

DOCUMENTATION

I. Blade files

Purpose:

- **Authenticated User:** If the user is authenticated (auth()->check()), a link to the **User Dashboard** is displayed. The URL is dynamically generated using the route helper, passing the authenticated user's ID as a parameter (auth()->user()->id).
- **Non-authenticated User:** If the user is not authenticated, a message prompts them to log in in order to access the dashboard.

resources > views > dashboard.blade.php

Dashboard.blade.php

```
1 <x-app-layout>
2 <x-slot name="header">
3 <h2 class="font-semibold text-xl text-gray-800 leading-tight">
4     {{ __('Dashboard') }}
5 </h2>
6 </x-slot>
7
8 <div class="py-12">
9     <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
10        <div class="bg-white overflow-hidden shadow-sm sm:rounded-lg">
11            <div class="p-6 text-gray-900">
12                {{ __('You're logged in!') }}
13
14                <!-- Check if user is authenticated -->
15                @if (auth()->check())
16                    <a href="{{ route('user-dashboard', ['userId' => auth()->user()->id]) }}"
17                        class="text-blue-500 hover:underline mt-4 inline-block">
18                        Go to User Dashboard
19                    </a>
20                @else
21                    <p>Please log in to access your dashboard.</p>
22                @endif
23            </div>
24        </div>
25    </div>
26 </div>
27 </x-app-layout>
```

resources > views > user-dashboard.blade.php

user-dashboard.blade.php

```
1 <x-app-layout>
2 <x-slot name="header">
3 <h2 class="font-semibold text-xl text-gray-800 leading-tight">
4     {{ __('User Dashboard') }}
5 </h2>
6 </x-slot>
7
8 <div class="py-12">
9     <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
10        <div class="bg-white overflow-hidden shadow-sm sm:rounded-lg">
11            <div class="p-6 text-gray-900">
12                <h1>Welcome to the user-dashboard, {{ $userId }}!</h1>
13
14                <p>This page is under development...</p>
15
16                <a href="{{ route('dashboard') }}" class="inline-block border border-blue-500 text-b
17                    Back to Dashboard
18                </a>
19
20                <form action="{{ route('logout') }}" method="POST" style="margin-top: 10px;">
21                    @csrf
22                    <button type="submit" class="bg-red-500 text-white px-4 py-2 rounded hover:
23                        Logout
24                    </button>
25                </form>
26            </div>
27        </div>
28    </div>
```

The **welcome message** dynamically greets the user using the **{{ \$userId }}** placeholder, which would be replaced by the actual user ID passed from the controller.

Session Status Component: Displays any session status messages like login success or errors.

Action Links and Button:

- **Forgot Password:** A link to reset the password.
- **Login Button:** Submits the login form.

resources > views > auth > login.blade.php

login.blade.php

```
1 <x-guest-layout>
2 <div class="flex flex-col justify-center min-h-screen">
3     <div class="text-center">
4         <h1>Login</h1>
5     </div>
6     <div class="mt-10">
7         <div class="text-left">
8             <div class="text-gray-600">Email Address</div>
9             <input type="text" value="" class="form-control w-100">
10        </div>
11        <div class="text-left">
12            <div class="text-gray-600">Password</div>
13            <input type="password" value="" class="form-control w-100">
14        </div>
15        <div class="text-left">
16            <div class="text-gray-600">Remember Me</div>
17            <input type="checkbox" class="form-check-input" />
18        </div>
19        <div class="text-center">
20            <button type="submit" class="btn btn-primary">Login</button>
21        </div>
22        <div class="text-center">
23            <a href="#" class="text-decoration: none; color: #007bff; font-weight: normal;">
24                <small>Forgot your password?</small>
25            </a>
26        </div>
27    </div>
28 </div>
```

```
resources > views > auth > register.blade.php
1 <x-guest-layout>
2 <form method="POST" action="{{ route('register') }}">
3 @csrf
4
5 <!-- Name -->
6 <div>
7 <x-input-label for="name" :value="__('Name')"/>
8 <x-text-input id="name" class="block mt-1 w-full" type="text" name="name" :value="old('name')" required autofocus autocomplete="name" />
9 <x-input-error :messages="$errors->get('name')" class="mt-2" />
10 </div>
11
12 <!-- Email Address -->
13 <div class="mt-4">
14 <x-input-label for="email" :value="__('Email')"/>
15 <x-text-input id="email" class="block mt-1 w-full" type="email" name="email" :value="old('email')" required autocomplete="username" />
16 <x-input-error :messages="$errors->get('email')" class="mt-2" />
17 </div>
18
19 <!-- Password -->
20 <div class="mt-4">
21 <x-input-label for="password" :value="__('Password')"/>
22 <x-text-input id="password" class="block mt-1 w-full" type="password" name="password" required autocomplete="new-password" />
23 <x-input-error :messages="$errors->get('password')" class="mt-2" />
24 </div>
25
26 <!-- Confirm Password -->
27 <div class="mt-4">
28 <x-input-label for="password_confirmation" :value="__('Confirm Password')"/>
29 <x-text-input id="password_confirmation" class="block mt-1 w-full" type="password" name="password_confirmation" required autocomplete="new-password" />
30 <x-input-error :messages="$errors->get('password_confirmation')" class="mt-2" />
31 </div>
32
33 <div class="flex items-center justify-end mt-4">
34 <a class="underline text-sm text-gray-600 hover:text-gray-900 rounded-md focus:outline-none focus:ring-2 focus:ring-indigo-500 focus:ring-offset-2" href="{{ route('login') }}">
35 {{ __('Already registered?') }}
36 </a>
37
38 <x-primary-button class="ms-4">
39 {{ __('Register') }}
40 </x-primary-button>
41 </div>
42 </form>
43 </x-guest-layout>
```

