

## BLADE FILES

**Dashboard.blade.php**

```
resources > views > 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 </x-app-layout>
27
28
```

**Authenticated User Check:**

If the user is logged in (auth()->check()), it shows a link to the User Dashboard. This link's URL is generated using the route helper, passing the logged-in user's ID (auth()->user()->id) as a parameter.

**Non-authenticated User:**

If the user is not logged in, a prompt appears, asking them to log in to access the dashboard.

**user-dashboard.blade.php**

```
resources > views > 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 <h3>Welcome to the user-dashboard, {{ $userId }}!</h3>
13
14 <p>This page is under development...</p>
15
16 <a href="{{ route('dashboard') }}" class="inline-block border border-blue-500
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>
29 </div>
30 </x-app-layout>
31
32
```

The welcome message greets the user dynamically by using the {{ \$userId }} placeholder, which is replaced with the actual user ID provided by the controller.

**Login.blade.php**

```
resources > views > login.blade.php
1 <x-guest-layout>
2 <!-- Session Status -->
3 <x-auth-session-status class="mt-4" status="session('status')"/>
4
5 <form method="POST" action="{{ route('login') }}">
6 @csrf
7
8 <!-- Email Address -->
9 <div>
10 <input-label for="email" value="{{ __('Email') }}" />
11 <input id="email" class="block w-full type="email" name="email" value="{{ old('email') }}" required autofocus autocomplete="username" />
12 <input-error messages="errors.get('email')"/>
13 </div>
14
15 <!-- Password -->
16 <div class="mt-4">
17 <input-label for="password" value="{{ __('Password') }}" />
18 <input id="password" class="block w-full type="password" name="password" value="{{ old('password') }}" required autocomplete="current-password" />
19 <input-error messages="errors.get('password')"/>
20 </div>
21
22 <!-- Remember Me -->
23 <div class="block mt-4">
24 <label for="remember_me" class="inline-flex items-center">
25 <input id="remember_me" type="checkbox" class="rounded border-gray-300 text-indigo-600 shadow-sm focus:ring-indigo-500 name="remember"/>
26 Remember Me
27 </label>
28 </div>
29
30 <div class="flex items-center justify-end mt-4">
31 <button type="submit" class="bg-indigo-600 text-white px-4 py-2 rounded-md focus:outline-none focus:ring-2 focus:ring-indigo-500">
32 Log In
33 </button>
34
35 <a href="{{ route('forgot-password') }}" class="text-indigo-600 text-sm hover:underline">
36 Forgot Password
37 </a>
38
39 <div class="text-center mt-4">
40 <input type="checkbox" class="checkbox" value="1" />
41 </div>
42
43 <div class="text-center mt-4">
44 <input type="checkbox" class="checkbox" value="1" />
45 </div>
46 </div>
47
```

**A Session Status Component:**

Shows messages related to the session, such as login success or errors.

**Action Links and Button:**

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

## Register.blade.php

```
resources > views > auth > register.blade.php
1 <@guest-layout>
2 <form method="POST" action="{{ route('register') }}">
3     @csrf
4
5     <!-- Name -->
6     <div>
7         <input-label form="name" :value="__('Name')"/>
8         <input id="name" class="block mt-1 w-full" type="text" name="name" :value="old('name')" required autofocus autocomplete="name" />
9         <input-error :messages="$errors->get('name')" class="mt-2"/>
10    </div>
11
12    <!-- Email Address -->
13    <div class="mt-4">
14        <input-label form="email" :value="__('Email')"/>
15        <input id="email" class="block mt-1 w-full" type="email" name="email" :value="old('email')" required autocomplete="username" />
16        <input-error :messages="$errors->get('email')" class="mt-2"/>
17    </div>
18
19    <!-- Password -->
20    <div class="mt-4">
21        <input-label form="password" :value="__('Password')"/>
22        <input id="password" class="block mt-1 w-full" type="password" name="password" required autocomplete="new-password" />
23        <input-error :messages="$errors->get('password')" class="mt-2"/>
24    </div>
25
26    <!-- Confirm Password -->
27    <div class="mt-4">
28        <input-label form="password_confirmation" :value="__('Confirm Password')"/>
29        <input id="password_confirmation" class="block mt-1 w-full" type="password" name="password_confirmation" required autocomplete="new-password" />
30        <input-error :messages="$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-900 rounded-md focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-blue-500" href="#">
35            {{ __('Already registered?') }}
36        </a>
37    </div>
38</form>
```

