**Microfront end using vue.js**

**Microfront End:**

Micro frontends is an architectural style of frontend web development where an application is split into features – micro frontends – and delivered independently.

Implementing a microfrontend architecture with Vue.js involves breaking down a large frontend application into smaller, independently developed, and deployable pieces (microfrontends).

**Choose one of these approaches for integrating microfrontends:**

**Module Federation (Webpack 5):** Share and load apps/modules dynamically.

**Single-SPA Framework:** A popular framework for microfrontend orchestration.

**Here is the step to implement using Module Federation(Webpack 5)**

**Step1:**

Create a vue project using the command

**Npx create-mf-app**

In this it will ask about

Name of your app? (Any Like eg:**HOST**)

Project Type? (Choose **Application**)

Port number? (Choose different port for each project)

Framework? (Choose **Vue3**)

Language ?

Css?

Note: **You need minimum of two application to build a microfrontend.**

**Select different port for each project  
 Run the create command again to create a 2nd vue project (eg:REMOTE)**

**Step2:**

Go to webpack.config.js in remote project (2nd project). There under plugins you have file name as       "**remoteEntry**.**js**".Copy the file name we need to paste it in **Host** project.

This is the webpack file that has a javascript code.we need to run this js file along with **Remote** application running localhost and paste that in **Host** application.

Step 3:

Then go to webpack.config.js in **HOST** Application and paste the link like this

 remotes: {

        "remote": "remote@http://localhost:8081/remoteEntry.js",

      },

**Remote** is the name that should match with 2nd project plugin name.

localhost:8081-Remote project running port.

**Step3:**

Now we can access the components in **Remote** Application From **Host** Application. But Both applications are independent.

<script>

import Header from "remote/Header"

import { RouterView } from "vue-router";

export default {

  components: {

    Header,

  }

}

</script>

Like this we can acces from remote application and use the component like

<template>

  <Header/>--------------The Header component from Remote application will displayed here.

</template>

**Step 4:(Router setup)**

We can set the path for components and route into that components.

We need to install the router package first.

**Yarn add vue-router@4**

By this command we can add router to vue application .1st ensure you have yarn installed in your system.

After adding the router import the router in bootloader.js in **HOST** application.

Like this

**import \* as vuerouter from "vue-router";**

**Then set the path for the components**

const routes = [

  { path: "/", component: Home },

  {path:"/about", component:About}

];

**Then initialize the router and use it before mounting the app**

const router = vuerouter.createRouter({

  routes,

  history:vuerouter.createWebHashHistory(),

});

createApp(App).use(router).mount("#app");

When we load the link like this

[**http://localhost:8080/about#/**](http://localhost:8080/about#/)

**based on the path it hit the component.**