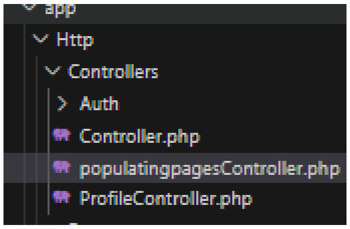
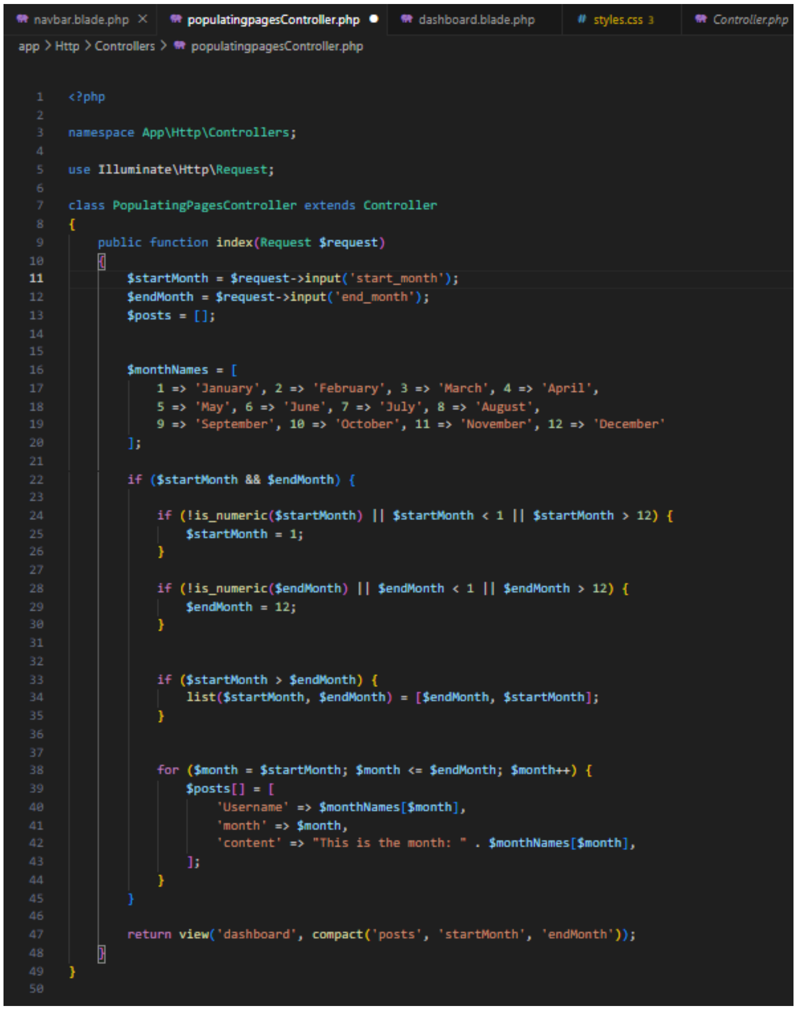
Quintanilla, Lord Zaro Fiber A.
BS IT – 3 C
Web Dev Populating the webpages

The controllers that we made:



Here is the populating the web pages controller:

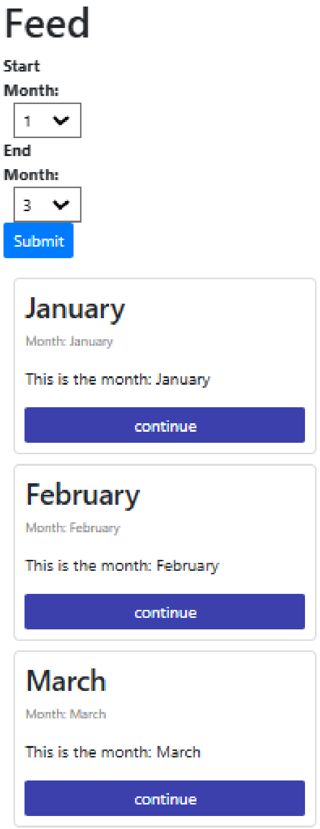

navbar.blade.php
populatingpagesController.php
dashboard.blade.php
styles.css 3
Controller.php
app > Http > Controllers > populatingpagesController.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class PopulatingPagesController extends Controller
{
    public function index(Request $request)
    {
        $startMonth = $request->input('start_month');
        $endMonth = $request->input('end_month');
        $posts = [];