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

### Action Links and Button:

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

## Homepage.blade.php

```
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                 </div>
15                 <div>
16                     @auth
17                         <p>You are logged in! Go to your <a href="{{ route('dashboard') }}" class="inline-block">
18                             @else
19                             <p>Please <a href="{{ route('login') }}" class="inline-block border border-blue-500">
20                             @endauth
21                     </div>
22                 </div>
23     </div>
24 </x-app-layout>
```

This conditional section uses the @auth directive to check if the user is authenticated.

- If authenticated, it shows a message confirming the user is logged in and provides a link to the dashboard.
- If not authenticated, it prompts the user to log in or register, with links directing to the login and registration routes.

## delete-user-form.blade.php

```
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     </header>
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-onclick.prevent="dispatch('open-modal', 'confirm-user-deletion')">
14        {{ __('Delete Account') }}</x-danger-button>
15
16    <x-modal name="confirm-user-deletion" :show="$errors->userDeletion->isNotEmpty()" focusable>
17        <form method="post" action="{{ route('profile.destroy') }}" class="p-6">
18            @csrf
19            @method('delete')
20
21            <h2 class="text-lg font-medium text-gray-900">
22                {{ __('Are you sure you want to delete your account?') }}
23            </h2>
24
25            <p class="mt-1 text-sm text-gray-600">
26                {{ __('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 delete your account.') }}
27            </p>
28
29            <div class="mt-6">
30                <input-label form="password" :value="__('Password')"/>
31                <input id="password" class="block mt-1 w-full" type="password" name="password" required />
32            </div>
33        </form>
34    </x-modal>
```

This section enables the user to delete their account from the platform.

## update-password-form.blade.php

```
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>
36     </div>
```

This section allows the user to change their account password.

