

DOCUMENTATION: Laboratory 1 - Controllers

I. 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
21 @else
22 <p>Please log in to access your dashboard.</p>
23 @endif
24 </div>
25 </div>
26 </div>
27 </x-app-layout>
28

```

Renders the dashboard page and checks if the user is authenticated using **auth()->check()**

- **Authenticated Users:** If the user is logged in, a link to their personalized dashboard is displayed, passing the **userId** as a route parameter.
- **Guest Users:** If the user is not logged in, a message prompting them to log in is shown.

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 <h1>Welcome to the user-dashboard, {{ $userId }}!</h1>
13
14
15 <p>This page is under development...</p>
16
17
18 <a href="{{ route('dashboard') }}" class="inline-block border border-blue-500 text-blue-700 font-
19 Back to Dashboard
20 </a>
21
22 <form action="{{ route('logout') }}" method="POST" style="margin-top: 10px;">
23 @csrf
24 <button type="submit" class="bg-red-500 text-white px-4 py-2 rounded hover:bg-red-600 trans-
25 Logout
26 </button>
27 </form>
28 </div>
29 </div>
30 </div>
31 </x-app-layout>
32

```

Welcome Message: Displays a welcome message with the dynamic **userId** to personalize the page.

Session Status: Displays any session status messages like successful login notification

Login Form:

- **Email Field:** Collects the user's email with error handling
- **Password Field:** Collects the user's password with error handling
- **Remember Me Checkbox:** Allows users to stay logged in
- **Forgot Password Link:** Redirects to password reset
- **Log In Button:** Submits the form for user authentication

login.blade.php

```

resources > views > auth > login.blade.php
1 <!-- Session Status -->
2 @auth-session-status class="mb-4" :status="session('status')" />
3
4 <form method="POST" action="{{ route('login') }}">
5 @csrf
6
7 <!-- Email Address -->
8 <div>
9 <input-label form="email" :value="__('Email')"/>
10 <input id="email" class="block w-full" type="email" name="email" :value="old('email')" required autofocus autocomplete="username" />
11 <input-error :messages="errors->get('email')" class="mt-2" />
12 </div>
13
14 <!-- Password -->
15 <div class="mt-4">
16 <input-label form="password" :value="__('Password')"/>
17 <input id="password" class="block w-full" type="password" name="password" required autocomplete="current-password" />
18 <input-error :messages="errors->get('password')" class="mt-2" />
19 </div>
20
21 <!-- Remember Me -->
22 <div class="block mt-4">
23 <label form="remember_me" class="inline-flex items-center">
24 <input id="remember_me" type="checkbox" class="rounded border-gray-300 text-indigo-600 shadow-sm focusing:ring-indigo-500" name="remember" />
25 <span class="ml-2 text-sm text-gray-600">{{ __('Remember me') }}</span>
26 </label>
27 </div>
28
29 <div class="flex items-center justify-end mt-4">
30 @if (Route::has('password.request'))
31 <a class="underline text-sm text-gray-600 hover:text-gray-900 rounded-md focus:outline-none focusing:ring-2 focusing:ring-indigo-500" href="{{ route('password.request') }}">
32 {{ __('Forgot your password?') }}
33 </a>
34 @endif
35
36 <input type="submit" value="Log In" />
37 </div>
38 </form>
39
40 <!-- Guest User -->
41 </x-guest-layout>
42

```

register.blade.php

Provides a user registration form with fields for name, email, password, and password confirmation. It includes error handling for each field, a link to the login page for existing users, and a submit button to register a new account.

Displays a welcome message and options based on the user's authentication status.