register.blade.php

Form: The registration form (<form>) uses the POST method to submit the data to the register route.

Action Links and Button:

- **Login Link:** Directs users to the login page if they already have an account.
- **Register Button:** Submits the form to create a new user.

```
resources > views > homepage.blade.php
1 <x-app-layout>
2 <x-slot name="header">
3 <h2 class="font-semibold text-xl text-gray-800 leading-tight">
4 {{ __('Home') }}
5 </h2>
6 </x-slot>
7
8 <div class="py-12">
9 <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
10 <div class="bg-white overflow-hidden shadow-sm sm:rounded-lg">
11 <div class="p-6 text-gray-900">
12 <h1>WELCOME</h1>
13 <p>Enjoy your visit here.</p>
14
15 @auth
16 <p>You are logged in! Go to your <a href="{{ route('dashboard') }}" class="inline-
17 @else
18 <p>Please <a href="{{ route('login') }}" class="inline-block border border-blue-50
19 @endauth
20 </div>
21 </div>
22 </div>
23 </div>
24 </x-app-layout>
25
```

homepage.blade.php

Purpose: This conditional section checks if the user is authenticated using the @auth directive.

- If **authenticated**, it displays a message confirming login and a link to the dashboard.
- If **not authenticated**, it prompts the user to log in or register, with links that lead to the login and registration routes.

Delete Account Blade:

- **Purpose:** This section allows the user to delete their account from the platform.

```
resources > views > profile > partials > delete-user-form.blade.php
1 <section class="space-y-6">
2 <header>
3 <h2 class="text-lg font-medium text-gray-900">
4 {{ __('DELETE ACCOUNT') }}
5 </h2>
6
7 <p class="mt-1 text-sm text-gray-600">
8 {{ __('Once your account is deleted, all of its resources and data will be permanently deleted. Before deleting your account, please download any data or info
9 </p>
10 </header>
11
12 <x-danger-button
13 x-data=""
14 x-on:click.prevent="$dispatch('open-modal', 'confirm-user-deletion')">
15 >{{ __('Delete Account') }}</x-danger-button>
16
17 <x-modal name="confirm-user-deletion" :show="$errors->userDeletion->isNotEmpty()" focusable>
18 <form method="post" action="{{ route('profile.destroy') }}" class="p-6">
19 @csrf
20 @method('delete')
21
22 <h2 class="text-lg font-medium text-gray-900">
23 {{ __('Are you sure you want to delete your account?') }}
24 </h2>
25
26 <p class="mt-1 text-sm text-gray-600">
27 {{ __('Once your account is deleted, all of its resources and data will be permanently deleted. Please enter your password to confirm you would like to pe
28 </p>
29
30 <div class="mt-6">
31 <x-input-label for="password" :value="__('Password')"/>
32 <x-text-input
33 id="password"
34 name="password"
35 type="password"
36 </x-text-input>
37 </div>
```

delete-user-form.blade.php

Update Password Blade:

- **Purpose:** This section allows the user to update their account password.

```
resources > views > profile > partials > update-password-form.blade.php
1 <section>
2 <header>
3 <h2 class="text-lg font-medium text-gray-900">
4     {{ __('Update Password') }}
5 </h2>
6
7 <p class="mt-1 text-sm text-gray-600">
8     {{ __('Ensure your account is using a long, random password to stay secure.') }}
9 </p>
10 </header>
11
12 <form method="post" action="{{ route('password.update') }}" class="mt-6 space-y-6">
13     @csrf
14     @method('put')
15
16     <div>
17         <x-input-label for="update_password_current_password" :value="__('Current Password')"/>
18         <x-text-input id="update_password_current_password" name="current_password" type="password" class="mt-1 block w-full" aut
19         <x-input-error :messages="$errors->updatePassword->get('current_password')"/>
20     </div>
21
22     <div>
23         <x-input-label for="update_password_password" :value="__('New Password')"/>
24         <x-text-input id="update_password_password" name="password" type="password" class="mt-1 block w-full" autocomplete="new-p
25         <x-input-error :messages="$errors->updatePassword->get('password')"/>
26     </div>
27
28     <div>
29         <x-input-label for="update_password_password_confirmation" :value="__('Confirm Password')"/>
30         <x-text-input id="update_password_password_confirmation" name="password_confirmation" type="password" class="mt-1 block w
31         <x-input-error :messages="$errors->updatePassword->get('password_confirmation')"/>
32     </div>
33
34     <div class="flex items-center gap-4">
35         <x-primary-button>{{ __('Save') }}</x-primary-button>
```

update-password-form.blade.php

```
resources > views > profile > partials > update-profile-information-form.blade.php
1 <section>
2 <header>
3 <h2 class="text-lg font-medium text-gray-900">
4     {{ __('Profile Information') }}
5 </h2>
6
7 <p class="mt-1 text-sm text-gray-600">
8     {{ __('Update your account's profile information and email address.') }}
9 </p>
10 </header>
11
12 <form id="send-verification" method="post" action="{{ route('verification.send') }}">
13     @csrf
14 </form>
15
16 <form method="post" action="{{ route('profile.update') }}" class="mt-6 space-y-6">
17     @csrf
18     @method('patch')
19
20     <div>
21         <x-input-label for="name" :value="__('Name')"/>
22         <x-text-input id="name" name="name" type="text" class="mt-1 block w-full" :value="old('name', $user->name)" required auto
23         <x-input-error class="mt-2" :messages="$errors->get('name')"/>
24     </div>
25
26     <div>
27         <x-input-label for="email" :value="__('Email')"/>
28         <x-text-input id="email" name="email" type="email" class="mt-1 block w-full" :value="old('email', $user->email)" required
29         <x-input-error class="mt-2" :messages="$errors->get('email')"/>
30
31         @if ($user instanceof \Illuminate\Contracts\Auth\MustVerifyEmail && ! $user->hasVerifiedEmail())
32             <div>
33                 <p class="text-sm mt-2 text-gray-800">
34                     {{ __('Your email address is unverified.') }}
35
36                 <button form="send-verification" class="underline text-sm text-gray-600 hover:text-gray-900 rounded-md focus:outline-none">
```

update-profile-information-form.blade.php

Profile Information Blade:

- **Purpose:** This section allows the user to update their profile information, including their name and email address.

II. Routes

web.php

```
routes > web.php
6
7 // Homepage Route
8 Route::get('/', [HomeController::class, 'index']);
9
10 // Dashboard Route
11 Route::get('/dashboard', function () {
12     return view('dashboard');
13 })->middleware(['auth', 'verified'])->name('dashboard');
14
15 // Profile Routes
16 Route::middleware('auth')->group(function () {
17     Route::get('/profile', [ProfileController::class, 'edit'])->name('profile.edit');
18     Route::patch('/profile', [ProfileController::class, 'update'])->name('profile.update');
19     Route::delete('/profile', [ProfileController::class, 'destroy'])->name('profile.destroy');
20 });
21
22 // User Dashboard Route
23 Route::get('/user-dashboard/{userId}', [UserDashboardController::class, 'show'])
24     ->middleware(['auth'])
25     ->name('user-dashboard');
26
27 require __DIR__.'/auth.php';
28
29
```

