

SjouN

README: COMO
RODAR O PROJETO

INSTALL

composer install

npm install

npm run build

cp .env.example .env

php artisan key:generate

SETUP DATABASE

No arquivo .env, ajuste os valores que começam com DB_, como DB_CONNECTION, DB_HOST, etc.

RODAR MIGRATIONS E SEEDS

php artisan migrate --step --seed

RODAR O SERVIDOR

php artisan serve



INFNET 2024

Sbuv

APRESENTAÇÃO DE CONCLUSÃO DE
SEMESTRE

Por Luisa Hering Bell de Otero



SOBRE O PROJETO SBUN

Parte de um ambicioso esforço individual, o SBUN nasceu da necessidade de modernizar a gestão bibliotecária. O SBUN posiciona-se como parte do Projeto de Bloco que conclui o semestre de Engenharia Disciplinada de Softwares no Instituto infnet, graduação de Engenharia de Software.

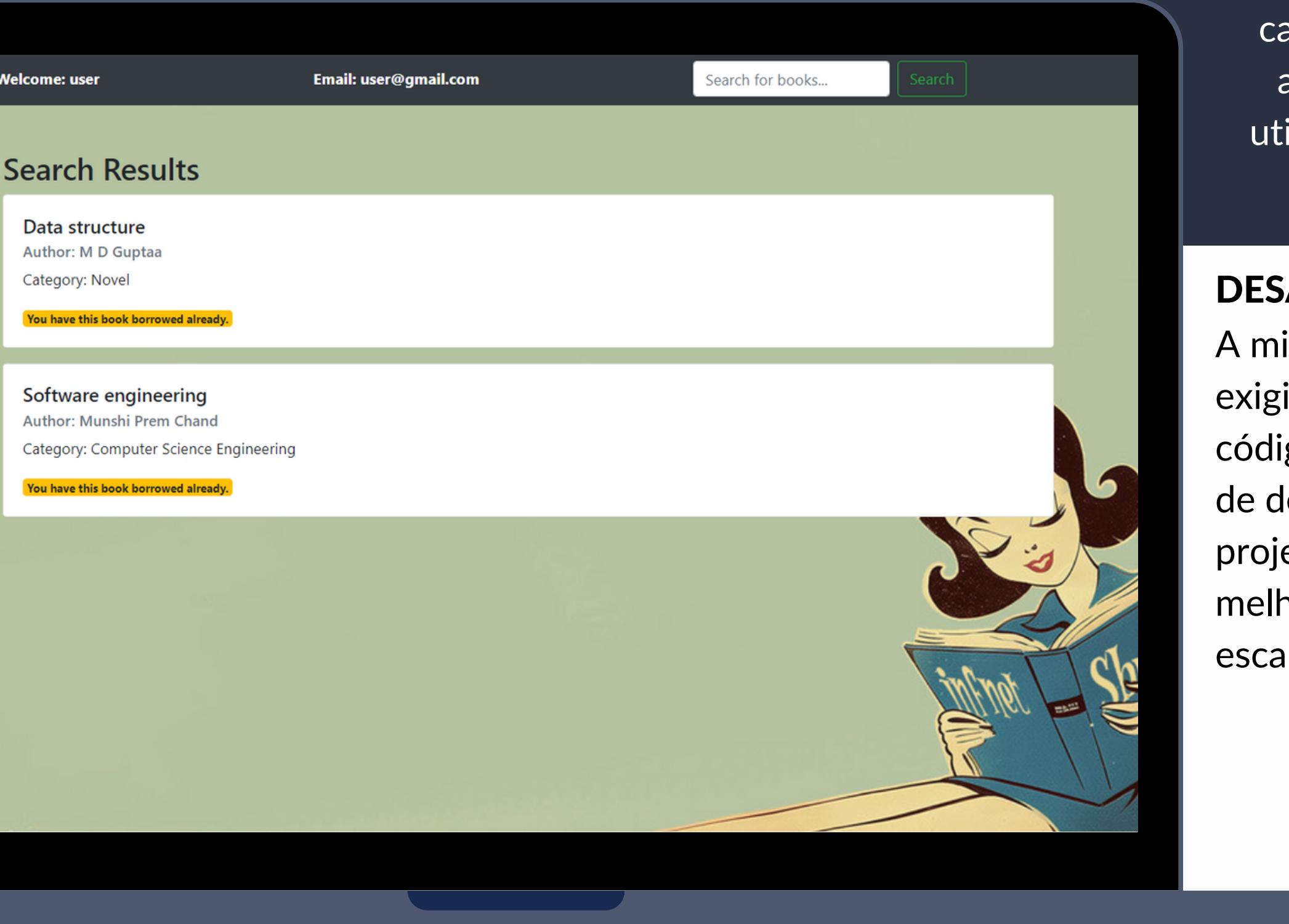
IMAGEM MAIOR:



O projeto SBUN configura-se como uma plataforma de gestão bibliotecária, destinada ao uso em bibliotecas públicas, com a finalidade de otimizar a interação entre o acervo disponível e seus usuários. Equipado com módulos essenciais para o gerenciamento de livros, empréstimos, devoluções, e pesquisa; o projeto cobre todas as necessidades fundamentais de operação de uma biblioteca moderna. Cada função, desenvolvido com um propósito específico dentro do contexto bibliotecário, contribui para a criação de um sistema “360°” capaz de gerenciar de maneira eficiente a circulação de materiais bibliotecários, trazendo uma experiência agradável tanto para o Bibliotecário quanto o usuário.



COMO COMEÇOU:



The screenshot shows a web application interface. At the top, there's a header with "Welcome: user" and "Email: user@gmail.com". A search bar contains "Search for books..." and a "Search" button. Below the header, the title "Search Results" is displayed. Two book entries are listed: "Data structure" by M D Gupta (Author: M D Gupta, Category: Novel) and "Software engineering" by Munshi Prem Chand (Author: Munshi Prem Chand, Category: Computer Science Engineering). Each entry includes a small thumbnail image of a person reading a book. A yellow banner at the bottom of each entry says "You have this book borrowed already."

O SBUN começou como um projeto monolítico, uma aplicação em PHP puro e HTML que embora fosse simples, carecia de escalabilidade. Agora, transformou-se em uma arquitetura moderna, separando front-end e back-end, utilizando o Laravel, que orquestra o back-end com PHP e uma interface de usuário reativa com JavaScript.

DESAFIOS DE REFATORAÇÃO:

A migração de um monólito para o Laravel exigiu uma reestruturação profunda do código, implementando melhores práticas de design de software e padrões de projeto que o Laravel promove para melhorar a manutenibilidade e escalabilidade.



Cada etapa do desenvolvimento foi uma oportunidade de aprendizado, ajustando a abordagem e a arquitetura para se alinhar com as melhores práticas e padrões emergentes da indústria.

COMO ELE SERÁ UTILIZADO?



Através de computadores disponibilizados na própria biblioteca, os usuários podem navegar pelo catálogo, verificar a disponibilidade de livros e fazer solicitações de empréstimo de maneira simples e rápida. Uma vez escolhido o livro, o usuário finaliza o processo de empréstimo no balcão, com a assistência de um bibliotecário, que confirma a transação no sistema. Além disso, o SBUN permite que os usuários consultem o status dos seus empréstimos atuais, incluindo a visualização de quaisquer atrasos na devolução. O SBUN é uma solução digital inovadora projetada especificamente para bibliotecas públicas, facilitando a interação entre os usuários e o acervo disponível.

COMO FUNCIONA?

O SBUN foi criado em PHP utilizando o framework Laravel, reconhecido por sua robustez e recursos de segurança de alta qualidade. Seguindo o modelo MVC (model-view-controller) para uma arquitetura de aplicativo limpa e o Eloquent ORM para interações eficientes com o banco de dados MySQL. A combinação do Tailwind com as blades do Laravel e a reatividade do Alpine.js cria uma interface de usuário fluida e intuitiva. A integridade visual e funcional é assegurada com a aplicação de padrões de design responsivo e interatividade cliente-side.

Esta fusão de tecnologias coloca o SBUN na vanguarda das aplicações web, oferecendo uma experiência de usuário que é ao mesmo tempo moderna, segura e altamente adaptável às necessidades emergentes de bibliotecas públicas e seus frequentadores.



STACK DE TECNOLOGIA

BACK-END ROBUSTO COM LARAVEL



Estruturação eficiente com o framework PHP Laravel para gerenciar o back-end, aproveitando seu ORM Eloquent, autenticação e segurança aprimoradas.

VERSIONAMENTO COM GIT



Uso do sistema de controle de versão Git para gerenciamento eficiente do código fonte e colaboração em equipe (se houver no futuro).



BANCO DE DADOS MYSQL



Armazenamento seguro e eficaz de dados com o sistema de gerenciamento de banco de dados MySQL, facilitando consultas e transações.

