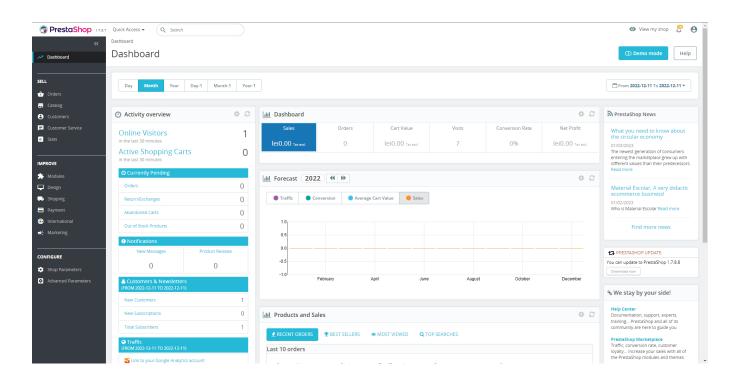
Partea de front-office este ceea ce clientii vor vedea.

Prestashop ofera un template default pentru site-ul nostru, cu un set de functionalitati predefinite. Noi putem adauga sau elimina diferite elemente sau functionalitati din partea de administrator.



Back-office

Partea de back-office a magazinului nostru este locul unde vom gestiona tot continutul, unde vom publica produsele si unde vom determina cum arata site-ul nostru. Este de asemenea locul unde vom gestiona si comenzile primite.



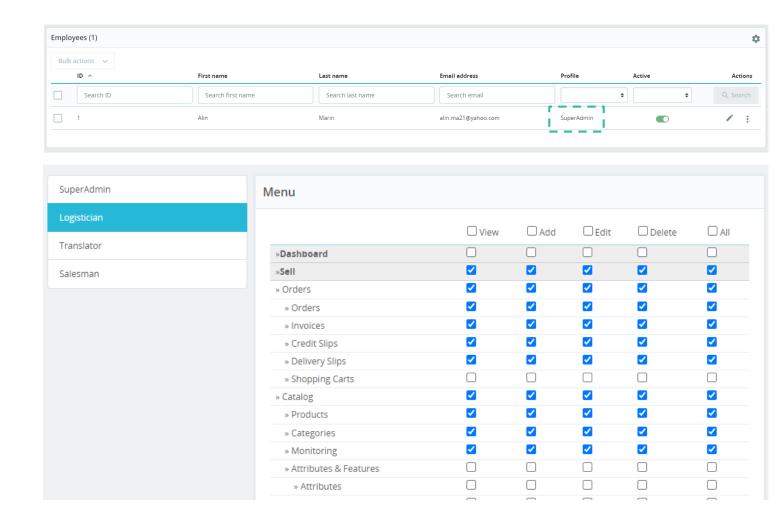
Roluri

Pentru partea de roluri este evident ca front-office este pentru clienti, iar dashboard-ul este destinat administratiei.

Clientii sunt impartiti in utilizatori si vizitatori, in functie de statutul de autentificare.

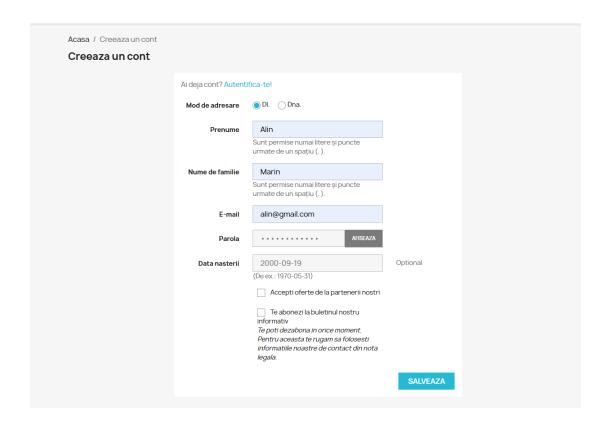
Si in partea de administrare putem avea mai multe roluri. Toti administatorii site-ului pot fi vazuti in view-ul "Team". Putem observa ca este deja inregistrat un SuperAdmin. SuperAdmin-ul este persoana care a creat site-ul.

Putem sa avem mai multe profiluri de angajati, fiecare profil avand permisiuni specifice de gestionare a resurselor site-ului.



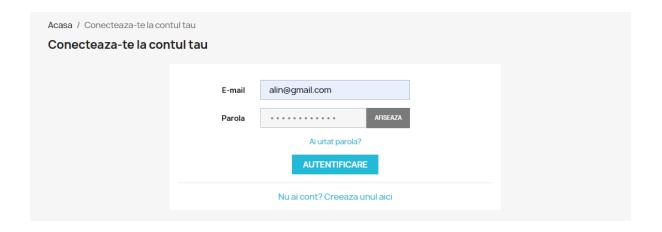
Crearea unui cont

Pentru a putea sa aiba acces la toata functionalitatea site-ului, mai ales la cea de plasare de comenzi, este necesara crearea unui cont prin completarea unor credentiale.



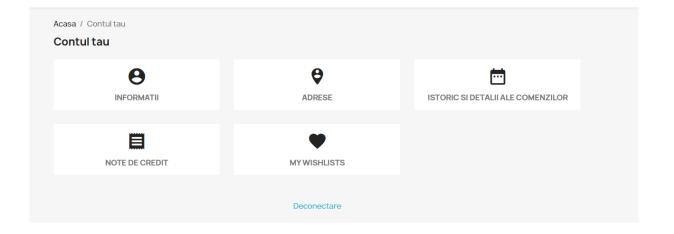
Autentificarea

Autentificarea se face pe baza email-ului si a parolei introduse la inregistrare.



Pagina profilului

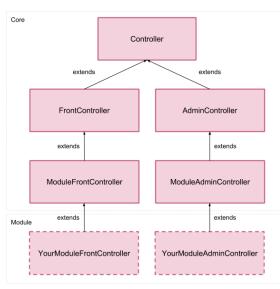
Fiecare utilizator are propria pagina a profilului, unde poate sa isi gestioneze informatiile personale, sa adauge adresele care vor fi folosite ca punct pentru livrare, sa verifice comenziile plasate de acesta sau sa vizualizeze lista de dorinte.



Controllere

Toate paginile prezentate mai devreme, de la cele de autentificare, pana la cele ce gestionare a profilului, sunt construite folosind controllere.

Atunci cand un modul trebuie sa implementeze mai mult decat poate oferi o pagina de configurare, controllerele vor avea grija de acest aspect. Intr-o arhitectura MVC, un Controller gestioneaza evenimentele de sincronizare intre View si Model. Acesta primeste toate evenimentele utilizatorului si declanseaza actiunile ce trebuie efectuate.



```
class AuthControllerCore extends FrontController

{    class AuthControllerCore extends
}

public Salt = true;
public function checkAccess()

}

class AuthControllerCore extends

{    class AuthControllerCore
public function checkAccess()

}

class AuthControllerCore
public function (context)

{    class AuthContextCore
public function (context)
}

class AuthControllerCore
public function into Context()

{    class AuthContextCore
public function (context)
}

class AuthContextCore
public function (context
```

Template-uri

Sunt fisiere cu extensia .tpl care determina aspectul unei pagini. Salveaza setarile implicite pentru aspectul paginii unui document si alte formatari. Fisierele tpl sunt folosite pentru a crea documente cu un stil si o structura similara.