- **Welcome Message:** Shows a general welcome to all users.
- **Authenticated User Message:** If the user is logged in, a message confirms this and provides a link to the dashboard.
- **Guest Message:** If the user is not logged in, prompts to log in or register, with links to the login and

homepage.blade.php

This enables account deletion with a two-step confirmation process. It displays a **"Delete Account"** button that opens a modal, where users must enter their password to confirm. Users can either confirm deletion or cancel the action, ensuring security and data retention reminders.

delete-user.blade.php

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 <@input-label for="update_password_current_password" value="{{ __('Current Password') }}" />
18 <@text-input id="update_password_current_password" name="current_password" type="password" class="mt-1 block w-full autocomplete="current-password" />
19 <@input-error messages="errors.updatePassword.get('current_password')" class="mt-2" />
20 </div>
21
22 <div>
23 <@input-label for="update_password_password" value="{{ __('New Password') }}" />
24 <@text-input id="update_password_password" name="password" type="password" class="mt-1 block w-full autocomplete="new-password" />
25 <@input-error messages="errors.updatePassword.get('password')" class="mt-2" />
26 </div>
27
28 <div>
29 <@input-label for="update_password_password_confirmation" value="{{ __('Confirm Password') }}" />
30 <@text-input id="update_password_password_confirmation" name="password_confirmation" type="password" class="mt-1 block w-full autocomplete="new-password" />
31 <@input-error messages="errors.updatePassword.get('password_confirmation')" class="mt-2" />
32 </div>
33
34 <div class="flex items-center gap-4">
35 <@primary-button {{ __('Save') }} />
36
37 <@if (session('status')) == 'password-updated'>
38 <p>
39 <@x-data="{ show: true }">
40 <@x-show="show">
41 <@x-transition>
42 <@x-init="setTimeout(() => show = false, 2000)">
43 <@class="text-sm text-gray-600">
44 <{{ __('Saved.') }} />
45 </p>
46 </endif>
47 </div>
48 </form>
49 </section>
```

This allows users to update their password. It includes fields for the current password, new password, and password confirmation. Upon successful submission, a success message is displayed briefly.

update-profile-information.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 <@input-label for="name" value="{{ __('Name') }}" />
22 <@text-input id="name" name="name" type="text" class="mt-1 block w-full" value="{{ old('name', $user->name) }}" required autofocus autocomplete="name" />
23 <@input-error class="mt-2" messages="errors.get('name')" />
24 </div>
25
26 <div>
27 <@input-label for="email" value="{{ __('Email') }}" />
28 <@text-input id="email" name="email" type="email" class="mt-1 block w-full" value="{{ old('email', $user->email) }}" required autocomplete="username" />
29 <@input-error class="mt-2" messages="errors.get('email')" />
30 </div>
31
32 <@if ($user instanceof \Illuminate\Contracts\Auth\CanVerifyEmail && ! $user->hasVerifiedEmail())>
33 <div>
34 <p class="text-sm text-gray-600">
35 <{{ __('Your email address is unverified.') }}>
36
37 <@button form="send-verification" class="underline text-sm text-gray-600 hover:text-gray-900 rounded-md focus:outline-none focus:ring-2 focus:ring-indigo-500">
38 <{{ __('Click here to re-send the verification email.') }}>
39 </button>
40 </p>
41 </div>
42
43 <@if (session('status')) == 'verification-link-sent'>
44 <p class="mt-2 font-medium text-sm text-green-600">
45 <{{ __('A new verification link has been sent to your email address.') }}>
46 </p>
47 </endif>
48 </endif>
49 </div>
50
51 <div class="flex items-center gap-4">
52 <@primary-button {{ __('Save') }} />
53
54 <@if (session('status')) == 'profile-updated'>
55 <p>
56 <@x-data="{ show: true }">
57 <@x-show="show">
58 <@x-transition>
59 <@x-init="setTimeout(() => show = false, 2000)">
60 <@class="text-sm text-gray-600">
61 <{{ __('Saved.') }} />
62 </p>
63 </endif>
64 </div>
```

This allows users to update their profile information (name and email). It includes an option to resend the email verification link if the user's email is unverified. After saving changes, a success message is displayed briefly.

II. Routes

```
routes > web.php
1  <?php
2  use App\Http\Controllers\HomeController;
3  use App\Http\Controllers\UserDashboardController;
4  use App\Http\Controllers\ProfileController;
5  use Illuminate\Support\Facades\Route;
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
```

Homepage Route:

- Loads the homepage view (HomeController@index), accessible to all visitors.

