



**BICOL UNIVERSITY**  
**COLLEGE OF SCIENCE**

Legaspi City



# Populating Pages

## Laboratory Activity


RICHARD D. BILAN JR.

BSIT 3-C

- Part 1: Create Views using Blades templates
  - Create [controllers](#) that load the dashboard, feed, or equivalent pages.
  - Register [controllers](#) in [routes](#) to link methods to URLs.
  - You may create and register more [controllers](#) if needed.
- Part 2: Create [controllers](#) that load the content for each page (Simulate access to a DB).
  - Load information such as posts, ratings, comments, pictures, etc.
  - The information must be in the form of an array.
- Part 3: Documentation (Individual)
  - Take screenshots of rendered pages and their accompanying code.
  - Write brief explanations of controller logic, parameter handling, and route assignments.
  - Cite other things you learned while making this.

Create a git release with the following name format: SystemName\_Lab#

In this activity, we do a multiplying pages base on user input called "**Populating Pages**" and we relate it to our chosen Final project "**Financial Tracker**", we implemented a dashboard that dynamically generates content based on user input. It allows users to specify a range of months, and the system displays posts related to those months.

 Dashboard

chad ▾

Dashboard

MONTH TRACKER

Start Month:  End Month:

March

Month: March

This is the month: March

Continue

April

Month: April

This is the month: April

Continue

May

Month: May

This is the month: May

Continue

This is How it works. The user must select where to start and end the Month of he/she wants to track.

The selection was set by numbers 1-12 instead of month January to December. Also to read the number of POST.

## HOW WE DO IT?

### First Controller Logic.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

2 references | 0 implementations
class populatingpagesController extends Controller
{
    1 reference | 0 overrides
    public function index(Request $request): Factory|View
    {
        $startMonth = $request->input(key: 'start_month');
        $endMonth = $request->input(key: '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) {
```

We created a controller named **populatingpagesController** with a method `index()`. The method processes two query parameters `start_month` and `end_month`. It validates the input to ensure the months are between 1 and 12. If invalid values are provided, it defaults to the full range (January to December). If the `start_month` is greater than the `end_month`, the values are swapped to ensure proper ordering.

```
        if ($startMonth && $endMonth) {
            if (!is_numeric(value: $startMonth) || $startMonth < 1 || $startMonth > 12) {
                $startMonth = 1;
            }