## update-profile-information-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">
```

This section enables the user to update their profile information, such as their name and email address.

## ROUTES

### web.php

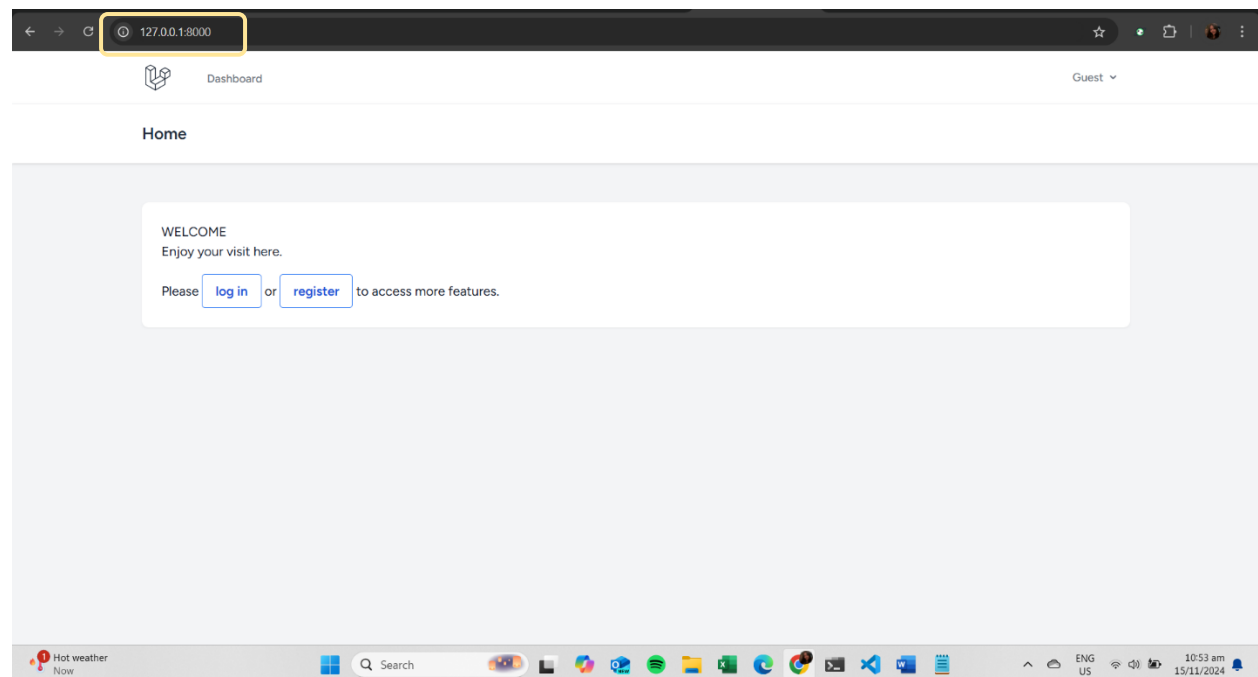
```
routes/web.php
// Routes for the application (supporting routes)
6
7 // Homepage Route
8 Route::get('/', [HomeController::class, 'index']);
9
10 // Dashboard Route
11 Route::get('/dashboard', function () {
12     return view('dashboard');
13 })->middleware(['auth', 'verified'])->name('dashboard');
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_once __DIR__.'/auth.php';