Dashboard Route:

- Provides access to the dashboard view.
- Protected by auth and verified middleware, ensuring only authenticated and email-verified users can access it.

Profile Routes (Requires authentication):

- **View Profile:** Displays the user profile page (GET /profile).
- **Update Profile:** Updates user profile information (PATCH /profile).
- **Delete Profile:** Deletes the user account (DELETE /profile).

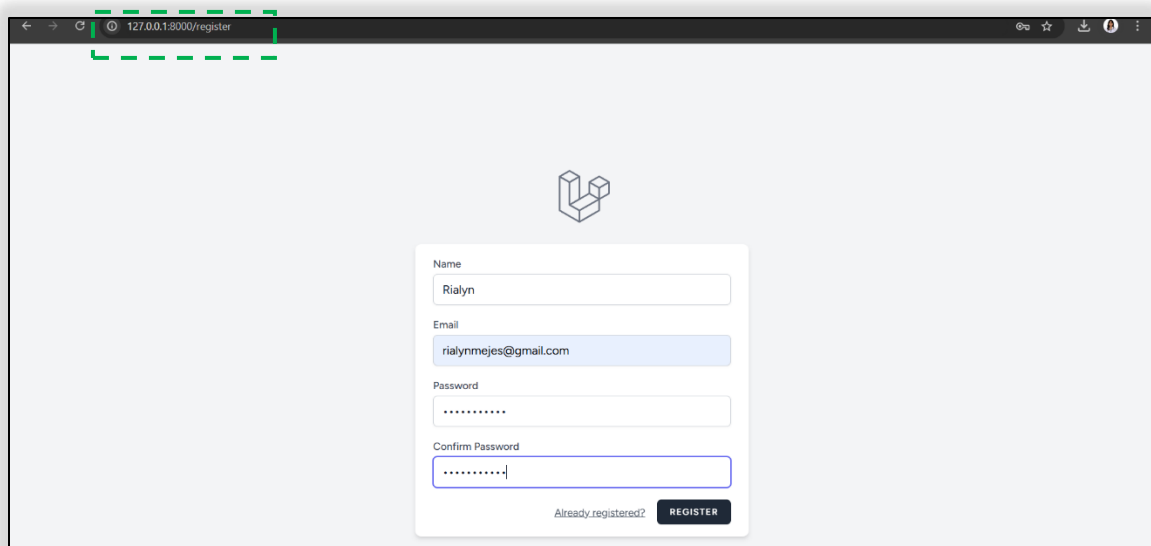
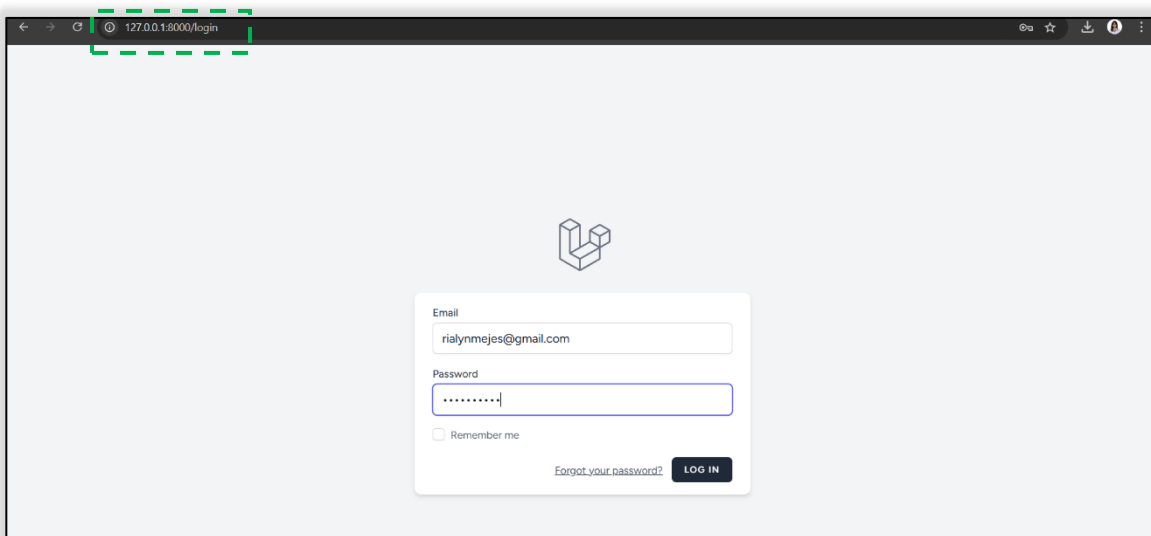
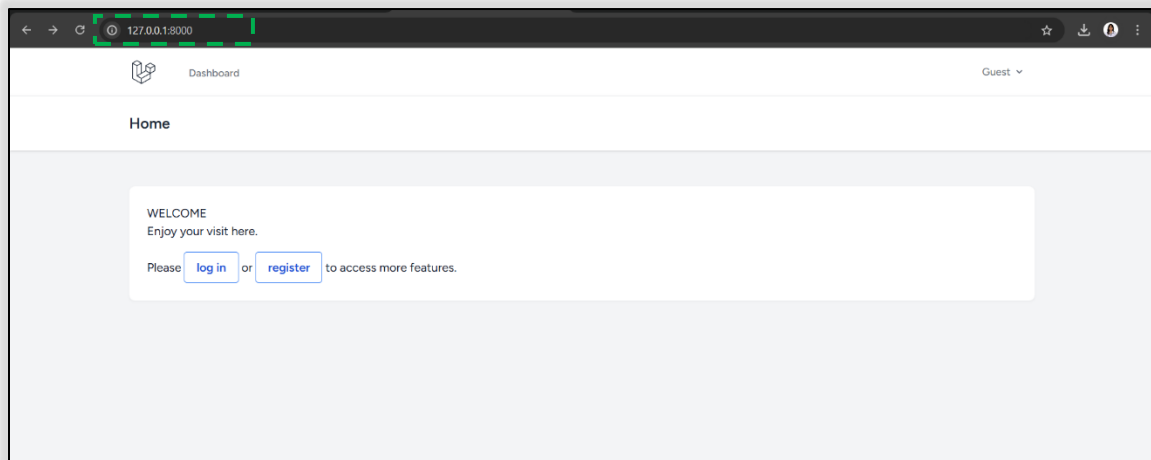
User Dashboard Route:

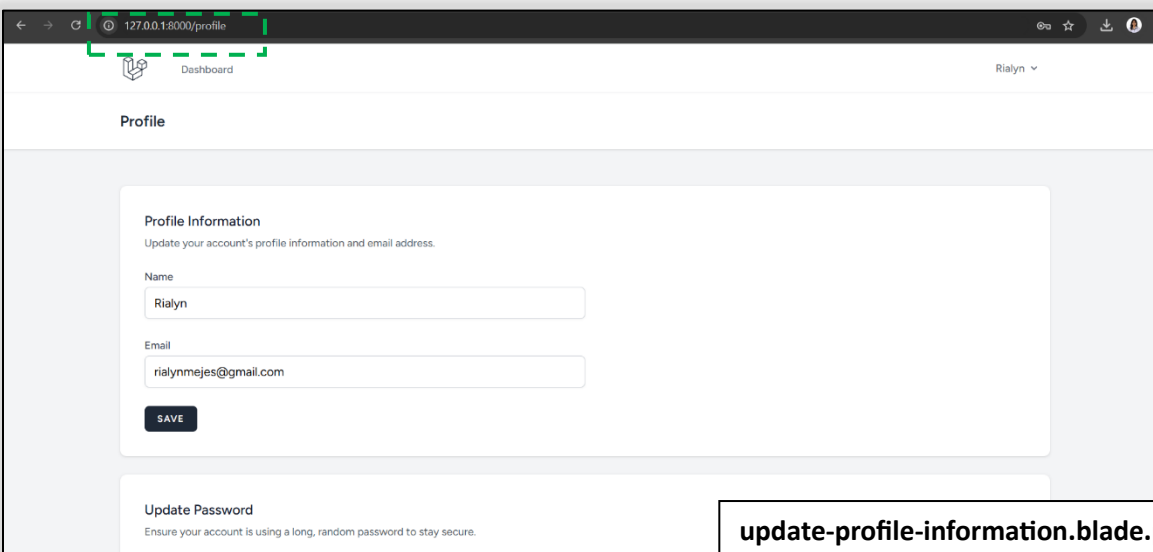
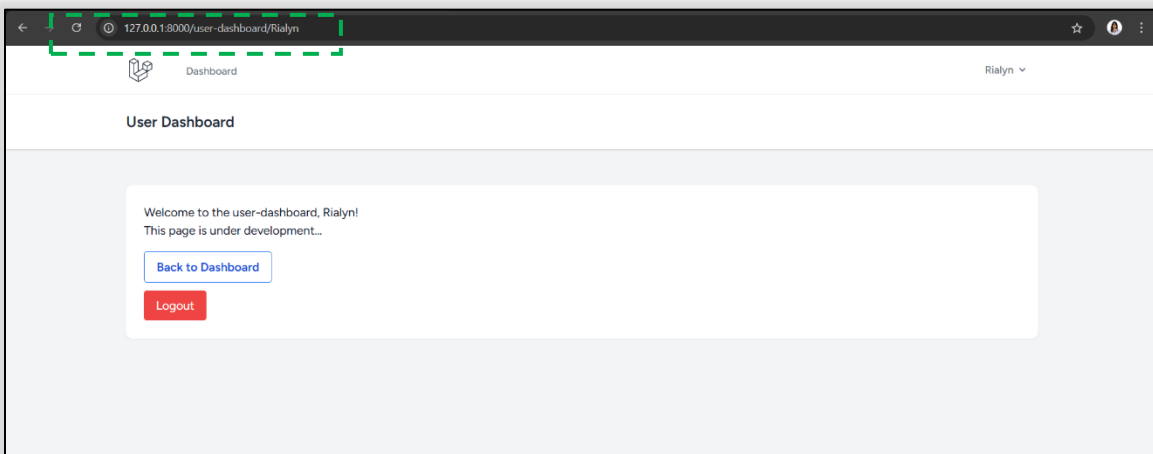
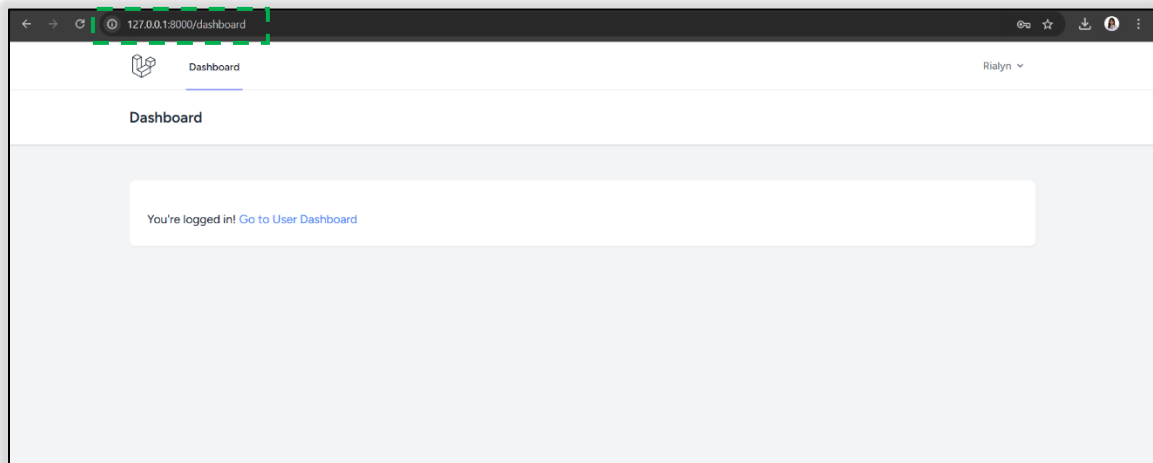
- Displays a personalized dashboard for a specific user, identified by the userId parameter.
- Protected by auth middleware to ensure the user is logged in.