FRONT-END COM BLADE E ALPINE.JS



Combinação do Blade, o sistema de templates do Laravel, para renderização dinâmica, e Alpine.js para adicionar interatividade leve e responsiva.

ESTILOS COM TAILWIND CSS



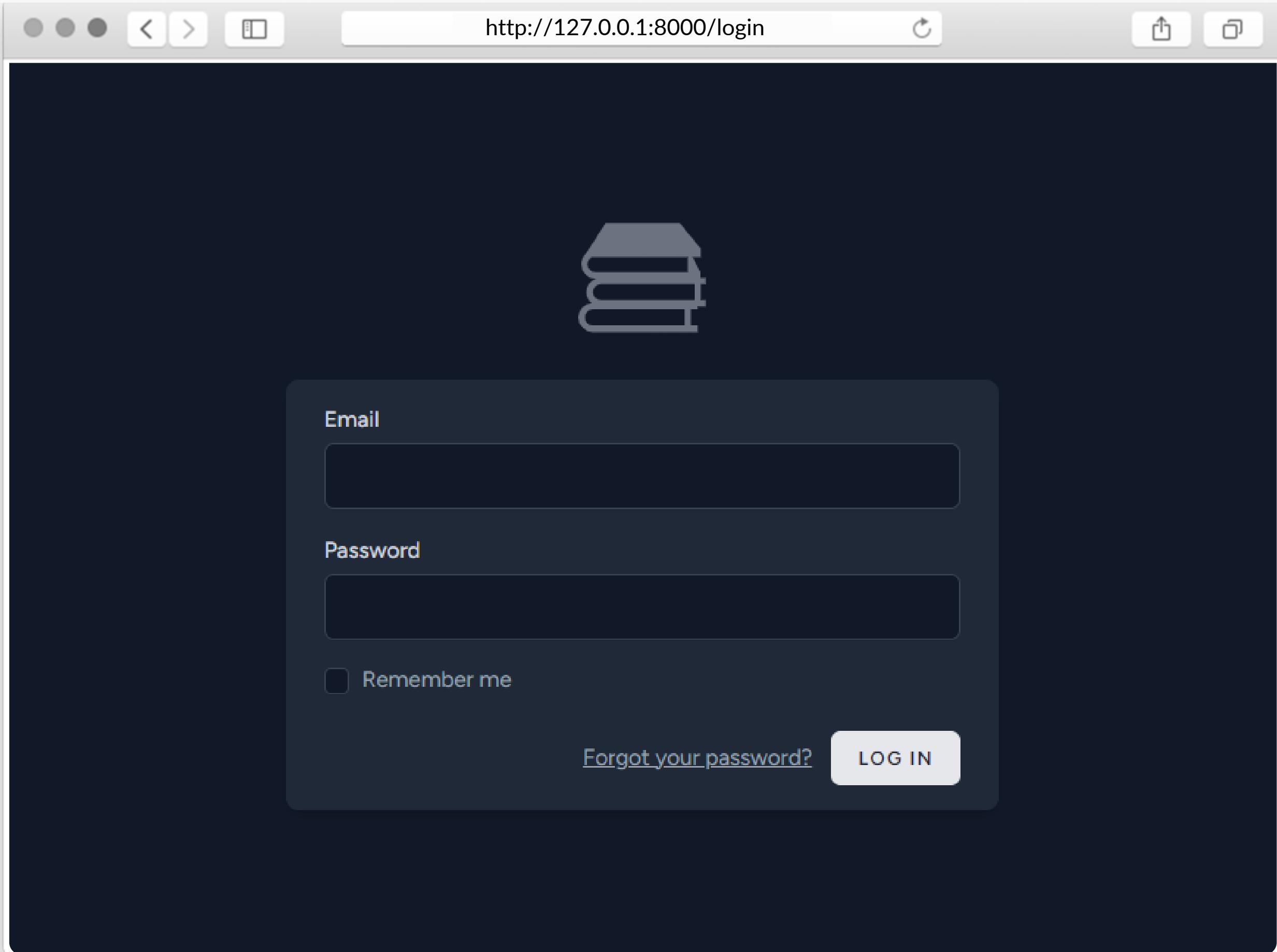
Utilização do framework de CSS Tailwind para um design responsivo e customizável, acelerando o desenvolvimento front-end.

TESTES COM PHPUNIT



Utilização do framework de CSS Tailwind para um design responsivo e customizável, acelerando o desenvolvimento front-end.

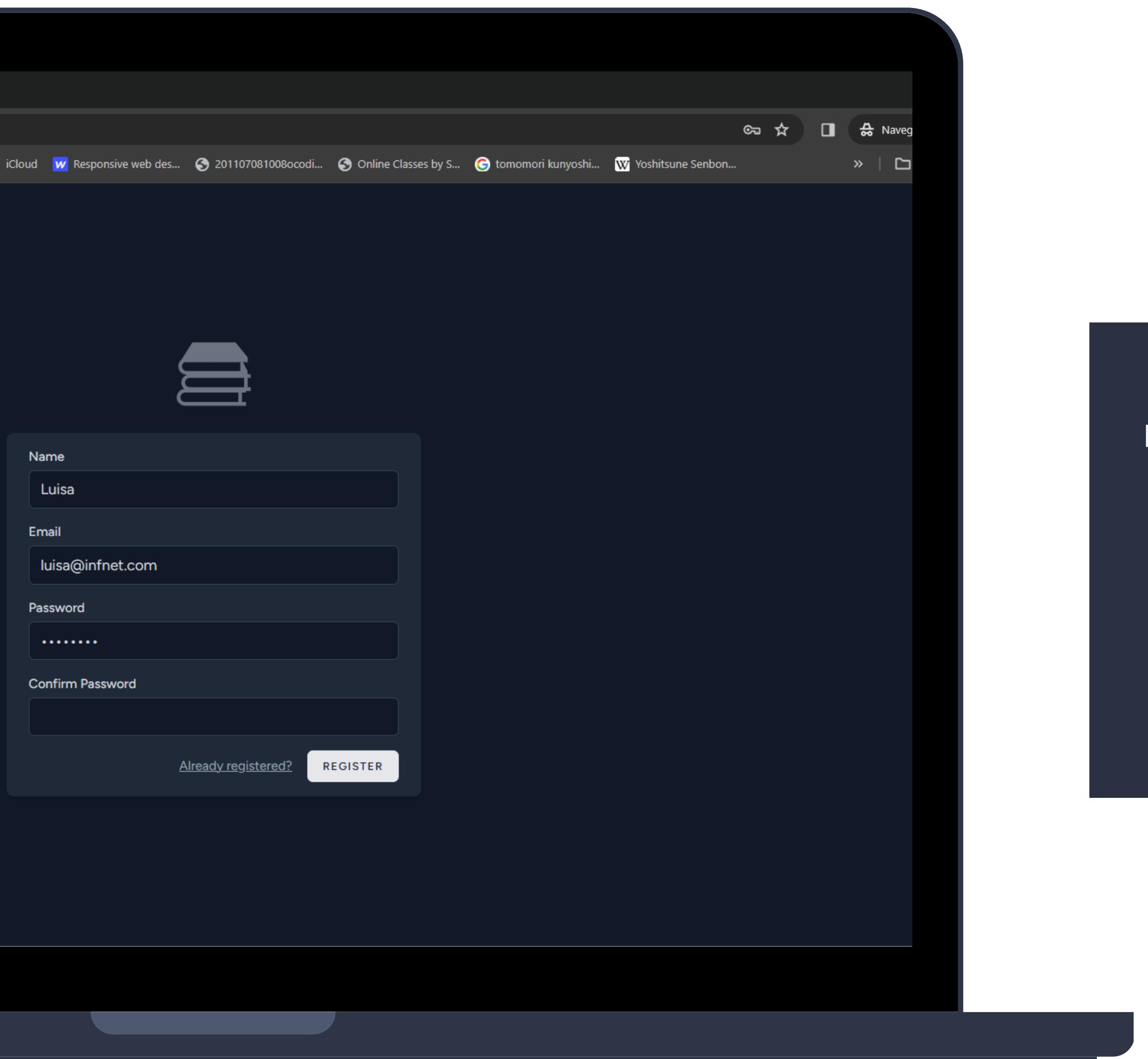
MACRO:



Login

um ponto de entrada seguro que estabelece a interação inicial com o sistema. Esta etapa realiza a comunicação direta com o banco de dados MySQL, aproveitando as robustas capacidades de autenticação fornecidas pelo Laravel, incluindo o uso do Laravel Sanctum para gestão de tokens API, reforçando a segurança através de um controle de acesso detalhado e personalizado. O design da UX ramifica-se em Login, Register e Forgot My Password.

