1. Laravel(update version) Project Creation

```
laravel new my-app
```

Inertia - Server Side Install

(a) Inertia composer installation

```
composer require inertiajs/inertia-laravel
```

(b) Configure Blade Template Engine - Prepare Master Blade Layout
`/views/app.blade.php`

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0" />
    @vite('resources/js/app.js')
    @inertiaHead
</head>
<body>
@inertia
</body>
</html>
```

(c) Create Inertia Middleware

```
php artisan inertia:middleware
```

(d) Add Inertia Middleware With Your Route Middleware
`bootstrap/app.php`

```
use App\Http\Middleware\HandleInertiaRequests;

$middleware->web(append: [
  HandleInertiaRequests::class,
]);
```

Inertia - Client Side Install

(a) Install Vue JS

```
npm install @inertiajs/vue3
```

Import `resources/js/app.js`

```
import { router } from '@inertiajs/vue3'
```

(b) configure vite

```
npm i @vitejs/plugin-vue
```

Import `vite.config.js`

```
import { defineConfig } from "vite";
import vue from "@vitejs/plugin-vue";
import laravel from "laravel-vite-plugin";
// import tailwindcss from "@tailwindcss/vite";

export default defineConfig({
    plugins: [
        laravel({
            input: ["resources/css/app.css", "resources/js/app.js"],
            refresh: true,
        }),
        vue(),
        // tailwindcss(),
    ],
    build: {
        manifest: true,
        outDir: "public/build",
    },
});
```

update

```
plugins: [
    vue(),
],
build: {
    manifest: true,
```

```
    outDir: "public/build",
},
```

(c) configure app.js

```
npm i vue
```

Import `resources/js/app.js`

```
import { createApp, h } from 'vue'
import { createInertiaApp } from '@inertiajs/vue3'
createInertiaApp({
    resolve: name => {
      const pages = import.meta.glob('./Pages/**/*.vue', { eager: true })
      return pages[`./Pages/${name}.vue`]
    },
    setup({ el, App, props, plugin }) {
      createApp({ render: () => h(App, props) })
        .use(plugin)
        .mount(el)
    },
})
```

---

4. Vue js frontend structure inside `resources/js/`

- Components (Naming Convention Optional)
- Utility (Naming Convention Optional)
- Pages (Naming Convention Mandatory)

5. Lets create our first website

- Home Page `[AppNavbar+Hero+Footer]`
- Profile page `[AppNavbar+ProfileForm+Footer]`
- Lets create a controller SiteController
- Lets create function for Render Vue.js Pages
- Lets define route in web.php

6. Let's Run Our `Laravel+Vue` Project

- php artisan serve
- npm run dev
- Access From Running Via Vite
- Access From Running Via Artisan

7. How Rendering Works Here

- `"/" Route-->Middleware-->Controller-->Render--->HomePage.vue`
- `"/profile" Route-->Middleware-->Controller-->Render--->ProfilePage.vue`

8. Set Our First SPA Navigation

- Use Inertia Build In Link component

```
<script setup>
    import { Link } from '@inertiajs/vue3';
</script>

<template>
    <h1 class="text-danger">This is Test page</h1>
    <Link href="/">Home</Link>
</template>
```

9. How to set Routing Progress Like vue JS

```
npm i nprogress
```

- Add CSS CDN
- `resources/view/app.blade.php`

```
<link rel="stylesheet" href="
{{url('https://cdnjs.cloudflare.com/ajax/libs/nprogress/0.2.0/nprogress.min.
css')}}" />
```

- Add Route Start End of `@app.js`

Added `resources/js/app.js`

```
import NProgress from 'nprogress'

router.on('start', () => {
```

```
    NProgress.start()
})

router.on('finish', () =>{
    NProgress.done()
})
```

Vue.js Packages Installation Guide

---

1. Toast Notifications: `@meforma/vue-toaster`

```
npm i @meforma/vue-toaster
```

A lightweight toast notification library for Vue 3.

2. Rich Text Editor: `@vueup/vue-quill`

```
npm i @vueup/vue-quill
```

Vue 3 wrapper for the Quill rich-text editor to add WYSIWYG functionality.

3. Data Table: `vue3-easy-data-table`

```
npm i vue3-easy-data-table
```

Import `resources/js/app.js`

```
import Vue3EasyDataTable from 'vue3-easy-data-table';
import 'vue3-easy-data-table/dist/style.css';

setup({ el, App, props, plugin }) {
    const app = createApp({ render: () => h(App, props) })
    app.use(plugin)
    app.component('EasyDataTable', Vue3EasyDataTable)
    app.mount(el)
},
```

Simple and easy-to-use data table component for Vue 3 applications.

4. Chart Libraries: `chart.js`, `vue-chartjs`, `apexcharts`

```
npm i chart.js
```


```
npm i chart.js
```


```
npm i apexcharts
```

- **chart.js**: JavaScript library for creating interactive charts.
- **vue-chartjs**: Vue wrapper for Chart.js.
- **apexcharts**: Modern charting library for data visualization.

5. Skeleton Screens: `vue-content-loader`

```
npm i vue-content-loader
```

Easily create skeleton screens for Vue 3 applications.

6. Circular Progress Bar: `vue3-circle-progress`

```
npm i vue3-circle-progress
```

A customizable circular progress bar component for Vue 3.

7. Lottie Animations: `vue3-lottie`

```
npm i vue3-lottie
```

A Vue 3 component for rendering Lottie animations.

8. CSS Framework: `bootstrap`

```
npm i bootstrap
```

Popular CSS framework for building responsive and mobile-first websites.