LABORATORY on Populating Pages.

DOCUMENTING THE ROUTES:

```
Route::get('/home', [HomeController::class, 'home'])->name('homepage');
```

1. When a user visits **/home**, the home method in **HomeController** is invoked. This method prepares and sends data like username and events to the homepage Blade view.

```
Route::get('/user/form', [HomeController::class, 'form'])->name('form');
```

2. Directs users to a form page. The form method loads the form Blade view, which likely contains an HTML form for user to create an event.

```
Route::get('/', function () {
   return view('user_role');
})->name('role');
```

3. Displays the user_role view, that allow users to choose their role (admin or organizer).

```
Route::post('/user/organizer', [RoleController::class, 'organizerLogin'])->name('login.organizer');
Route::post('/user/admin', [RoleController::class, 'adminLogin'])->name('login.admin');
```

4. Manages login attempts for organizers. The controller validates the credentials given in the request (username and password). Upon success, the user is sent to /home with their username saved during the session, if not they are sent to an error page. The adminLogin Similar to the organizer route, it validates credentials like if the password is correct.

DOCUMENTING THE CONTROLLERS

HomeController

- **home** Method:
    - Fetches username from the session or set its defaults to "Guest".
    - Prepares a list of dummy event data with titles, descriptions, and image paths.
    - Returns the homepage view, passing the username and events data to it.
- **form Method:**
    - Returns the form view for user to customize their event.

RoleController

- **organizerLogin** Method:
    - Retrieves username and password from the POST request.
    - Verifies the password (**organizer123**).
    - Redirects to the homepage with username stored in the session if successful, otherwise, to the error page.
- **adminLogin Method:**
    - Similar to organizerLogin, but the password check is for admin.

Part 1: Create Views using Blades templates

- ✓ Create controllers that load the dashboard, feed, or equivalent pages.
  HomeController: Responsible for rendering the homepage and the form page.
- ✓ Register controllers in routes to link methods to URLs.

```
Route::get('/home', [HomeController::class, 'home'])->name('homepage');
```

- ✓ You may create and register more controllers if needed.
  In our case we did not create additional Controller for loading the page since its more efficient to put it on the **HomeController** with the **home** method for the mean time.

Part 2: Create controllers that load the content for each page (Simulate access to a DB).

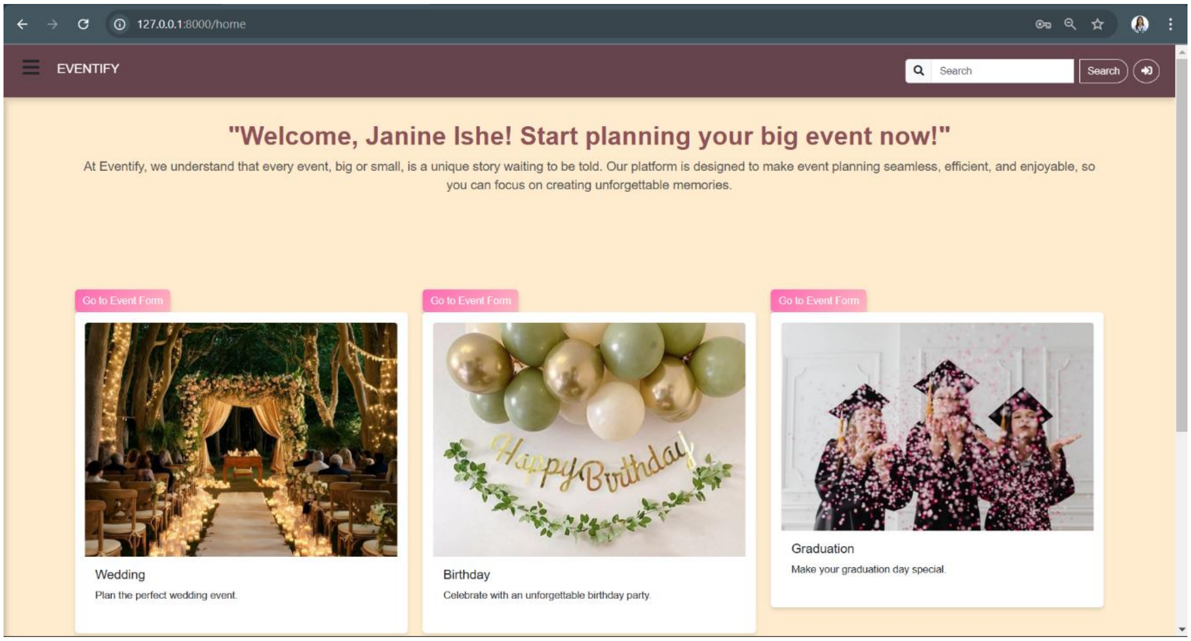
- ✓ Load information such as posts, ratings, comments, pictures, etc.
  For the **homepage**, **event** data was created as an array in the **HomeController@home** method:

```
// Dummy data for events
    $events = [
        [
            'title' => 'Wedding',
            'description' => 'Plan the perfect wedding event.',
            'image' => asset('assets/images/wedding.jpg')
        ],
        [
            'title' => 'Birthday',
            'description' => 'Celebrate with an unforgettable birthday party.',
            'image' => asset('assets/images/baptism.jpg')
        ],
        [
            'title' => 'Graduation',
            'description' => 'Make your graduation day special.',
            'image' => asset('assets/images/birthday.jpg')
        ],

    ];
```

```
<div class="content-container" id="mainContent">
    <div class="row">
        @foreach ($events as $event)
            <div class="col-md-4 col-sm-6 mb-4">
                <a href="{{ route('form') }}" class="btn btn-primary">Go to Event Form</a>
                <div class="card p-3">
                    <img src="{{ $event['image'] }}" alt="{{ $event['title'] }}" class="card-img-top img-
fluid">
                    <div class="card-body">
                        <h5 class="card-title">{{ $event['title'] }}</h5>
                        <p>{{ $event['description'] }}</p>
                    </div>
                </div>
            </div>
        @endforeach
    </div>
</div>
```

