1.Провести **валидацию** значений, записываемых в таблицу **carts** (по **полям** и **правилам**, **аналогичным** тем, что были **рассмотрены на занятии**):

**app\Http\Requests\CartRequest.php + Request.php + resources\views\shop\components (copy-paste)**

**resources\views\shop\product.blade.php (в необходимое место верстки)**

@if ($errors->any())

@component('shop.components.alert')

@slot('type')

danger

@endslot

@foreach ($errors->all() as $error)

{{ $error }}<br>

@endforeach

@endcomponent

@endif

**app\Controllers\ShopController.php**

use App\ {

…

**Http\Requests\CartRequest**

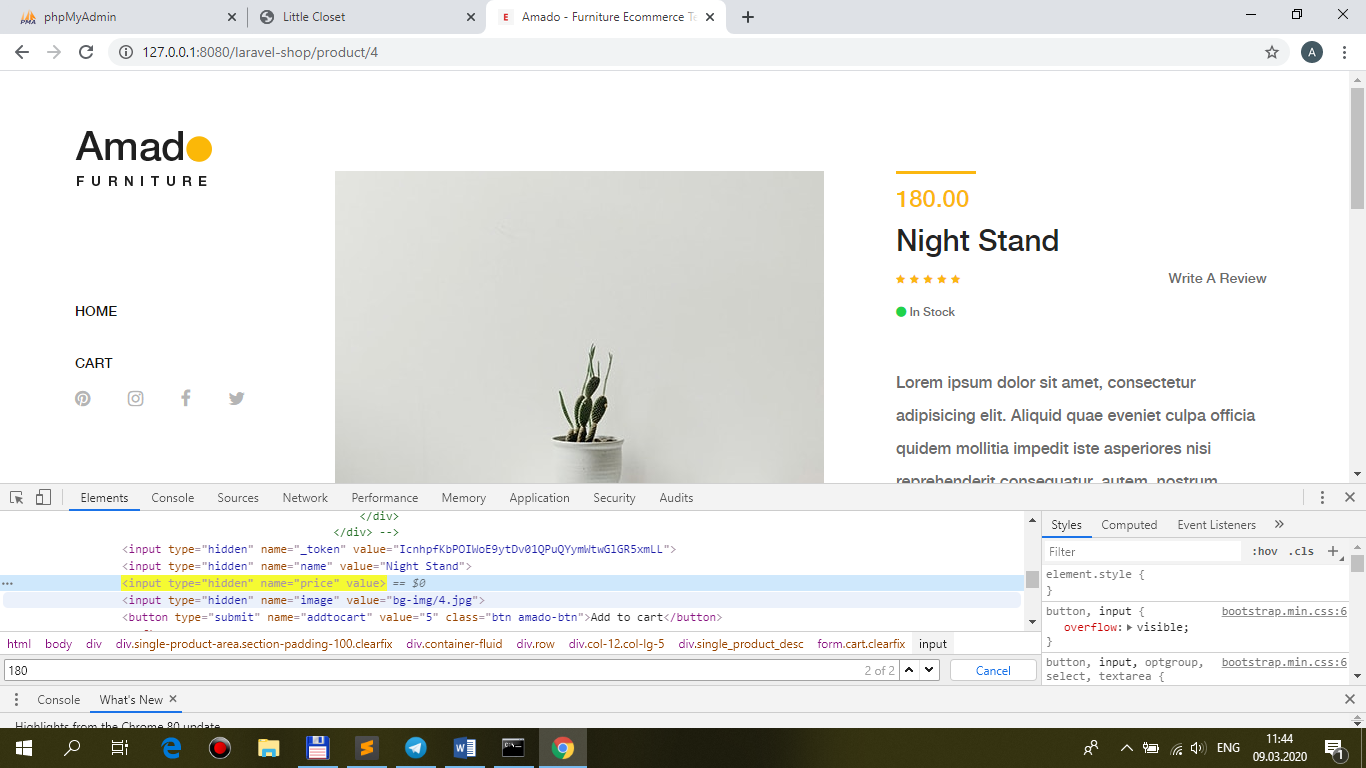
};