Registrar

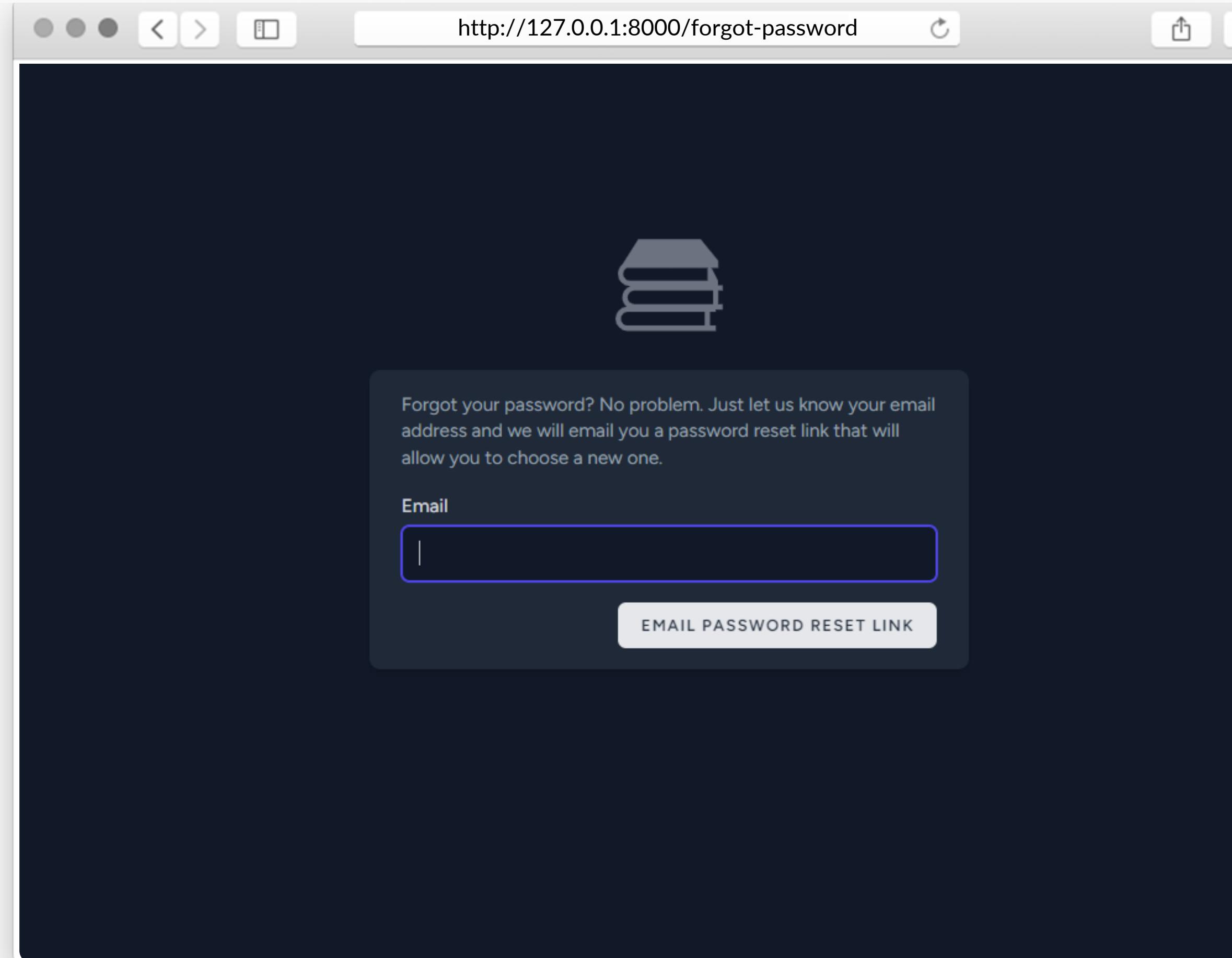


Tela fundamental, o usuário preenche os dados e o front valida esse input, garantindo a coleta de dados “saudáveis” para o cadastro subsequente no banco de dados.

Neste, guardamos a senha criptografada com Bcrypt, garantindo segurança antes de armazenar no banco de dados.

Esqueci minha senha

A funcionalidade "Esqueci minha senha" é projetada para contornar os problemas de esquecimento de senha, algo comum entre os usuários. Com a ajuda do Laravel, ao fornecer seu e-mail, o usuário pode solicitar a redefinição de senha. O sistema, então, envia um link seguro via e-mail, permitindo que o usuário crie uma nova senha de forma confidencial e segura. Isso assegura que o acesso à conta possa ser recuperado sem comprometer a segurança da conta do usuário no SBUN.



DASHBOARD PARA USUÁRIO FINAL

A tela “casa” da nossa aplicação abriga uma das visões mais importantes do nosso projeto, os Livros Emprestados do usuário. Além de comportar a navbar que auxilia o usuário na navegação do SBUN.

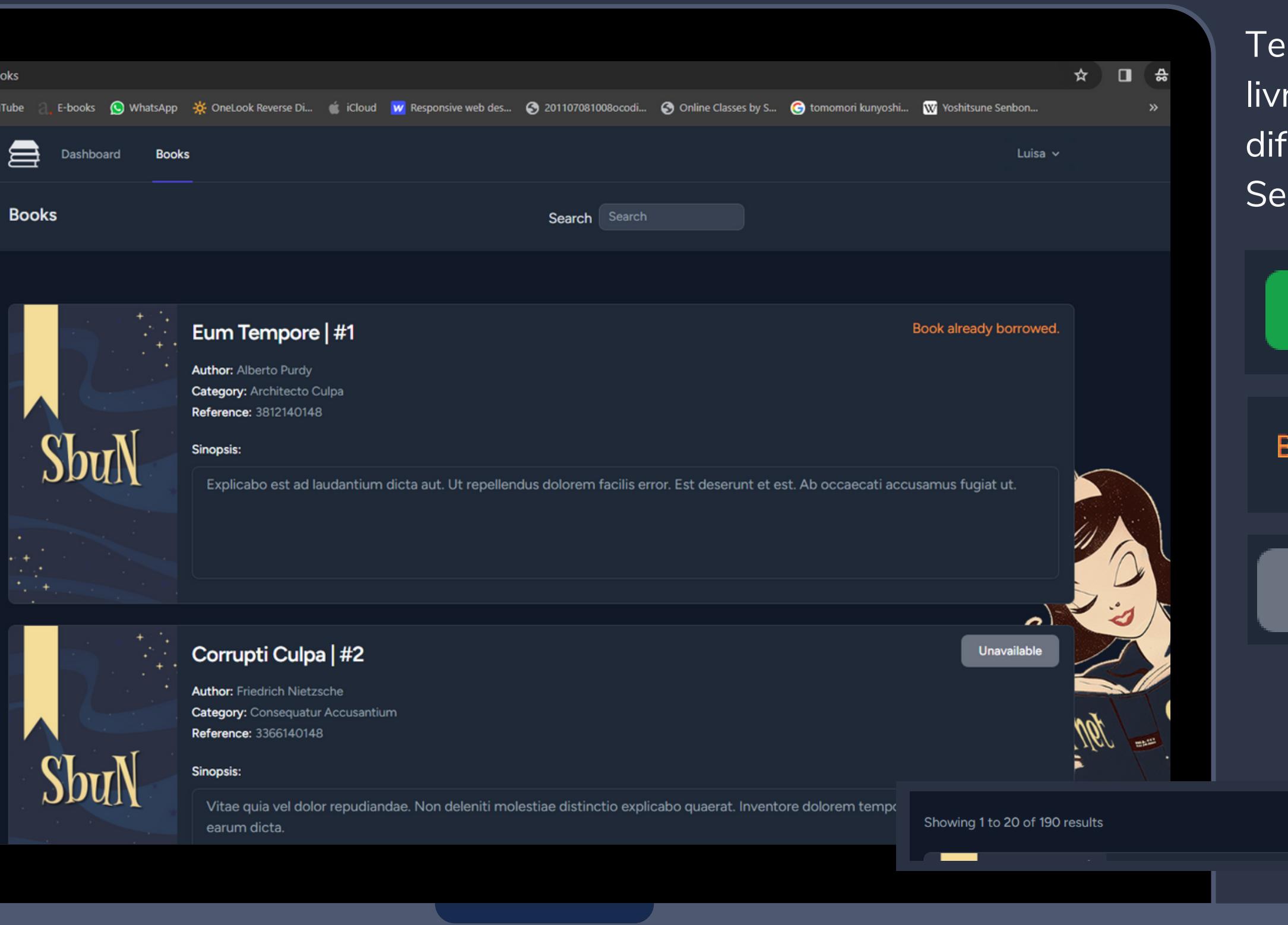
The screenshot displays the user dashboard interface. At the top, there's a navigation bar with a logo, the URL '7.0.0.1:8000/dashboard', and tabs for 'Dashboard' and 'Books'. Below the navigation is a header with the text 'Dashboard' and a sub-section titled 'My Borrowed Books' with a 'View all items' link. This section lists five books with their titles, authors, and status buttons: 'Além do bem e do mal' by Friedrich Nietzsche with a 'RETURN' button; 'Adipisci Possimus' by Alda Waters Sr. with a 'RETURN' button; 'O estrangeiro' by Albert Camus with a 'RETURN' button; 'Corrupti Autem' by Albert Camus with a 'PENDING' button; and 'Eum Tempore' by Alberto Purdy with a 'LATE RETURN' button. To the right of the dashboard, a sidebar shows a user profile for 'Luisa' with 'Profile' and 'Log Out' options. The bottom right corner features a cartoon illustration of a woman reading a book.