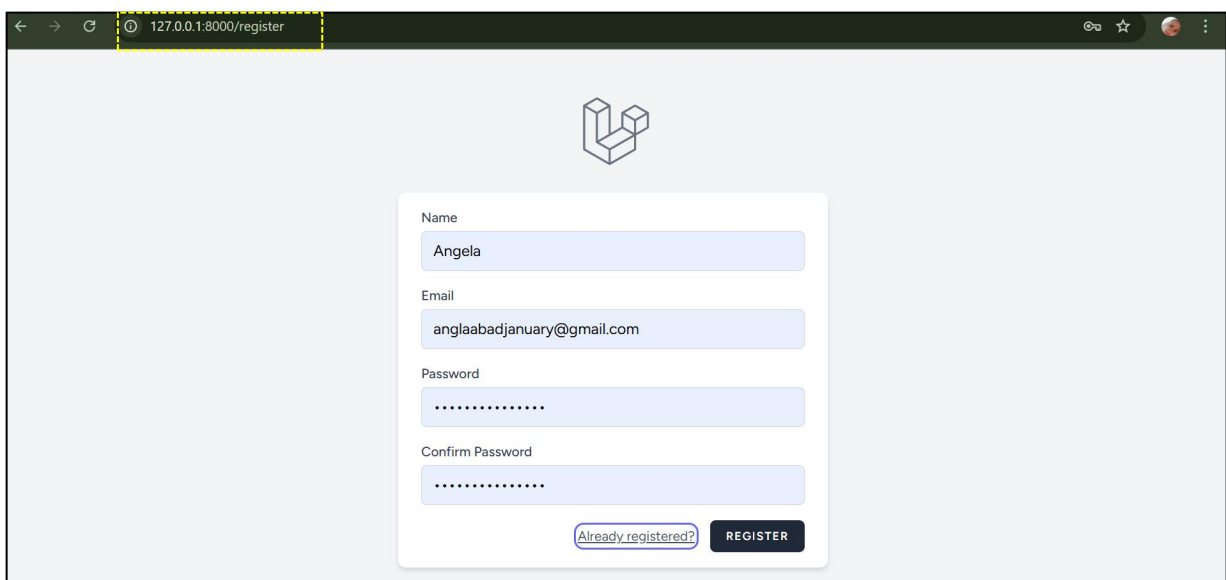
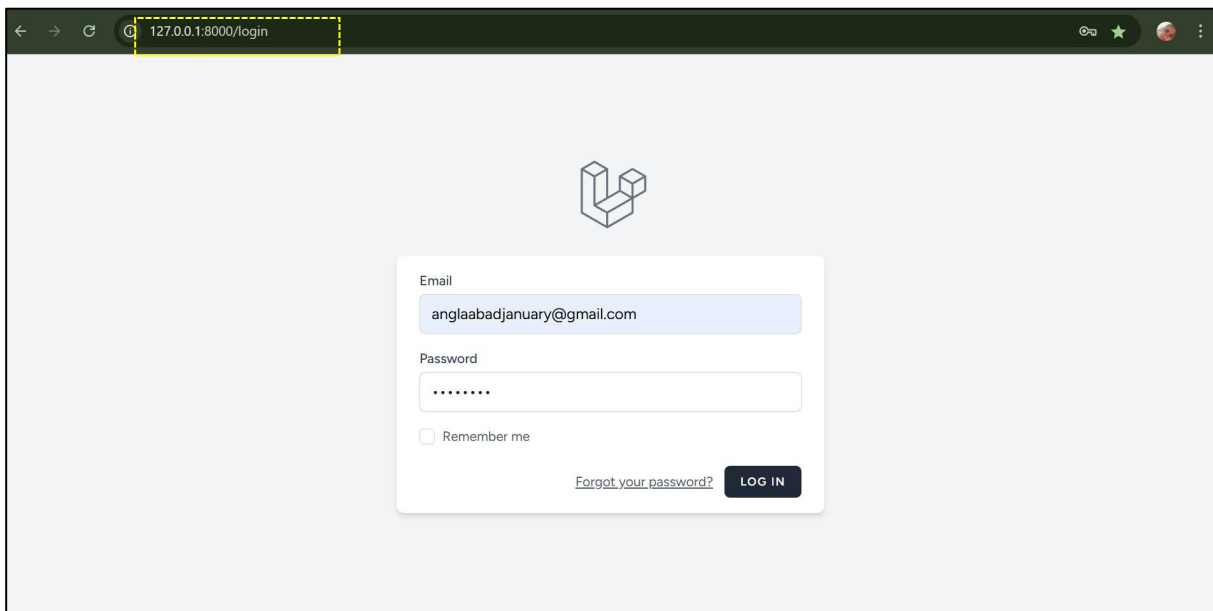
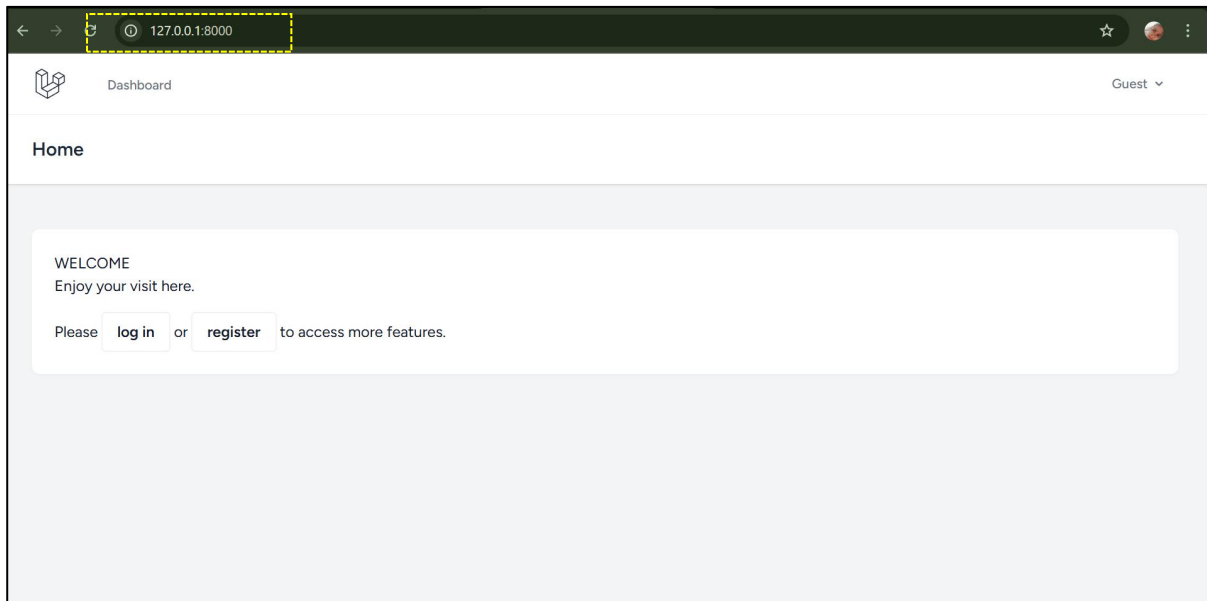
/: Handled by the *HomeController@index* method

