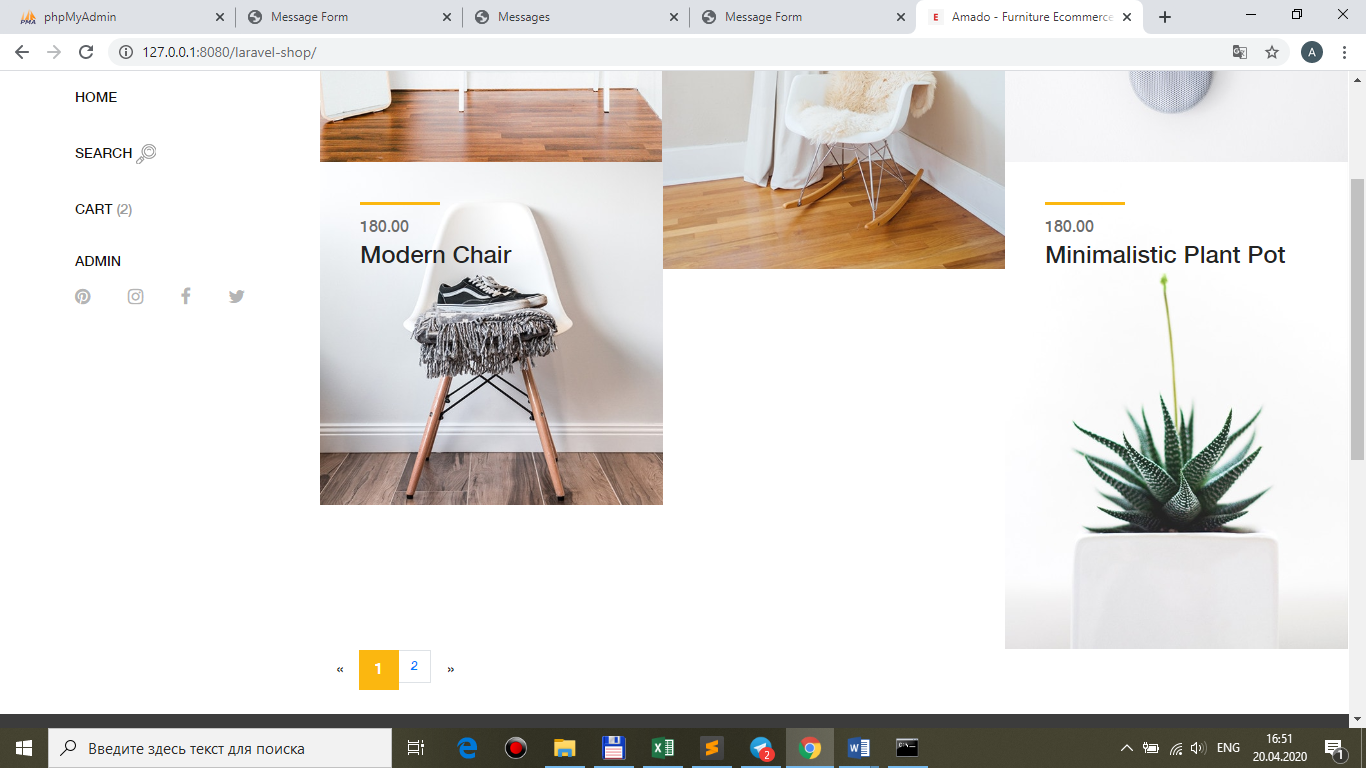
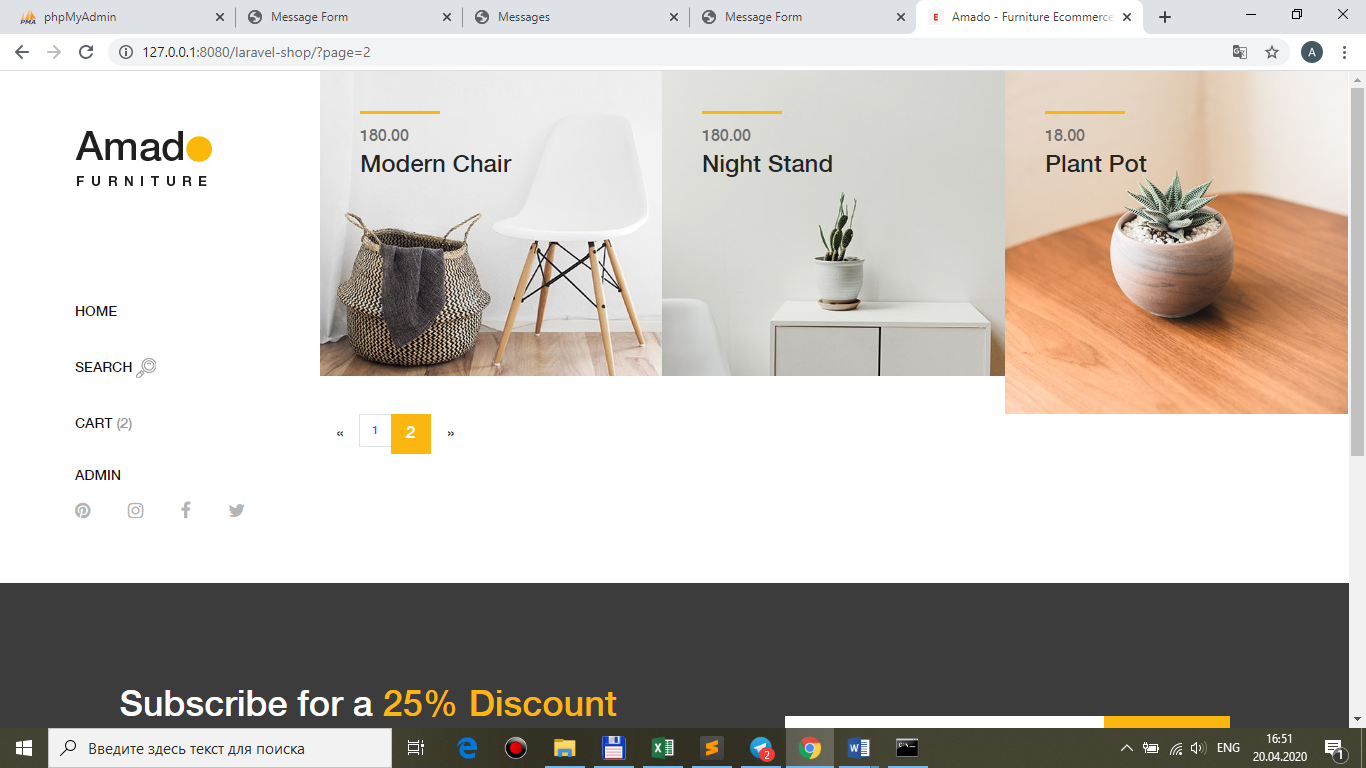