        $monthNames = [
            1 => 'January', 2 => 'February', 3 => 'March', 4 => 'April',
            5 => 'May', 6 => 'June', 7 => 'July', 8 => 'August',
            9 => 'September', 10 => 'October', 11 => 'November', 12 => 'December'
        ];

        if ($startMonth && $endMonth) {

            if (!is_numeric($startMonth) || $startMonth < 1 || $startMonth > 12) {
                $startMonth = 1;
            }

            if (!is_numeric($endMonth) || $endMonth < 1 || $endMonth > 12) {
                $endMonth = 12;
            }


            if ($startMonth > $endMonth) {
                list($startMonth, $endMonth) = [$endMonth, $startMonth];
            }


            for ($month = $startMonth; $month <= $endMonth; $month++) {
                $posts[] = [
                    'Username' => $monthNames[$month],
                    'month' => $month,
                    'content' => "This is the month: " . $monthNames[$month],
                ];
            }
        }

        return view('dashboard', compact('posts', 'startMonth', 'endMonth'));
    }
}

```

In this one we made it as to populate the dashboard by simply adding the months that the user would want to track the expense so we have put on the start month and the end month here and the dash board simply ask for the user on what month should they start to track the expense.
January = 1
….
December = 12

when the user starts at the 1st month to the 3rd month it shows the months January to March

Here is the result



Feed

Start Month: 1

End Month: 3

January

Month: January

This is the month: January

February

Month: February

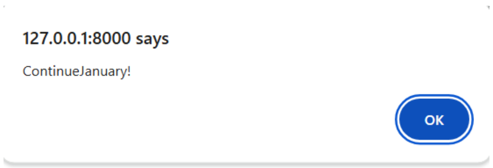
This is the month: February

March

Month: March

This is the month: March

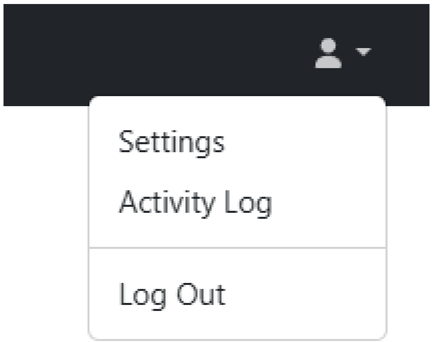
When pressed continue:

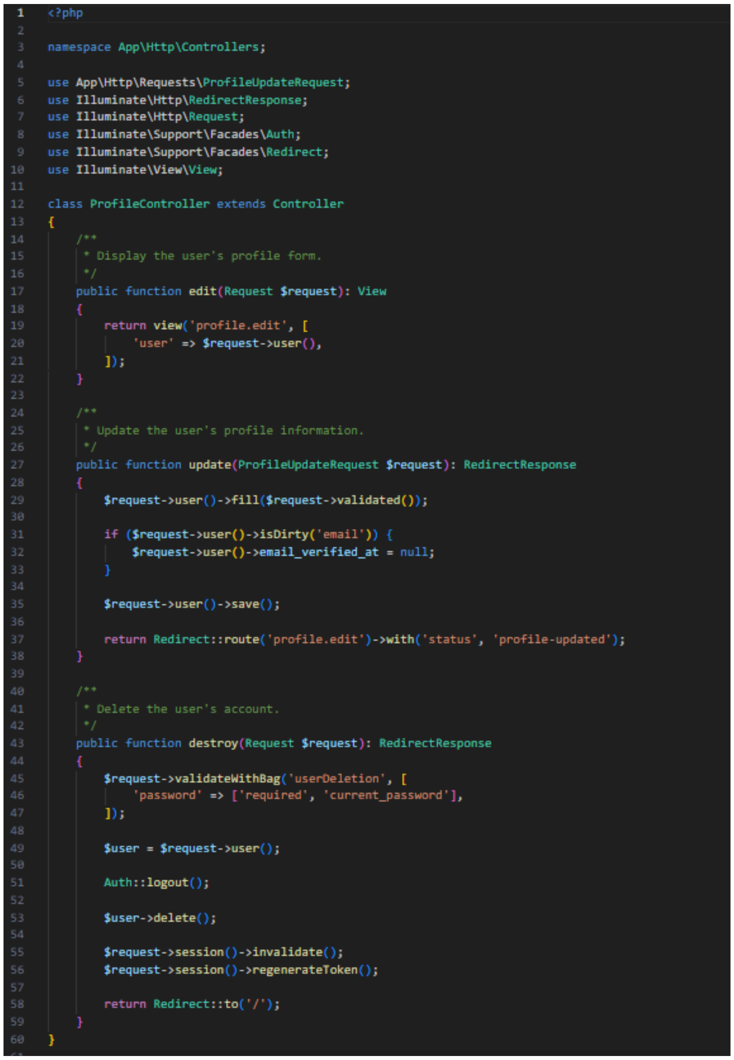


127.0.0.1:8000 says

ContinueJanuary!

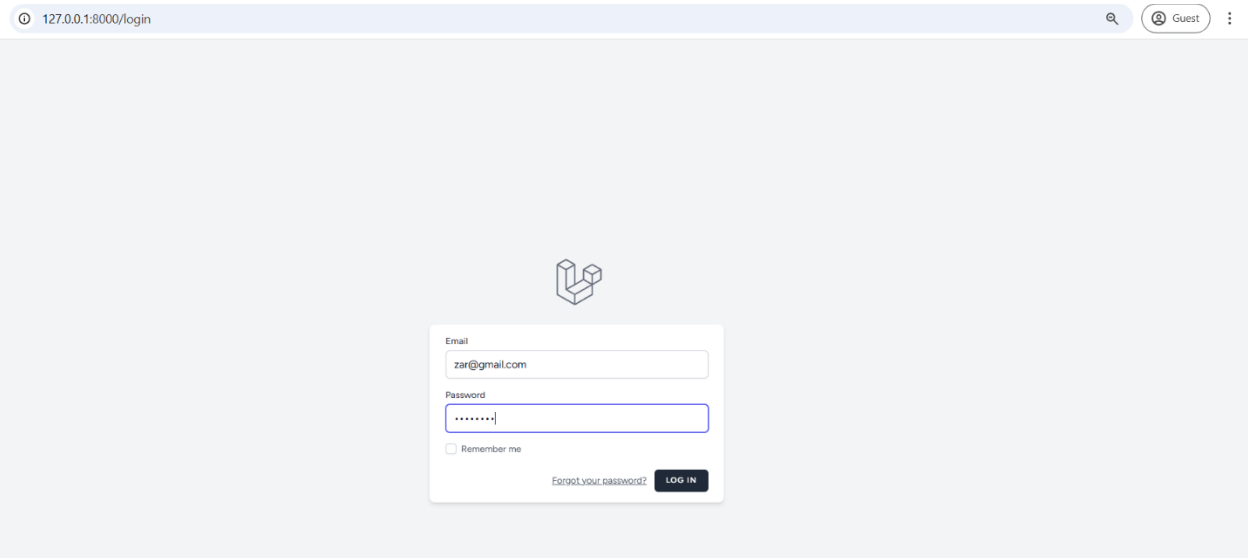
Next one is the profile controller It lets the user edits the profile info like the email, update the password and etc (CRUD).