OUTPUT



"Welcome, Janine Ishe! Start planning your big event now!"

At Eventify, we understand that every event, big or small, is a unique story waiting to be told. Our platform is designed to make event planning seamless, efficient, and enjoyable, so you can focus on creating unforgettable memories.

Wedding

Plan the perfect wedding event.

Birthday

Celebrate with an unforgettable birthday party.

Graduation

Make your graduation day special.

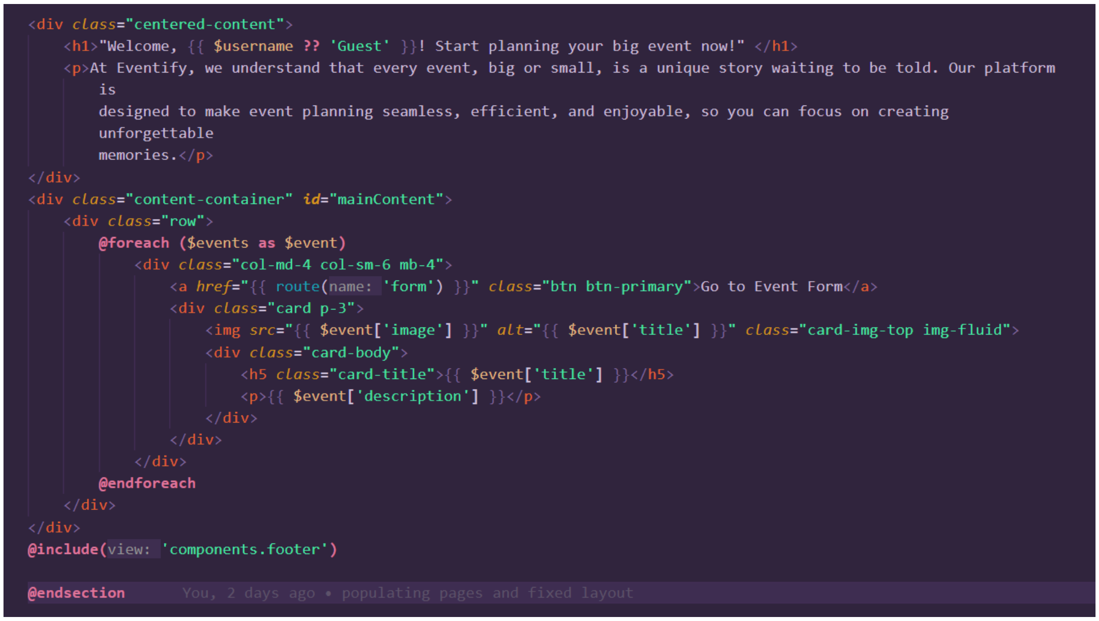
- ✓ The information must be in the form of an array.