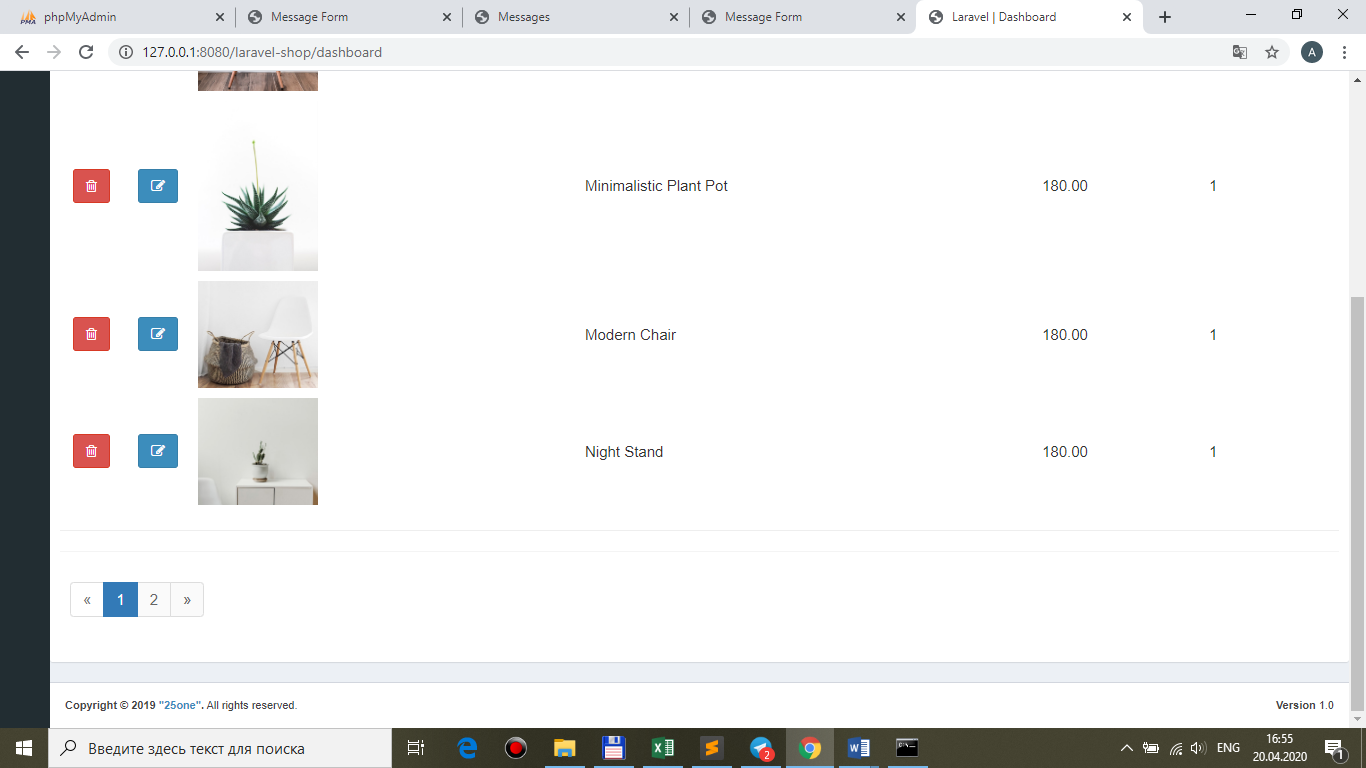
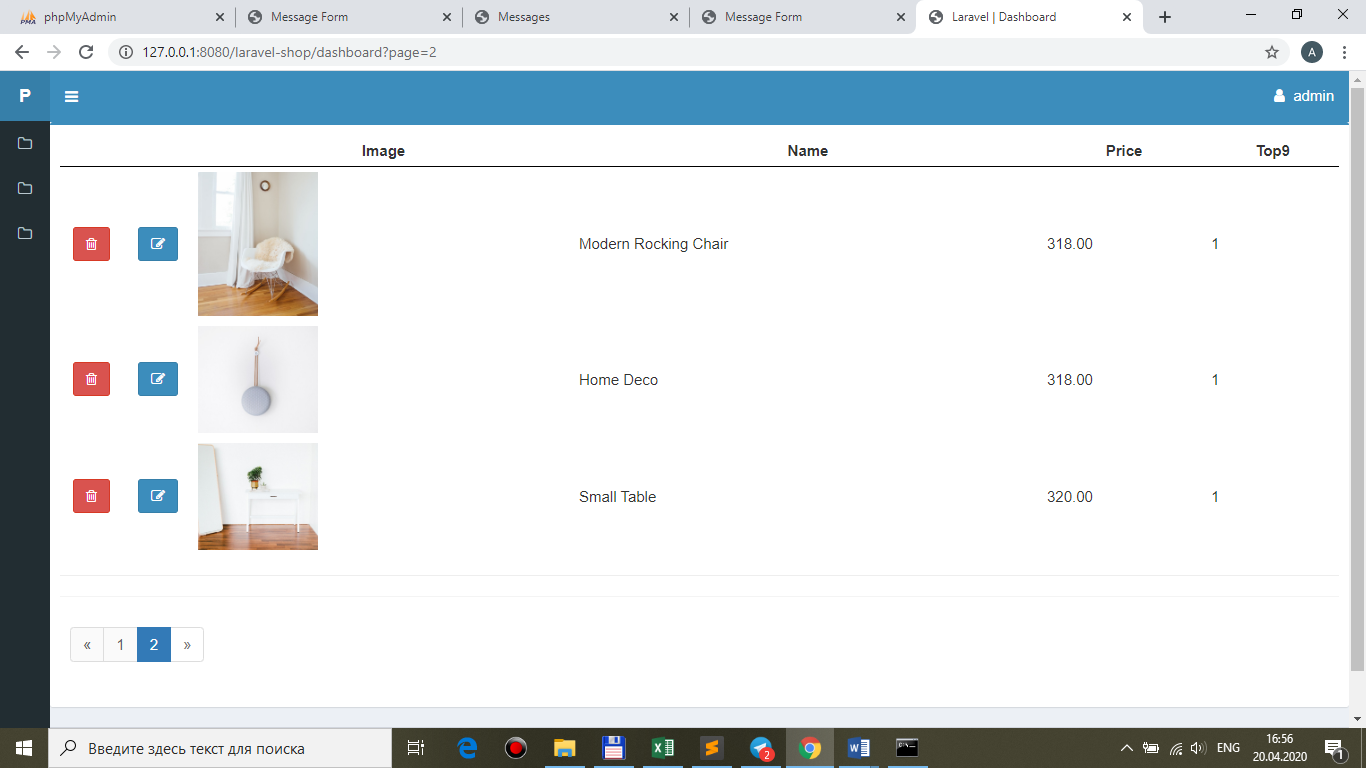
В **laravel-shop** реализовать **пагинацию** на **front-** и **back(dashboard)-**страницах:









**copy pagination.blade.php to views\front and views\back**

**config\app.php**

'nbrPages' => [

**'front' => [**

**'products' => 5, //!!!front**

**],**

**'back' => [**

**'products' => 5, //!!!back**

**],**

],

**!!!+ php artisan config:cache**

Controllers\**Front\ShopController.php**

public function \_\_construct(ShopRepository $repository)

{

…

**$this->nbrPages = config('app.nbrPages.front.products'); //!!!front**

}

Controllers\**Back\AdminController.php**

public function \_\_construct(ShopRepository $repository)

{

…

**$this->nbrPages = config('app.nbrPages.back.products'); //!!!back**

}

**Traits\Indexable.php**

/\*\*

\* The pagination number.

\*

\* @var int

\*/

**protected $nbrPages;**

public function index(Request $request)

{

$products = $this->repository->funcSelect($request, $this->getParameters(), **$this->nbrPages**);

**$links = $products ->links($this->namespace . '.pagination');**

// Ajax response

if ($request->ajax()) {

return response()->json([

'table' => view($this->namespace . ".brick-standard", ['products' => $products])->render(),

**'pagination' => $links->toHtml(),**

]);

}

return view($this->namespace . '.index', ['products' => $products**, 'links' => $links**]);

}

**ShopRepository.php && AdminRepository.php**

public function funcSelect($request, $parameters**, $nbrPages**)