```php
<?php

namespace App\Http\Controllers;

use App\Http\Requests\ProfileUpdateRequest;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Redirect;
use Illuminate\View\View;

class ProfileController extends Controller
{
    /**
     * Display the user's profile form.
     */
    public function edit(Request $request): View
    {
        return view('profile.edit', [
            'user' => $request->user(),
        ]);
    }

    /**
     * Update the user's profile information.
     */
    public function update(ProfileUpdateRequest $request): RedirectResponse
    {
        $request->user()->fill($request->validated());

        if ($request->user()->isDirty('email')) {
            $request->user()->email_verified_at = null;
        }

        $request->user()->save();

        return Redirect::route('profile.edit')->with('status', 'profile-updated');
    }

    /**
     * Delete the user's account.
     */
    public function destroy(Request $request): RedirectResponse
    {
        $request->validateWithBag('userDeletion', [
            'password' => ['required', 'current_password'],
        ]);

        $user = $request->user();

        Auth::logout();

        $user->delete();

        $request->session()->invalidate();
        $request->session()->regenerateToken();

        return Redirect::to('/');
    }
}
```

Here is the sample log in page.



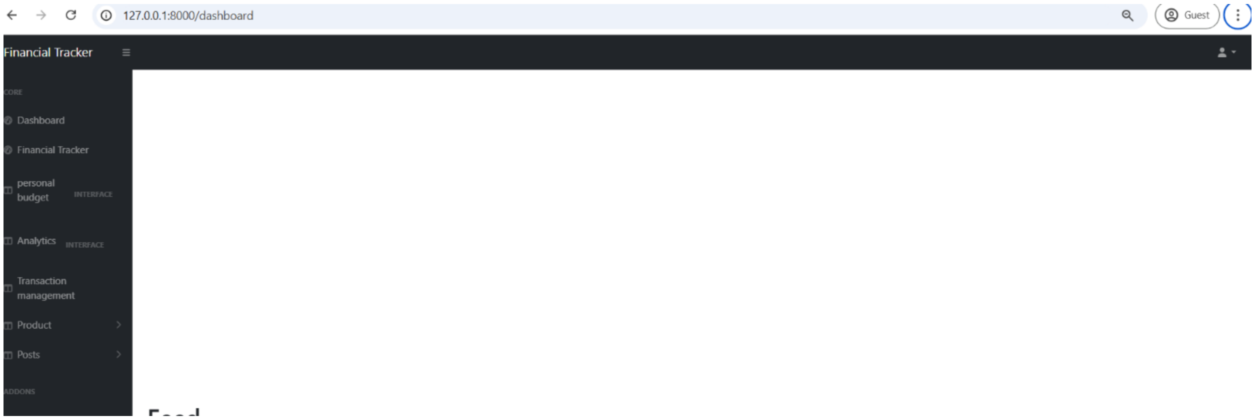
127.0.0.1:8000/login

Email: zar@gmail.com

Password: ••••••••

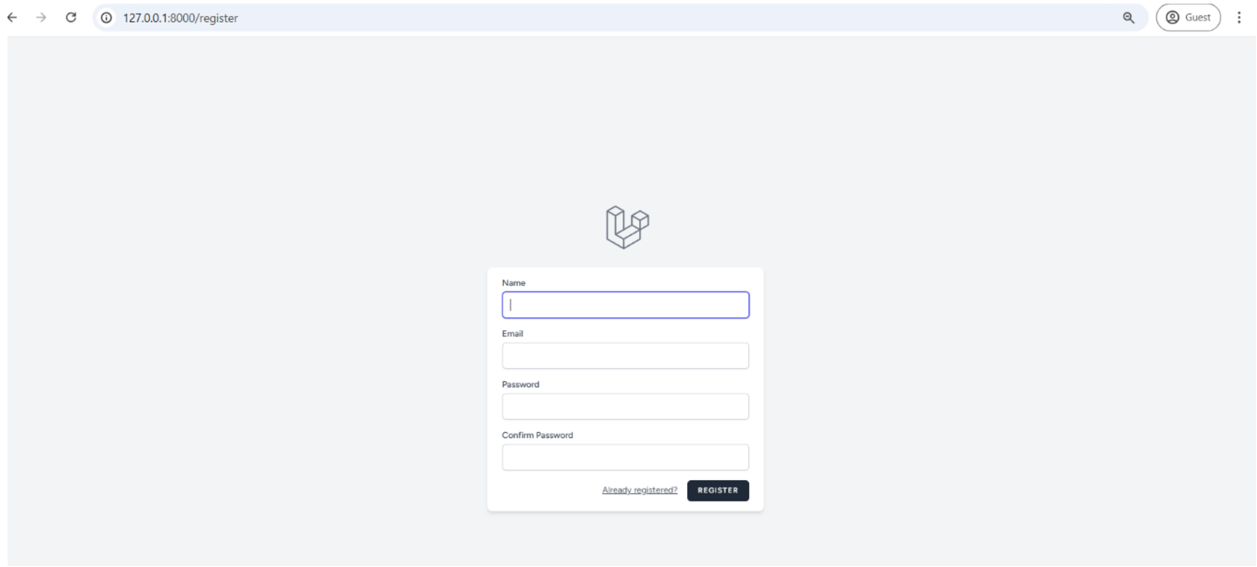
□ Remember me

after logging in here is the dashboard



127.0.0.1:8000/dashboard

The user can then choose to register an account



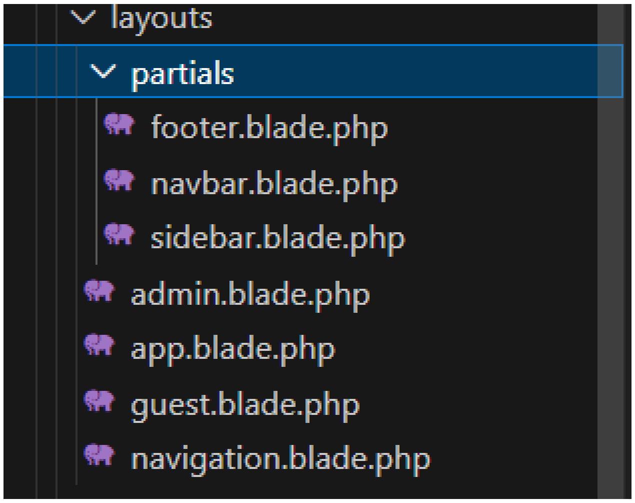
127.0.0.1:8000/register

Name

Email

Password

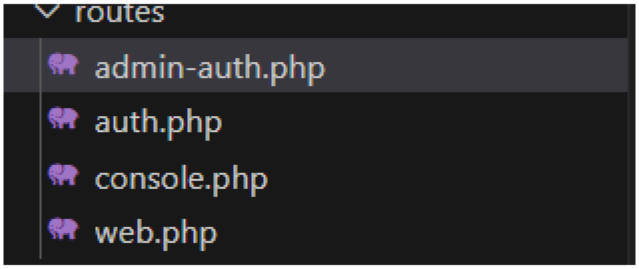
Confirm Password



