1.登录注册

用户进入网站，可以点击首页右上角的登录注册或者底部tabber“我的”，来到用户详情页面（用户登陆成功后右上角将不再显示）。

用户登录可以通过手机短信验证码，也可以账号密码登录，所有的短信验证码和图形验证码都是一次性动态的。用户登录我们的网站后可以修改个人信息和注销，网站是允许

没有登录的用户浏览的，如果想下单等其他操作，就必须要注册。

2.首页部分

用户首页要美观大方，色调以橙色为主，给人舒适感比较有食欲。首页部分主要由轮播导航，店铺列表和底部导航。

轮播分页导航使用了swiper插件，安装相应的依赖，组件中引入文件后直接调用。

this.$nextTick(() => {

// 一旦完成界面更新, 立即执行回调

new Swiper('.swiper-container', {

autoplay: true,

pagination: {

el: '.swiper-pagination',

clickable: true

}

})

})

}

商家列表页分装成ShopList组件，粗略地显示商家的头像，名称，评分等信息，点击店铺可以通过路由传参跳转到相应的详情页，可以供用户快速地选择自己看中的店铺下单

3.商家详情页

点击商家列表（ShopList）里某一项进入商家店铺的一级路由界面（Shop），商家店铺界面包括顶部的一个头部的一般组件(ShopHeader)和下面三个可以切换的路由子组件(ShopGoods、ShopInfo、ShopRatings)

商家店铺包括点餐，评价和商家信息三个部分的数据都是ajax请求后台动态生成的。点餐，评价和商家这三个部分，通过路由来切换到不同的组件。数据中不同分类的食品，用数组goods来存放，

它们没有直接顺序的关系，但都是同一类型。每一类的食品除了名称name，还有foods数据来存放这一类食品对象。用户的评价信息

统一放在一个大数组里，页面加载的时候可以直接遍历渲染到页面上，用户也可以选择性筛选评论来判断（好评，差评，有无评价）

点击详情页头部底部会弹出该店铺的优惠活动。食品列表需要完成用户交互功能，当用户点击左侧的食品分类列表时，右侧列表滑动到指定位置（导航的底部），同理，

点击右侧食品列表，左侧当前分类切换不同颜色。在滑动过程中，实时监听scrollY（右侧食品列表滑动的Y轴坐标），列表第一次显示后得到tops（所有右侧分类标题到屏幕顶部的距离）

准备计算属性：currentIndex，

currentIndex () { // 初始和相关数据发生了变化

// 得到条件数据

const {scrollY, tops} = this

// 根据条件计算产生一个结果

const index = tops.findIndex((top, index) => {

// scrollY>=当前top && scrollY<下一个top

return scrollY >= top && scrollY < tops[index + 1]

})

// 返回结果(也就是当前的scrollY值属于第几个li区间)

return index

}

在左侧分类中绑定点击事件，

clickMenuItem (index) {

// 得到目标位置的scrollY

const scrollY = this.tops[index]

// 立即更新scrollY(让点击的分类项成为当前分类)

this.scrollY = scrollY

// 平滑滑动右侧列表 better-scroll里的方法

this.foodsScroll.scrollTo(0, -scrollY, 300)

}

用户点击商品列表页会弹出对应的商品详情以供用户来参考，在列表页和详情弹框用户都可以往购物车里加订单。

4.购物车及提交订单

购物车是前端展示的重点，封装购物车组件Shopcart和删减按钮CartControl,CartControl在购物车组件和商品列表里都需要用到，

为了防止点击加减按钮同时触发弹出食品弹框，每个CartControl 组件都使用@click.stop 来阻止了事件冒泡。购物车里存放的数据是要大于零的

，包括食品的名称，价格，数量，总价等。这里通过vuex来管理数据，购物车初始值设置为空数组。点击加号按钮，往state里的cartFoods数组增加食品信息

画个vuex）

定义Getters来设置购物车相关数据信息

export default {

totalCount (state) {

return state.cartFoods.reduce((preTotal, food) => preTotal + food.count, 0)

},

totalPrice (state) {

return state.cartFoods.reduce((preTotal, food) => preTotal + food.count \* food.price, 0)

},

positiveSize (state) {

return state.ratings.reduce((preTotal, rating) => preTotal + (rating.rateType === 0 ? 1 : 0), 0)

}

}

通过计算属性和watch监听来设置购物车不同的样式和提示文字。在用户没有加入购物车的时候，购物车无法触发点击事件，当然样式也没有变化

用户点击商品加入购物车后，如果金额未满足，右侧会提示还差多少元起送，点击你所选的食品信息（价格，数量等），满足一定金额后才可以去支付。

5.店家添加数据

6.后台管理员部分