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

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

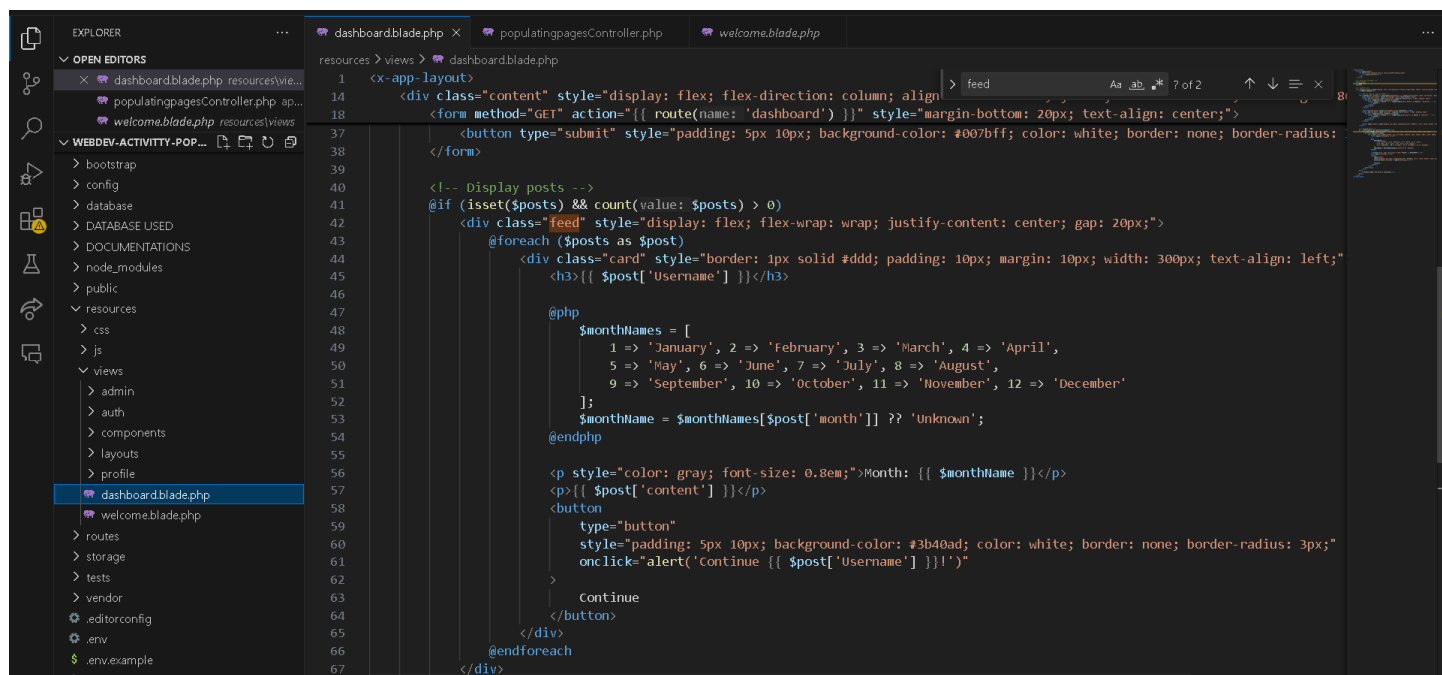
            for ($month = $startMonth; $month <= $endMonth; $month++) {
                $posts[] = [
                    'Username' => $monthNames[$month],
                    'month' => $month,
                    'content' => "This is the month: " . $monthNames[$month],
                ];
            }
      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

        return view(view: 'dashboard', data: compact(var_name: 'posts', var_names: 'startMonth', 'endMonth'));
    }
}
```

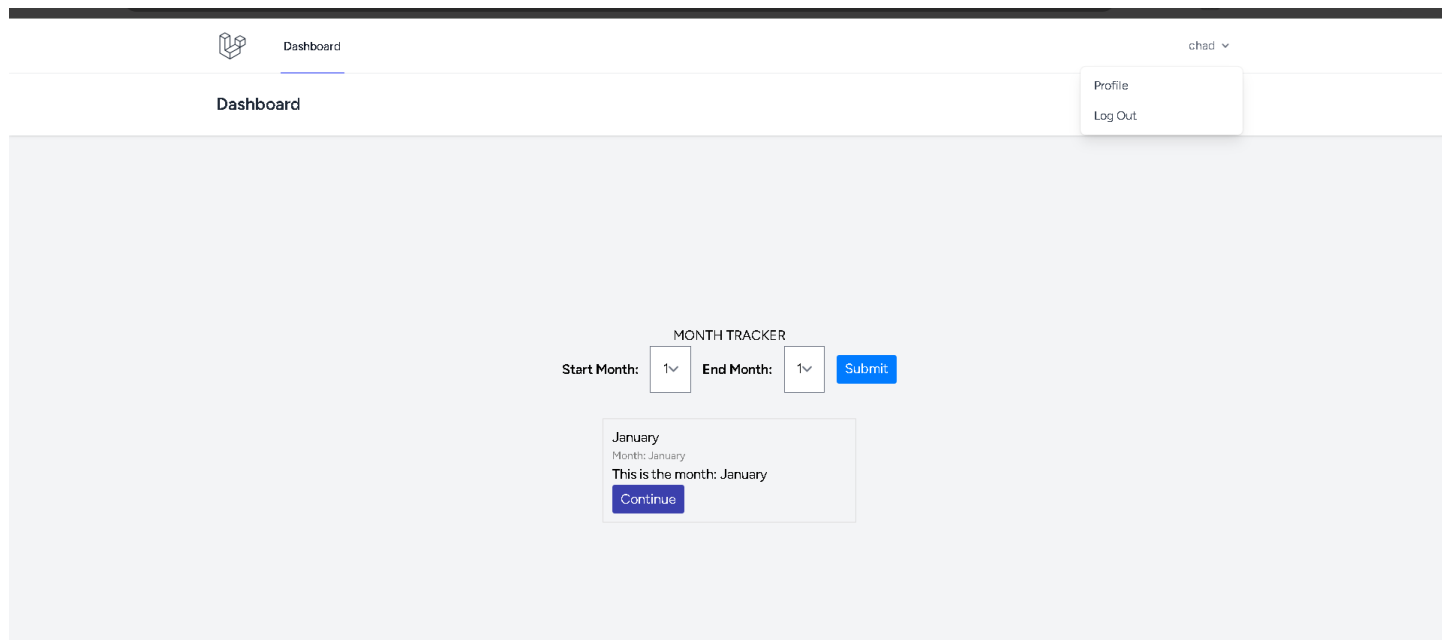
Using a for loop, I generated an array of posts for the selected range of months, with each post containing

- Username:** The name of the month.
- month:** The numeric value of the month.
- content:** A description related to the month.

## Second, Blade View



