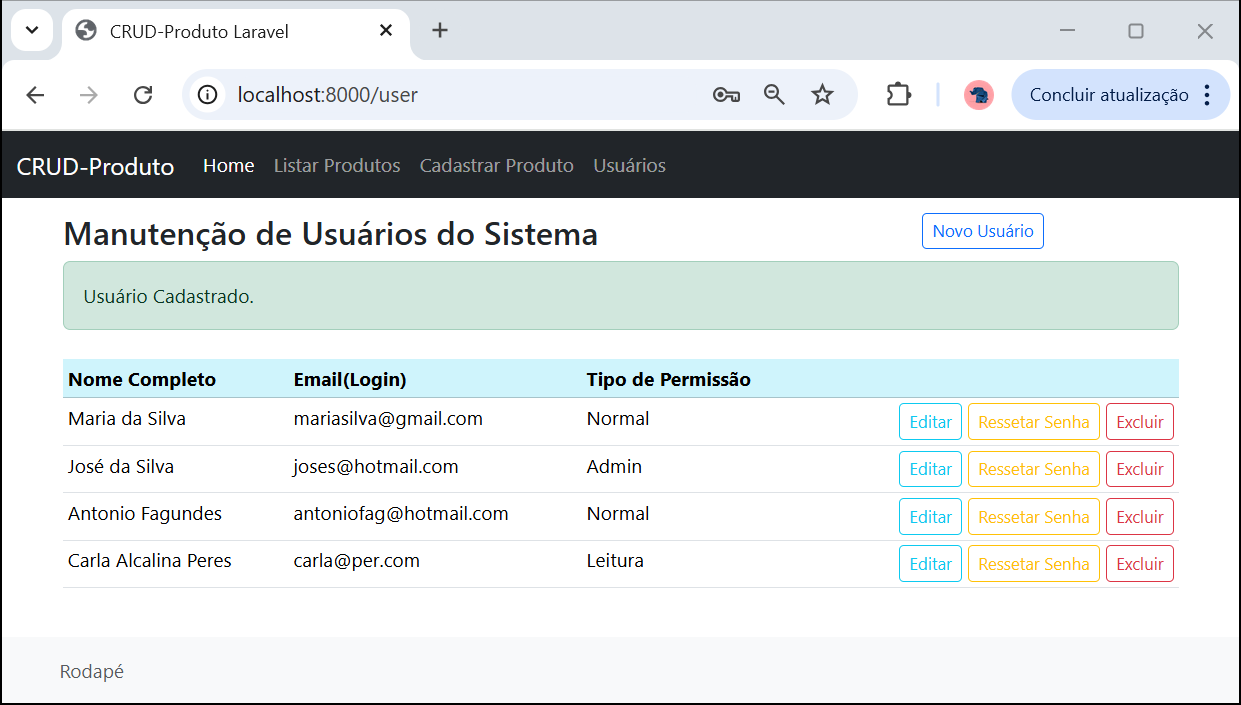
**Aula 6 – CRUD-Produto em Laravel**

**Manutenção de Usuário com Modal**

**Objetivos:**

* Implementar as funcionalidades de Cadastro, Consulta, Edição e Exclusão (CRUD) de usuários do sistema;
* No frontend será utilizado o recurso de Modal;

Ao final desta aula devemos ter já em funcionamento a Manutenção de Usuários do Sistema, onde todas as funcionalidades serão concentradas em uma única página com a utilização de Modal.



1. **Tabela de Usuários**

O Laravel já cria automaticamente uma tabela de Usuários chamada users a qual podemos usar para controle de acesso à nossa Aplicação. Essa tabela já tem um modelo (app\Models\User.php) e também já tem uma migração (database\migrations\0001\_01\_01\_000000\_create\_users\_table.php)

A tabela users do Laravel possui a estrutura da figura a seguir. Note que não existe um campo login e que temos o campo id (chave primária) e o campo email que deve ser único. Assim, podemos utilizar o campo email para login na Aplicação.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Vamos acrescentar a coluna permissao que será utilizada para implementar nosso sistema de permissão de acesso às funcionalidades da Aplicação. Para isso, devemos:

1. Criar uma migração para acrescentar a coluna. Execute o seguinte comando no terminal:

php artisan make:migration add\_permissao\_to\_users\_table --table=users

Texto

O conteúdo gerado por IA pode estar incorreto.

1. Editar a migration criada. Abra o arquivo de migration gerado em database\migrations\ e edite os métodos up() e down()e acrescente as linhas em vermelho a seguir:

database\migrations\2025\_04\_09\_121900\_add\_permissao\_to\_users\_table.php

public function up(): void

{

Schema::table('users', function (Blueprint $table) {

**$table->string('permissao', 50)->after('email');**

});

}

/\*\*

\* Reverse the migrations.

\*/

public function down(): void

{

Schema::table('users', function (Blueprint $table) {

**$table->dropColumn('permissao');**

});

}

1. O próximo passo é executar a migration. Salve o arquivo anterior e execute o seguinte comando no prompt:

php artisan migrate

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

1. Por fim, é necessário atualizar o modelo User para permitir *mass assignment* (atribuição em massa) do campo permissao. Então edite o arquivo app\Models\User.php e acrescente a linha em vermelho:

app\Models\User.php

protected $fillable = [

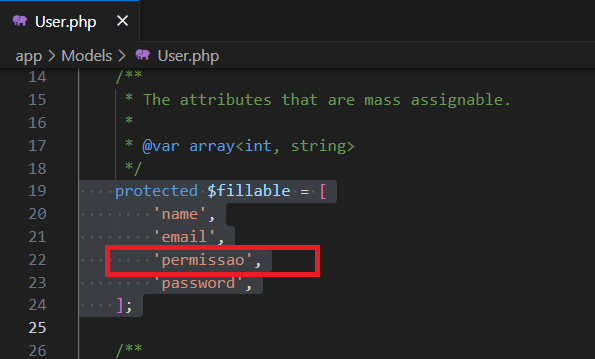
'name',

'email',

**'permissao',**

'password',

];



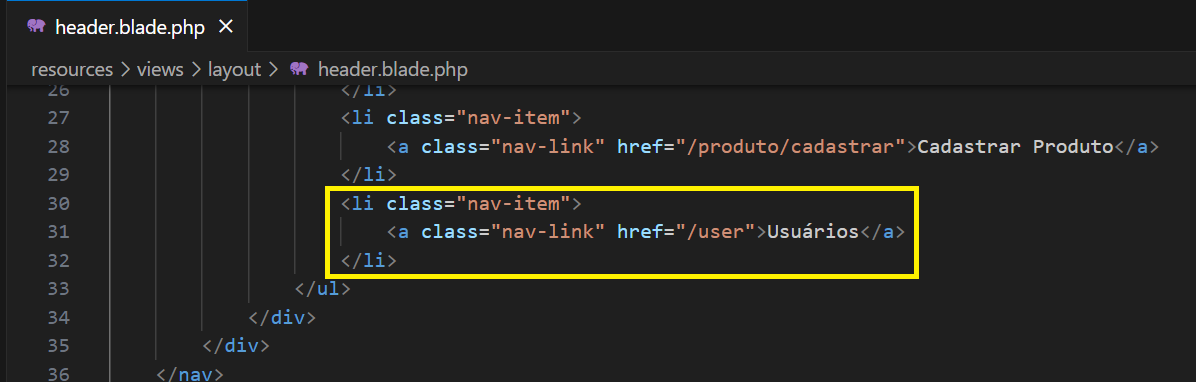
1. **Inserir o item Usuários no menu de opções da Aplicação**

