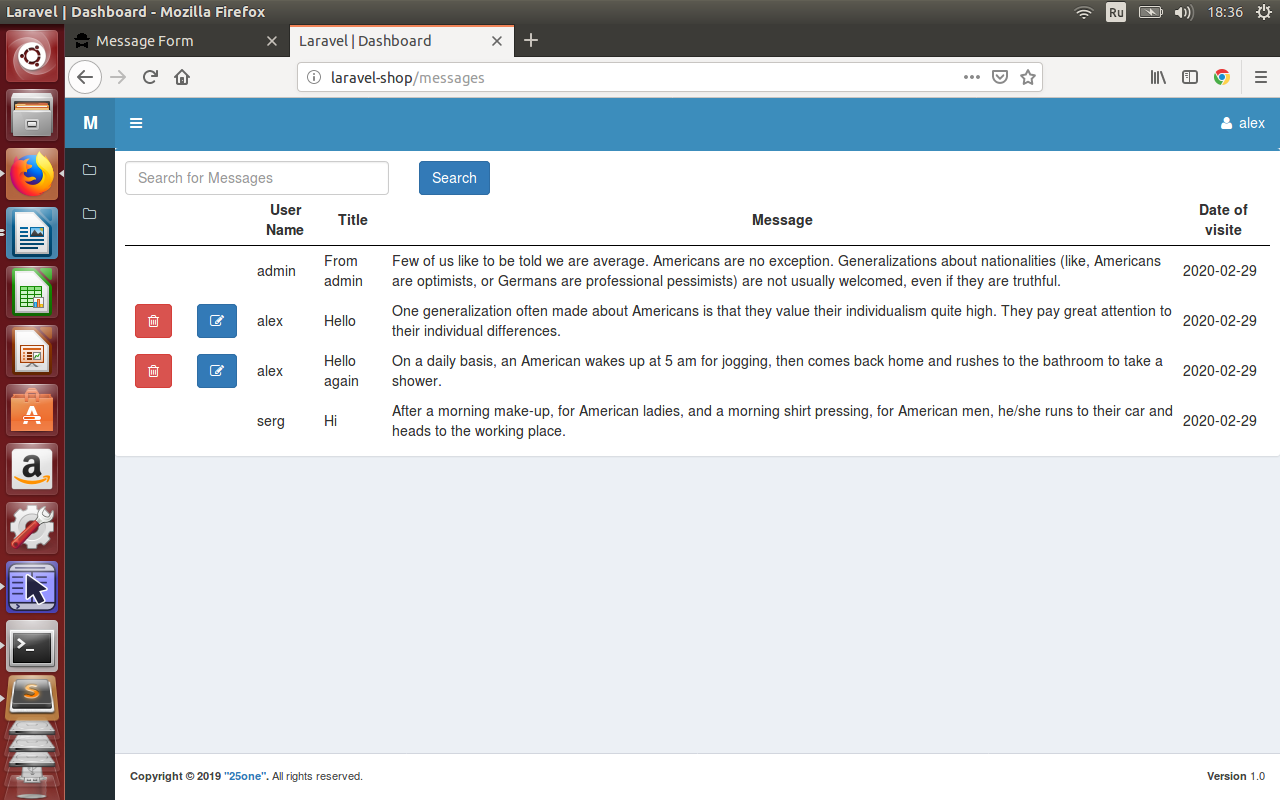
1.**Front-message-auth** — показывать иконки **update/destroy только** для сообщений **auth**-пользователя, **остальные** сообщения — показывать **без** иконок:



@foreach($messages as $message)

<tr>

**@if($message->user\_id == auth()->user()->id)**

<td class="center widthbutton"><a class="btn btn-danger listbuttonremove" href="#"><i class="fa fa-trash-o" aria-hidden="true"></i></a></td>

<td class="col-md-6 col-sm-6 col-xs-6 center widthbutton"><a class="btn btn-primary listbuttonupdate" href="#"><i class="fa fa-edit" aria-hidden="true"></i></a></td>

**@else**

<td class="center widthbutton"></td>

<td class="col-md-6 col-sm-6 col-xs-6 center widthbutton"></td>

**@endif**

<td>{{ $message->user->name }}</td>

<td>{{ $message->title }}</td>

<td>{{ $message->message }}</td>

<td>{{ $message->datevisit }}</td>

</tr>

@endforeach

2.**Back-message-auth** — создать **политику** для дальнейшего использования при **update/destroy** (т. е. - **в независимости от** **Front-показа** иконок **невозможность update/destroy чужих** сообщений):

**app\Policies\ApimessagePolicy.php**

<?php

namespace App\Policies;

use App\Models\ {User, **Apimessage**};

class **Apimessage**Policy extends Policy

{

/\*\*

\* Determine whether the user can manage the comment.

\*

\* @param \App\Models\User $user

\* @param \App\Models\Apimessage $message

\* @return mixed

\*/

public function manage(User $user, **Apimessage $message**)

{

return $user->id === **$message**->user\_id;

}

}

**app\Providers\AppServiceProvider.php**

…

**use App\Models\Apimessage;**

**use App\Policies\ApimessagePolicy;**

protected $policies = [

// 'App\Model' => 'App\Policies\ModelPolicy',

Cart::class => CartPolicy::class,

**Apimessage::class => ApimessagePolicy::class,**

];