#	BOOK	AUTHOR	CATEGORY	IS LATE	BORROWED AT	RETURNED AT	RETURN BY	RETURN?
199	Além do bem e do mal	Friedrich Nietzsche	Perspiciatis Recusandae	OK ●	2024-04-08 17:02:54		2024-04-23 17:02:54	<button>RETURN ↵</button>
198	Adipisci Possimus	Aldo Waters Sr.	Vel Soluta	OK ●	2024-04-08 17:00:37		2024-04-23 17:00:37	<button>RETURN ↵</button>
197	O estrangeiro	Albert Camus	Romance	OK ●	2024-04-08 17:00:28		2024-04-23 17:00:28	<button>RETURN ↵</button>
196	Corrupti Autem	Albert Camus	Filosofia	OK ●	2024-04-08 00:36:30		2024-04-23 00:36:30	<button>PENDING</button>
193	Eum Tempore	Alberto Purdy	Architecto Culpa	LATE ●	2024-04-08 14:02:03		2024-04-08 02:11:07	<button>LATE RETURN ↵</button>

TELA DE LIVROS EMPRESTADOS DO USUÁRIO

Tela mostrando todos os livros emprestados do usuário, incluindo os botões para return e os sinalizadores de status de atraso do livro, mostra também a data que o livro foi emprestado e quando o usuário deve devolver.

TELA DE LIVROS PARA O USUÁRIO



Tela mostrando todos os livros da aplicação. Com diferentes status e com Search*.

Borrow

*Livro disponível para empréstimo

Book requested.

*Livro já emprestado pela pessoa

Unavailable

*Livro sem quantidade disponível

Paginação de 20 itens, utilizando o Tailwind.

< 1 2 3 4 5 6 7 8 9 10 >

*Livro requisitado para empréstimo pelo usuário

DASHBOARD PARA O ADMIN (BIBLIOTECÁRIO)

A tela de navegação das ferramentas do Bibliotecário(Admin) na aplicação. As rotas são protegidas. Diferença de cor na nav-bar para diferenciação entre Users normais e Bibliotecários.



The screenshot shows the Admin Dashboard interface. At the top, there's a navigation bar with various links and a user profile for 'Luisa'. Below the navigation bar, the main content area is titled 'Dashboard'. It contains three main buttons: 'Borrow Requests' with a 'View' link, 'Return Requests' with a 'View' link, and 'Add Book'. Each button has a small icon next to it. To the right of the dashboard, there's a cartoon illustration of a woman with dark hair and a bow, wearing a white dress and holding a book. The book has the word 'infnet' written on it. The overall theme is professional and user-friendly.

TELA DE LIVROS PARA O ADMIN (BIBLIOTECÁRIO)

A tela de "controle" principal da aplicação para os Admins (Bibliotecários). Com search, create, edit e delete. Quantidade do livro e quantidade disponível do livro. O centro de controle dos livros.

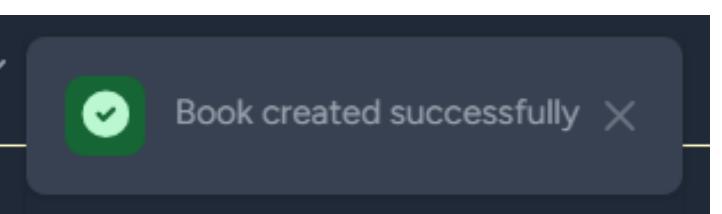
The screenshot shows a dark-themed web application interface for managing books. At the top, there's a navigation bar with tabs: Dashboard, Books (which is selected), Borrow Requests, and Return Requests. On the far right, it shows the user 'Lu Admin' and a dropdown menu. Below the navigation is a search bar with placeholder 'Search' and a green 'CREATE +' button. The main area is titled 'Books' and contains a table with the following data:

#	TITLE	QUANTITY	REFERENCE	AUTHOR	CATEGORY	AVAILABLE QUANTITY	AVAILABLE	ACTIONS?
1	Eum Tempore	2	3812140148	Alberto Purdy	Architecto Culpa	1	● Yes	EDIT DELETE SHOW
2	Corrupti Culpa	0	3366140148	Friedrich Nietzsche	Suspense	0	● No	EDIT DELETE SHOW
3	Molestiae Sed	4	3033140148	Dr. Gust Torphy	Autoajuda	4	● Yes	EDIT DELETE SHOW
4	Corrupti Autem	1	4501140148	Albert Camus	Filosofia	1	● Yes	EDIT DELETE SHOW
5	Dolor Non	1	1644140148	Warren Nicolas	Poesia	1	● Yes	EDIT DELETE SHOW
6	Esse Consectetur	0	2404140148	Prof. Green Emmerich	Perspiciatis Recusandae	0	● No	EDIT DELETE SHOW
7	Minus Molestias	0	6003140148	Mr. Ansley Lesch	Qui Et	0	● No	EDIT DELETE SHOW

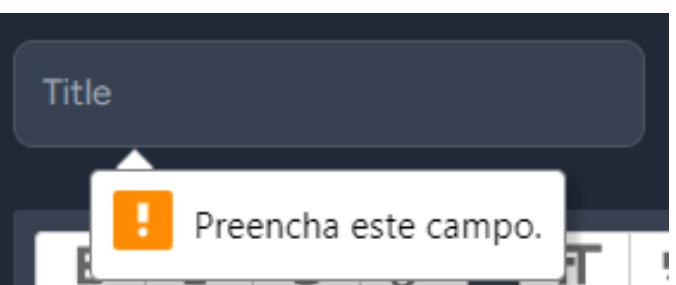
The screenshot shows a dark-themed web application interface for managing books. At the top, there's a navigation bar with links for Dashboard, Books, Borrow Requests, and Return Requests, along with a user profile for 'Lu Admin'. Below the navigation is a 'Create' section. On the left, there are three input fields: 'Title', 'Quantity', and 'Reference', each with its own input field and a rich text editor below it. On the right, there's a 'Relations' section with dropdown menus for 'Author' and 'Category'. A large, stylized illustration of a woman reading a book is centered at the bottom of the page. A blue 'Submit' button is located at the bottom left. A green toast message 'Book created successfully' with a checkmark icon appears in the bottom right corner.

TELA DE CRIAÇÃO DE LIVRO

Tela importante para o crud, que se comunica com o banco - para a inserção de um novo livro na database.



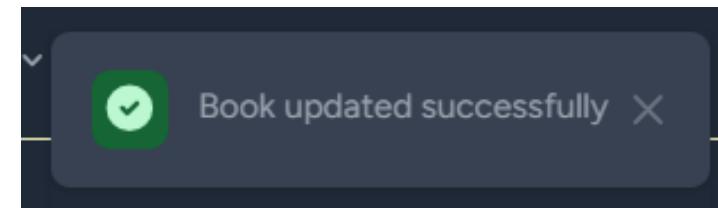
*Toast de sucesso com redirection



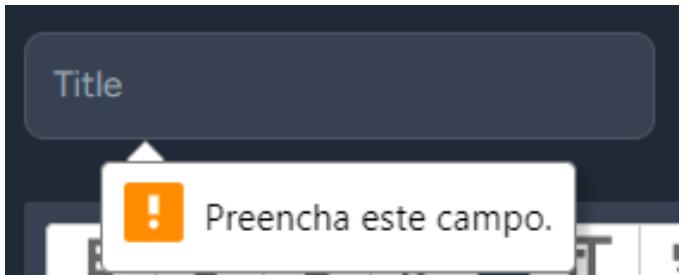
*Validações

TELA DE EDIÇÃO DE LIVRO

Tela importante para o crud, que se comunica com o banco - para a edição do livro.