{

…

**//**return $query->get();

**return $query->paginate($nbrPages);**

}

views\**front\index.blade.php**

…

**<div id="pagination" class="box-footer">**

**{{ $links }}**

**</div>**

<br>

</div>

<!-- Product Catagories Area End -->

views\**back\index.blade.php**

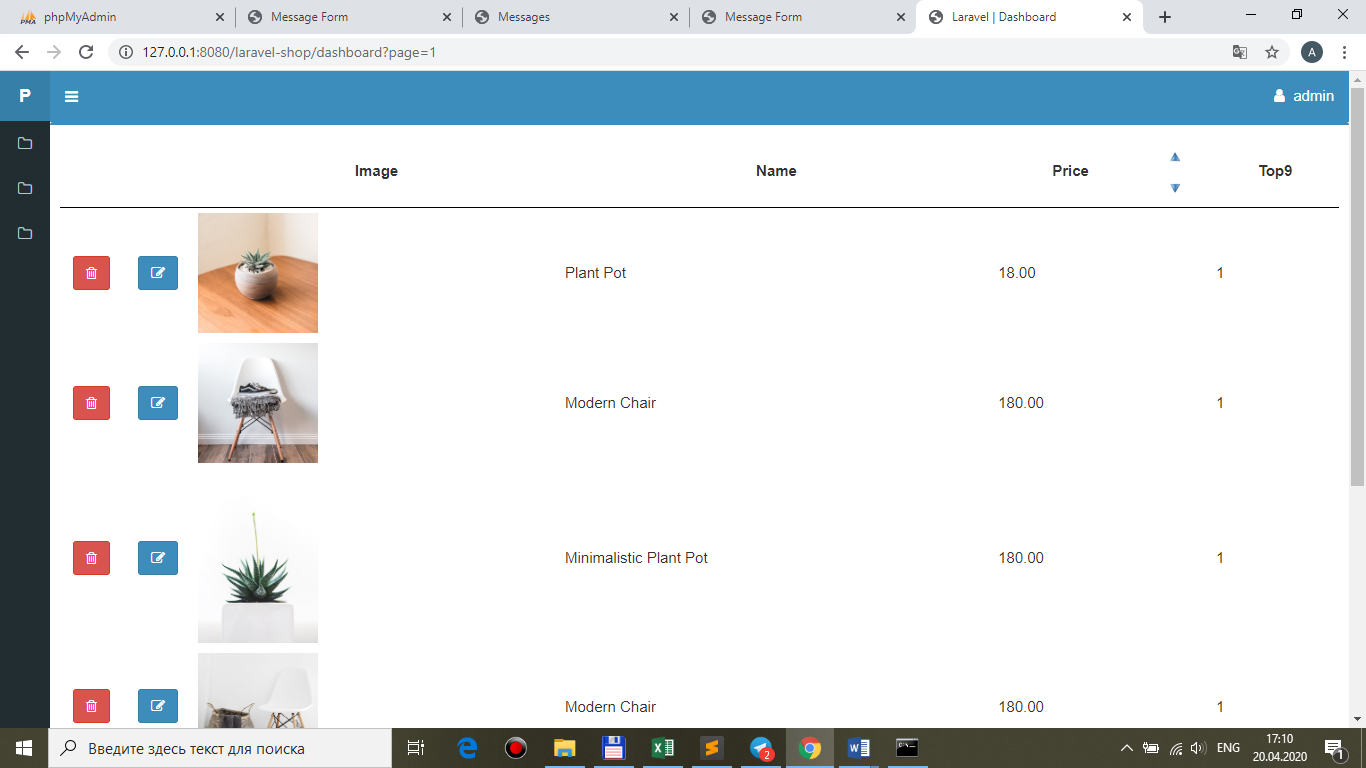
<hr>

**<div id="pagination" class="box-footer">**

**{{ $links }}**

**</div>**

2.На **back(dashboard)**-странице поместить элементы **сортировки** для поля **price**:



views\**back\index.blade.php**

…

<td>

<table>

<tr>

<td rowspan="2">**Price**</td>

<td ><a href="#" class="sort" **data-order="price" data-direction="asc"> <!-- like config\parameters.php -->**

<img src="{{ asset('public/images/top.jpg') }}" alt />

</a></td>

</tr>

<tr>

<td><a href="#" class="sort" **data-order="price" data-direction="desc"> <!-- like config\parameters.php -->**