```
layouts
  partials
    footer.blade.php
    navbar.blade.php
    sidebar.blade.php
  admin.blade.php
  app.blade.php
  guest.blade.php
  navigation.blade.php
```

And in the layout contains the footer, navigaton,sidebar and other design of the web app

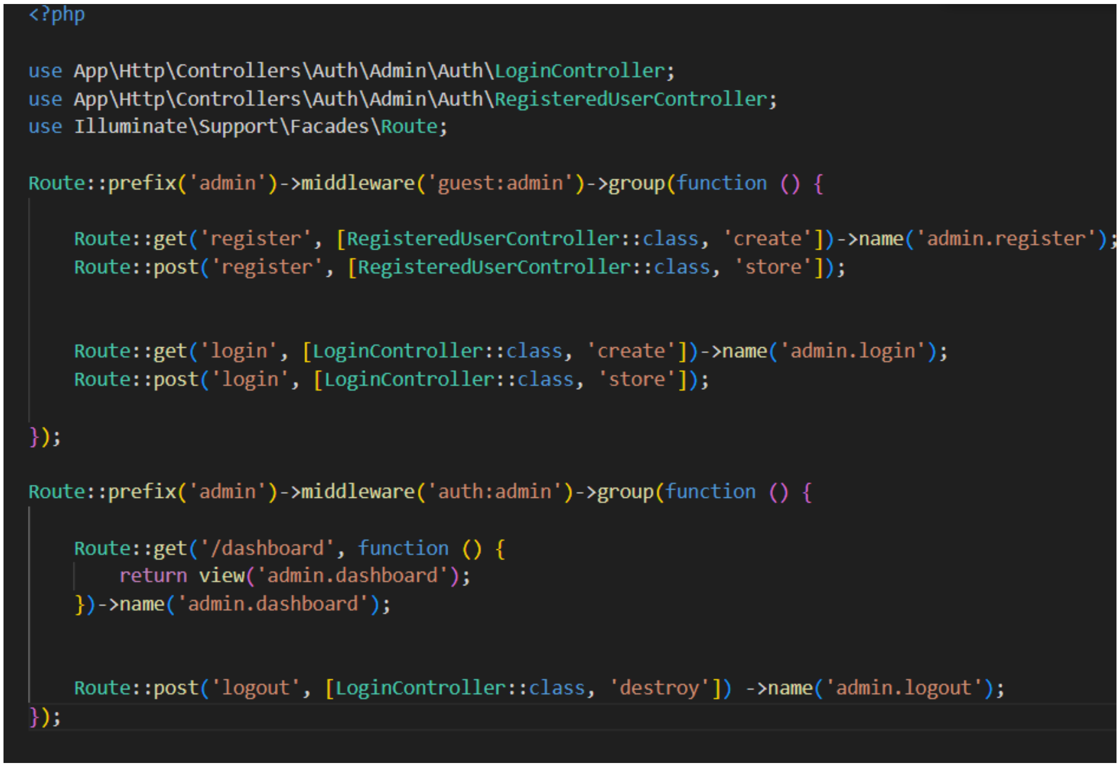
here are the routes:



```
routes
  admin-auth.php
  auth.php
  console.php
  web.php
```

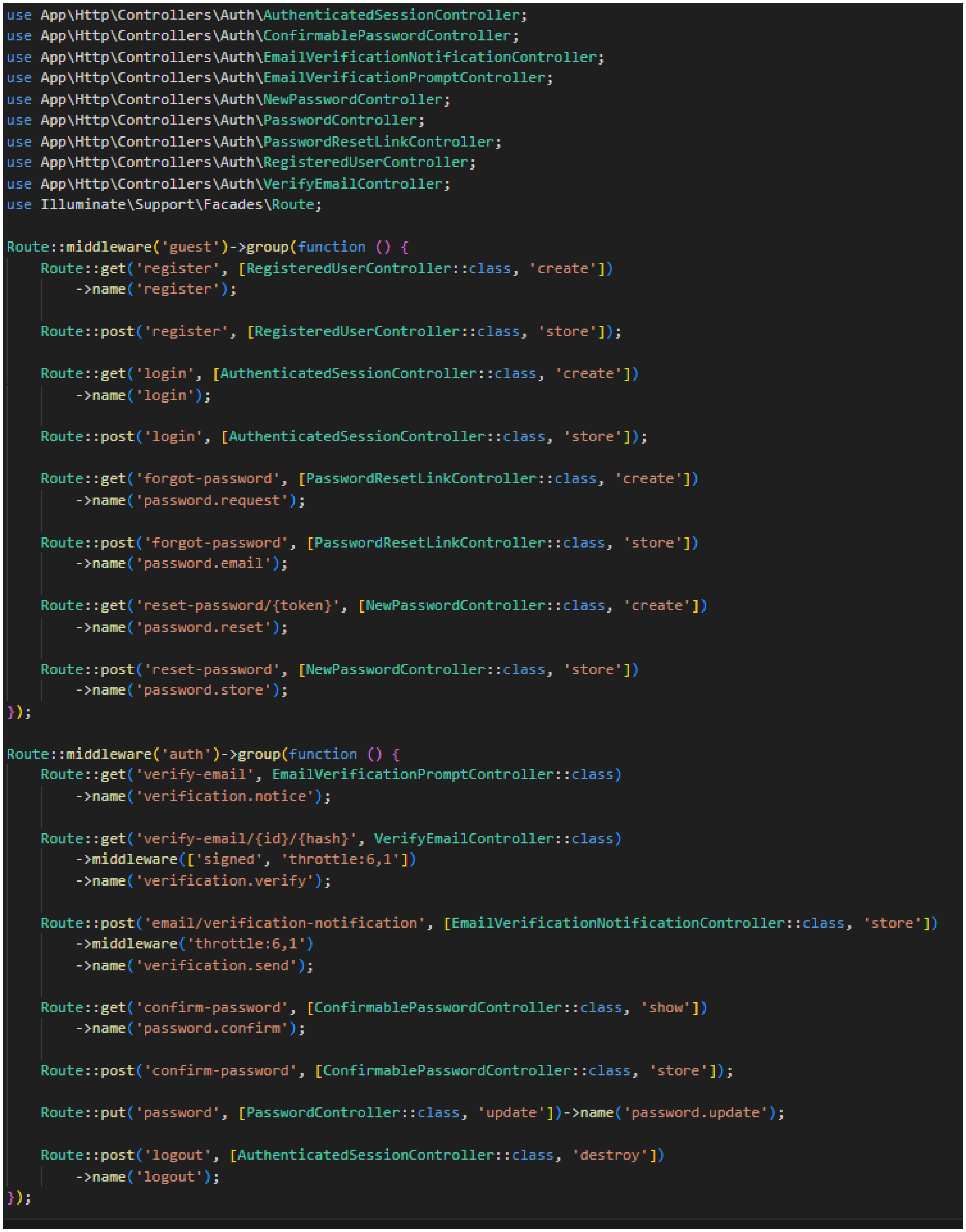
Made a separate one that would be used for the admin routes

also have the admin/login where the user can chose to login as a user or admin



```php
<?php

use App\Http\Controllers\Auth\Admin\Auth\LoginController;
use App\Http\Controllers\Auth\Admin\Auth\RegisteredUserController;
use Illuminate\Support\Facades\Route;

Route::prefix('admin')->middleware('guest:admin')->group(function () {

    Route::get('register', [RegisteredUserController::class, 'create'])->name('admin.register');
    Route::post('register', [RegisteredUserController::class, 'store']);