Primeiramente, inserir o item Usuários no menu da Aplicação para chamar o método index() do UserController. Isso é feito no arquivo resources\views\layout\header.blade.php. insira um novo item como a seguir, logo após o item Cadastrar Produto :

<li class="nav-item">

<a class="nav-link" href="/user">Usuários</a>

</li>

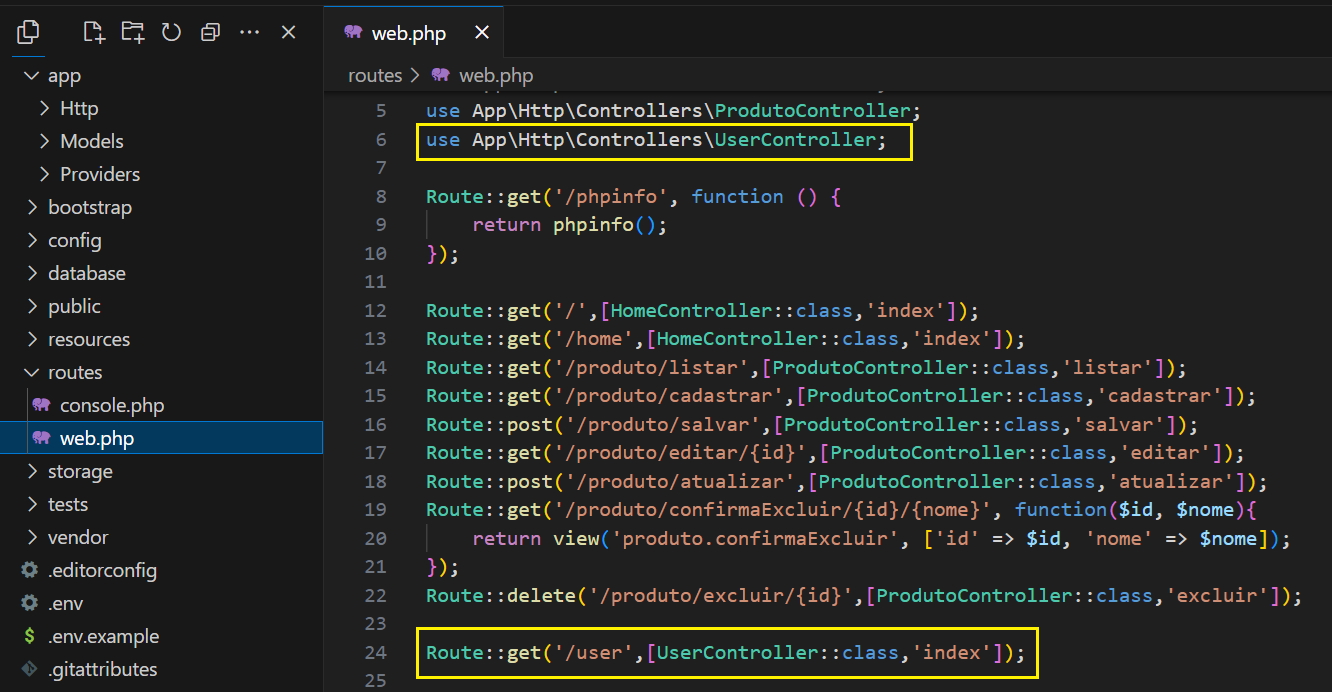


1. **Definir a Rota**

Agora é preciso definir a rota no arquivo routes\web.php, acrescentando a seguinte linha:

use App\Http\Controllers\UserController;

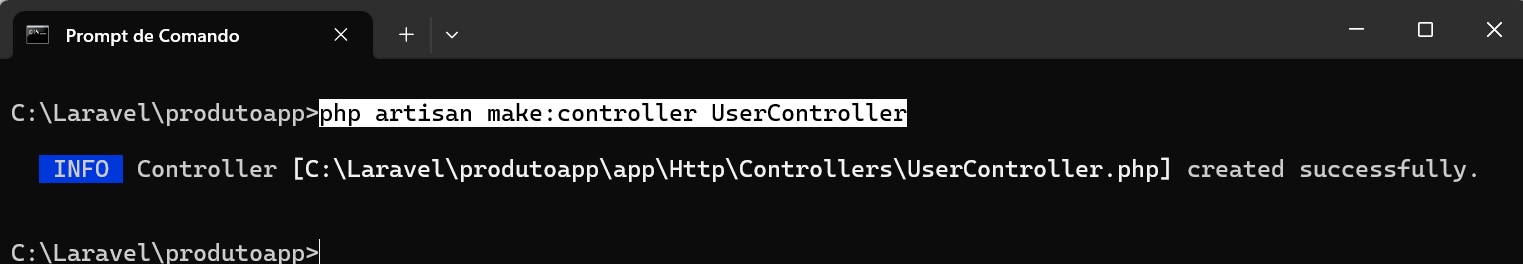
Route::get('/user',[UserController::class,'index']);



1. **Criar o Controlador UserController**

É necessário criar o Controlller do Usuário, caso ainda não tenha sido criado. Para tanto, no console da Aplicação, execute:

php artisan make:controller UserController



1. **Criando o método index() do UserController**

Em app\Http\Controllers\UserController.php criar o método index() e dar um use no modelo User:

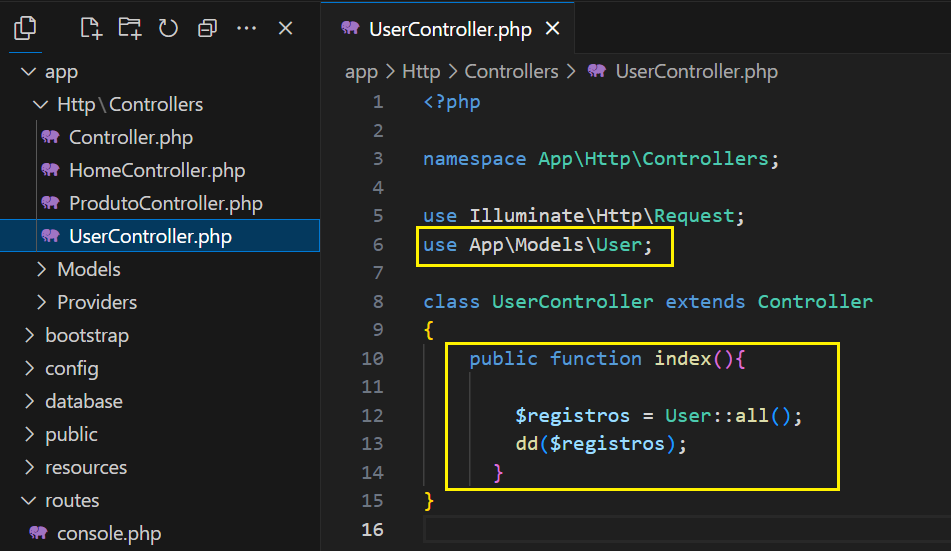
use App\Models\User;