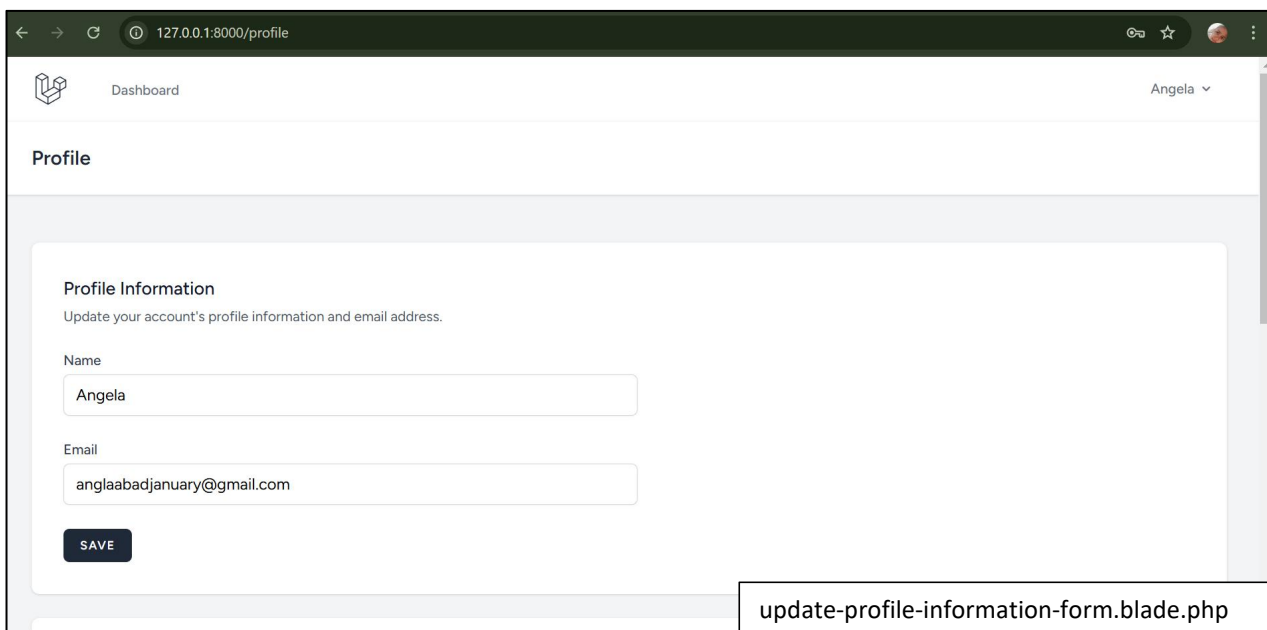
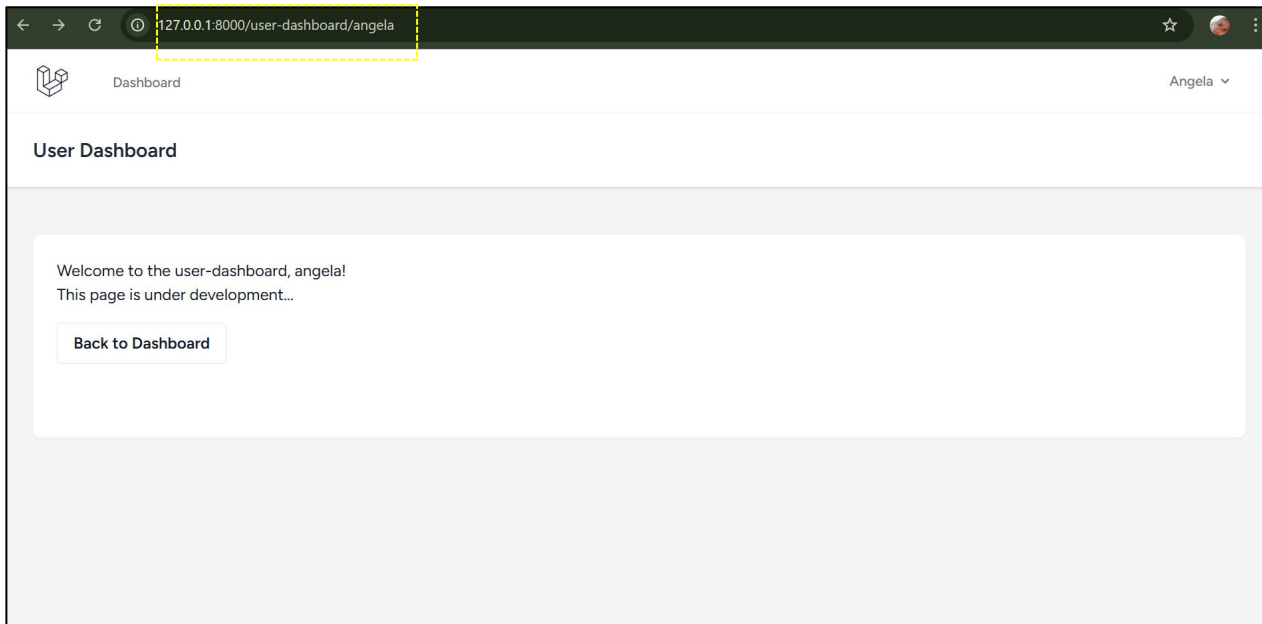
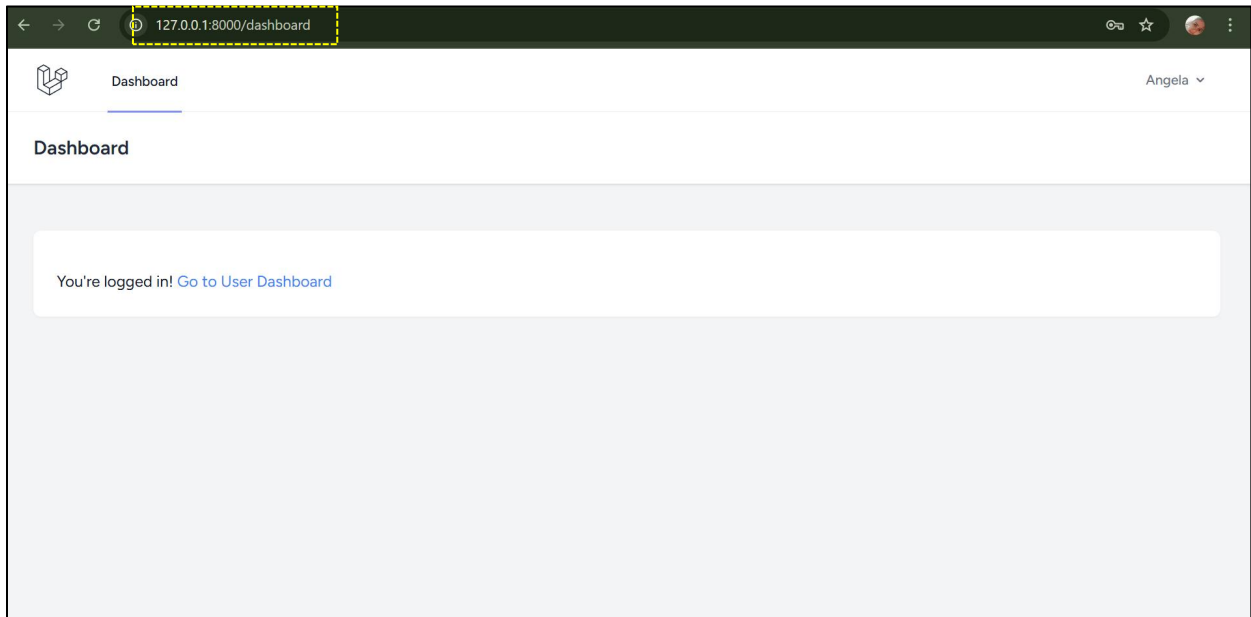
/dashboard: Uses an anonymous function (direct view return), protected by the auth and verified middleware.