```

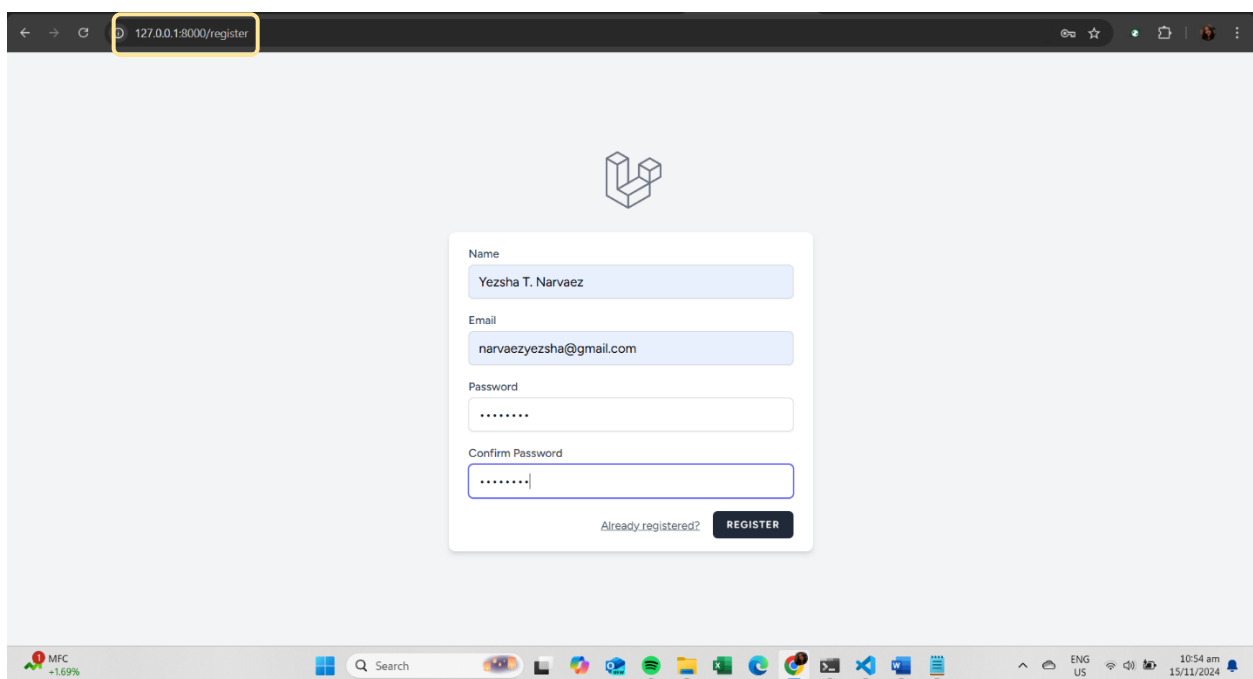
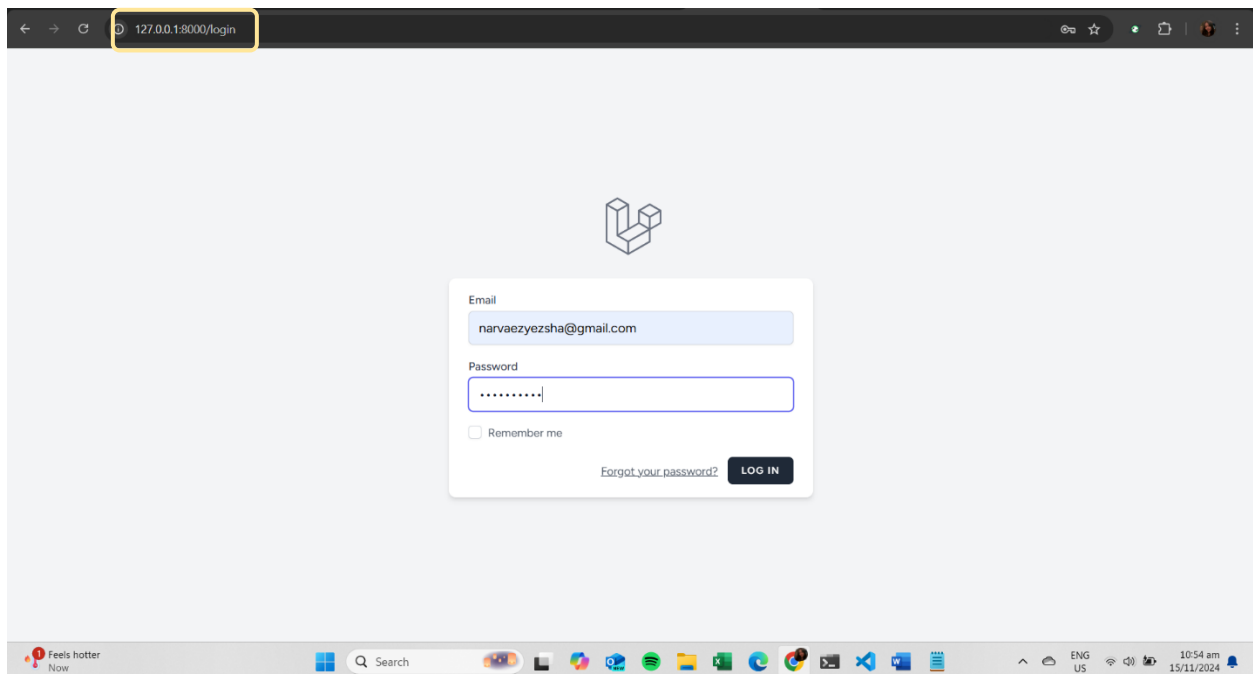
**/:** Managed by the HomeController@index method.

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

**/profile:** Managed by the ProfileController@edit, ProfileController@update, and ProfileController@destroy methods, all protected by the auth middleware.

## RENDERED PAGES







## Dashboard

You're logged in! [Go to User Dashboard](#)



## User Dashboard

Welcome to the user-dashboard, Yezsha!  
This page is under development...

[Back to Dashboard](#)

[Logout](#)