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.

Code Reference:

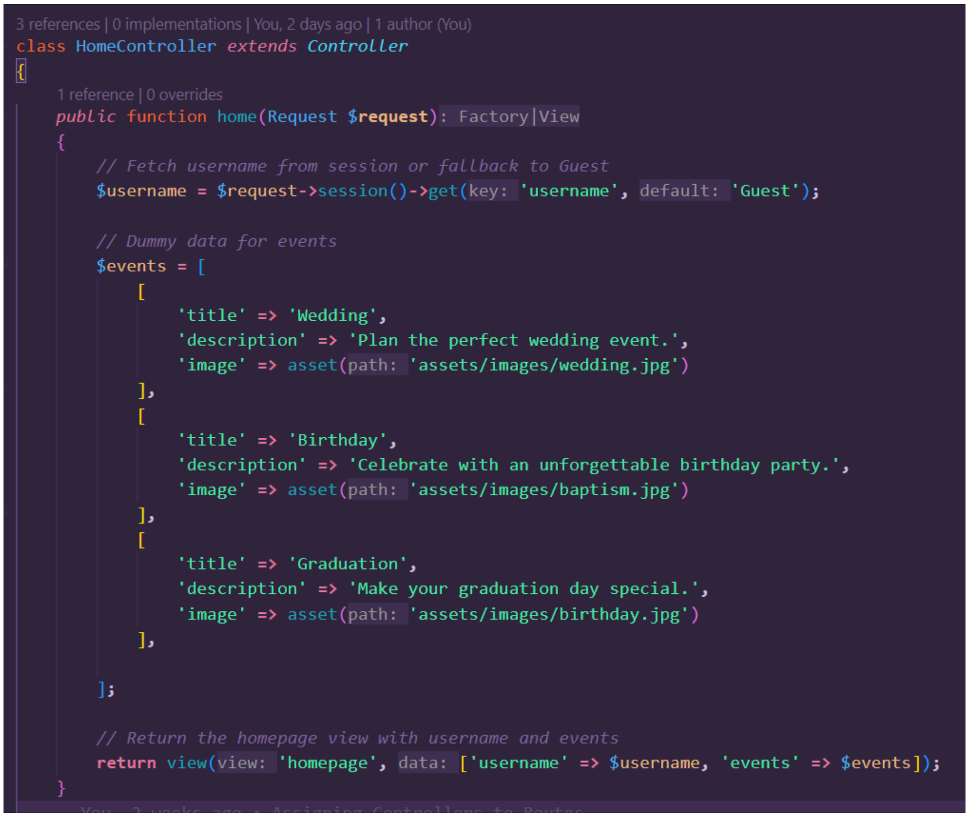
View (homepage.blade.php): Renders events using a foreach loop.



```
<div class="centered-content">
    <h1>"Welcome, {{ $username ?? 'Guest' }}! Start planning your big event now!" </h1>
    <p>At Eventify, we understand that every event, big or small, is a unique story waiting to be told. Our platform
        is
        designed to make event planning seamless, efficient, and enjoyable, so you can focus on creating
        unforgettable
        memories.</p>
</div>
<div class="content-container" id="mainContent">
    <div class="row">
        @foreach ($events as $event)
            <div class="col-md-4 col-sm-6 mb-4">
                <a href="{{ route(name: 'form') }}" class="btn btn-primary">Go to Event Form</a>
                <div class="card p-3">
                    <img src="{{ $event['image'] }}" alt="{{ $event['title'] }}" class="card-img-top img-fluid">
                    <div class="card-body">
                        <h5 class="card-title">{{ $event['title'] }}</h5>
                        <p>{{ $event['description'] }}</p>
                    </div>
                </div>
            </div>
        @endforeach
    </div>
</div>
@include(view: 'components.footer')

@endsection      You, 2 days ago • populating pages and fixed layout
```

Controller (HomeController@home): Passes username and events to the view.