/profile: Handled by *ProfileController@edit*, *ProfileController@update*, and *ProfileController@destroy* methods with **auth** middleware.

/user-dashboard/{userId}: Handled by *UserDashboardController@show* with **auth** middleware.

III. Rendered Pages





Update Password

Ensure your account is using a long, random password to stay secure.

Current Password

New Password

Confirm Password

SAVE

update-password-form.blade.php

Delete Account

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.

DELETE ACCOUNT

delete-user-form.blade.php

IV. Controllers Logic

```
app > Http > Controllers > DashboardController.php
1  <?php
2
3
4
5  namespace App\Http\Controllers;
6
7  use Illuminate\Http\Request;
8
9  class DashboardController extends Controller
10 {
11     public function index()
12     {
13         return view('dashboard');
14     }
15
16     public function show($id)
17     {
18         return view('user-dashboard', ['userId' => $id]); // in resources/views/user-dashboard.blade.php
19     }
20 }
21
```

Purpose:

The **DashboardController** is responsible for handling and displaying the general user dashboard as well as user-specific dashboards. It offers two main functionalities: loading the general dashboard view for authenticated users and showing a personalized dashboard for a specific user, identified by their id.

Methods:

- **index()** -Loads and displays the general dashboard view for authenticated users.
- **show(\$id)** - Displays the user-specific dashboard for a particular user, identified by the id parameter.

Purpose:

The `UserDashboardController` is responsible for managing the user-specific dashboard page. It is designed to show personalized content or data based on the user's ID passed as a parameter.

Methods:

`show($userId)`

Purpose: Displays the user dashboard for a specific user, identified by the `userId` parameter.

Functionality: This method retrieves the `userId` passed as a parameter and passes it to the `user-dashboard` view.