## updated-profile-information-form.blade.php

Dashboard Yezsha

### Profile

#### Profile Information

Update your account's profile information and email address.

Name

Email

SAVE

#### Update Password

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

Current Password

## update-password-form.blade.php

### Update Password

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

Current Password

New Password

Confirm Password

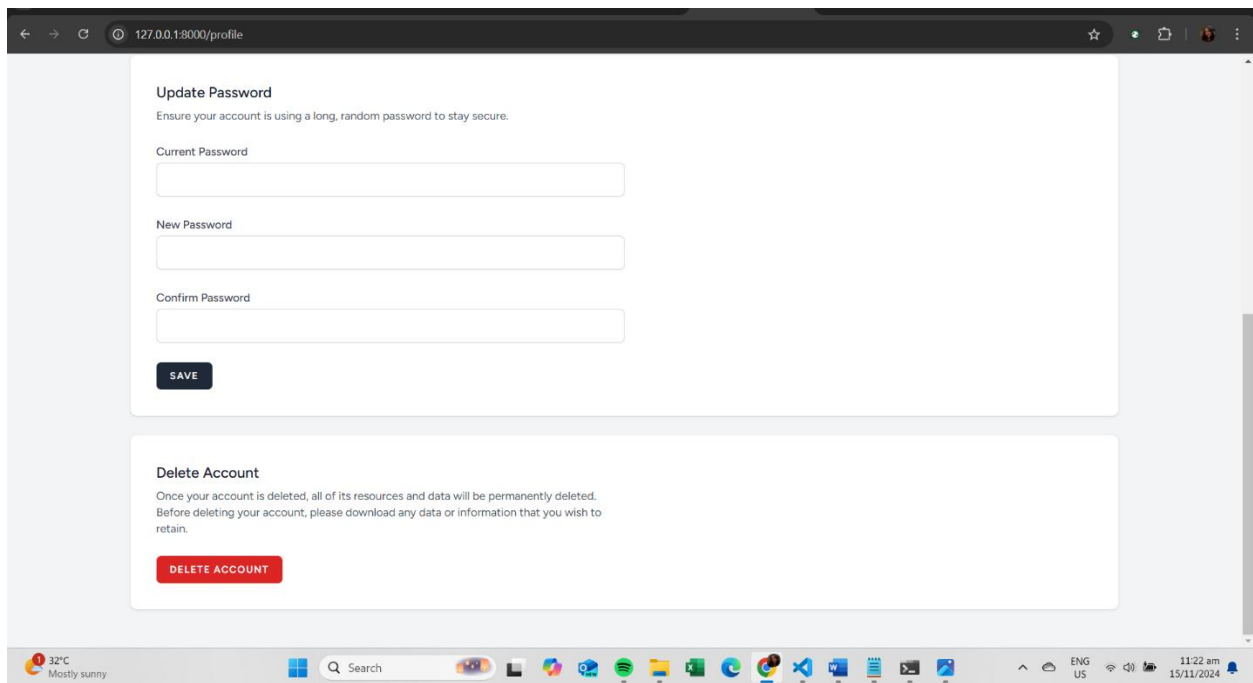
SAVE

#### 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



## CONTROLLERS LOGIC

### *DashboardController.php*