*Toast de sucesso com redirection



*Validações

127.0.0.1:8000/admin/books/6/edit

Books Borrow Requests Return Requests

Edit

Title: Além do bem e do mal

Quantity: 2

Reference: 2404140148

A obra mais polêmica e emblemática de Friedrich Nietzsche, considerada por ele a mais importante, e um prelúdio a uma filosofia do futuro Nesta obra, o pensador adota um tom mais crítico do que em seus textos anteriores, iniciando a fase destrutiva de sua produção,

Submit

Relations

Author: Friedrich Nietzsche

Category: * Perspicacis Recusandae

http://127.0.0.1:8000/admin/books

Dashboard Books Borrow Requests Return Requests

← Book

Eum Tempore

EDIT **DELETE**

Quantity: 2
Reference: 3812140148
Author: Alberto Purdy
Category: Architecto Culpa
Sinopsis:
Explicabo est ad laudantium dicta aut. Ut repellendus dolorem facilis error. Est deserunt et est. Ab o...

Do you really want to delete this item?

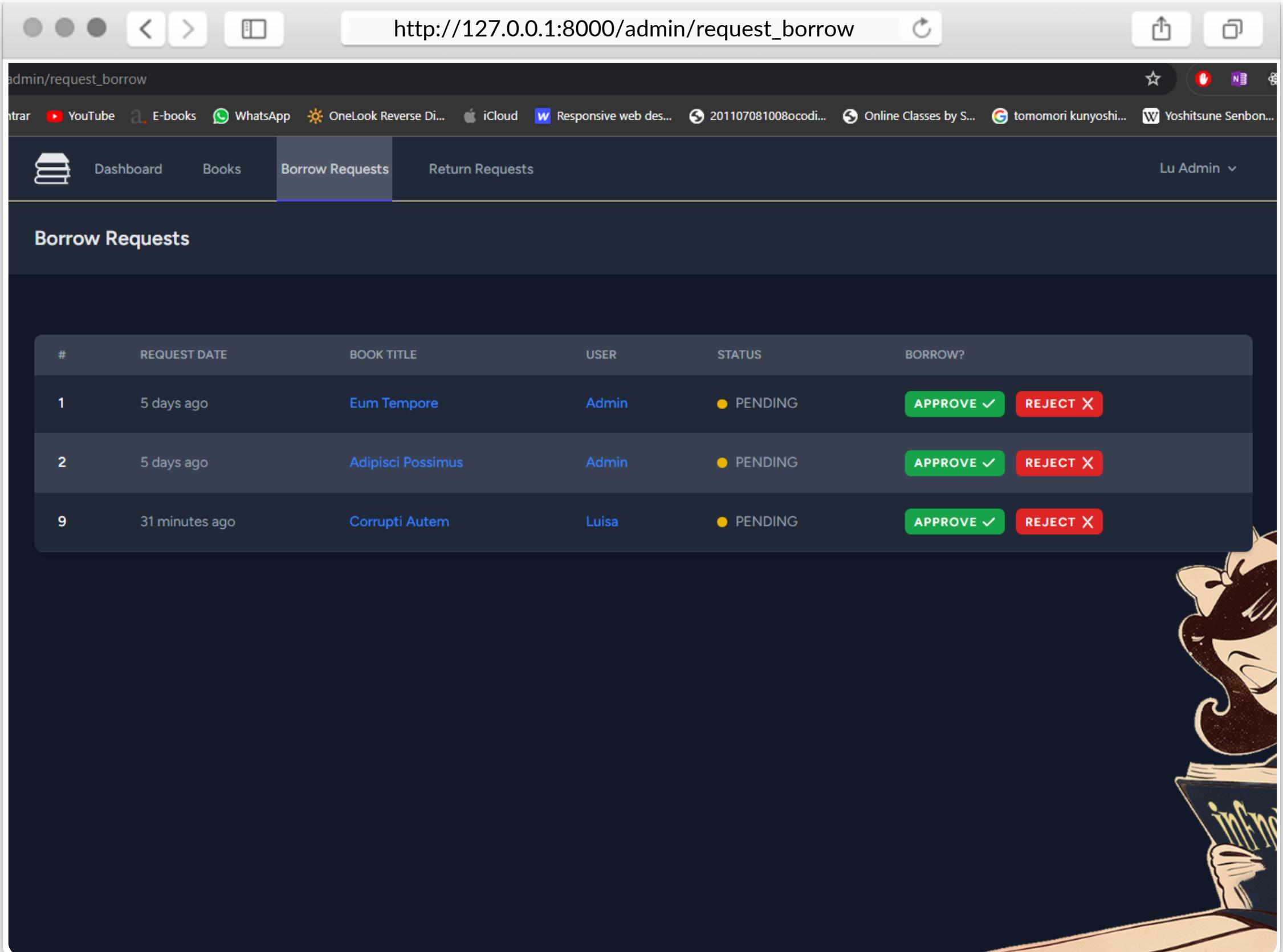
OK Cancelar

TELA DE VISUALIZAÇÃO DO LIVRO

Tela mostrando o livro selecionado, utilizando o ID do livro, incluindo Edit e Delete e um callback para a tela de livros.

*Double check com o usuário para o delete

TELA GESTÃO DE PEDIDOS DE EMPRÉSTIMO



The screenshot shows a web browser window with the URL http://127.0.0.1:8000/admin/request_borrow. The page title is "Borrow Requests". The navigation bar includes links for "Dashboard", "Books", "Borrow Requests" (which is active), and "Return Requests". A dropdown menu for "Lu Admin" is visible. The main content area displays a table of borrow requests:

#	REQUEST DATE	BOOK TITLE	USER	STATUS	BORROW?
1	5 days ago	Eum Tempore	Admin	PENDING	<button>APPROVE ✓</button> <button>REJECT X</button>
2	5 days ago	Adipisci Possimus	Admin	PENDING	<button>APPROVE ✓</button> <button>REJECT X</button>
9	31 minutes ago	Corrupti Autem	Luisa	PENDING	<button>APPROVE ✓</button> <button>REJECT X</button>

Tela mostrando todos os pedidos de empréstimos feitos pelos usuários do SBUN. Por trás de cada listagem:

Utilizamos o ORM Eloquent do Laravel para consultar e manipular dados de maneira segura e eficiente. A listagem de pedidos é realizada com uma filtragem pelo status, mostrando apenas os pedidos pendentes, garantindo foco nas ações imediatas.

O RequestBorrowController orquestra a lógica de negócios. Ao acessar o índice, um eloquente modelo RequestBorrow é consultado com pré-carregamento (eager loading) dos dados relacionados do usuário e livro, otimizando o desempenho e minimizando as consultas ao banco de dados.

As ações de aprovar ou rejeitar são tratadas com validações estritas, assegurando que apenas dados coerentes sejam processados. Se aprovado, um objeto borrow é criado para registrar o empréstimo efetivo, refletindo imediatamente no sistema.

TELA GESTÃO DE PEDIDOS DE RETORNO DE LIVRO

#	REQUEST DATE	BOOK TITLE	USER	STATUS	RETURN?
4	2 hours ago	Corrupti Autem	Luisa	PENDING	<button>CONFIRM ✓</button> <button>REJECT X</button>
5	11 seconds ago	Adipisci Possimus	Luisa	PENDING	<button>CONFIRM ✓</button> <button>REJECT X</button>

O RequestReturnController é responsável por gerenciar os pedidos de devolução, onde a primeira operação é a listagem de pedidos pendentes, realizada através de um modelo RequestReturn com pré-carregamento (eager loading) das relações necessárias. Esse método assegura uma busca rápida e otimizada dos dados, fundamental para uma gestão eficaz.

O processo de atualização de status de cada pedido é estritamente validado para assegurar que apenas transições de estado válidas sejam permitidas. As transições de estados são gerenciadas de maneira dinâmica, permitindo que o sistema reflita em tempo real as aprovações ou rejeições dos pedidos de devolução.

Tela mostrando todos os pedidos de retorno de empréstimo feitos pelos usuários do SBUN. Por trás de cada listagem:

Quando um pedido é aprovado, o sistema executa uma ação para confirmar a devolução do livro, atualizando o status de disponibilidade do livro e fechando o ciclo do empréstimo.

Busca

A busca é uma funcionalidade importante da aplicação, contemplando tanto o usuário quanto o bibliotecário - ambas acontecem em suas respectivas telas de Book e com suas próprias lógicas que contemplam o pipeline de comportamento de cada modalidade (user vs. bibliotecário).

SEARCH DO BIBLIOTECÁRIO (ADMIN)

The screenshot shows a dark-themed web application interface for library management. At the top, there's a navigation bar with links for 'Dashboard', 'Books' (which is currently selected), 'Borrow Requests', and 'Return Requests'. A user profile 'Lu Admin' is visible on the right. Below the navigation is a search bar with the placeholder 'Search' and the term 'além'. A green 'CREATE +' button is located to the right of the search bar. The main content area displays a table of books. The columns are labeled '#', 'TITLE', 'QUANTITY', 'REFERENCE', 'AUTHOR', 'CATEGORY', 'AVAILABLE QUANTITY', 'AVAILABLE', and 'ACTIONS?'. One row is highlighted, showing the book 'Além do bem e do mal' by Friedrich Nietzsche, categorized under 'Persiciatis Recusandae'. The 'AVAILABLE QUANTITY' is 1, and the status is 'Yes'. Action buttons for 'EDIT' and 'DELETE' are shown next to the row.