```
1 <x-app-layout>
14 <div class="content" style="display: flex; flex-direction: column; align-items: center; padding: 20px;>
18 <form method="GET" action="{{ route(name: 'dashboard') }}" style="margin-bottom: 20px; text-align: center;">
37 <button type="submit" style="padding: 5px 10px; background-color: #007bff; color: white; border: none; border-radius: 5px;">Continue</button>
38 </form>
39
40 <!-- Display posts -->
41 @if (isset($posts) && count($posts) > 0)
42 <div class="feed" style="display: flex; flex-wrap: wrap; justify-content: center; gap: 20px;">
43 @foreach ($posts as $post)
44 <div class="card" style="border: 1px solid #ddd; padding: 10px; margin: 10px; width: 300px; text-align: left;">
45 <h3>{{ $post['Username'] }}</h3>
46
47 @php
48 $monthNames = [
49 1 => 'January', 2 => 'February', 3 => 'March', 4 => 'April',
50 5 => 'May', 6 => 'June', 7 => 'July', 8 => 'August',
51 9 => 'September', 10 => 'October', 11 => 'November', 12 => 'December'
52 ];
53 $monthName = $monthNames[$post['month']] ?? 'Unknown';
54 @endphp
55
56 <p style="color: gray; font-size: 0.8em;">Month: {{ $monthName }}</p>
57 <p>{{ $post['content'] }}</p>
58 <button
59 type="button"
60 style="padding: 5px 10px; background-color: #3b40ad; color: white; border: none; border-radius: 3px;"
61 onclick="alert('Continue {{ $post['Username'] }}!')"
62 >Continue
63 </button>
64 </div>
65 @endforeach
66 </div>
67 </x-app-layout>
```



This is the code and the output it self.

It includes a form with dropdowns for selecting the start and end months. Each dropdown is dynamically populated with values from 1 to 12.

The view displays the generated posts in a card format

- The month's name.
- A brief description of what month is this
- A "Continue" button with an interactive message.

### Third, **Routing**:

```
// Update to point to the correct method
Route::get(uri: '/dashboard', action: [populatingpagesController::class, 'index'])->middleware(middleware: ['auth', 'verified'])->name(name: 'dashboard');
```

A route was added in web.php to map /dashboard to the index method of populatingpagesController.

This route is protected by auth and verified middleware to ensure only authenticated and verified users can access it.