Sunt definite blocuri ce vor fi folosite in mai multe parti documente, in interiorul carora poate fi adaugat cod HTML/CSS/JS.

```
{extends file=$layout}
{block name='content'}
 <section id="main">
    {block name='page header container'}
      {block name='page title' hide}
       <header class="page-header">
         <h1>{$smarty.block.child}</h1>
       </header>
     {/block}
    {/block}
    {block name='page content container'}
     <div id="content" class="page-content card card-block">
        {block name='page_content_top'}{/block}
       {block name='page content'}
         <!-- Page content -->
       {/block}
     </div>
    {/block}
    {block name='page footer container'}
     <footer class="page-footer">
       {block name='page footer'}
         <!-- Footer content -->
       {/block}
     </footer>
   {/block}
 </section>
{/block}
```

Formulare

Formularele sunt tot niste fisiere de tip template create cu scopul de a configura niste formulare standard ce vor fi apelate de catre controllere.

```
{elseif $field.type === 'password'}
  {block name='form field item password'}
   <div class="input-group js-parent-focus">
     <input
       id="field-{$field.name}"
       class="form-control js-child-focus js-visible-password"
       name="{$field.name}"
       title="{| s='At least 5 characters long' d='Shop.Forms.Help'}"
       aria-label="{| s='Password input of at least 5 characters' d='Shop.Forms.Help'}"
       type="password"
       {if $field.autocomplete}autocomplete="{$field.autocomplete}"{/if}
       pattern=".{literal}{{/literal}5,{literal}}{/literal}"
       {if $field.required}required{/if}
     <span class="input-group-btn">
       <button
         class="btn"
         type="button"
         data-action="show-password"
         data-text-show="{| s='Show' d='Shop.Theme.Actions'}"
         data-text-hide="{l s='Hide' d='Shop.Theme.Actions'}"
          {l s='Show' d='Shop.Theme.Actions'}
       </button>
     </span>
   </div>
  {/block}
```

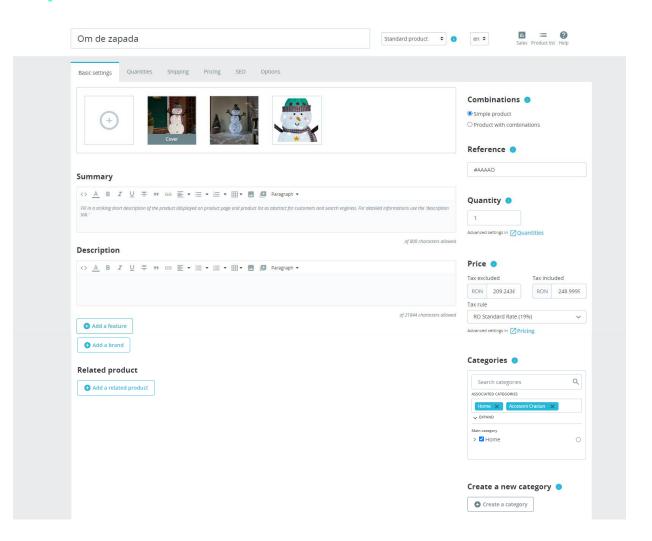
Formulare

Formularele sunt tot niste fisiere de tip template create cu scopul de a configura niste formulare standard ce vor fi apelate de catre controllere.

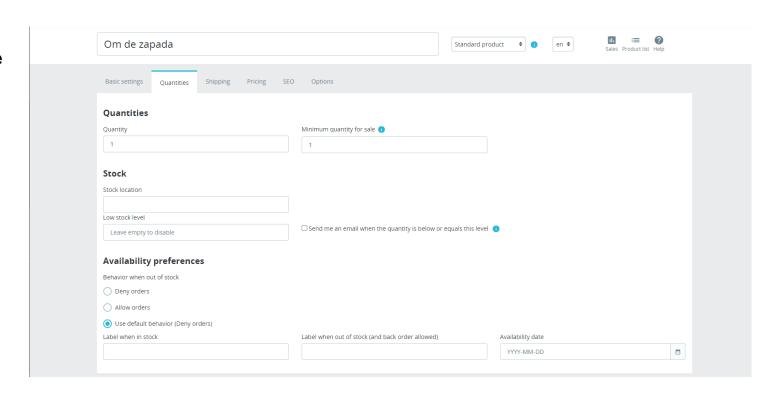
```
{elseif $field.type === 'password'}
  {block name='form field item password'}
   <div class="input-group js-parent-focus">
     <input
       id="field-{$field.name}"
       class="form-control js-child-focus js-visible-password"
       name="{$field.name}"
       title="{| s='At least 5 characters long' d='Shop.Forms.Help'}"
       aria-label="{| s='Password input of at least 5 characters' d='Shop.Forms.Help'}"
       type="password"
       {if $field.autocomplete}autocomplete="{$field.autocomplete}"{/if}
       pattern=".{literal}{{/literal}5,{literal}}{/literal}"
       {if $field.required}required{/if}
     <span class="input-group-btn">
       <button
         class="btn"
         type="button"
         data-action="show-password"
         data-text-show="{| s='Show' d='Shop.Theme.Actions'}"
         data-text-hide="{l s='Hide' d='Shop.Theme.Actions'}"
          {l s='Show' d='Shop.Theme.Actions'}
       </button>
     </span>
   </div>
  {/block}
```

Prestashop faciliteaza prin intermediul back-office-ului gestionarea produselor. Astfel este mai usor de creat sau editat un produs completand campurile puse la dispozitie acolo.

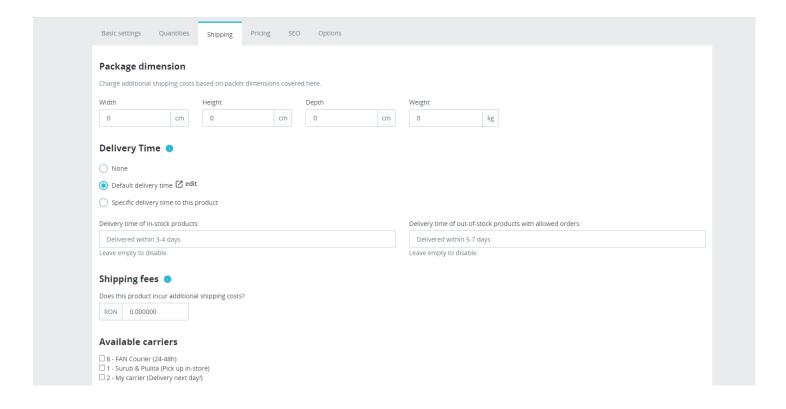
In field-ul de Basic Settings configuram aspectele de baza ale produsului, cum ar fi alegerea numelui pentru produs, adaugarea imaginilor, descrierea produsului. Tot aici putem sa stabilim un id al produsului, cantitatea si pretul sau sa adaugam produsul la categoriile din care face parte.