public function index(){

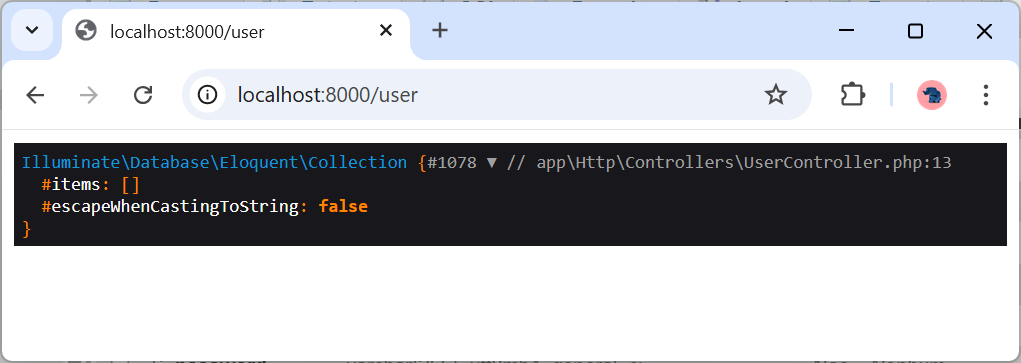
$registros = User::all();

dd($registros);

}



Ainda não temos usuários cadastrados, mas já é possível testar o acesso à tabela de usuários. Abra um Navegador e acesse <http://localhost:8000/user>



1. **Criar a Página de Manutenção de Usuários do Sistema**

Primeiro, altere o método index() do UserController.php para o seguinte:

public function index(){

$registros = User::all();

if ($registros->count()==0){

session(['erro' => 'Nenhum Usuário Encontrado!']);

}

return view('user.index', compact('registros'));

}

A view a ser chamada ficará dentro de resources\views\user com o nome index.blade.php. Crie a pasta user, caso não exista. O conteúdo do arquivo index.blade.php será:

@extends('layout.site')

@if ($errors->any())

<script>

document.addEventListener("DOMContentLoaded", function () {

var modal = new bootstrap.Modal(document.getElementById("{{ session('Modal') }}"));

modal.show();

});

</script>

@endif

@section('main')

<div class="container">

<div class="row">

<div class="col-md-9">

<h3 class="h3 mt-2">Manutenção de Usuários do Sistema</h3>

</div>

<div class="col-md-3">

<!-- Button trigger modal Cadastro Novo Usuário-->

<button type="button" class="btn btn-outline-primary btn-sm mb-2 mt-2" id='btnNovoUsuario' data-bs-toggle="modal" data-bs-target="#ModalNovoUsuario" > Novo Usuário </button>

</div>

</div>

<span> <!-- Mensagens de Erro e Sucesso -->

@if (session()->has('sucesso'))

<div class="alert alert-success">

{{ session('sucesso') }}

</div>

{{ session()->forget('sucesso') }}

@endif

@if (session('erro'))

<div class="alert alert-danger">

{{ session('erro') }}

</div>

{{ session()->forget('erro') }}

@endif

</span>

<!-- Modal Novo Usuário -->

<div class="modal fade" id="ModalNovoUsuario">

<div class="modal-dialog modal-dialog-centered">

<div class="modal-content">

<div class="modal-header">

<h4 class="modal-title" id="ModalNovoUsuario">Cadastrar Novo Usuário</h4>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<form id="formNovoUsuario" action="/user/salvarnovo" method="post">

@csrf

<div class="modal-body">

@if ($errors->any())

<div id="msg-erros" class="alert alert-danger">

<ul>

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

<li>{{ $error }}</li>

@endforeach

</ul>

</div>

@endif

<div class="form-group">

<label for="email" class="col-form-label">Email</label>

<input type="text" class="form-control" name="email" id="email" value="{{ old('email') }}" required>

</div>

<div class="form-group">

<label for="name" class="col-form-label">Nome Completo</label>

<input type="text" class="form-control" name="name" id="name" value="{{ old('name') }}" required>

</div>

<div class="form-group">

<label for="password" class="col-form-label">Senha</label>

<input type="password" class="form-control" name="password" id="password" placeholder="{{ session('Modal') === 'ModalNovoUsuario'? 'Por segurança, digite a senha novamente.' : 'Digite a senha' }}" required>

</div>

<div class="form-group">

<label for="permissao" class="col-form-label">Tipo de Permissão</label>

<select class="form-control" name="permissao" id="permissao">

<option value="Selecione" {{ old('permissao') == 'Selecione' ? 'selected' : '' }} >Selecione</option>

<option value="Admin" {{ old('permissao') == 'Admin' ? 'selected' : '' }}>Administrador</option>

<option value="Normal" {{ old('permissao') == 'Normal' ? 'selected' : '' }}>Normal</option>

<option value="Leitura" {{ old('permissao') == 'Leitura' ? 'selected' : '' }}>Somente Leitura</option>

</select>

</div>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancelar</button>

<button type="submit" id="btnConfirmar" class="btn btn-outline-primary">Salvar</button>

</div>

</form>

</div>

</div>

</div> <!-- Fim ModalNovoUsuario -->

@endsection

Já é possível testar o funcionamento. Salve os arquivos e teste. Na interface da Aplicação, clique em Usuários e depois no botão Novo Usuário.

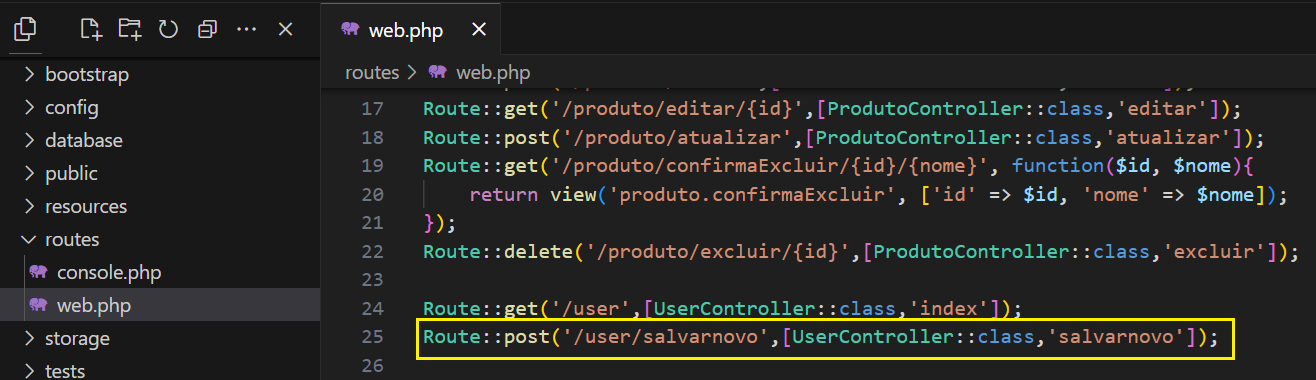
Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