```
use App\Http\Controllers\populatingpagesController;
use App\Http\Controllers\ProfileController;
use Illuminate\Support\Facades\Route;
```

This is the class to call the controllers

Profile

Log Out



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

New Password

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

## USER LOGIN



Email

Password

☐ Remember me

[Forgot your password?](#)

LOG IN

## USER REGISTER



Name

Email

Password

Confirm Password

[Already registered?](#)

REGISTER



## ADMIN REGISTER



### Admin Register Page

Name

Email

Password

Confirm Password

[Already registered?](#)

**REGISTER**

## ADMIN LOGIN



### Admin Login Page

Email

Password

☐ Remember me

[Forgot your password?](#)

**LOGIN**

DATABASE

financialdb

New

admins

cache

cache\_locks

failed\_jobs

jobs

job\_batches

migrations

password\_reset\_tokens

sessions

users

financialtracker

information\_schema

lab1

mysql

performance\_schema

Table	Action	Rows	Type	Collation	Size	Overhead
admins		2	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
cache		12	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
cache_locks		0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
failed_jobs		0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
jobs		0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
job_batches		0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
migrations		4	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
password_reset_tokens		1	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
sessions		1	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
users		6	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
10 tables	Sum	26	InnoDB	utf8mb4_general_ci	256.0 KiB	0 B

☐ Check all

With selected:

ADMINS

Extra options

Edit

Copy

Delete

1

richard

chard@laravel.com

NULL

\$2y\$12\$gt44C9LjVNRKSLNIG1cKge4bFTSxogHnMurA5jMOJf8...

NULL

2024-11-29 08:01:40

2024-11-29 08:01:40

Edit

Copy

Delete

4

admin chad

admin@laravel.com

NULL

\$2y\$12\$WRZGyWwChK2G7QkgR01m5Oa9n.AasTQbbeGuT7mbXwE...

NULL

2024-12-01 00:39:37

2024-12-01 00:39:37

☐ Check all

With selected:

Edit

Copy

Delete

Export

☐ Show all

Number of rows: 25

Filter rows: Search this table

Sort by key: None

USER

Edit

Copy

Delete

1

Richard

chardzbilan@gmail.com

NULL

\$2y\$12\$P3lL8RDvO887sturmc6GeODks5 p/Ojp944QLLvelliY...

NULL

2024-11-29 04:51:46

2024-11-29 04:51:46

Edit

Copy

Delete

2

Richard

richardbilan2@gmail.com

NULL

\$2y\$12\$5Dgh/pr7sDBedePMfvgvju7ECAI68Yteb2/sXX1zwQ...

NULL

2024-11-29 08:05:57

2024-11-29 08:05:57

Edit

Copy

Delete

3

admin

richard@admin.com

NULL

\$2y\$12\$Vz5l5hGmM2rWQAZfKfNEa.O5gMb11G2OHDYMQHngi3z...

NULL

2024-11-29 08:08:14

2024-11-29 08:08:14

Edit

Copy

Delete

4

juan

user@mail.com

NULL

\$2y\$12\$BD5X8NoLaBsvXs41A5i1zOCV56bDHzwi.SLI2oOSauf...

NULL

2024-11-29 08:43:50

2024-11-29 08:43:50

Edit

Copy

Delete

5

chad

user@laravel.com

NULL

\$2y\$12\$Qr85OIXPNIVISKEQY3QZHe/nmIONpC.2bbFukbnHOKV...

NULL

2024-12-01 00:28:48

2024-12-01 00:28:48

Edit

Copy

Delete

6

juan

juan@laravel.ccom

NULL

\$2y\$12\$QRV3bTzuUjzGp2WYgkZ6teEstMxahHiU9kkc1m8HFz...

NULL

2024-12-02 03:43:39

2024-12-02 03:43:39

☐ Check all

With selected:

Edit

Copy

Delete

Export

# ADMIN DASHBOARD

Financial Tracker

CORE

Dashboard

Financial Tracker

personal budget

INTERFACE

Analytics

INTERFACE

Transaction management

Product

>

Posts

>

ADDONS

Charts

Tables

Settings

Activity Log

Log Out

Dashboard

Dashboard

127.0.0.1:8000/admin/dashboard#