    Route::get('login', [LoginController::class, 'create'])->name('admin.login');
    Route::post('login', [LoginController::class, 'store']);

});

Route::prefix('admin')->middleware('auth:admin')->group(function () {

    Route::get('/dashboard', function () {
        return view('admin.dashboard');
    })->name('admin.dashboard');


    Route::post('logout', [LoginController::class, 'destroy']) ->name('admin.logout');
});
```

(CRUD)



```php
use App\Http\Controllers\Auth\AuthenticatedSessionController;
use App\Http\Controllers\Auth\ConfirmablePasswordController;
use App\Http\Controllers\Auth\EmailVerificationNotificationController;
use App\Http\Controllers\Auth\EmailVerificationPromptController;
use App\Http\Controllers\Auth\NewPasswordController;
use App\Http\Controllers\Auth\PasswordController;
use App\Http\Controllers\Auth\PasswordResetLinkController;
use App\Http\Controllers\Auth\RegisteredUserController;
use App\Http\Controllers\Auth\VerifyEmailController;
use Illuminate\Support\Facades\Route;

Route::middleware('guest')->group(function () {
    Route::get('register', [RegisteredUserController::class, 'create'])
        ->name('register');

    Route::post('register', [RegisteredUserController::class, 'store']);

    Route::get('login', [AuthenticatedSessionController::class, 'create'])
        ->name('login');

    Route::post('login', [AuthenticatedSessionController::class, 'store']);

    Route::get('forgot-password', [PasswordResetLinkController::class, 'create'])
        ->name('password.request');

    Route::post('forgot-password', [PasswordResetLinkController::class, 'store'])
        ->name('password.email');

    Route::get('reset-password/{token}', [NewPasswordController::class, 'create'])
        ->name('password.reset');

    Route::post('reset-password', [NewPasswordController::class, 'store'])
        ->name('password.store');
});

Route::middleware('auth')->group(function () {
    Route::get('verify-email', EmailVerificationPromptController::class)
        ->name('verification.notice');

    Route::get('verify-email/{id}/{hash}', VerifyEmailController::class)
        ->middleware(['signed', 'throttle:6,1'])
        ->name('verification.verify');

    Route::post('email/verification-notification', [EmailVerificationNotificationController::class, 'store'])
        ->middleware('throttle:6,1')
        ->name('verification.send');

    Route::get('confirm-password', [ConfirmablePasswordController::class, 'show'])
        ->name('password.confirm');

    Route::post('confirm-password', [ConfirmablePasswordController::class, 'store']);

    Route::put('password', [PasswordController::class, 'update'])->name('password.update');

    Route::post('logout', [AuthenticatedSessionController::class, 'destroy'])
        ->name('logout');
});
```

Also have the CRUD in register a user where one can make an account update delete and log out

We also tried to connect the database and see how it works. The registered user and admin would be put in there and we also plan to put there the tracker of the budget, expenses and etc. of the user it is

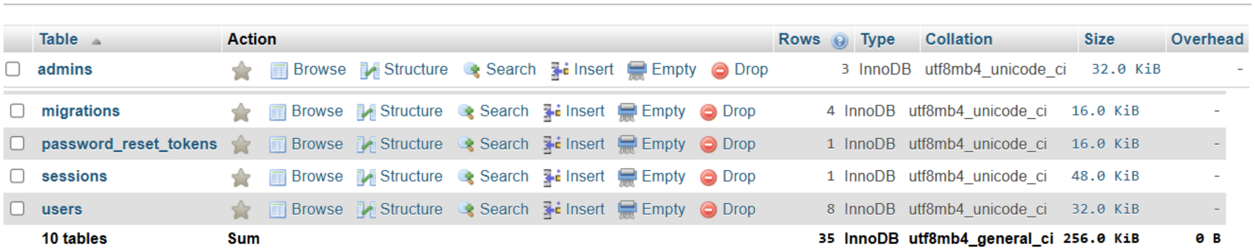


	Table ▲	Action	Rows	Type	Collation	Size	Overhead
□	admins	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
□	migrations	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
□	password_reset_tokens	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
□	sessions	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
□	users	★ Browse Structure Search Insert Empty Drop	8	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
	10 tables	**Sum**	**35**	**InnoDB**	**utf8mb4_general_ci**	**256.0 KiB**	**0 B**