Vue.Js Setup Part 4 :

**++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**links :**[**https://www.youtube.com/watch?v=MQZ4PQGw-sc&list=PL8p2I9GklV45qwTH-mdzllUuFRJO-euYn&index=28**](https://www.youtube.com/watch?v=MQZ4PQGw-sc&list=PL8p2I9GklV45qwTH-mdzllUuFRJO-euYn&index=28)

**page not found | 404 page | Vue router part 4 >>>>>**

create one pageNotfound component and set url in main.js file if not matched url then open 404 page ..

**Main.Js >>>**

// Import Vue and App component

import { createApp } from 'vue';

import App from './App.vue';

// Import Bootstrap and BootstrapVue 3 CSS files

import 'bootstrap/dist/css/bootstrap.css';

import 'bootstrap-vue-3/dist/bootstrap-vue-3.css';

// Import BootstrapVue 3

import BootstrapVue3 from 'bootstrap-vue-3';

// Import Vue Router and necessary methods from vue-router

import { createRouter, createWebHistory } from 'vue-router';

// Import the components for routing

import Home from './components/Home.vue';

import About from './components/About.vue';

import Login from './components/Login.vue';

import User from './components/User.vue';

**import PageNotFound from './components/PageNotFound.vue';**

// Define the routes

const routes = [

{ path: '/', component: Home },

{ path: '/about', component: About },

{ path: '/login', component: Login },

{ path: '/user/:id', component: User },

// { path: '\*', component: PageNotFound }, // not used its use in vue 2 not this \* is not exits

**{ path: '/:pathMatch(.\*)\*', component: PageNotFound }, // Catch-all for 404 Page Not Found**

];

// Create the router instance

const router = createRouter({

history: createWebHistory(),

routes,

});

// Create the Vue app instance

const app = createApp(App);

// Use BootstrapVue 3 globally

app.use(BootstrapVue3);

// Use Vue Router globally

app.use(router);

// Define the global "filter-like" function (for example: capitalize)

app.config.globalProperties.$filters = {

capitalize(val) {

if (!val) return '';

return val.toString().toUpperCase();

},

};

// Mount the app to the #app element

app.mount('#app');

**PageNotFound.vue >>>>**

<template>

<div>

<h2>Page Not Found</h2>

<h3>Go to home page</h3>

<router-link to="/">Home</router-link>

</div>

</template>

<script>

export default {

name:"PageNotFound",

}

</script>

**now if you test with wrong or not exist url then send 404 page ..**

**++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**dynamic component >>>>>**

**SignIn.vue >>>**

<template>

<div>

<h3>SignIn Component</h3>

</div>

</template>

<script>

export default {

name:"SingIn",

}

</script>

**SignUp.vue >>>**

<template>

<div>

<h3>SignUp Component</h3>

</div>

</template>

<script>

export default {

name:"SingUp",

}

</script>

**App.vue >>>>**

**while load dynamic component then i create 2 button when you click then component is loaded.**

<template>

<div id="app">

<h2>Dynamic Component</h2>

<!-- when i am use dynamic component then not use this other way in next lines -->

<!-- <signin />

<signup /> -->

**<component v-bind:is="cmp"></component>**

**<button @click="updatecmp('signin')">Load SignIn</button>**

**<button @click="updatecmp('signup')">Load SignUp</button>**

</div>

</template>

<script>

**import SignIn from './components/SignIn.vue';**

**import SignUp from './components/SignUp.vue';**

**export default {**

**name: 'App',**

**components: {**

**'signin':SignIn,**

**'signup':SignUp,**

**},**

**data(){**

**return {cmp:null}**

**},**

**methods:{**

**updatecmp(item){**

**this.cmp = item;**

**}**

**}**

**};**

</script>

<style>

#app {

font-family: Avenir, Helvetica, Arial, sans-serif;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

text-align: center;

color: #2c3e50;

margin-top: 60px;

}

</style>

by default no component load when i click on load signin then load signin page & same singup.

default data function need to load & after update component then its works .

**+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**make custom directive >>>>**