<img src="{{ asset('public/images/bottom.jpg') }}" alt />

</a></td>

</tr>

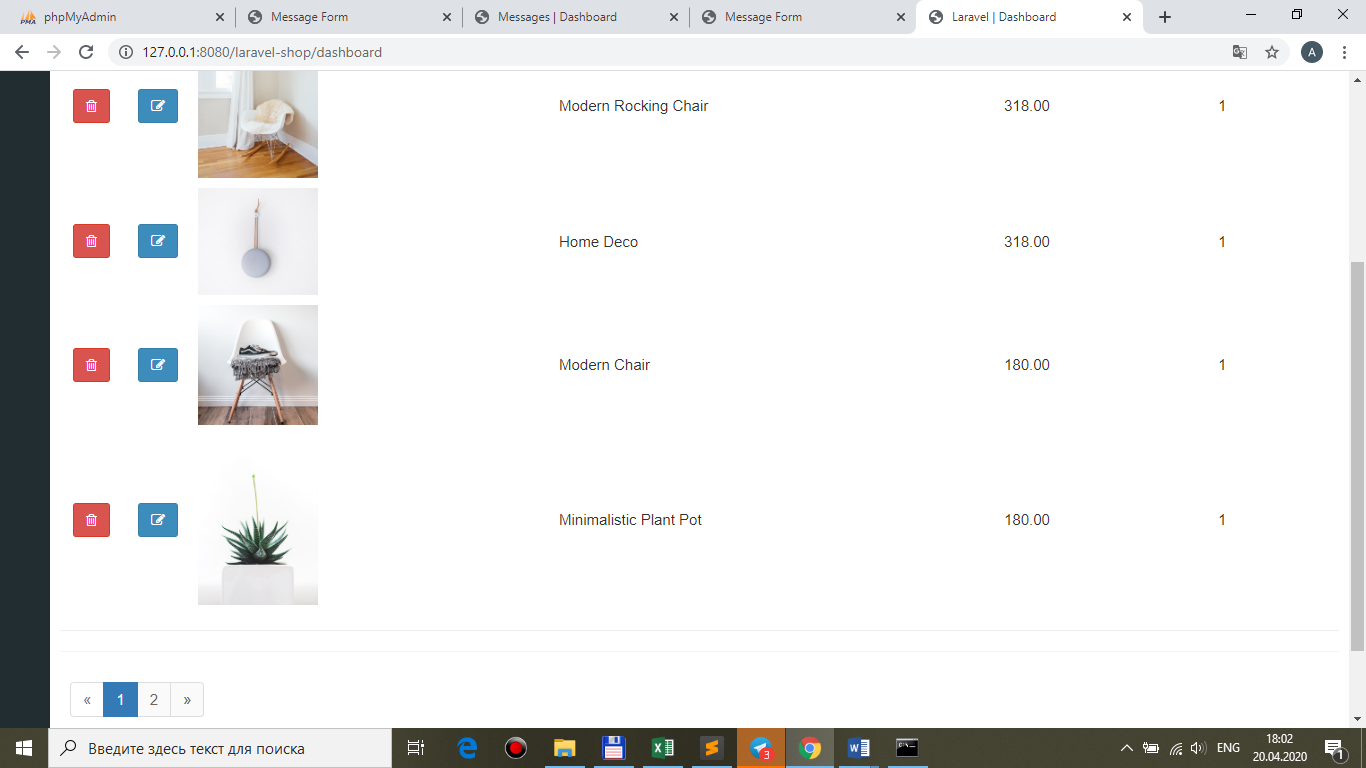
</table>

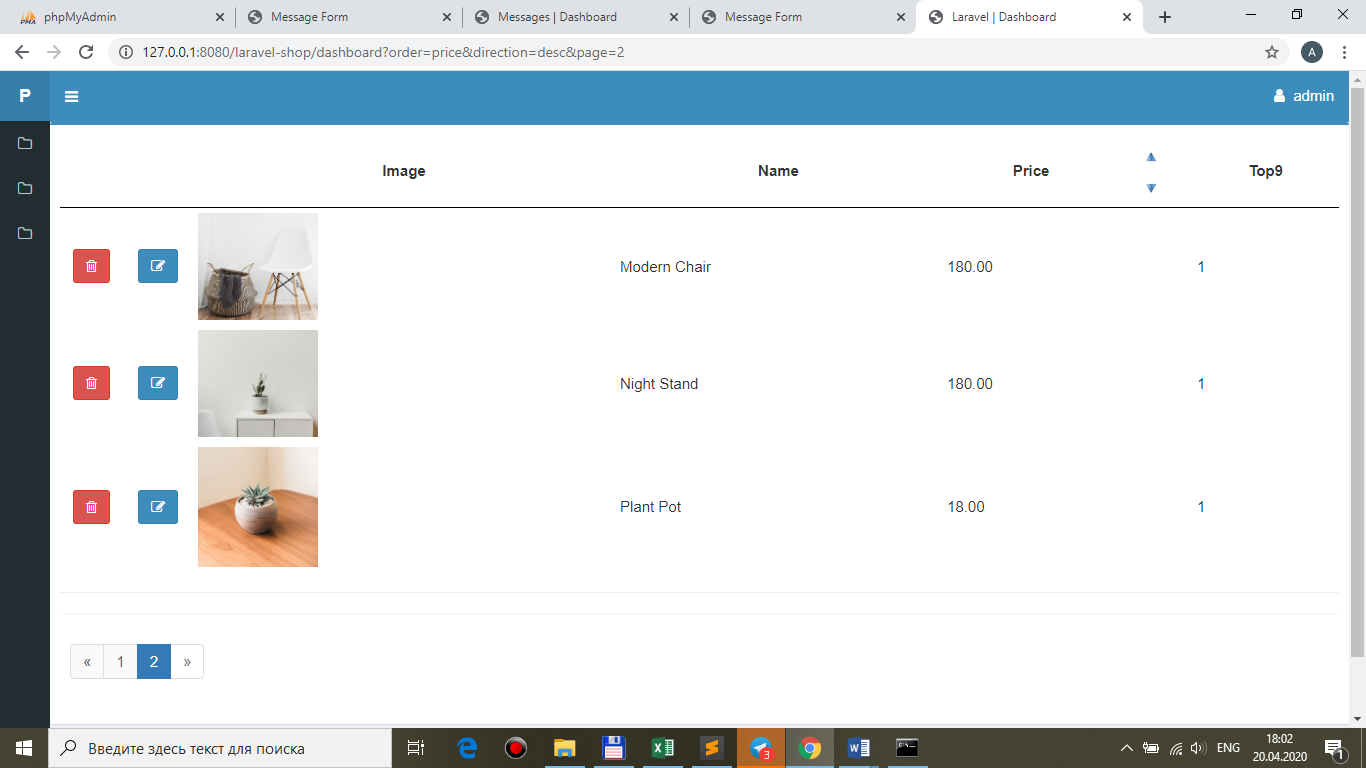
</td>

…

**+ public\images\bottom.jpg && top.jpg**

…и, используя **config\parameters.php** реализуйте алгоритм **ajax**-сортировки по данному полю:





в config\**parameters.php есть**

**'back' => [**

**'order' => 'price',**

**'direction' => 'asc',**

**],**

views\**back\index.blade.php**

$(document).ready(function(){

…

**$('.sort').click(function(){**

**BaseRecord.order=$(this).attr('data-order');**

**BaseRecord.direction=$(this).attr('data-direction');**

**BaseRecord.dashboardparameters();**

**return false;**

**});**

});

**public\js\main.js**

**order: 'price',**

**direction: 'desc',**

**dashboardparameters: function(){**

var ajaxSetting={

**method: 'get',**

**url: './dashboard',**

**data: {**

**order: BaseRecord.order,**

**direction: BaseRecord.direction,**

**},**

success: function(data){

//alert(data);

**$('.back-pannel').html(data.table); //!!!.back-pannel**

**$('#pagination').html(data.pagination); //!!!**

**$('.listbuttonremove').click(function(){**

**BaseRecord.destroy($(this).attr('id'));**

**return false;**

**});**

},

error: function(data){

alert(data.responseText);

},

};

$.ajax(ajaxSetting);

},

**Traits\Indexable.php**

public function index(Request $request)

{

$products = $this->repository->funcSelect($request, $this->getParameters(**$request**), $this->nbrPages);

…

protected function getParameters(**$request**)

{

// Default parameters

$parameters = config("parameters.".$this->namespace); //$parameters['order'], $parameters['direction']

**foreach($parameters as $key => &$value){**

**if(isset($request->$key)) {**

**$value = $request->$key;**

**}**

**}**

return $parameters;

}

Traits\**Indexable.php**

public function index(Request $request)

{

$products = $this->repository->funcSelect($request, $this->getParameters($request), $this->nbrPages);

$links = $products->**appends($this->getParameters($request))->**links($this->namespace . '.pagination');

…

в **AdminRepository.php ЕСТЬ**

public function funcSelect($request, $parameters, $nbrPages)

{

…

**->orderBy($parameters['order'], $parameters['direction']);**