Authentication Routes:

- Includes Laravel's built-in authentication routes (auth.php) for login, registration,

III. Rendered Pages





The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/profile". The page contains a form for updating a password. At the top, there is an email address field with "rialynmejess@gmail.com" and a "SAVE" button. Below this, the "Update Password" section is highlighted with a green dashed border. It includes a sub-header "Update Password", a note "Ensure your account is using a long, random password to stay secure.", and three input fields: "Current Password", "New Password", and "Confirm Password". A "SAVE" button is located at the bottom of this section. Below the highlighted section, there is a "Delete Account" link.

rialynmejess@gmail.com

SAVE

Update Password

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

Current Password

New Password

Confirm Password

SAVE

Delete Account

update-password-form.blade.php

The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/profile". The page contains a form for deleting an account. At the top, there is a "SAVE" button. Below this, the "Delete Account" section is highlighted with a green dashed border. It includes a sub-header "Delete Account", a note "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.", and a red "DELETE ACCOUNT" button.

SAVE

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

IV. Controllers Logic

```
user-dashboard.blade.php DashboardController.php X
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
```

DashboardController: Handles the logic for displaying the dashboard and user-specific pages.

- **index():** Renders the main dashboard view.
- **show(\$id):** Renders the user-dashboard view with a dynamic `userId` passed to the view.

```
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
```

UserDashboardController manages user-specific dashboard views. Its **show** method takes a **userId** parameter to load and display a personalized dashboard page for that user.

- Method (**show**):
 - **Purpose:** Displays the user-specific dashboard.
 - **Parameter:** Accepts **\$userId** to identify the user.
 - **Return:** Loads the **user-dashboard** view and passes **userId** data for personalized content.

```
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
```

HomeController is responsible for managing the display of the homepage

- Method (**index**):
 - **Purpose:** Loads and displays the **homepage** view.
 - **Return:** Returns the **homepage** view


```

app > Http > Controllers > ProfileController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Http\Requests\ProfileUpdateRequest;
6  use Illuminate\Http\RedirectResponse;
7  use Illuminate\Http\Request;
8  use Illuminate\Support\Facades\Auth;
9  use Illuminate\Support\Facades\Redirect;
10 use Illuminate\View\View;
11
12 class ProfileController extends Controller
13 {
14     /**
15      * Display the user's profile form.
16      */
17     public function edit(Request $request): View
18     {
19         return view('profile.edit', [
20             'user' => $request->user(),
21         ]);
22     }
23
24     /**
25      * Update the user's profile information.
26      */
27     public function update(ProfileUpdateRequest $request): RedirectResponse
28     {
29         $request->user()->fill($request->validated());
30
31         if ($request->user()->isDirty('email')) {
32             $request->user()->email_verified_at = null;
33         }
34
35         $request->user()->save();
36
37         return Redirect::route('profile.edit')->with('status', 'profile-updated');
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 }

```

ProfileController manages user profile operations, including viewing, updating, and deleting the profile.

- **Method (edit):**
 - **Purpose:** Displays the profile edit form.
 - **Parameter:** Uses the **Request** object to retrieve the current user.
 - **Return:** Loads the **profile.edit** view, passing the user's data.
- **Method (update):**
 - **Purpose:** Updates the user's profile information.
 - **Parameter:** Uses a **ProfileUpdateRequest** for validated profile data.
 - **Logic:** Sets new profile data, resets email verification if email changes, and saves updates.
 - **Return:** Redirects back to **profile.edit** with a success message.
- **Method (destroy):**
 - **Purpose:** Deletes the user's account.
 - **Parameter:** Uses **Request** to validate the user's password for security.
 - **Logic:** Validates password, logs out the user, deletes the account, and invalidates the session.
 - **Return:** Redirects to the homepage after account deletion.