Al doilea field este cel dedicat cantitatii si stocului, unde putem configura cate produse avem in stoc sau care este cantitatea maxima de produse care pot fi puse la vanzare.

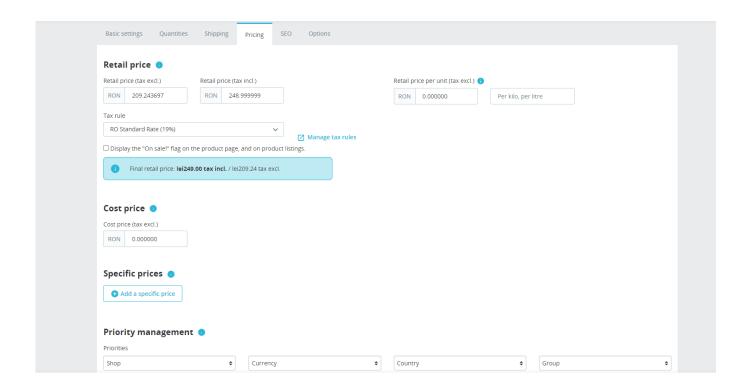


Urmatorul field este cel dedicat livrarii. Aici putem specifica dimensiunile si greutatea coletului, putem sa specificam timpul de livrare, daca este standard sau implica alte modificari de timp, costuri aditionale de transport sau diferite moduri de livrare.



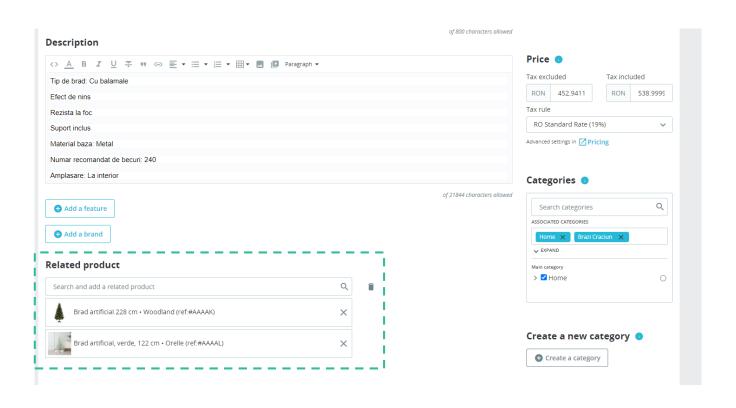
In field-ul Pricing sunt puse la dispozitie preturile produsului.

Putem configura pretul cu sau fara taxe, modul de calculare a taxelor sau putem specifica costul de producere, pentru a putea vedea profitul facut in urma vanzarii.

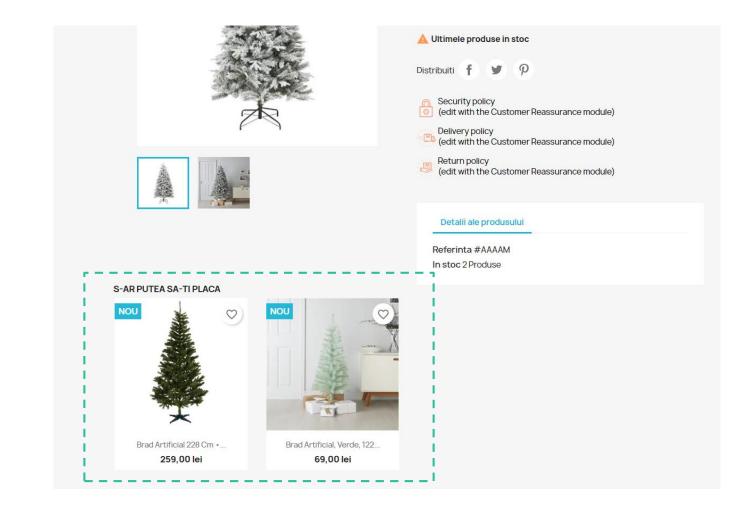


Produse de acelasi tip

In campul Related products putem adauga produse care s-ar putea sa il intereseze pe clientul atunci cand vizioneaza un anumit produs.

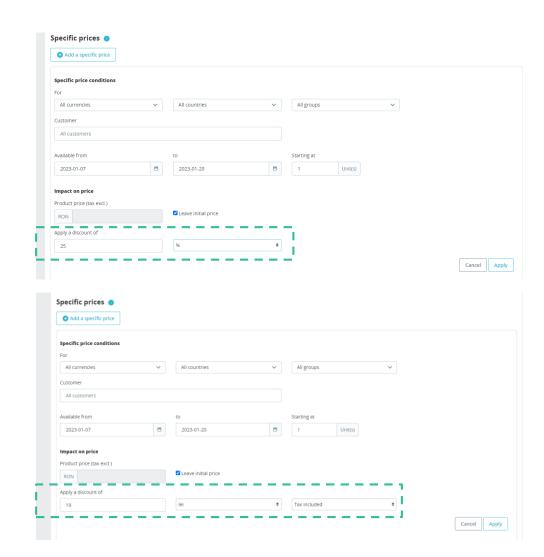


Produse de acelasi tip

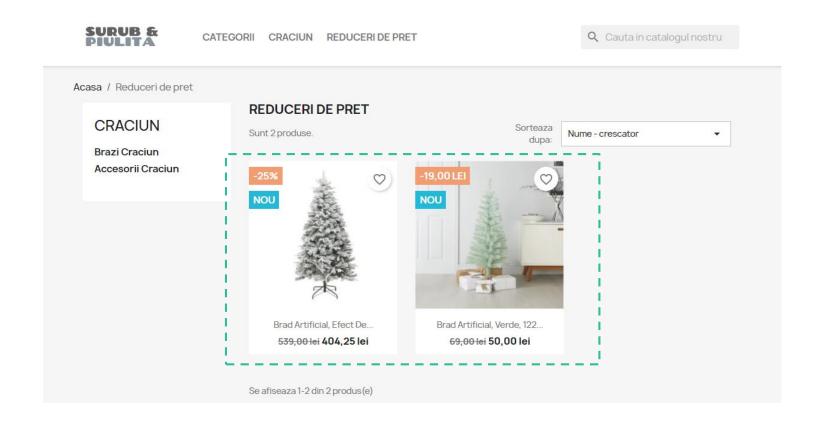


Reducerea pretului pentru un anumit produs

Putem adauga o reducere pentru un produs in campul Specific prices. Putem edita reducerea ce va fi aplicata, daca va fi un procent redus sau o suma de bani si de asemenea si data in care se va aplica reducerea.