1. **Configurar a rota e receber os dados do Formulário**

Edite o arquivo routes\web.php, acrescentando a seguinte linha:

Route::post('/user/salvarnovo',[UserController::class,'salvarnovo']);



Como especificamos no Form action, o método que vai receber os dados do Formulário e gravá-los no banco é salvarnovo. Assim, acrescente o método salvarnovo() na classe UserController, arquivo app\Http\Controllers\UserController.php com o seguinte código:

public function salvarnovo(Request $req){

//Coloco uma flag na Sessão Flash (só vale até a próxima requisição) para

//indicar qual modal na view deve ser reaberto caso aconteça erro de validação

session()->flash('Modal','ModalNovoUsuario');

// Validação com mensagens personalizadas

$req->validate([

'email' => 'required|max:255|unique:users,email',

'name' => 'required|max:255',

'password' => 'required|min:4|max:255',

'permissao' => 'required|in:Admin,Normal,Leitura',

], [

'email.unique' => 'Já existe um Usuário cadastrado com este Email!',

'email.required|max' => 'O campo Email é obrigatório e deve ter no máximo 255 caracteres!',

'name.\*' => 'O campo Nome é obrigatório e deve ter no máximo 255 caracteres!',

'password.\*' => 'O campo Senha é obrigatório e deve ter no mínimo 4 caracteres!',

'permissao.\*' => 'Selecione uma permissão!'

]

);

//Grava os dados no Banco

User::create($req->all());

return redirect('/user')->with('sucesso', 'Usuário Cadastrado.');

}//Fim salvarnovo()

O método acima faz a validação dos dados do Formulário através do validate() do Laravel, já utilizado antes. Note aqui a validação do campo email, o qual será utilizado para “logar” na Aplicação e deve ser único. Ou seja, não posso ter dois ou mais usuários com o mesmo login. Assim, é preciso verificar se já não existe um usuário cadastrado anteriormente com este email. O validate() do Laravel já faz isso através da validação unique:users,email, como colocado no código anterior. Faça o teste. Tente cadastrar um novo usuário com um email já cadastrado!

Também é importante notar no início do método a linha de comando:

session()->flash('Modal','ModalNovoUsuario');

na qual setamos uma flag na Sessão para indicar qual Modal deverá ser reaberto caso tenhamos que reapresentar o Modal por algum erro de validação. Aqui estou utilizando a Sessão flash do Laravel, a qual só tem validade até a próxima requisição. Depois disso, o Laravel já a removerá.

Neste ponto, já é possível cadastrar um Usuário. Então, salve tudo, abra o Navegador e aponte para:

http:8000//user Clique no botão Novo Usuário para abrir o Modal com o Formulário de Cadastro. Preencha com os dados e clique no botão Salvar.

Interface gráfica do usuário, Texto, Aplicativo

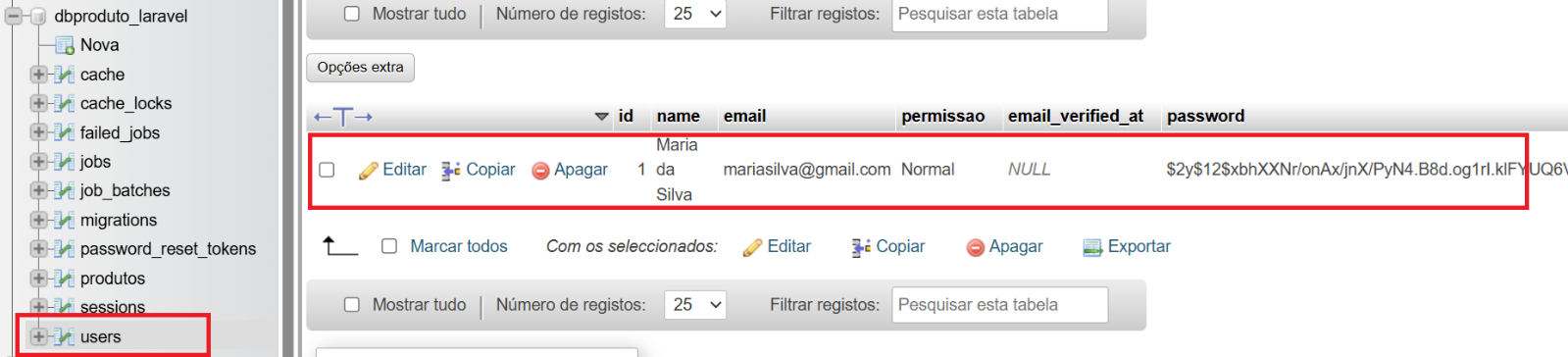
O conteúdo gerado por IA pode estar incorreto.

Se você digitou todos os dados corretamente para passar pelas validações, o usuário será cadastrado e a seguinte tela será:

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Note que o usuário foi cadastrado, mas não aparece na Listagem de Usuários. Isso é feito no próximo item. Mas você pode verificar no Banco de Dados se o usuário foi cadastrado na tabela users. Veja:



1. **Montar a Lista de Usuários Cadastrados**

Troque todo o conteúdo do arquivo resources\views\user\index.blade.php pelo seguinte:

@extends('layout.site')

@if ($errors->any())

<script>

document.addEventListener("DOMContentLoaded", function () {

var modal = new bootstrap.Modal(document.getElementById("{{ session('Modal') }}"));

modal.show();

});

</script>

@endif

@section('main')

<div class="container">

<div class="row">

<div class="col-md-9">

<h3 class="h3 mt-2">Manutenção de Usuários do Sistema</h3>

</div>

<div class="col-md-3">

<!-- Button trigger modal Cadastro Novo Usuário-->

<button type="button" class="btn btn-outline-primary btn-sm mb-2 mt-2" id='btnNovoUsuario' data-bs-toggle="modal" data-bs-target="#ModalNovoUsuario" > Novo Usuário </button>

</div>

</div>

<span> <!-- Mensagens de Erro e Sucesso -->

@if (session()->has('sucesso'))

<div class="alert alert-success">

{{ session('sucesso') }}

</div>

{{ session()->forget('sucesso') }}

@endif

@if (session('erro'))

<div class="alert alert-danger">

{{ session('erro') }}

</div>

{{ session()->forget('erro') }}

@endif

</span>

**<div class="table-responsive">**

**<table class="table table-hover table-sm mt-2">**

**<thead>**

**<tr class="table-info">**

**<th class="info">Nome Completo</th>**

**<th class="info">Email(Login)</th>**

**<th class="info">Tipo de Permissão</th>**

**<th class="info"></th>**

**</tr>**

**</thead>**

**<tbody>**

**@foreach($registros as $registro)**

**<tr>**

**<td>{{ $registro->name }}</td>**

**<td>{{ $registro->email }}</td>**

**<td>{{ $registro->permissao }}</td>**

**<td style="text-align: right;">**

**<!-- Button trigger modal Editar-->**

**<button type="button" class="btn btn-outline-info btn-sm mr-2" id="btnEditarUsuario" data-bs-toggle="modal" data-bs-target="#ModalEditarUsuario" data-ided="{{ $registro->id }}" data-name="{{ $registro->name }}" data-email="{{ $registro->email }}" data-permissao="{{ $registro->permissao }}">Editar</button>**