SEARCH DO USUÁRIO

The screenshot shows a dark-themed web application interface for library management. At the top, there's a navigation bar with links for 'Dashboard' and 'Books' (which is currently selected). A user profile 'Luisa' is visible on the right. Below the navigation is a search bar with the placeholder 'Search' and the term 'fried'. The main content area displays a detailed view of a book. On the left, there's a thumbnail image of the book cover, which features the letters 'SbuN'. To the right of the image, the book title is 'Corrupti Culpa | #2', the author is 'Friedrich Nietzsche', the category is 'Suspense', and the reference number is '3366140148'. Below this information is a section titled 'Sinopsis:' containing the text: 'Vitae quia vel dolor repudianda. Non deleniti molestiae distinctio explicabo quaerat. Inventore dolorem tempore molestias quia earum dicta.'

BUSCA ABRANGENTE E EFICIENTE

Visão Completa do Acervo: Administradores têm uma busca avançada que permite uma gestão detalhada do inventário, incluindo campos como ID, título e referência.

```
$records = Book::orderBy($orderBy, $sortDir)
->when($search, function ($query, $s) {
    return $query
        ->orWhere(DB::raw('LOWER(id)'), 'like', "%{$s}%")
        ->orWhere(DB::raw('LOWER(title)'), 'like', "%{$s}%")
        ->orWhere(DB::raw('LOWER(reference)'), 'like', "%{$s}%");
})
->paginate(20);
```

*A consulta de busca para administradores ordena os registros da tabela books pelo campo especificado (id, title, reference, etc.) e direção (asc ou desc) fornecidos nas entradas do usuário. Quando há um termo de busca, a consulta utiliza LIKE para filtrar os resultados com base nesse termo nos campos id, title, e reference.

BUSCA FOCADA E CONVENIENTE

Busca Simplificada: Usuários podem pesquisar livros utilizando palavras-chave relevantes, como título, referência ou autor.

```
$booksQuery = Book::with('author', 'category')
->when($search, function ($query, $search) {
    return $query->join('authors', 'books.author_id', '=', 'authors.id')
        ->where(function ($query) use ($search) {
            $query->whereRaw('LOWER(books.title) LIKE ?', ["%" . $search . "%"])
            ->orWhereRaw('LOWER(books.reference) LIKE ?', ["%" . $search . "%"])
            ->orWhereRaw('LOWER(authors.name) LIKE ?', ["%" . $search . "%"]);
        })
        ->select('books.*');
});
```

*Nesta consulta, o sistema realiza um join com a tabela authors para incluir nomes de autores na busca e usa LIKE para comparar o termo de busca em minúsculas com os campos title, reference da tabela books e name da tabela authors.

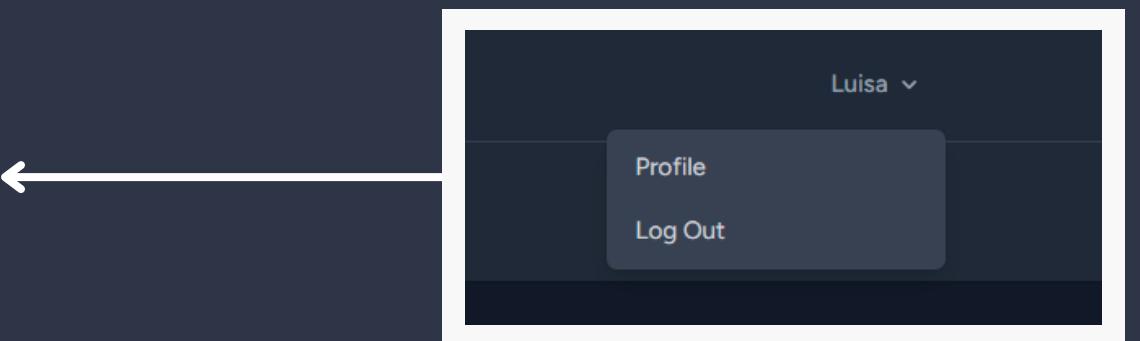
PROFILE

The screenshot shows a web application interface for managing a user profile. At the top, the URL bar displays `http://127.0.0.1:8000/profile`. The page has a dark background with light-colored text and input fields.

Email: `luisa@infnet.com`

Update Password: A section with instructions: "Ensure your account is using a long, random password to stay secure." It contains fields for **Current Password**, **New Password**, and **Confirm Password**, each with a placeholder "Enter password". Below these fields is a **SAVE** button.

Delete Account: A section with a warning: "Once your account is deleted, all of its resources and data will be permanently deleted. Before deleting your account, please download any data or information that you wish to retain." It contains a prominent **DELETE ACCOUNT** button.



Tela de visualização das informações do perfil, com a possibilidade de alteração. Além também de contar com a opção de Deletar a conta.

TESTES PHPUNIT

A aplicação conta com 26 testes, entre feature tests e testes unitários.

```
PASS Tests\Feature\AdminBooksRouteTest
✓ no admin cant access route
✓ route redirects

PASS Tests\Feature\Auth\AuthenticationTest
✓ login screen can be rendered
✓ users can authenticate using the login screen
✓ users can not authenticate with invalid password
✓ users can logout

PASS Tests\Feature\Auth\EmailVerificationTest
✓ email verification screen can be rendered
✓ email can be verified
✓ email is not verified with invalid hash

PASS Tests\Feature\Auth>PasswordConfirmationTest
✓ confirm password screen can be rendered
✓ password can be confirmed
✓ password is not confirmed with invalid password

PASS Tests\Feature\Auth>PasswordResetTest
✓ reset password link screen can be rendered
✓ reset password link can be requested
✓ reset password screen can be rendered
✓ password can be reset with valid token

PASS Tests\Feature\Auth>PasswordUpdateTest
✓ password can be updated
✓ correct password must be provided to update password

PASS Tests\Feature\Auth\RegistrationTest
✓ registration screen can be rendered
✓ new users can register

PASS Tests\Feature\ProfileTest
✓ profile page is displayed
✓ profile information can be updated
✓ email verification status is unchanged when the email address is unchanged
✓ user can delete their account
✓ correct password must be provided to delete account

Tests: 26 passed (79 assertions)
Duration: 147.81s
```

EXEMPLO DE TESTE UNITÁRIO - TDD

```
<?php

namespace Tests\Unit;

use Tests\TestCase;
use App\Models\User;
use Illuminate\Foundation\Testing\RefreshDatabase;

// use PHPUnit\Framework\TestCase;

0 references | 0 implementations
class AdminUserTest extends TestCase
{
    use RefreshDatabase;

    // Testa se a lógica de identificação de usuários administradores está funcionando corretamente.
    0 references | 0 overrides
    public function testIfUserIsAdmin(): void
    {
        // Cria um usuário de teste com o status de administrador definido como falso.
        $user = User::factory()->createOne([
            'is_admin' => false,
        ]);

        // Verifica se o método isAdmin() retorna falso para um usuário não-administrador.
        $this->assertFalse($user->isAdmin());

        // Altera o status do usuário para administrador.
        $user->is_admin = true;
        $user->save(); // Salva a alteração no banco de dados.

        // Verifica se o método isAdmin() agora retorna verdadeiro, indicando que o usuário é um administrador.
        $this->assertTrue($user->isAdmin());
    }
}
```

```
Luisa@PCLuisa MINGW64 /c/xampp/htdocs/sbun-laravel (master)
$ php artisan test --filter=Admin

PASS Tests\Unit\AdminUserTest
✓ if user is admin

PASS Tests\Feature\AdminBooksRouteTest
✓ no admin cant access route
✓ route redirects

Tests: 3 passed (20 assertions)
Duration: 13.49s
```

OBJETIVO DO TESTE: Confirmar se a lógica para verificar o status de administrador em um usuário funciona como esperado.

GARANTIA DE FUNCIONALIDADE: Assegura que o sistema corretamente identifica e altera o status de administração dos usuários.

SEGURANÇA E INTEGRIDADE

LARAVEL SANCTUM

Gerenciamento de autenticação stateful e APIs tokenizadas, oferecemos um mecanismo de login seguro e eficiente, com a capacidade de integrar facilmente a autenticação em SPAs, aplicativos móveis e serviços simples baseados em token.