```
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]); //
19     }
20 }
```

The **DashboardController** handles and displays both the general user dashboard and user-specific dashboards. It provides two key functions: loading the general dashboard view for authenticated users and showing a personalized dashboard for a specific user based on 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.



## UserDashboardController.php

```
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 }
```

The

**UserDashboardController** is responsible for managing the user-specific dashboard page. It is designed to display personalized content or data based on the user's ID, which is 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 uses it to load and pass personalized content to the user-dashboard view.

### Parameter Handling:

- **\$userId:** The show method accepts the \$userId parameter, which is used to fetch and display content specific to that user. This parameter is passed from the route and helps determine which user's data should be shown on their dashboard.

## HomeController.php

```
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 }
```

The **HomeController** handles the logic for the homepage of the application. Its main responsibility is to process requests directed to the homepage and render the corresponding view, which is the homepage view located at resources/views/homepage.blade.php.

### Methods:

#### index():

- **Purpose:** This method serves as the default handler for requests targeting the homepage of the site. It loads and returns the homepage view.

- **Functionality:** When a request is made to the route assigned to this controller (typically the root URL /), the index() method is called. It simply returns the

## ProfileController.php

```
app > Http > Controllers > ProfileController.php
13  {
38      }
39
40      /**
41       * Delete the user's account.
42       */
43      public function destroy(Request $request): RedirectResponse
44      {
45          $request->validateWithBag('userDeletion', [
46              'password' => ['required', 'current_password'],
47          ]);
48
49          $user = $request->user();
50
51          Auth::logout();
52
53          $user->delete();
54
55          $request->session()->invalidate();
56          $request->session()->regenerateToken();
57
58          return Redirect::to('/');
59      }
60  }
```

The **ProfileController** is responsible for managing actions related to the user's profile. It enables users to view, update, and delete their profile information.

### Methods:

- **edit(Request \$request): View**
  - **Purpose:** Displays the user's profile form, allowing them to view and edit their profile details.
  - **Functionality:** This method shows the profile editing form by retrieving the currently authenticated user and passing the user object to the profile.edit view for editing.
- **update(ProfileUpdateRequest \$request): RedirectResponse**
  - **Purpose:** Updates the authenticated user's profile information.
  - **Functionality:** This method handles the update of user profile details, such as name and email. It validates the request using ProfileUpdateRequest, saves the

updated information in the database, and manages email verification if the email changes. After updating, it redirects the user back to the profile edit page with a success message.

- **destroy(Request \$request): RedirectResponse**
    - **Purpose:** Deletes the authenticated user's account after confirming their current password.
    - **Functionality:** This method handles account deletion by first validating that the entered password matches the current password. If the password is correct, it deletes the user's account from the database, logs them out, invalidates the session, and redirects them to the homepage.
- 

#### PARAMETER HANDLING:

- **edit(Request \$request):**
    - The edit method accepts a Request parameter, which provides access to the authenticated user (\$request->user()). This parameter is automatically passed by Laravel and represents the current HTTP request, including any user-related data.
  - **update(ProfileUpdateRequest \$request):**
    - The update method accepts a ProfileUpdateRequest parameter, which is a custom request class designed to handle validation for updating the user's profile. It ensures the incoming data is valid before updating the user's information in the database.
  - **destroy(Request \$request):**
    - The destroy method also accepts a Request parameter, which is used to access the authenticated user and their session data. The password input is validated to confirm that the user has entered the correct password before their account is deleted.
- 

#### KEY FEATURES:

- **ProfileUpdateRequest:**
  - This is a custom request class used to validate the data when a user updates their profile. It ensures that the input adheres to the required formats (e.g., valid email, correct password) and passes any necessary validation checks before allowing the update to proceed.
- **Auth::logout():**
  - This function is used to log the user out of the application. It is called in the destroy method after the user's account is deleted to ensure they can no longer access the site with their previous session. It clears the session and effectively logs the user out.

- **Redirect::route('profile.edit'):**
  - This function is used to redirect the user back to the profile edit page after their profile has been updated successfully. It typically includes a success message to inform the user that the update was successful.
- **update(ProfileUpdateRequest \$request):**
  - The update method accepts a ProfileUpdateRequest parameter, which ensures the incoming data is validated before updating the user's profile information. The validation ensures that only correct and valid data is processed before making any changes.
- **destroy(Request \$request):**
  - The destroy method accepts a Request parameter, which is used to access the authenticated user and their session data. It ensures the user's password is correct before deleting their account. After the account is deleted, the user is logged out, and the session is invalidated.