Add CoreUI to an existing Laravel project

To add CoreUI to an existing Laravel project, you can follow these steps. CoreUI provides a variety of options, including a full admin template as well as individual components you can use. Here’s how to integrate it into your Laravel setup:

Step 1: Install CoreUI via NPM

CoreUI offers the ability to be installed via NPM. Run the following command:

```
npm install @coreui/coreui
```

Step 2: Integrate CoreUI Assets

Now that you have CoreUI installed, you need to import its CSS and JavaScript into your Laravel project.

1. Open the `resources/js/app.js` file (or wherever your JavaScript is initialized) and add:

   ```
   import '@coreui/coreui/dist/css/coreui.min.css';
   import '@coreui/coreui/dist/js/coreui.bundle.min.js';
   ```

2. Open the `resources/css/app.css` (or your SCSS file) and add:

   ```
   @import "~@coreui/coreui/dist/css/coreui.min.css";
   ```

Step 3: Compile Assets

Make sure your Laravel Mix is configured properly. Typically, this will be in the `webpack.mix.js` file.

Example:

```
mix.js('resources/js/app.js', 'public/js')
   .sass('resources/sass/app.scss', 'public/css')
```

```
    .version();
```

Compile the assets:

```
npm run dev
```

Step 4: Add CoreUI Layout to Laravel Views

Create a Blade layout file that will use CoreUI’s structure. You can make this in `resources/views/layouts/app.blade.php`.

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>{{ config('app.name', 'Laravel') }}</title>
    <link href="{{ mix('css/app.css') }}" rel="stylesheet">
</head>
<body class="c-app">
    <div class="c-wrapper">
        @include('layouts.sidebar')
        <div class="c-body">
            @include('layouts.header')
            <main class="c-main">
                <div class="container-fluid">
                    @yield('content')
                </div>
            </main>
            @include('layouts.footer')
        </div>
    </div>
    <script src="{{ mix('js/app.js') }}"></script>
</body>
</html>
```

Step 5: Create Sidebar, Header, and Footer

You can structure your CoreUI components as separate partials, like `resources/views/layouts/sidebar.blade.php`, `header.blade.php`, and `footer.blade.php`.

Example Sidebar:

```
<nav class="c-sidebar c-sidebar-dark c-sidebar-fixed" id="sidebar">
    <ul class="c-sidebar-nav">
        <li class="c-sidebar-nav-item">
            <a class="c-sidebar-nav-link" href="{{ route('home') }}">
                <i class="c-icon cil-speedometer"></i> Dashboard
            </a>
        </li>
        <!-- Add more sidebar items as needed -->
    </ul>
</nav>
```

Example Header:

```
<header class="c-header c-header-light c-header-fixed">
    <button class="c-header-toggler c-class-toggler d-lg-none mfe-auto" type="button">
        <i class="c-icon c-icon-lg cil-menu"></i>
    </button>
    <a class="c-header-brand d-lg-none" href="#">CoreUI</a>
</header>
```

Step 6: Use the Layout in Your Views

Make your Laravel views use this new CoreUI layout by extending the layout:

```
@extends('layouts.app')

@section('content')
    <h1>Welcome to CoreUI Dashboard!</h1>
    <!-- Your content goes here -->
@endsection
```

Step 7: Customize CoreUI Components

Customize CoreUI components and styles as needed. You can override the default CSS by adding your own styles or modifying the existing ones.

Step 8: Recompile Assets

After making any CSS or JavaScript changes, remember to recompile:

```
npm run dev
```

Conclusion

You have now integrated CoreUI into your existing Laravel project. CoreUI brings in a lot of reusable components, making it easy to set up an admin dashboard or application interface. You can customize the layout, navigation, and pages to suit your requirements.

Revision #1
Created 18 October 2024 04:51:29 by AKSHAY D JOSHI
Updated 18 October 2024 04:51:57 by AKSHAY D JOSHI