AUTORIZAÇÃO E ACESSO

Rotas e ações administrativas são protegidas pelo **middleware** do Laravel, assegurando que somente usuários com as devidas permissões possam acessar informações sensíveis ou realizar operações críticas.

```
Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
    return $request->user();
});
```

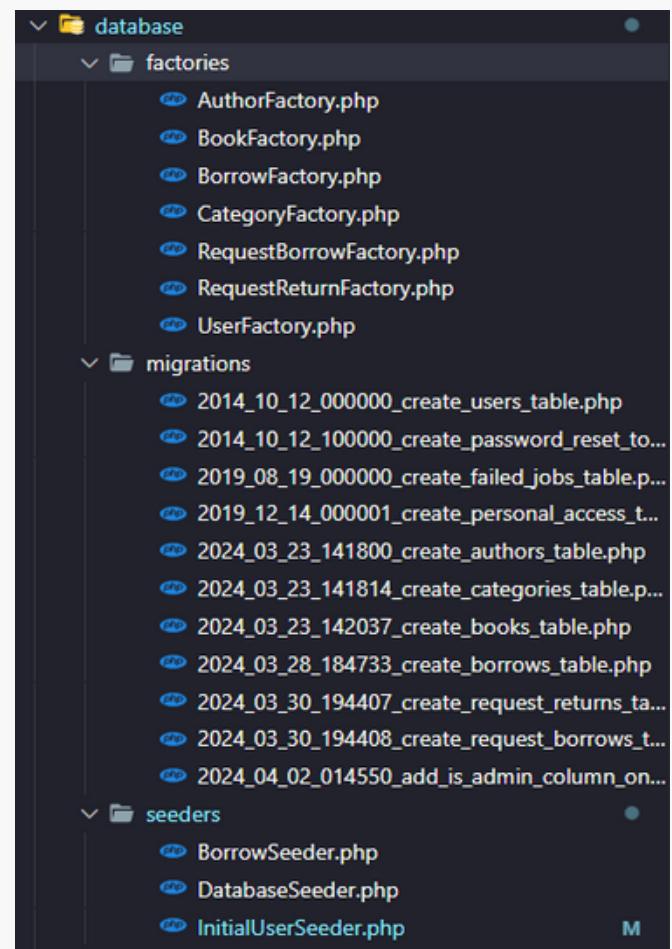
```
use Laravel\Sanctum\HasApiTokens;
use Illuminate\Notifications\Notifiable;
use Illuminate\Database\Eloquent\Relations\HasMany;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;

class User extends Authenticatable
{
    use HasApiTokens;
    use HasFactory;
    use Notifiable;
```

```
Middleware
AdminUserOnly.php
Authenticate.php
EncryptCookies.php
PreventRequestsDuringMaintenance.php
RedirectIfAuthenticated.php
TokenVerifier.php

Route::prefix('admin')
    ->name('admin.')
    ->middleware('admin')
    ->group(function () {
        require __DIR__ . '/admin.php';
    });

require __DIR__ . '/auth.php';
```



VALIDAÇÃO E SANITIZAÇÃO DE DADOS

Garantimos a integridade dos dados com as funcionalidades de validação e sanitização embutidas no Laravel, evitando inserções maliciosas.

```
<div class="mt-4">
    <x-input-label for="password" :value="__('Password')"/>
    <x-text-input id="password" class="block mt-1 w-full" type="password" name="password" required autocomplete="new-password"/>
    <x-input-error :messages="$errors->get('password')" class="mt-2"/>
</div>
```

MIGRATIONS E SEEDERS

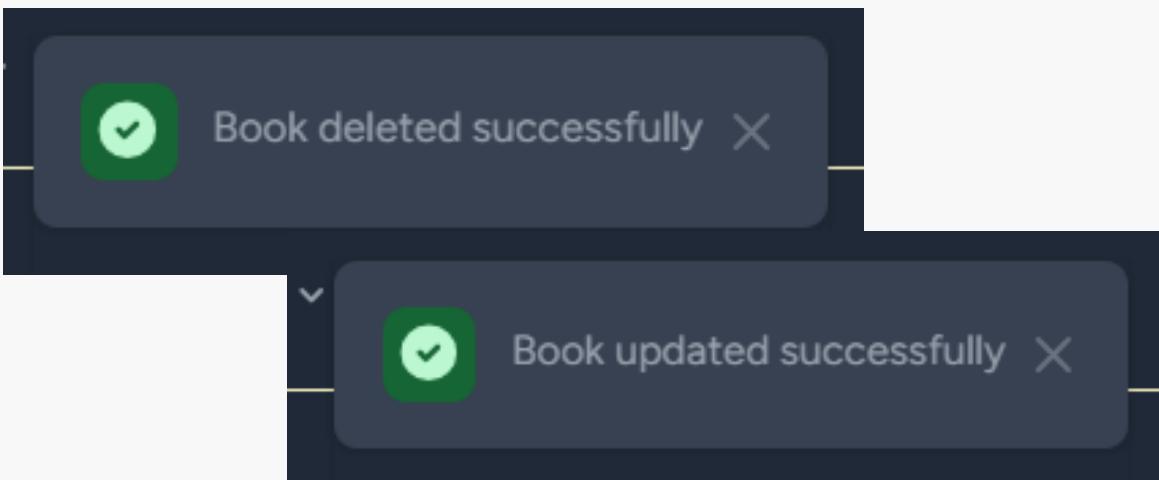
Estruturamos o banco de dados de forma segura e confiável com as migrations do Laravel, enquanto os seeders populam inicialmente o sistema com dados seguros.

TOAST (FEEDBACK DINAMICO) E VALIDAÇÕES

Optamos por criar nosso próprio toast, em detrimento de utilizar algum pronto como o Toastify e elevamos a interação do usuário através de notificações elegantes e informativas para ações, erros e alertas.

ESCUTA ATIVA DE EVENTOS

Captura eventos e reage instantaneamente com mensagens relevantes.

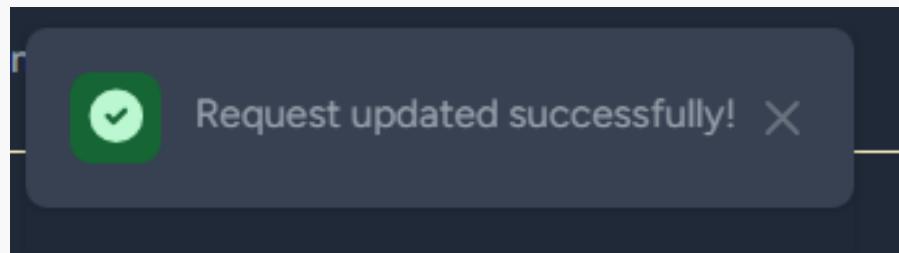


INTEGRAÇÃO COM O DOM

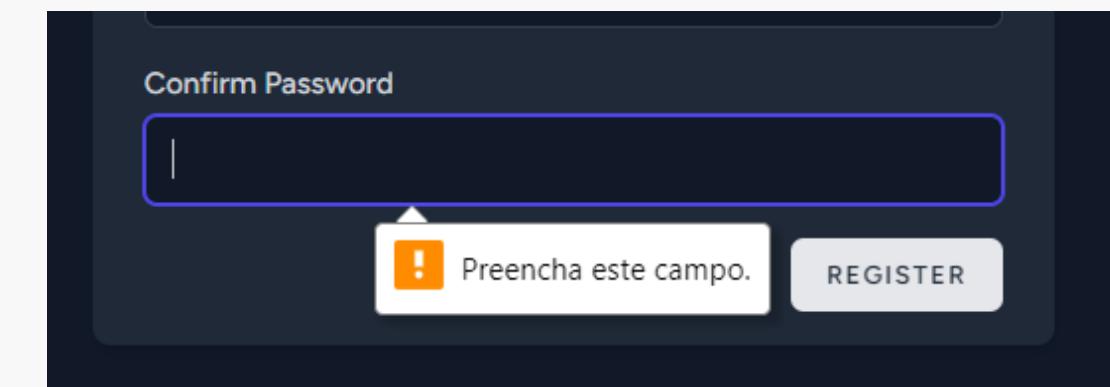
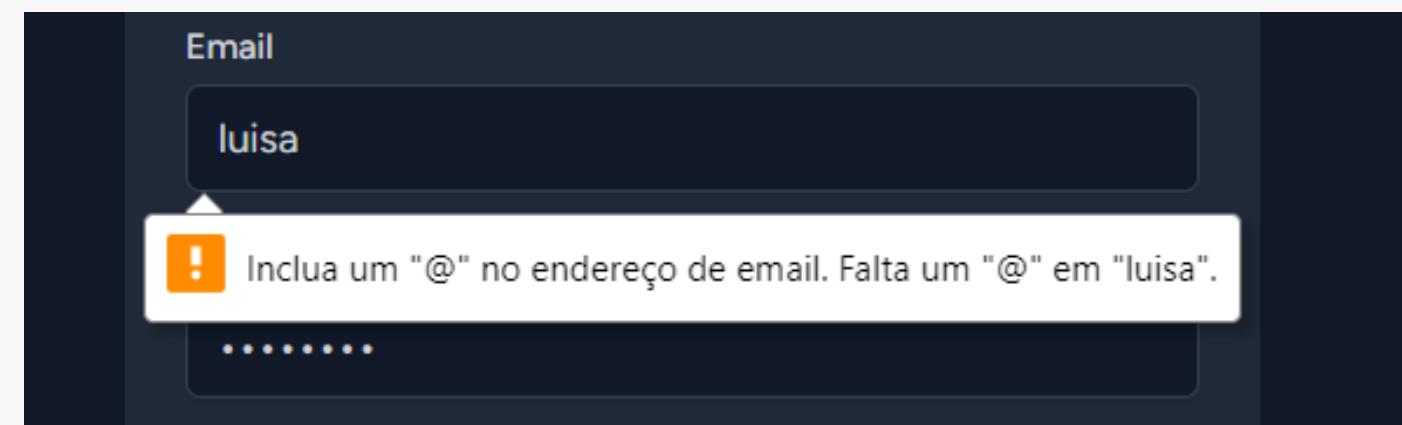
Os toasts são habilmente injetados no DOM, assegurando que a interface do usuário permaneça fluída e interativa, mesmo durante notificações.