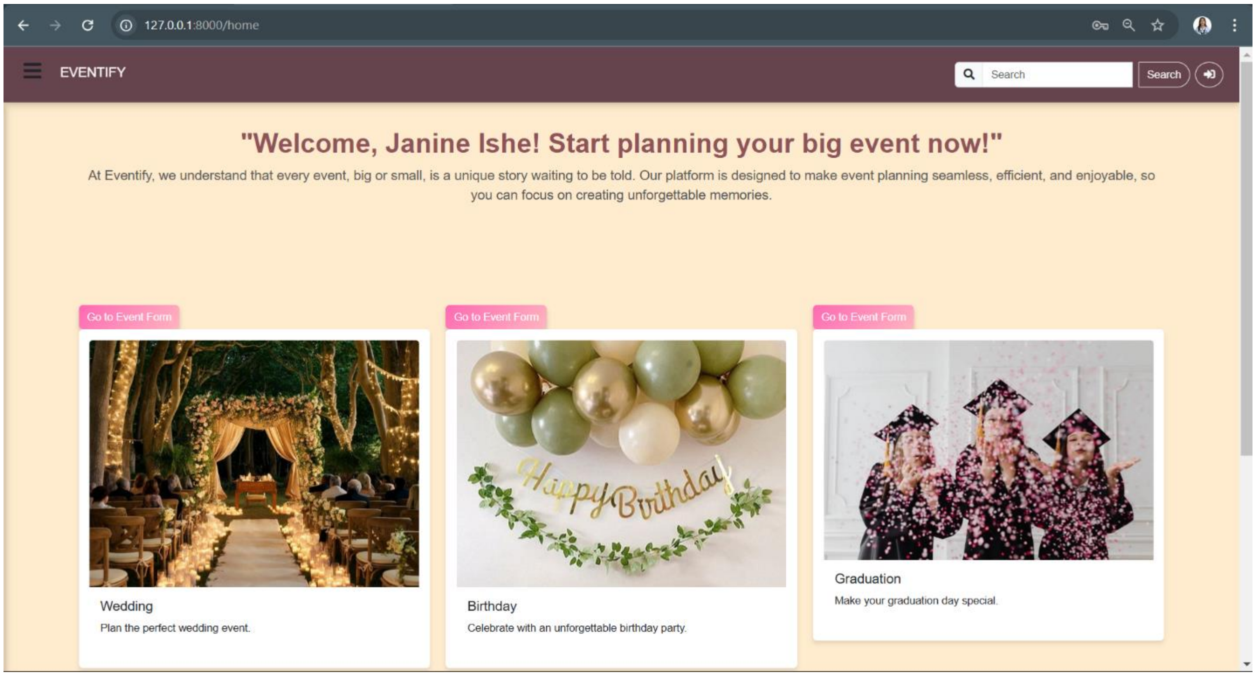
```
3 references | 0 implementations | You, 2 days ago | 1 author (You)
class HomeController extends Controller
{
    1 reference | 0 overrides
    public function home(Request $request): Factory|View
    {
        // Fetch username from session or fallback to Guest
        $username = $request->session()->get(key: 'username', default: 'Guest');

        // Dummy data for events
        $events = [
            [
                'title' => 'Wedding',
                'description' => 'Plan the perfect wedding event.',
                'image' => asset(path: 'assets/images/wedding.jpg')
            ],
            [
                'title' => 'Birthday',
                'description' => 'Celebrate with an unforgettable birthday party.',
                'image' => asset(path: 'assets/images/baptism.jpg')
            ],
            [
                'title' => 'Graduation',
                'description' => 'Make your graduation day special.',
                'image' => asset(path: 'assets/images/birthday.jpg')
            ],

        ];

        // Return the homepage view with username and events
        return view(view: 'homepage', data: ['username' => $username, 'events' => $events]);
    }
        You, 2 weeks ago • Assigning Controllers to Routes
```

OUTPUT



"Welcome, Janine Ishe! Start planning your big event now!"

At Eventify, we understand that every event, big or small, is a unique story waiting to be told. Our platform is designed to make event planning seamless, efficient, and enjoyable, so you can focus on creating unforgettable memories.

Wedding

Plan the perfect wedding event.

Birthday

Celebrate with an unforgettable birthday party.

Graduation

Make your graduation day special.

THINGS LEARNED:
Blade Templates let us create dynamic content on our website, like event cards. It's like using a simple language to tell the computer what to show, based on the information we provide. For example, we can say "show an event card for each event in this list," and Blade will automatically generate the correct HTML code.

Prepared by:

JANINE ISHE M. MATAMOROSA
BSIT 3C