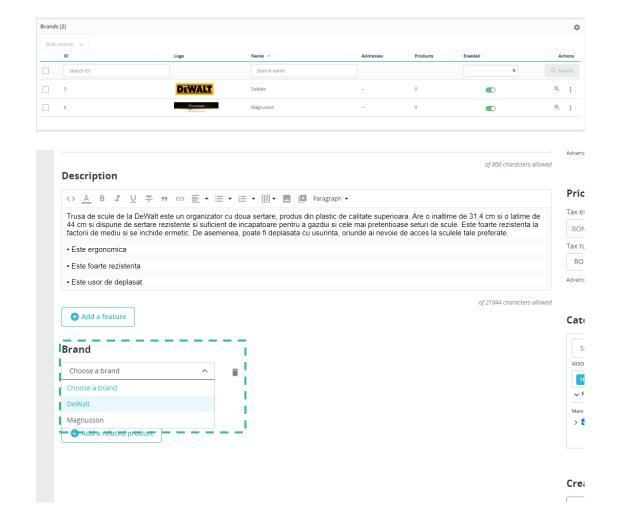


Reducerea pretului pentru un anumit produs

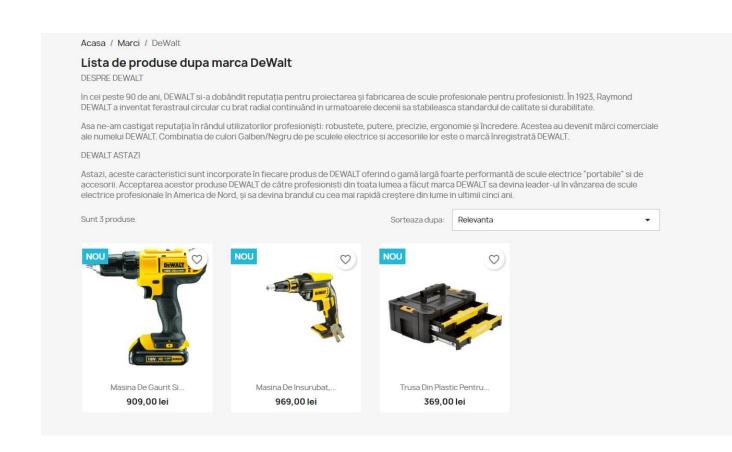


Adaugarea unui brand asociat produsului

Produsele pot fi asociate brand-urilor care le produc. Pentru aceasta este necesar sa adaugam brand-urile mai intai.



Adaugarea unui brand asociat produsului



Adaugarea produsului pe site

43

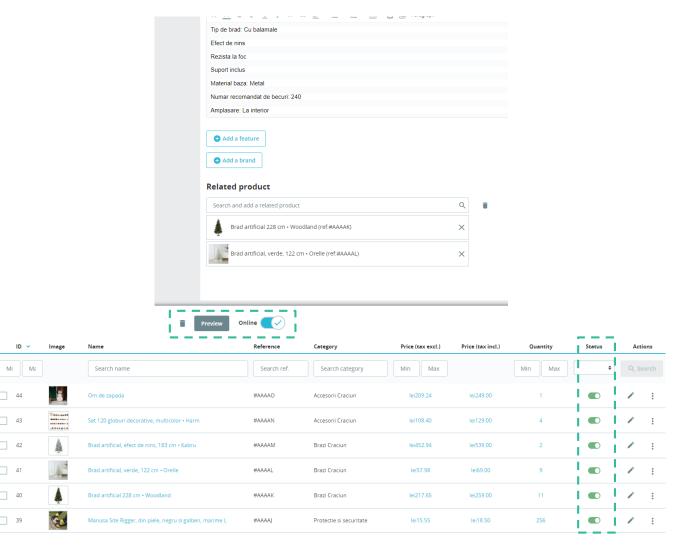
42

40

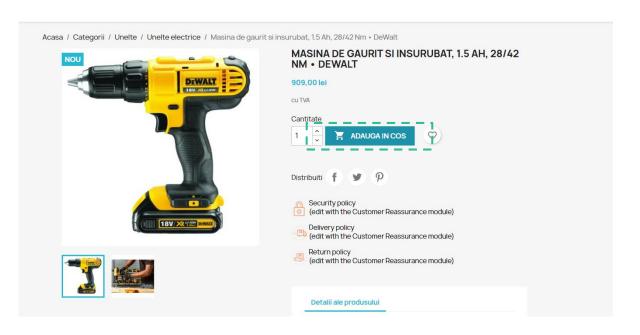
39

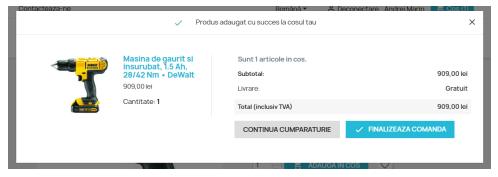
Crearea produsului nu il adauga automat pe site. Produsele pot fi configurate ca active sau nu. Clientii vor putea vizualiza doar produsele care sunt activate.

Acest lucru se poate face in 2 moduri.



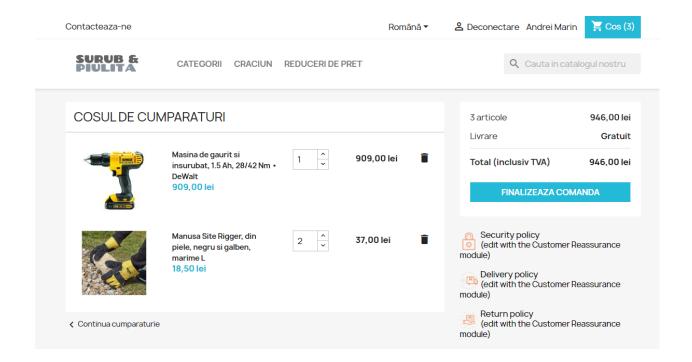
De fiecare data cand adaugam un produs in cos apare un pop-up care ne spune ca operatia s-a realizat cu succes.

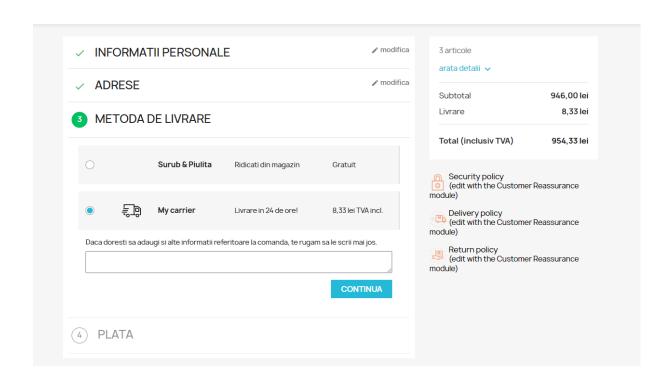


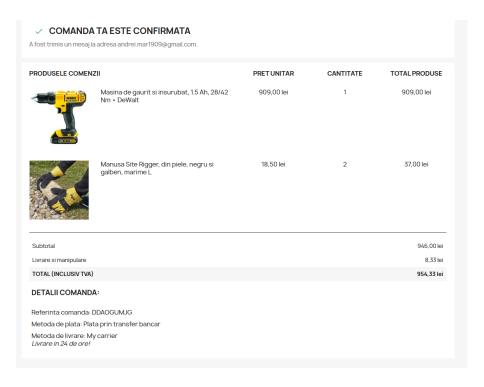


Butonul "Finalizeaza comanda" ne trimite pe pagina unde trebuie sa completam datele comenzii: adresa, metoda de livrare si metoda de plata.