TEMPLATES ADAPTÁVEIS

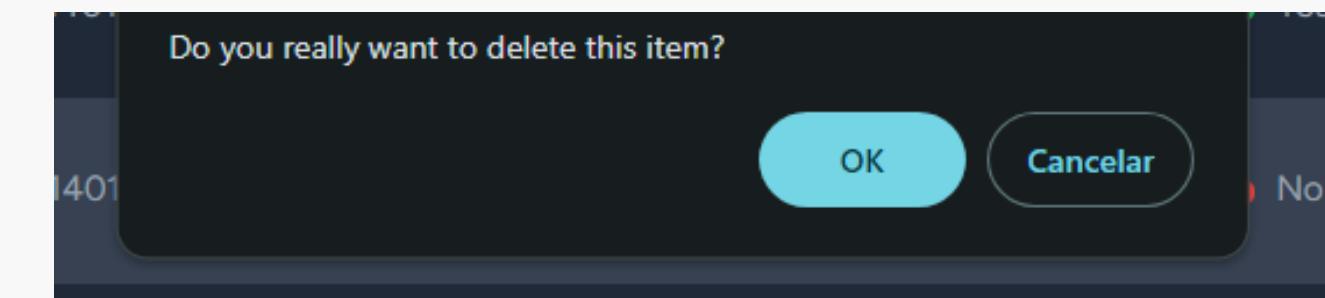
Exibe o toast correto, alinhado ao contexto do evento.



Todos os dados são validados com mensagens para garantir que o usuário saiba o que é esperado dele.



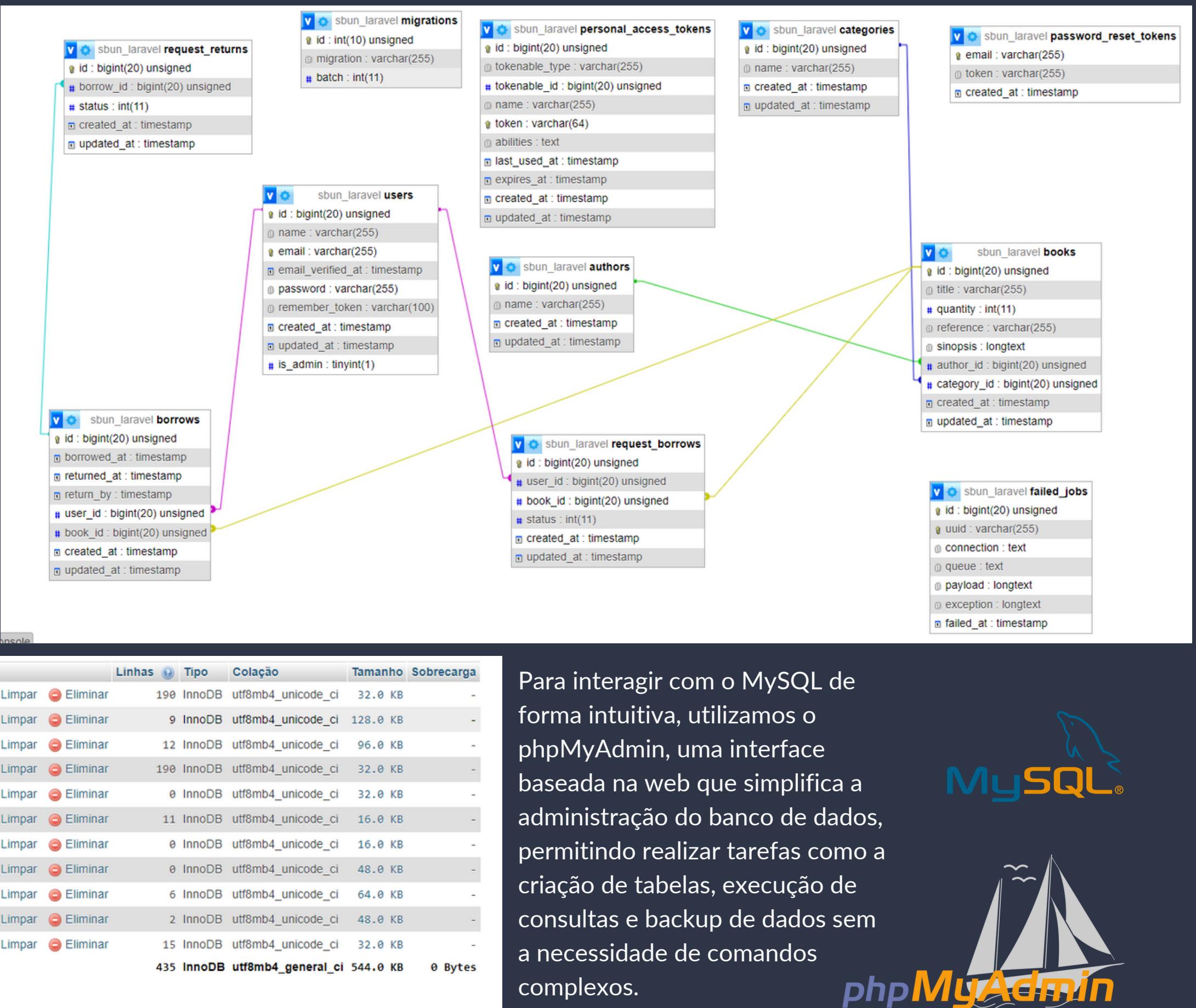
Ações "graves" como o delete possuem um *double check* com o usuário.



BANCO DE DADOS

Minha escolha pelo MySQL para o SBUN não é por acaso. Ele oferece desempenho excepcional, confiabilidade e facilidade de uso, seja administrando grandes volumes de dados de livros, usuários, empréstimos ou solicitações de devolução. Sua compatibilidade com PHP, a linguagem principal da nossa aplicação, garante uma integração suave e uma operação eficiente.

The screenshot shows the phpMyAdmin interface for the 'sbun_laravel' database. The left sidebar lists databases like 'information_schema', 'mysql', 'performance_schema', 'phpmyadmin', 'sbun', and 'sbun_laravel'. The 'sbun_laravel' database is selected, showing its tables: 'authors', 'books', 'borrows', 'categories', 'failed_jobs', 'migrations', 'password_reset_tokens', 'personal_access_tokens', 'request_borrows', 'request_returns', and 'users'. The main area displays the structure of the 'users' table, which includes columns for id, name, email, email_verified_at, password, remember_token, created_at, updated_at, and is_admin. Below this, a list of all tables in the database is shown with their respective details like type (InnoDB), character set (utf8mb4_unicode_ci), and sizes.



Para interagir com o MySQL de forma intuitiva, utilizamos o phpMyAdmin, uma interface baseada na web que simplifica a administração do banco de dados, permitindo realizar tarefas como a criação de tabelas, execução de consultas e backup de dados sem a necessidade de comandos complexos.



phpMyAdmin

FUTURO DO SBUN

O SBUN encontra-se ainda em estado embrionário (MVP), entretanto, várias funcionalidades foram idealizadas para seu futuro, como:

SbuN⁺

Módulo de Multas Automatizadas: Um futuro módulo permitirá o cálculo e a gestão de multas diretamente na dashboard administrativa, facilitando a administração de empréstimos.

Sistema de Doação de Livros: Usuários poderão doar e adquirir livros com créditos, fomentando o intercâmbio cultural e ajudando as bibliotecas a gerenciar acervos excedentes.

SbuN

Notificações de Devolução por E-mail: Planejamos notificar usuários sobre prazos de devolução, incentivando a responsabilidade e a pontualidade.

SbuN
Obrigada!

SbuN