**<!-- Button trigger modal RessetarSenha-->**

**<button type="button" class="btn btn-outline-warning btn-sm mr-2" data-bs-toggle="modal" data-bs-target="#ModalRessetarSenha" data-idrs="{{ $registro->id }}" data-namers="{{ $registro->name }}" data-emailrs="{{ $registro->email }}">Ressetar Senha</button>**

**<!-- Button trigger modal Excluir-->**

**<button type="button" class="btn btn-outline-danger btn-sm" data-bs-toggle="modal" data-bs-target="#ModalExcluirUsuario" data-idex="{{ $registro->id }}" data-nameex="{{ $registro->name }}" data-emailex="{{ $registro->email }}">Excluir</button>**

**</td>**

**</tr>**

**@endforeach**

**</tbody>**

**</table>**

**</div>**

<!-- Modal Novo Usuário -->

<div class="modal fade" id="ModalNovoUsuario">

<div class="modal-dialog modal-dialog-centered">

<div class="modal-content">

<div class="modal-header">

<h4 class="modal-title" id="ModalNovoUsuario">Cadastrar Novo Usuário</h4>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<form id="formNovoUsuario" action="/user/salvarnovo" method="post">

@csrf

<div class="modal-body">

@if ($errors->any())

<div id="msg-erros" class="alert alert-danger">

<ul>

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

<li>{{ $error }}</li>

@endforeach

</ul>

</div>

@endif

<div class="form-group">

<label for="email" class="col-form-label">Email</label>

<input type="text" class="form-control" name="email" id="email" value="{{ old('email') }}" required>

</div>

<div class="form-group">

<label for="name" class="col-form-label">Nome Completo</label>

<input type="text" class="form-control" name="name" id="name" value="{{ old('name') }}" required>

</div>

<div class="form-group">

<label for="password" class="col-form-label">Senha</label>

<input type="password" class="form-control" name="password" id="password" placeholder="{{ session('Modal') === 'ModalNovoUsuario'? 'Por segurança, digite a senha novamente.' : 'Digite a senha' }}" required>

</div>

<div class="form-group">

<label for="permissao" class="col-form-label">Tipo de Permissão</label>

<select class="form-control" name="permissao" id="permissao">

<option value="Selecione" {{ old('permissao') == 'Selecione' ? 'selected' : '' }} >Selecione</option>

<option value="Admin" {{ old('permissao') == 'Admin' ? 'selected' : '' }}>Administrador</option>

<option value="Normal" {{ old('permissao') == 'Normal' ? 'selected' : '' }}>Normal</option>

<option value="Leitura" {{ old('permissao') == 'Leitura' ? 'selected' : '' }}>Somente Leitura</option>

</select>

</div>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancelar</button>

<button type="submit" id="btnConfirmar" class="btn btn-outline-primary">Salvar</button>

</div>

</form>

</div>

</div>

</div> <!-- Fim ModalNovoUsuario -->

@endsection

Note que o código acrescentado está em vermelho. Nele, é feito a listagem em tabela dos usuários cadastrados, juntamente com os botões de Editar, Ressetar Senha e Excluir.

Salve e teste.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

**Explicando o Código:**

@foreach($registros as $registro)

. . .

@endforeach

O trecho de código dentro do comando Laravel @foreach ... @endforeach acima é um laço de repetição que percorre o array de registros contendo os usuários que veio do Controlador. Dentro deste laço está sendo montada a tabela que exibe as informações de cada Usuário e os botões Editar, Ressetar Senha e Excluir.

<td>{{ $registro->name }}</td>

<td>{{ $registro->email }}</td>

<td>{{ $registro->permissao }}</td>

As linhas de código acima montam as colunas com as informações de nome, email e permissão de cada linha da tabela. Note que $registro é um Objeto da Classe app\Models\User e note a necessidade de usar {{ }}, pois estamos inserindo código Laravel/PHP em um comando HTML.

<!-- Button trigger modal Editar-->

<button type="button" class="btn btn-outline-info btn-sm mr-2" id="btnEditarUsuario" data-bs-toggle="modal" data-bs-target="#ModalEditarUsuario" data-ided="{{ $registro->id }}" data-name="{{ $registro->name }}" data-email="{{ $registro->email }}" data-permissao="{{ $registro->permissao }}">Editar

</button>

O trecho anterior é responsável por montar o botão Editar, o qual é responsável por mostrar o Modal de Edição dos dados do Usuário. Note as instruções data-bs-toggle="modal" e data-bs-target="#ModalEditarUsuario" que são atributos css do Bootstrap e que são responsáveis por identificar o Modal que será acionado(#ModalEditarUsuario). Note também os atributos data-ided, data-name, data-email e data-permissao os quais contém as informações do usuário que serão carregadas para o Formulário(via Javascript). Dessa forma, não é necessário um acesso ao Banco de Dados (backend) para carregar as informações do Usuário. O mesmo esquema foi adotado para os botões Ressetar Senha e Excluir.

1. **Implementando as funcionalidades dos botões Editar, Ressetar Senha e Excluir**

Na interface anterior, os botões já foram colocados, porém ainda não funcionam, pois falta o Modal de cada interface. Assim, troque todo o conteúdo do arquivo resources\views\user\index.blade.php pelo seguinte:

@extends('layout.site')

@if ($errors->any())

<script>

document.addEventListener("DOMContentLoaded", function () {

var modal = new bootstrap.Modal(document.getElementById("{{ session('Modal') }}"));

modal.show();

});

</script>

@endif

@section('main')

<div class="container">

<div class="row">

<div class="col-md-9">

<h3 class="h3 mt-2">Manutenção de Usuários do Sistema</h3>

</div>

<div class="col-md-3">

<!-- Button trigger modal Cadastro Novo Usuário-->

<button type="button" class="btn btn-outline-primary btn-sm mb-2 mt-2" id='btnNovoUsuario' data-bs-toggle="modal" data-bs-target="#ModalNovoUsuario" > Novo Usuário </button>

</div>

</div>

<span> <!-- Mensagens de Erro e Sucesso -->

@if (session()->has('sucesso'))

<div class="alert alert-success">

{{ session('sucesso') }}

</div>

{{ session()->forget('sucesso') }}

@endif

@if (session('erro'))

<div class="alert alert-danger">

{{ session('erro') }}

</div>

{{ session()->forget('erro') }}

@endif

</span>

<div class="table-responsive">

<table class="table table-hover table-sm mt-2">

<thead>

<tr class="table-info">

<th class="info">Nome Completo</th>

<th class="info">Email(Login)</th>

<th class="info">Tipo de Permissão</th>

<th class="info"></th>

</tr>

</thead>

<tbody>

@foreach($registros as $registro)