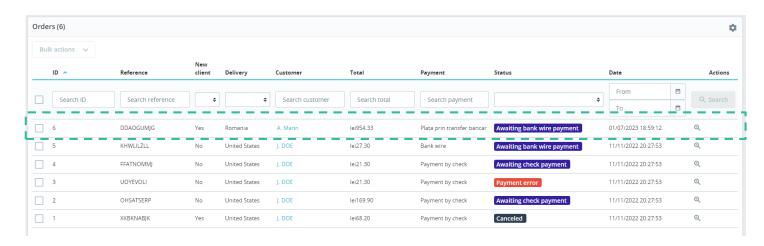
Dupa ce a fost totul completat si clientul plaseaza comanda, acesta va primi o confirmare.







In back-office putem observa cum comanda a fost inregistrata, iar statusul acesteia ne spune ca plata prin virament bancar este asteptata.

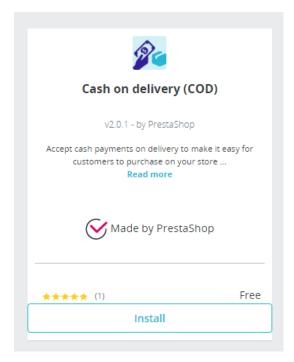


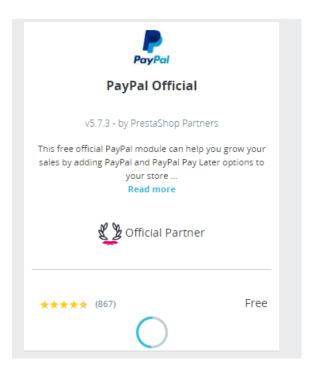
Instalarea unui modul nou

Template-ul default oferit de Prestashop ofera doar 2 metode posibile de plata: prin cec si prin transfer bancar.

Putem totusi sa instalam alte module puse la dispozitie de dezvoltatori pentru a adauga si alte metode de plata.

Vom opta pentru adaugarea platii la livrare si plata prin PayPal.





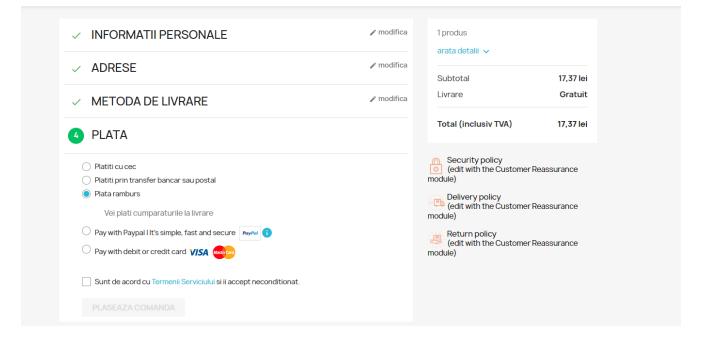
Instalarea unui modul nou

Se observa ca acum avem mai multe optiuni de plata.

Plata folosind PayPal este configurata, dar nu poate fi folosita deoarece site-ul nu suporta momentan HTTPS, nu avem un certificat SSL valid !!!



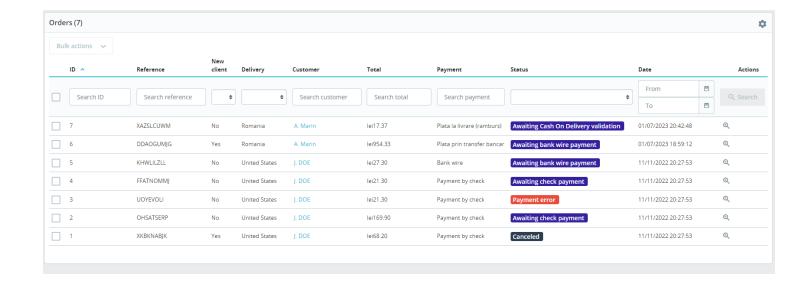
Contacteaza-ne



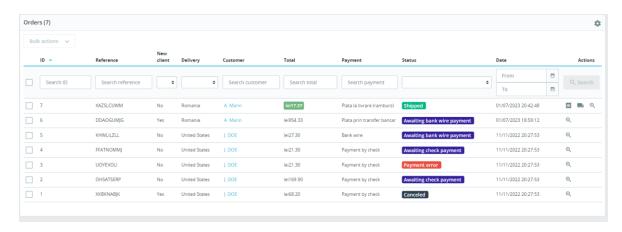
Gestionarea comenzilor

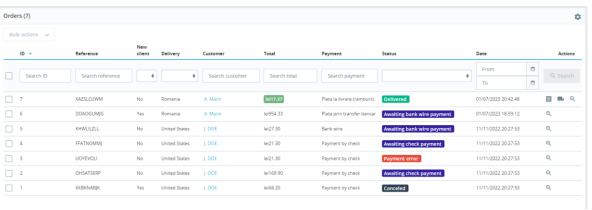
Atunci cand am instalat modulul pentru plata ramburs, au fost adaugate si proprietati pentru atributul "Status".

Astfel, comanda noastra poate avea statusul "In curs de livrare", "Asteptarea platii la livrare" sau "Livrata".



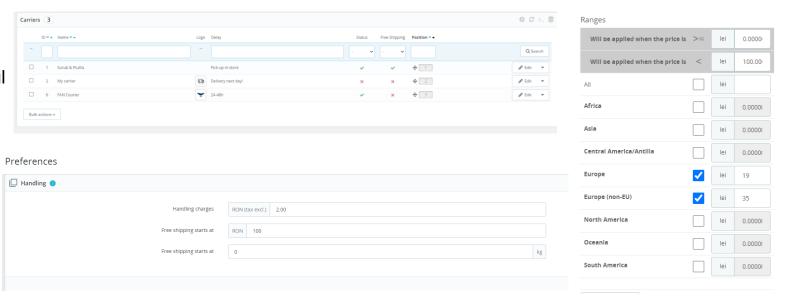
Gestionarea comenzilor



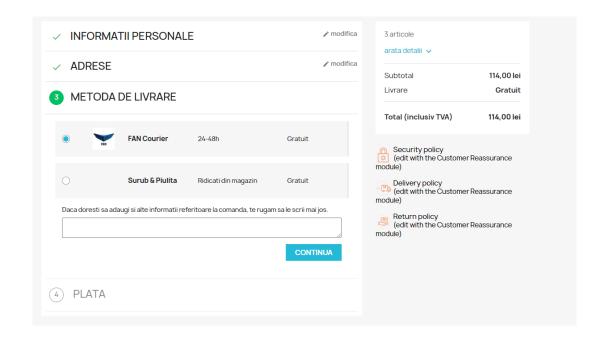


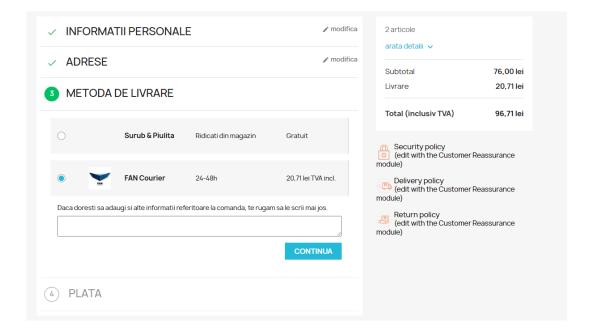
Adaugarea unui serviciu de curierat personalizat

Putem adauga o firma de curierat pe care clientii o pot selecta atunci cand este plasata o comanda. Putem configura pretul de livrare in functie de zona unde se livreaza si de asemenea putem configura ca transportul sa fie gratuit daca comanda depaseste o anumita suma.



