1.搭建一个新的手脚架；

2.页面构建出一个大致布局（导航栏），一级路由，二级路由，三级路由

3.将登录、注册与导航栏相连。

4.进行数据交互：

安装vue-resource.

man.js中引入组件

import VueResource from ‘VueResource’

Vue.use(VueResource)

登录使用get请求，key关键词，value值

this.$http.get(‘https://www.easy-mock.com/mock/5c061eec8f1be27163bf6e22/vue/getData’,{params:{name:’kavion’,age:18(key:value)}})

.then(

res=>{console.log(res)},//请求成功回调函数

err=>{ console.log(err)}//请求失败回调函数

)

注册使用post请求

this.$http. post (‘https://www.easy-mock.com/mock/5c061eec8f1be27163bf6e22/vue/getData’,{{name:’kavion’,age:18(key:value)}})

.then(

res=>{console.log(res)},//请求成功回调函数

err=>{ console.log(err)}//请求失败回调函数

)

但之后不在进行维护了，所以使用axios。

安装vue-resource.

man.js中引入组件

import Axios from ‘axios’

Vue.prototype.$axios=Axios;// 把axios存到Vue的原型身上是为了在其它的组件里面都可以使用

操作与vue-resource相似只是改为this.$axios.post(‘url’,{ key:value })

5.在注册里面将注册数据使用axios发送POST请求到数据库中。

v-model数据双向绑定。

const formDate={username:this.username,password:this.password}

//把要传递的数据存储在一个对象中。作为数据交互的第二个字段

6.登录里面发送GET请求获取数据。

因为取到的数据是所有的用户名和密码。放在了res.data中；所以想把数据放在数组当 =>const keys=Object.keys(res.data);

通过for( let v of keys){}1.定义一个登录是否成功的状态let logined=false;2.进行遍历循环比较，用当前输入的用户、密码与所有users数据比较，判断登录是否成功。成功后用break跳出循环。

7.菜单模块中设定5个2级路由，并通过div等，用设定出死的数据进行展示。但死数据的显示要通过js显示。如：

<ul class="boxList">

        <li v-for="(item, index) in dishes" :key="index">

            <img :src="item.img" alt="">

            <div class="menuName">

                <div class="left">

                    <span>{{item.name}}</span>

                    <p>RMB<strong>{{item.price}}/份</strong></p>

                </div>

                <div class="right">

                    <a href="javascript:;" @click="order(item)"></a>

                </div>

            </div>

        </li>

    </ul>

8.在菜品、订单、管理页面的死数据样式修改后，在管理界面的添加模块进行真正数据交互操作。

9.通过post方法将添加请求发送到数据库。

10.在管理界面的删除模块操作：

a.先将菜单中的信息展示出来。通过created(){this.$axios.get请求数据} b.删除。需要找到删除的key.(会发现点击删除后页面没有变，但数据库里已经删查出来。这是后面需要解决的问题。)

11.将管理里面连接的菜单数据库域菜单模块进行操作，通过GET请求变成活的数据。

将获取到的数据（全部菜品）进行过滤分成不同种类的菜品，在不同分类展示。

12.将菜单中不同分类做成一个模板，并通过在子模板里使用props:[‘fathername’]传递。

13.点餐功能：

a.需要找到点击的是哪个菜？order(item);

b.将点餐数据发送到数据库中。

1.发现num为0.

2.当连续点击时发现出现新的多条数据，仅仅num不同。

于是在点击前判断该菜品是否点击过，如果点击过，则更新数据，如果没有点击过，需要增加数据。

通过get请求，拿到所有的预定菜信息，拿到所有的数据后，去判断当前点击的那个菜是否被点击过（通过状态判定）

但可能出现拿到的数据为空，即没有数据的情况。于是需要先进行判断拿到的数据不为空。

如要被击过，那就通过put请求更新，如果没有那就通过post请求增加

vuex:管理数据的。可以对数据进行集中管理操作。

const store = new Vuex.Store({

state: { //存储数据 },

getters:{//获取数据},

mutations: {//管理数据

increment (state) { state.count++ }

}

})

用到的东西：

props：可以在调用组件的时候传递数据

语法：

组件里 props:[‘属性名’,’属性名’…]（在components即定义组件那里进行进行定义）

模板里 {{属性名}}（在前面需要重复使用的模块里将要改的内容改为可变的属性名）

标签里 <组件名 属性名=’值’ 属性名=’值’ ></组件名>（直接在组件里添加修改属性名。<toheader name=’kuang’ ..></ toheader >）

<https://ke.qq.com/webcourse/index.html#cid=373724&term_id=100444911&taid=3041841868288988&vid=v14307mgz5h>

注意：

尽量使用局部组件。

new Vue({

components:{

tagName:options(组件名，ID里的名字，可以一样)

}

})

全局组件，局部组件

引入组件：在<script>import \*\*\* from ‘\*\*\*\*路径’

components:{ \*\*\*}

</script>

路由，一级路由，二级路由。。。（可以把一个页面当做一个组件）

路由重定向，beforecreated

全局守卫、后置钩子的是在man.js中配置

路由独享守卫是在router.js中的相应路由下进行配置，

const router = new VueRouter({ ... })

router.beforeEach((to, from, next) => {

// ...

})

组件内守卫是放在组件内的。