<tr>

<td>{{ $registro->name }}</td>

<td>{{ $registro->email }}</td>

<td>{{ $registro->permissao }}</td>

<td style="text-align: right;">

<!-- Button trigger modal Editar-->

<button type="button" class="btn btn-outline-info btn-sm mr-2" id="btnEditarUsuario" data-bs-toggle="modal" data-bs-target="#ModalEditarUsuario" data-ided="{{ $registro->id }}" data-name="{{ $registro->name }}" data-email="{{ $registro->email }}" data-permissao="{{ $registro->permissao }}">Editar</button>

<!-- Button trigger modal RessetarSenha-->

<button type="button" class="btn btn-outline-warning btn-sm mr-2" data-bs-toggle="modal" data-bs-target="#ModalRessetarSenha" data-idrs="{{ $registro->id }}" data-namers="{{ $registro->name }}" data-emailrs="{{ $registro->email }}">Ressetar Senha</button>

<!-- Button trigger modal Excluir-->

<button type="button" class="btn btn-outline-danger btn-sm" data-bs-toggle="modal" data-bs-target="#ModalExcluirUsuario" data-idex="{{ $registro->id }}" data-nameex="{{ $registro->name }}" data-emailex="{{ $registro->email }}">Excluir</button>

</td>

</tr>

@endforeach

</tbody>

</table>

</div>

<!-- Modal Novo Usuário -->

<div class="modal fade" id="ModalNovoUsuario">

<div class="modal-dialog modal-dialog-centered">

<div class="modal-content">

<div class="modal-header">

<h4 class="modal-title" id="ModalNovoUsuario">Cadastrar Novo Usuário</h4>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<form id="formNovoUsuario" action="/user/salvarnovo" method="post">

@csrf

<div class="modal-body">

@if ($errors->any())

<div id="msg-erros" class="alert alert-danger">

<ul>

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

<li>{{ $error }}</li>

@endforeach

</ul>

</div>

@endif

<div class="form-group">

<label for="email" class="col-form-label">Email</label>

<input type="text" class="form-control" name="email" id="email" value="{{ old('email') }}" required>

</div>

<div class="form-group">

<label for="name" class="col-form-label">Nome Completo</label>

<input type="text" class="form-control" name="name" id="name" value="{{ old('name') }}" required>

</div>

<div class="form-group">

<label for="password" class="col-form-label">Senha</label>

<input type="password" class="form-control" name="password" id="password" placeholder="{{ session('Modal') === 'ModalNovoUsuario'? 'Por segurança, digite a senha novamente.' : 'Digite a senha' }}" required>

</div>

<div class="form-group">

<label for="permissao" class="col-form-label">Tipo de Permissão</label>

<select class="form-control" name="permissao" id="permissao">

<option value="Selecione" {{ old('permissao') == 'Selecione' ? 'selected' : '' }} >Selecione</option>

<option value="Admin" {{ old('permissao') == 'Admin' ? 'selected' : '' }}>Administrador</option>

<option value="Normal" {{ old('permissao') == 'Normal' ? 'selected' : '' }}>Normal</option>

<option value="Leitura" {{ old('permissao') == 'Leitura' ? 'selected' : '' }}>Somente Leitura</option>

</select>

</div>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancelar</button>

<button type="submit" id="btnConfirmar" class="btn btn-outline-primary">Salvar</button>

</div>

</form>

</div>

</div>

</div> <!-- Fim ModalNovoUsuario -->

<!-- Modal Editar Usuário -->

<div class="modal fade" id="ModalEditarUsuario">

<div class="modal-dialog modal-dialog-centered">

<div class="modal-content">

<div class="modal-header">

<h4 class="modal-title" id="ModalEditarUsuario">Editar Cadastro do Usuário</h4>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<form id="formEditarUsuario" action="/user/salvarEditado" method="post">

@csrf

<div class="modal-body">

@if ($errors->any())

<div id="msg-erros-editar" class="alert alert-danger">

<ul>

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

<li>{{ $error }}</li>

@endforeach

</ul>

</div>

@endif

<div class="form-group">

<input type="hidden" name="id" id="ided">

</div>

<div class="form-group">

<label for="emailed" class="col-form-label">Email</label>

<input type="text" class="form-control" name="email" id="emailed" required>

</div>

<div class="form-group">

<label for="nameed" class="col-form-label">Nome Completo</label>

<input type="text" class="form-control" name="name" id="nameed" required>

</div>

<div class="form-group">

<label for="permissaoed" class="col-form-label">Tipo de Permissão</label>

<select class="form-control" name="permissao" id="permissaoed">

<option value="Selecione">Selecione</option>

<option value="Admin">Administrador</option>

<option value="Normal">Normal</option>

<option value="Leitura">Somente Leitura</option>

</select>

</div>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancelar</button>

<button type="submit" class="btn btn-outline-primary">Confirmar</button>

</div>

</form>

</div>

</div>

</div> <!--Fim ModalEditarUsuario -->

<!-- Modal Ressetar Senha Usuário -->

<div class="modal fade" id="ModalRessetarSenha">

<div class="modal-dialog modal-dialog-centered">

<div class="modal-content">

<div class="modal-header">

<h4 class="modal-title" id="ModalRessetarSenha">Ressetar a Senha do Usuário</h4>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close">

</div>

<form id="formRessetarSenha" action="/user/atualizarSenha" method="post">

@csrf

<div class="modal-body">

@if ($errors->any())

<div id="msg-erros-ressetar-senha" class="alert alert-danger">

<ul>

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

<li>{{ $error }}</li>

@endforeach

</ul>

</div>

@endif

<div class="form-group">

<input type="hidden" name="id" id="idrs">

</div>

<div class="form-group">

<label for="emailrs" class="col-form-label">Email</label>

<input type="text" class="form-control" id="emailrsshow" readonly disabled>

<input type="hidden" class="form-control" name="email" id="emailrshidden">

</div>

<div class="form-group">

<label for="namers" class="col-form-label">Nome Completo</label>

<input type="text" class="form-control" id="namersshow" readonly disabled>

<input type="hidden" class="form-control" name="name" id="namershidden">

</div>

<div class="form-group">

<label for="senhars" class="col-form-label">Senha Provisória</label>

<input type="password" class="form-control" name="password" id="senhars" value="" >

</div>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-primary" data-bs-dismiss="modal">Cancelar</button>

<button type="submit" class="btn btn-outline-danger">Confirmar</button>

</div>

</form>

</div>

</div>

</div> <!--Fim ModalRessetarSenha -->

<!-- Modal Excluir Usuário -->

<div class="modal fade" id="ModalExcluirUsuario">

<div class="modal-dialog modal-dialog-centered">

<div class="modal-content">

<div class="modal-header">

<h4 class="modal-title">Confirma Exclusão do Usuário?</h4>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close">