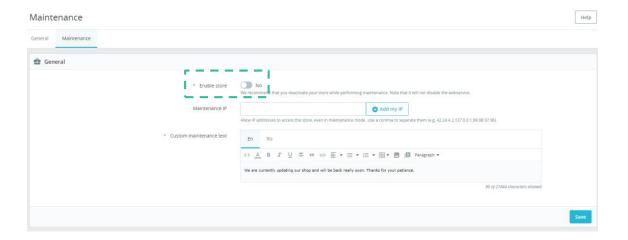
Adaugarea unui serviciu de curierat personalizat





Mentenanta site-ului

Atunci cand instalam sau configuram module care aduc un impact major site-ului, este recomandat sa punem magazinul in starea de mentenanta. Clientii vor avea acces in continuare pe site, dar vor fi redirectionati automat catre pagina care ii atentioneaza despre faptul ca se lucreaza la site.

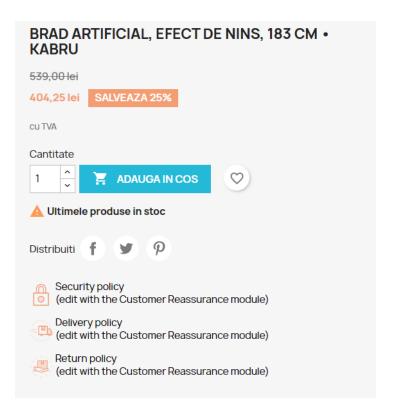


Revenim curand.

We are currently updating our shop and will be back really soon. Thanks for your patience.

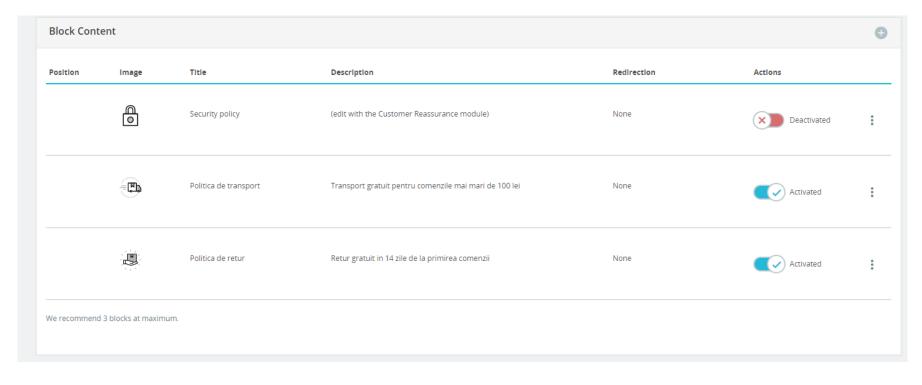
Configurarea unui modul

Observam ca atunci cand vizualizam un produs, pentru anumite detalii ne este sugerat sa configuram modului Customre Reassurance.



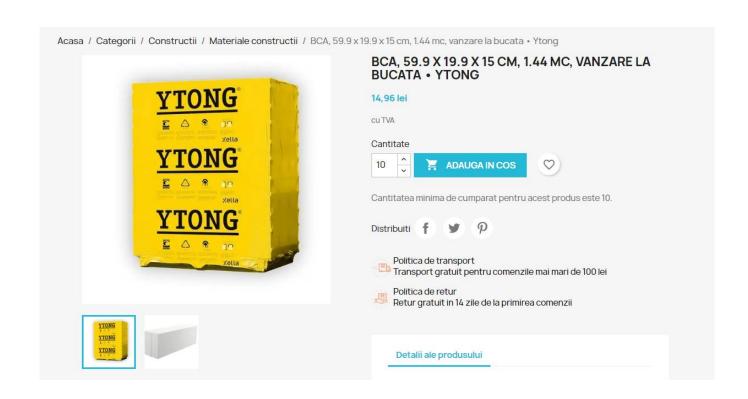
Configurarea unui modul





Configurarea unui modul

Blocurile de continut configurate de catre noi sunt afisate acum pentru fiecare produs in parte.

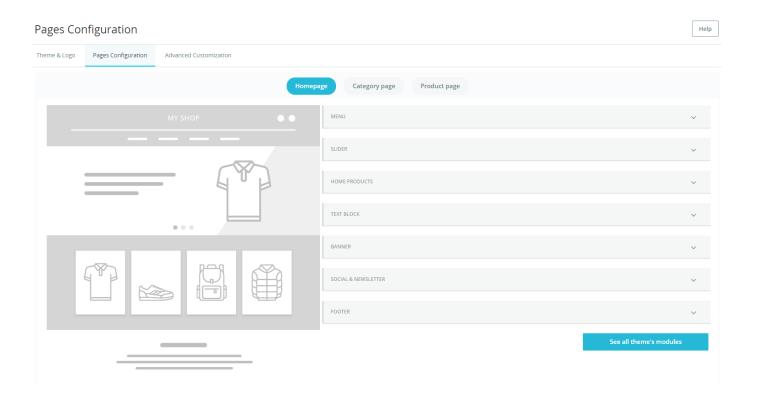


Configurarea unei pagini

Template-ul oferit de Prestashop ofera 3 pagini principale:

- Pagina de acasa
- Pagina unei categorii
- □ Pagina unui produs

Fiecare pagina este impartita in mai multe sectiuni/blocuri, iar fiecare bloc este construit folosinduse mai multe module.



In partea de back-office, stilizarea site-ului poate fi modificata foarte putin fata de ceea ce ofera deja template-ul. Din acest motiv, pentru stilizari suplimentare trebuie creat un fisier .css customizat.

Fisierul va fi creat la calea specificata pentru a putea fi vazut din exterior, laolalta cu alte fisiere .css. Aceasta cale nu este obligatorie, dar faciliteaza declararea ulterioara a acesteia.