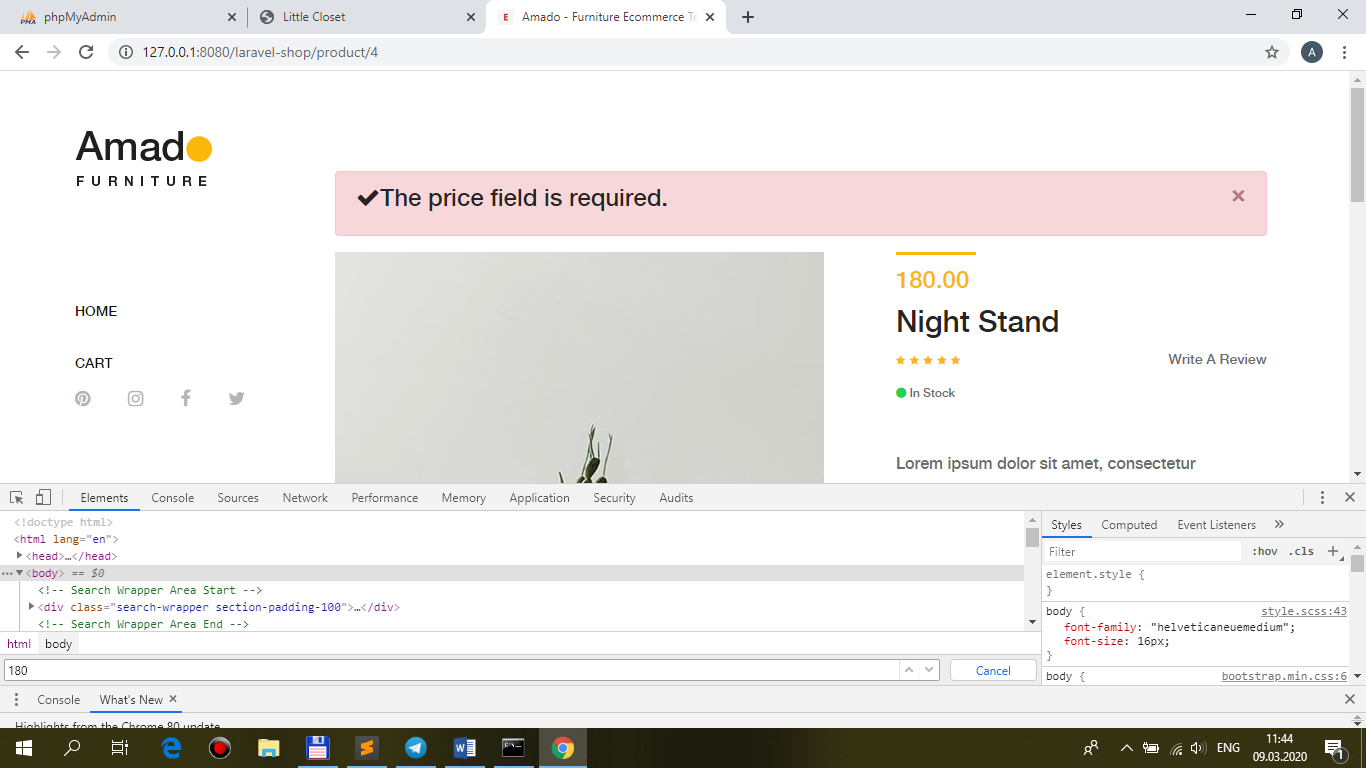
…

public function **tocart(CartRequest $request**, ShopRepository $repository)

{

…





2.При переходе на **cart-page** вывести (пока **print\_r**) **перечень товаров корзины**:

**app\Controllers\ShopController.php**

public function **cart(ShopRepository $repository)**

{

**$carts = $repository->fromCart();**

return view('shop.cart'**, compact('carts')**);

}

**app\Repositories\ShopRepository.php**

protected $model**\_product**;

**protected $model\_cart;**

public function \_\_construct(Product $product, **Cart $cart**)

{

$this->model**\_product** = $product;

**$this->model\_cart = $cart;**

}

public function funcSelect($request)

{

$query = $this->**model\_product**…

/\*\*

\* Create a query for **Cart**.

\*

\* @return \Illuminate\Database\Eloquent\Builder

\*/

**public function fromCart()**

**{**

**$query = $this->model\_cart**

**->select('id', 'name', 'price', 'image');**

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

**}**