</div>

<form id="formExcluirUsuario" action="/user/excluir" method="post">

@csrf

<div class="modal-body">

@if ($errors->any())

<div id="msg-erros-excluir" class="alert alert-danger">

<ul>

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

<li>{{ $error }}</li>

@endforeach

</ul>

</div>

@endif

<div class="form-group">

<input type="hidden" name="id" id="idex">

</div>

<div class="form-group">

<label for="emailrs" class="col-form-label">Email</label>

<input type="text" class="form-control" id="emailexshow" readonly disabled>

<input type="hidden" class="form-control" name="email" id="emailexhidden">

</div>

<div class="form-group">

<label for="namers" class="col-form-label">Nome Completo</label>

<input type="text" class="form-control" id="nameexshow" readonly disabled>

<input type="hidden" class="form-control" name="name" id="nameexhidden">

</div>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-primary" data-bs-dismiss="modal">Cancelar</button>

<button type="submit" id="btnExcluir" name="btnExcluir" value="btnExcluir" class="btn btn-outline-danger">Confirmar</button>

</div>

</form>

</div>

</div>

</div> <!--Fim ModalExcluirUsuario -->

</div>

<script>

//Aplica autofocus no campo Email do Modal Novo Usuário

const modal = document.getElementById('ModalNovoUsuario');

modal.addEventListener('shown.bs.modal', function () {

document.getElementById('email').focus();

});

</script>

<script>

//Passa parâmetro para o Modal Editar Usuário

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

if (event.relatedTarget) { // Modal foi aberto por um clique em botão Editar

var button = $(event.relatedTarget); //Botão que ativou o Modal

var recipientid = button.data('ided');

var recipientName = button.data('name');

var recipientEmail = button.data('email');

var recipientPermissao = button.data('permissao');

// Remove a div de mensagens de erro dentro do Modal

const mensagensErro = document.querySelector('#msg-erros-editar');

if (mensagensErro) {

mensagensErro.remove();

}

}else{

var recipientid = "{{ old('id') }}";

var recipientName = "{{ old('name') }}";

var recipientEmail = "{{ old('email') }}";

var recipientPermissao = "{{ old('permissao') }}";

}

//alert(recipientid);

//alert(recipientName);

var modal = $(this);

//Pega o valor armazenado no recipient e substitui no modal onde o #id = o id do campo no modal

modal.find('#ided').val(recipientid);

modal.find('#nameed').val(recipientName);

modal.find('#emailed').val(recipientEmail);

modal.find('#permissaoed').val(recipientPermissao);

$('#emailed').focus();

});

</script>

<script>

//Passa parâmetro para o Modal Ressetar Senha

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

if (event.relatedTarget) { // Modal foi aberto por um clique em botão Ressetar Senha

var button = $(event.relatedTarget); //Botão que ativou o Modal

var recipientid = button.data('idrs');

var recipientName = button.data('namers');

var recipientEmail = button.data('emailrs');

// Remove a div de mensagens de erro dentro do Modal

const mensagensErro = document.querySelector('#msg-erros-ressetar-senha');

if (mensagensErro) {

mensagensErro.remove();

}

}else{ //Modal aberto automaticamente para reapresentação do Formulário

var recipientid = "{{ old('id') }}";

var recipientName = "{{ old('name') }}";

var recipientEmail = "{{ old('email') }}";

}

//alert(recipientid);

//alert(recipientName);

var modal = $(this);

//Pega o valor armazenado no recipient e substitui no modal onde o #id = o id do campo no modal

modal.find('#idrs').val(recipientid);

modal.find('#namersshow').val(recipientName);

modal.find('#namershidden').val(recipientName);

modal.find('#emailrsshow').val(recipientEmail);

modal.find('#emailrshidden').val(recipientEmail);

//Coloca o foco no campo Senha

const modalRessetarSenha = document.querySelector('#senhars');

modalRessetarSenha.focus();

});

</script>

<script>

//Passa parâmetro para o Modal Excluir Usuario

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

if (event.relatedTarget) { // Modal foi aberto por um clique em botão Excluir

var button = $(event.relatedTarget); //Botão que ativou o Modal

var recipientid = button.data('idex');

var recipientName = button.data('nameex');

var recipientEmail = button.data('emailex');

// Remove a div de mensagens de erro dentro do Modal

const mensagensErro = document.querySelector('#msg-erros-excluir');

if (mensagensErro) {

mensagensErro.remove();

}

}else{ //Modal aberto automaticamente para reapresentação do Formulário

var recipientid = "{{ old('id') }}";

var recipientName = "{{ old('name') }}";

var recipientEmail = "{{ old('email') }}";

}

//alert(recipientid);

//alert(recipientName);

var modal = $(this);

//Pega o valor armazenado no recipient e substitui no modal onde o #id = o id do campo no modal

modal.find('#idex').val(recipientid);

modal.find('#nameexshow').val(recipientName);

modal.find('#nameexhidden').val(recipientName);

modal.find('#emailexshow').val(recipientEmail);

modal.find('#emailexhidden').val(recipientEmail);

});

</script>

<script>

//Limpa o Formulário de Cadastro de Novo Usuário, caso tenha informações na Sessão

document.querySelector('#btnNovoUsuario').addEventListener('click', function() {

// Seleciona todos os inputs dentro do modal

const inputs = document.querySelectorAll('#ModalNovoUsuario input');

const select = document.querySelector('#permissao');

const mensagensErro = document.querySelector('#msg-erros');

// Limpa cada input

inputs.forEach(input => {

if (input.name != "\_token"){ //Preciso disso senão remove o token csrf

input.value = ''; // Limpa o valor

}

if (input.name == "password"){//Troca o placeholder do campo de senha

input.placeholder = "Digite a Senha";

}

});

// Remove a div de mensagens de erro dentro do Modal

if (mensagensErro) {

mensagensErro.remove();

}

//Seta o select da Permissão para Selecione

select.value = 'Selecione';

// Abre o modal

});

</script>

<script>

//Limpa o Formulário de Editar Usuário, caso tenha informações na Sessão

document.querySelector('#EditarUsuario').addEventListener('click', function() {

// Seleciona todos os inputs dentro do modal

const mensagensErro = document.querySelector('#msg-erros-editar');

// Remove a div de mensagens de erro dentro do Modal

if (mensagensErro) {

mensagensErro.remove();

alert("removeu");

}

// Abre o modal

});

</script>

@endsection

Salve tudo e Teste. Note que o Modal associado a cada botão é aberto, porém sem dados! Isso porque o carregamento dos dados no Formulário é feito por Javascript/JQuery e falta carregar o JQuery. Isso é feito no próximo item.

1. **Baixar o JQuery usado para preenchimento do Modal**

Na tabela de listagem dos usuários montada anteriormente já programamos os botões Editar, Ressetar Senha e Excluir com as informações de cada usuário. Estas informações serão enviadas para o formulário dentro do Modal quando o botão for clicado. Para que este processo de preenchimento funcione é necessário usar o JQuery. Portanto, baixe o JQuery em <https://code.jquery.com/jquery-3.7.1.min.js> e salve na pasta public\js da Aplicação com o nome jquery-3.7.1.min.js

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Agora é necessário carregar o script. Isso deve ser feito no arquivo header.blade.php. Insira a sequinte linha no arquivo resources\views\layout\header.blade.php

<script src="{{ asset('js/jquery-3.7.1.min.js') }}"></script>

Texto

O conteúdo gerado por IA pode estar incorreto.