^		. – – – – – .	
ume	Data modificării	Tip	Dimensiune
64041bf784575b30a79fa31900c79cf6.otf	20.07.2022 19:46	Fişier font OpenTy	69 KB
83131b9daba3e9a7b2c7ae7e47d2d503.woff	20.07.2022 19:46	Fișier WOFF	31 KB
479700ea91d964d295d75c6b67bd0b28.eot	20.07.2022 19:46	Fișier EOT	68 KB
52260854eea2d38939bf40e9e9687830.woff2	20.07.2022 19:46	Fișier WOFF2	30 KB
a8af1233a958bd1e96edcad65791a0d3.woff	20.07.2022 19:46	Fișier WOFF	33 KB
a7360927bb3673ed7814d6f607d122cb.wo	20.07.2022 19:46	Fișier WOFF2	30 KB
b2efa71e4ca1cd26b0f4df4e71e531a8.svg	20.07.2022 19:46	Microsoft Edge H	1 KB
b252849e892c264f4f220c7e7cb0207a.svg	20.07.2022 19:46	Microsoft Edge H	2 KB
c80d4fb11dcc2140b26007ce39cfa521.woff	20.07.2022 19:46	Fișier WOFF	32 KB
c472ea2ef3e5b37438730a4ab00513b4.svg	20.07.2022 19:46	Microsoft Edge H	1 KB
cbeba06698c00a61e82219b76357ce42.svg	20.07.2022 19:46	Microsoft Edge H	1 KB
dce0916af972e3040aa191f74b17dd46.woff2	20.07.2022 19:46	Fișier WOFF2	30 KB
de69cd9e672c81725abcde04ecf022ee.woff	20.07.2022 19:46	Fișier WOFF	33 KB
error.css	20.07.2022 19:46	Document foaie st	1 KB
f2a0933406f7830651524f477ba2f543.woff	20.07.2022 19:46	Fișier WOFF	78 KB
💽 f56d95737d55e2bdba95baa528b2583b.svg	20.07.2022 19:46	Microsoft Edge H	4 KB
💽 fa3993a997a0b84e9900ee168b18b919.svg	20.07.2022 19:46	Microsoft Edge H	1 KB
fd2d932b381003546821288ecde6e370.otf	20.07.2022 19:46	Fișier font OpenTy	70 KB
my-custom.css	08.01.2023 13:20	Document foaie st	1 KB
theme.css	20.07.2022 19:46	Document foaie st	199 KB

Adaugam continutul dorit in fisierul .css, apoi trebuie sa ii spunem site-ului sa il ia in considerare.

Acest lucru se face in metoda setMedia() din FrontController.

```
public function setMedia()

(

public function setMedia()

(

$this->registerStylesheet('theme-main', '/assets/css/theme.css', ['media' => 'all', 'priority' => 50]);

$this->registerStylesheet('theme-custom', '/assets/css/custom.css', ['media' => 'all', 'priority' => 1000]];

| $this->registerStylesheet('theme-my-custom', '/assets/css/my-custom.css', ['media' => 'all', 'priority' => 0]);

| $this->registerStylesheet('theme-rtl', '/assets/css/rtl.css', ['media' => 'all', 'priority' => 900]);

| $this->registerStylesheet('theme-rtl', '/assets/css/rtl.css', ['media' => 'all', 'priority' => 900]);

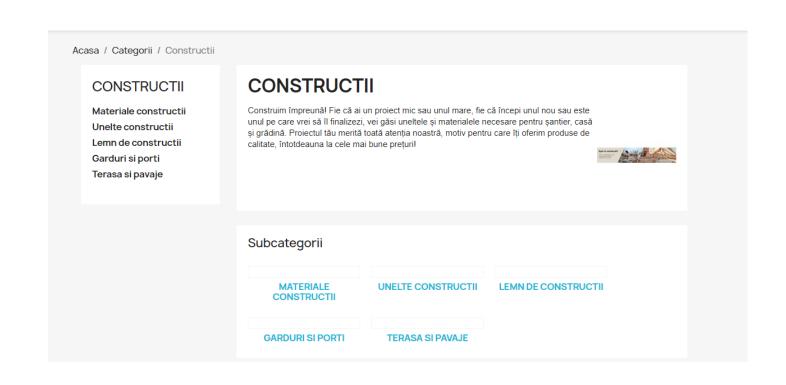
| $this->registerJavascript('corejs', '/themes/core.js', ['position' => 'bottom', 'priority' => 0]);

| $this->registerJavascript('theme-main', '/assets/js/theme.js', ['position' => 'bottom', 'priority' => 50]);

| $this->registerJavascript('theme-custom', '/assets/js/custom.js', ['position' => 'bottom', 'priority' => 1000]);
```

In acest fel era configurata partea de descriere a unei categorii inainte de modificarile noastre.

Dupa ce aducem modificari in afara site-ului este recomandat sa stergem cache-ul.



Observam ca acum site-ul ia in considerare si noua stilizare pe care am introdus-o.

Acasa / Categorii / Constructii

CONSTRUCTII

Materiale constructii
Unelte constructii
Lemn de constructii
Garduri si porti
Terasa si pavaje

CONSTRUCTII



Construim împreună! Fie că ai un proiect mic sau unul mare, fie că începi unul nou sau este unul pe care vrei să îl finalizezi, vei găsi uneltele și materialele necesare pentru șantier, casă și grădină. Proiectul tău merită toată atenția noastră, motiv pentru care îți oferim produse de calitate, întotdeauna la cele mai bune prețuri!

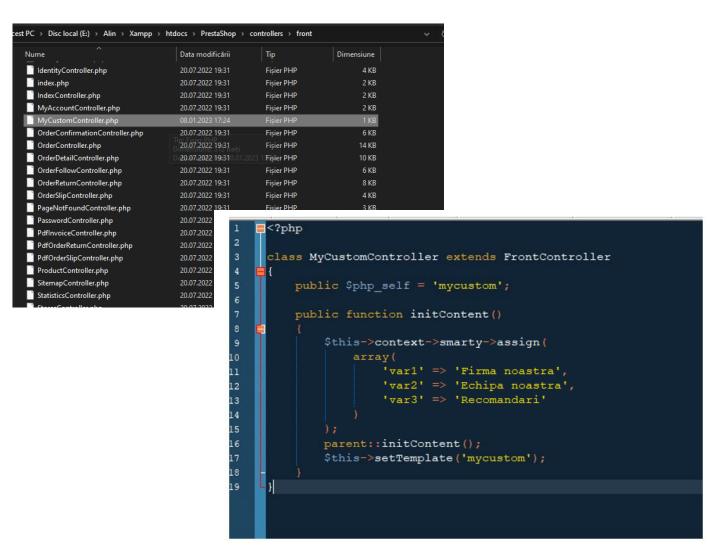
Crearea unei pagini customizate

Daca dorim sa adaugam si alte pagini site-ului nostru putem sa le cream noi. Pentru a face o pagina in Prestashop vom avea nevoie de un fisier template, dar pentru aceasta vom avea nevoie si de un controller care sa apeleze metoda setTemplate() pentru acel template.

Din acest motiv vom crea un controller nou care va apela template-ul pentru noua pagina.