Parameter Handling:

`$userId`: The `show` method accepts a `userId` as a parameter, which is used to load personalized content related to that specific user. This parameter is passed from the route, and it helps identify which user's data should be displayed on the dashboard.

```
app > Http > Controllers > UserDashboardController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class UserDashboardController extends Controller
8  {
9      /**
10       * Show the user dashboard.
11       *
12       * @param int $userId
13       * @return \Illuminate\View\View
14       */
15     public function show($userId)
16     {
17         return view('user-dashboard', ['userId' => $userId]);
18     }
19 }
20
```

```
app > Http > Controllers > HomeController.php
1  <?php
2
3
4
5
6
7  namespace App\Http\Controllers;
8
9  class HomeController extends Controller
10 {
11     public function index()
12     {
13         return view('homepage'); // in resources/views/home.blade.php
14     }
15 }
16
```

Purpose:

The `HomeController` is responsible for handling the logic for the homepage of the application. Its main purpose is to process requests that target the homepage of the site and then render the appropriate view, which in this case is the homepage view located at `resources/views/homepage.blade.php`.

Methods:

- `index()`:

Purpose: This method is the default method for handling requests to the homepage of the site. It loads and returns the homepage view.

Functionality: When a request is made to the route that this controller is assigned to (usually the root URL `/`), the `index()` method is invoked. The method simply returns a view (`homepage`) to be rendered in the browser.

```

4 use App\Http\Requests\ProfileUpdateRequest;
5 use Illuminate\Http\RedirectResponse;
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\Auth;
8 use Illuminate\Support\Facades\Redirect;
9 use Illuminate\View\View;
10
11 class ProfileController extends Controller
12 {
13     /**
14      * Display the user's profile form.
15      */
16     public function edit(Request $request): View
17     {
18         return view('profile.edit', [
19             'user' => $request->user(),
20         ]);
21     }
22
23     /**
24      * Update the user's profile information.
25      */
26     public function update(ProfileUpdateRequest $request): RedirectResponse
27     {
28         $request->user()->fill($request->validated());
29
30         if ($request->user()->isDirty('email')) {
31             $request->user()->email_verified_at = null;
32         }
33
34         $request->user()->save();
35
36         return Redirect::route('profile.edit')->with('status', 'profile-updat
37     }
38 }

```

Purpose:

The **ProfileController** is responsible for handling actions related to the user's profile. It allows users to view and update their profile information, as well as delete their account if they choose to.

Methods:

● **edit(Request \$request): View**

- Displays the user's profile form, where they can view and edit their profile details.

Functionality: This method is responsible for showing the profile editing form. It retrieves the currently authenticated user and passes the user object to the profile.edit view so the user can edit their profile.

● **update(ProfileUpdateRequest \$request): RedirectResponse**

- Updates the authenticated user's profile information.

Functionality: This method is used to handle the updating of the user's profile details, such as name and email. It validates the request data using the ProfileUpdateRequest, updates the user's profile in the database, and handles the email verification logic (if the email has changed). After saving the updated information, it redirects the user back to the edit profile page with a success message.

● **destroy(Request \$request): RedirectResponse**

- Deletes the authenticated user's account after confirming the current password.

Functionality: This method handles the deletion of the user's account. It first validates that the password entered by the user matches the current password. After validation, the user's account is deleted from the database, and the user is logged out. The session is invalidated to prevent the user from accessing the site after deletion. Finally, the user is redirected to the homepage.

Parameter Handling:

● **edit(Request \$request):**

The edit method takes a Request parameter, which is used to access the authenticated user (\$request->user()). This parameter is passed automatically by Laravel and represents the current HTTP request, including user data.

● **update(ProfileUpdateRequest \$request):**

The update method takes a ProfileUpdateRequest parameter, which is a custom request class that handles validation for updating the user's profile. It ensures that the incoming data is valid before updating the user's information.

● **destroy(Request \$request):**

The destroy method also takes a Request parameter to access the authenticated user and their session data. The password input field is validated to ensure the user has entered the correct password before their account is deleted.

Explanation of Key Features:

ProfileUpdateRequest:

This is a custom request class used to validate the data being submitted when the user updates their profile. It ensures that the input meets the required format (e.g., valid email, correct password) before proceeding with the update.

Auth::logout():

This function logs the user out of the application. It's used in the destroy method after the user account is deleted to ensure they can no longer access the site.

Redirect::route('profile.edit'):

This is used to redirect the user back to the profile edit page with a success message after their profile is updated.

update(ProfileUpdateRequest \$request):

The update method takes a ProfileUpdateRequest parameter, which is a custom request class that handles validation for updating the user's profile. It ensures that the incoming data is valid before updating the user's information.