Salve tudo e Teste. Não esqueça de atualizar o Navegador (CTRL F5). Clique nos botões Editar, Ressetar Senha e Excluir. Note agora que as informações são preenchidas. O próximo passo é implementar

1. **Implementando as funcionalidades dos botões Editar, Ressetar Senha e Excluir**

Acrescente as rotas a seguir no arquivo routes\web.php:

Route::post('/user/salvarEditado',[UserController::class,'salvarEditado']);

Route::post('/user/atualizarSenha',[UserController::class,'atualizarSenha']);

Route::post('/user/excluir',[UserController::class,'excluir']);

Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

Troque o conteúdo do arquivo app\Http\Controllers\UserController.php pelo seguinte.

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use Illuminate\Validation\Rule;

use Illuminate\Support\Facades\Validator;

use Illuminate\Support\Facades\Hash;

use App\Models\User;

class UserController extends Controller

{

public function index(){

$registros = User::all();

if ($registros->count()==0){

session(['erro' => 'Nenhum Usuário Encontrado!']);

}

return view('user.index', compact('registros'));

}

public function salvarnovo(Request $req){

//Coloco uma flag na Sessão Flash (só vale até a próxima requisição) para

//indicar qual modal na view deve ser reaberto caso aconteça erro de validação

session()->flash('Modal','ModalNovoUsuario');

// Validação com mensagens personalizadas

$req->validate([

'email' => 'required|max:255|unique:users,email',

'name' => 'required|max:255',

'password' => 'required|min:4|max:255',

'permissao' => 'required|in:Admin,Normal,Leitura',

], [

'email.unique' => 'Já existe um Usuário cadastrado com este Email!',

'email.required|max' => 'O campo Email é obrigatório e deve ter no máximo 255 caracteres!',

'name.\*' => 'O campo Nome é obrigatório e deve ter no máximo 255 caracteres!',

'password.\*' => 'O campo Senha é obrigatório e deve ter no mínimo 4 caracteres!',

'permissao.\*' => 'Selecione uma permissão!'

]

);

//Grava os dados no Banco

User::create($req->all());

return redirect('/user')->with('sucesso', 'Usuário Cadastrado.');

}//Fim salvarnovo()

public function salvarEditado(Request $req){

//Coloco uma flag na Sessão Flash (só vale até a próxima requisição)

// Para indicar qual modal na view deve ser reaberto caso aconteça erro de validação

session()->flash('Modal','ModalEditarUsuario');

// Validação com mensagens personalizadas

$req->validate([

'email' => [

'required',

'max:255',

Rule::unique('users','email')->ignore($req->id),

],

'name' => 'required|max:255',

'permissao' => 'required|in:Admin,Normal,Leitura',

], [

'email.unique' => 'Já existe um Usuário cadastrado com este Email!',

'email.required|max' => 'O campo Email é obrigatório e deve ter no máximo 255 caracteres!',

'name.\*' => 'O campo Nome é obrigatório e deve ter no máximo 255 caracteres!',

'permissao.\*' => 'Selecione uma permissão!'

]

);

///Atualiza os dados no Banco

User::find($req->input('id'))->update($req->all());

return redirect('/user')->with('sucesso', 'Usuário Atualizado.');

}//Fim salvarEditado()

public function atualizarSenha(Request $req){

//Coloco uma flag na Sessão Flash (só vale até a próxima requisição)

// Para indicar qual modal na view deve ser reaberto caso aconteça erro de validação

session()->flash('Modal','ModalRessetarSenha');

// Validação com mensagens personalizadas

$validator = Validator::make($req->all(), [

'id' => 'required',

'email' => 'required|max:255',

'name' => 'required|max:255',

'password' => 'required|min:4|max:255',

], [

'id.required' => 'Id do usuário é necessário!',

'email.\*' => 'O campo Email é obrigatório e deve ter no máximo 255 caracteres!',

'name.\*' => 'O campo Nome é obrigatório e deve ter no máximo 255 caracteres!',

'password.\*' => 'O campo Senha é obrigatório e deve ter no mínimo 4 caracteres!',

]

);

//Validação se existe um usuário com estes id, email e nome, pois embora estes dados

// não sejam editáveis na interface, o usuário pode utilizar F12 e alterá-los

$validator->after(function ($validator) use ($req) {

$existe = User::where('id', $req->id)

->where('email', $req->email)

->where('name', $req->name)

->exists();

if (! $existe) {

$validator->errors()->add('usernaoexiste', 'Não existe um Usuário com essas informações!');

}

});

//Validação se foi alterada a senha

//Busca o usuário pelo ID

$user = User::find($req->id);

$validator->after(function ($validator) use ($req, $user) {

if (Hash::check($req->password, $user->password)) {

$validator->errors()->add('senhaigual', 'A nova senha é igual a senha atual.');

}

});

if ($validator->fails()) {

return redirect()->back()

->withErrors($validator)

->withInput();

}

//Passou pelas Validações, então

//Atualiza o campo password do usuário

User::find($req->input('id'))->update([

'password' => Hash::make($req->input('password')),

]);

return redirect('/user')->with('sucesso', 'Senha do Usuário Atualizada.');

}//Fim satualizarSenha()

public function excluir(Request $req){

//Coloco uma flag na Sessão Flash (só vale até a próxima requisição)

// Para indicar qual modal na view deve ser reaberto caso aconteça erro de validação

session()->flash('Modal','ModalExcluirUsuario');

// Validação com mensagens personalizadas

$validator = Validator::make($req->all(), [

'id' => 'required',

'email' => 'required|max:255',

'name' => 'required|max:255',

], [

'id.required' => 'Id do usuário é necessário!',

'email.\*' => 'O campo Email é obrigatório e deve ter no máximo 255 caracteres!',

'name.\*' => 'O campo Nome é obrigatório e deve ter no máximo 255 caracteres!',

]

);

//Validação se existe um usuário com estes id, email e nome, pois embora estes dados

// não sejam editáveis na interface, o usuário pode utilizar F12 e alterá-los

$validator->after(function ($validator) use ($req) {

$existe = User::where('id', $req->id)

->where('email', $req->email)

->where('name', $req->name)

->exists();

if (! $existe) {

$validator->errors()->add('usernaoexiste', 'Não existe um Usuário com essas informações!');

}

});

if ($validator->fails()) {

return redirect()->back()

->withErrors($validator)

->withInput();

}

//Passou pelas Validações, então exclui o usuário

User::find($req->input('id'))->delete();

return redirect('/user')->with('sucesso', 'Usuário Excluído.');

}//Fim excluir()

}

Salve tudo, atualize o Navegador (CRTL F5) e teste.