Controllerul creat va fi unul simplu care sa satisfaca cele mai de baza cerinte: va mosteni FrontController, va avea un identificator de pagina. Metoda initContent() este mostenita si va fi executata imediat ce controller-ul este executat.

Putem sa preparam si cateva variabile pe care le vom folosi in fisierul .tpl.



Crearea unei pagini customizate

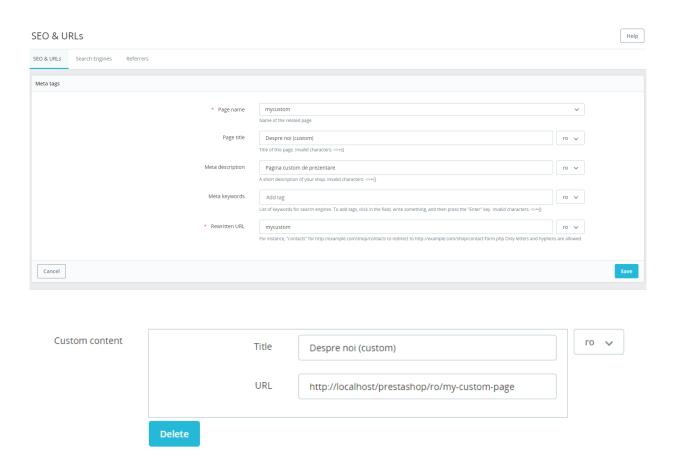
Urmatorul pas este crearea template-ului.

```
{block name='page title'}
  <span class="sitemap-title">{l s='Despre Noi - Custom' d='Shop.Theme'}</span>
{block name='page content container'}
  <section id="content" class="page-content page-cms page-cms-4">
      <div class="col-xs-12 col-sm-4">
      <div class="cms-block">
      <h3 class="page-subheading">{$var1}</h3>
       <strong class="dark">Lorem ipsum dolor sit amet conse ctetur adipisicing elit, sed do eiusmod tempor incididun.</strong>
      Lorem ipsum dolor sit amet conse ctetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam. Lorem ipsum dolor sit amet conse ctet
      <em class="icon-ok"></em>Produse de cea mai bună calitate
      <em class="icon-ok"></em>Cele mai bune servicii pentru clienții noștri
      <em class="icon-ok"></em>Garanție cu banii înapoi în 30 de zile
      </div>
      <div class="col-xs-12 col-sm-4">
      <div class="cms-box">
      <h3 class="page-subheading">{$var2}</h3>
      <strong class="dark">Lorem set sint occaecat cupidatat non </strong>
       Eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo.
      </div>
      <div class="col-xs-12 col-sm-4">
      <div class="cms-box">
      <h3 class="page-subheading">{$var3}</h3>
      <div class="testimonials">
      <div class="inner"><span class="before">"</span>Lorem ipsum dolor sit amet conse ctetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim.
       <strong class="dark">Lorem ipsum dolor sit</strong>
      <div class="inner"><span class="before">"</span>Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet cons
      <strong class="dark">Ipsum dolor sit</strong>
      </div>
   </section>
```

Crearea unei pagini customizate

Trebuie apoi sa adaugam pagina in back-office pentru a putea fi cautata. Ii atribuim un nume, o descirere si, cel mai important, un URL care sa ne ajute sa accesam pagina.

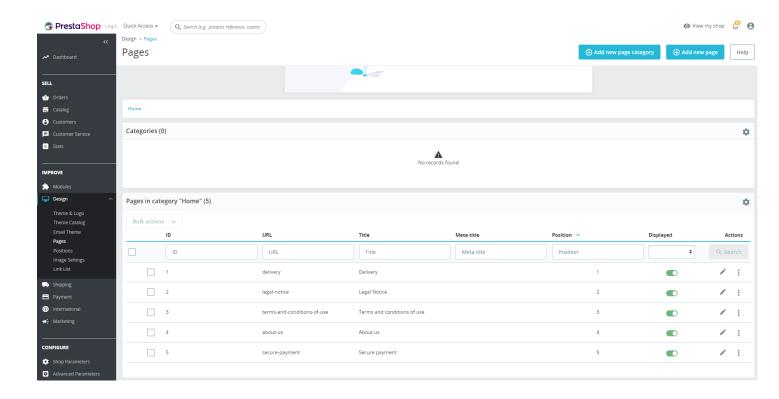
Apoi pagina si in footer, pentru a fi vizibila si accesibila.



Crearea unei pagini CMS

Daca dorim sa adaugam o pagina de prezentare, cu mai putin continut si cu mai putina functionalitate, putem sa adaugam o pagina noua din back-office mult mai rapid.

Trebuie insa tinut cont ca se poate adauga doar text, asa ca pentru stilizare si functionalitate trebuie facuta o implementare customizata.



JS customizat

Asemenea fisierelor .css, si in cazul fisierelor .js putem sa adaugam functionalitate suplimentara elementelor site-ului.

Nume	Data modificării	Тір	Dimensiune
🏽 error.js	20.07.2022 19:46	JavaScript File	0 KB
g my-custom.js	08.01.2023 16:35	JavaScript File	1 KB
👸 theme.js	20.07.2022 19:46	JavaScript File	180 KB

JS customizat

In noul fisier o sa adaugam urmatoare functionalitate: de fiecare data cand o pagina se incarca, logo-ul va avea un efect de fade pana cand va fi afisat cu opacitatea maxima.

Asemenea customizarii CSS, si in acest caz va fi nevoie sa specificam site-ului sa ia in considerare noul script. Acest lucru se face in acelasi loc, metoda setMedia(), din FrontController.

Daca dorim sa aducem modificari doar pentru o anumita pagina, atunci script-ul nu ar trebui introdus aici. In schimb, functionalitatea ar putea fi adaugata in fisierul template specific acelei pagini.

```
$\frac{1}{2}$$
$\text{this->registerJavascript('corejs', '/themes/core.js', ['position' => 'bottom', 'priority' => 0]);
$\text{this->registerJavascript('theme-main', '/assets/js/theme.js', ['position' => 'bottom', 'priority' => 50]);
$\text{this->registerJavascript('theme-custom', '/assets/js/custom.js', ['position' => 'bottom', 'priority' => 1000]);
$\text{this->registerJavascript('theme-my-custom', '/assets/js/my-custom.js', ['position' => 'bottom', 'priority' => 100]);}
$\text{this->registerJavascript('theme-my-custom', '/assets/js/my-custom.js', ['position' => 'bottom', 'priority' => 1000]);}
$\text{this->registerJavascript('theme-my-custom', '/assets/js/my-custom.js', ['position' => 'bottom', 'priority' => 100]);}
$\text{this->registerJavascript('theme-my-custom', '/assets/js/my-custom', 'priority' => 'bottom', 'priority' => 100]);}
$\text{this->registerJavascript('theme-my-custom', 'priority' => '
```

Modulul Custom CSS+JS PRO

O modalitate mult mai rapida si usoara de customizare este folosind modulul PRO. Direct in back-office poate fi adaugat codul css si codul js pentru paginile dorite.

