

在线点餐系统设计与实现

李思羽 黄媛媛

四川大学锦城学院计算机与软件学院 四川 成都 611731

【摘要】本文简述了一个基于 Web 开发的在线点餐系统的开发,顺应了当今时代的快节奏生活和互联网发展,为有线上网点餐需求的用户提供了简单快捷的选择。前端框架使用 Vue 搭建,网页界面使用了 HTML 语言和 Element-Ui 组件库,后端框架使用 Node.js 搭建,数据存储采用 MongoDB 数据库。在该系统中,用户可以登陆注册,线上浏览以及收藏商品,挑选需要的商品加入购物车,以及添加必要的订单信息后进行下单支付,支付完成后能看见自己的订单详情。

【关键词】在线点餐; Vue; Node.js

1 引言

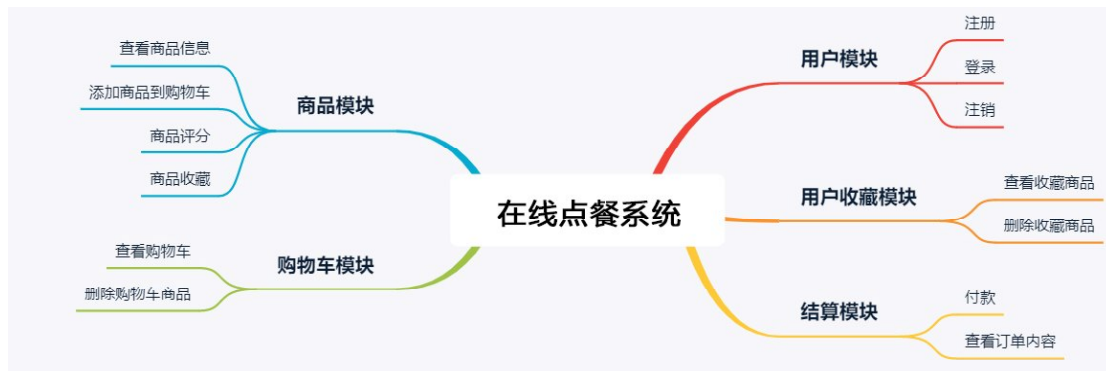
近年来,随着互联网科学技术的发展,越来越多的传统行业开始与互联网进行结合产生新的产业,其中在线点餐系统就是传统餐饮业与互联网结合的结果^[1],其顺应了当今时代下人们快节奏的生活习惯,在既可以节约时间成本的同时又能够获取更多的信息进行选择,餐厅也降低了运营成本和管理成本,更好的方便了人们的生活。

2 系统设计

2.1 技术选型

本系统前端使用 HTML 语言、Vue 框架和 Element-Ui

系统分为用户模块、收藏模块、商品模块、购物车模块、结算模块五个部分。



2.2.1 用户模块

包括登录、注册、注销三个功能,用户通过手机号和密码进行注册,系统将数据传入后端的 json 文件中,登录时将用户输入的账号和密码与 json 数据进行匹配,匹配成功即可登录。登录成功进入首页,点击右上角的退出功能可以注销登录状态。

2.2.2 商品模块

包括展示商品,收藏商品和添加购物车三个功能。该模块将商品图片、名字、价格数据从后端数据中取出并分类展示在网页上,每个商品可以点击加购按钮加入购物车,点击收藏按钮添加到个人收藏夹。

2.2.3 收藏模块

包括查看收藏和取消收藏两个功能。在个人中心点击我的收藏,可以看见商品信息和评分,点击删除可以取消该商品的收藏。

组件库,后端使用 Node.js 和 MongoDB 数据库。Vue 是近几年较为流行的一款渐进式框架,用来构建用户界面,具有轻量级、组件化、简单易学、双向数据绑定等优点^[2]。Element-Ui 则是以 Vue 为基础框架实现的一套组件库,提供了许多网页常用的组件模板,只需在文件中简单引入即可使用。后端使用的 Node.js 是一个 JavaScript 运行环境,基于谷歌 V8 引擎,具有由事件驱动,非阻塞 I/O 等特点,因此适合 I/O 密集的应用。MongoDB 数据库基于分布式文件存储,文件存储直接采用 BSON,因此在文档中可以直接存储更复杂的数据类型,更为灵活。

2.2 功能模块设计

2.2.4 购物车模块

包括加购和删除两个功能,在主页加购的商品可以显示在购物车内,商品数量可以增加或减少,也可以直接删除商品,购物车显示的总价根据购物车内商品的变化而改变。

2.2.5 结算模块

包括查看订单内容和付款两个功能。在购物车中点击去结算按钮进入订单页,该页获取并展示用户收货地址、订单内容、加配送费的总价、选择餐具及添加备注等信息,点击去支付可以进入选择支付方式网页,并在选择支付宝或微信支付后进入不同的网页进行付款,付款完成后可进入个人中心查看我的订单。

3 系统实现

3.1 用户模块

是用户打开网页看见的第一个界面, 实现了注册登录功能, 也只有进行注册登录之后才能进入后面的页面。

3.1.1 注册功能

注册规定使用手机号进行注册, 用户只能输入十一位的数字, 如果不符合要求输入框下方会提示“请输入正确手机号”, 密码则规定至少六位, 不符合要求也会提醒“请输入至少六位字符”, 点击注册按钮, 会获取当前网页输入框的 value 值, 通过 JSON.stringify 转换成 json 格式, 使用 fs 模块的 writeFile 存储到 json 文件。注册成功后可以填写自己的名字地址等收货信息, 也会存储在对应的 json 文件。

3.1.2 登录功能

在登录模块中, 输入手机号和密码, 点击登录按钮, 与 json 文件中的数据进行匹配, json 文件中不存在在该数据会显示“登录失败, 请重新输入”, 存在该数据则登录成功, 进入点餐系统首页, 在点餐系统中点左上角的退出按钮则触发函数切换路由路径, 即退出登录状态, 同时退回到登录界面, 实现注销功能。

3.2 商品模块

商品模块是用户登陆后看见的网站首页, 实现了商品展示功能和数据存储到浏览器本地功能。

3.2.1 商品展示功能

该模块从数据表中获取商品的数据, 转换为 json 格式, 使用了 Element-Ui 中的 Tabs 标签页组件展示不同类别的商品, container 布局容器和 layout 进行每个标签页内的布局, 设置了不限行数每行四列的布局样式, 在每个表格中放入卡片组件, 使用 v-for 动态渲染数据, 获取对象数组中的图片、名字、价格数据渲染到卡片上

3.4.1 数量和价格变化功能

该界面从使用 getItem 获取 shopObj 的 value 值, 也就是加购的商品信息, 并在 HTML 中使用 ul、li 列表布局, ul 上使用 v-for 动态渲染商品信息, 在 li 中插入商品图片、名字和价格, 以及控制数量加减的按钮, 每当按加或者减, 会触发 reduce 或 plus 两个不同的方法, 该方法将数据库中购物车对应的商品数量加一或减一, 同时总价 sum 会随着数量的更改重新将购物车内商品各自的数量和价格相乘并且加起来, 实现同步的总价变化。

3.5 结算模块

在购物车页面点击去结算按钮, 会触发绑定的点击方法切换路由, 跳到结算页, 该模块实现订单内容查看和付款功能。

3.5.1 订单查看功能

结算页在后端获取当前用户的名字、手机号以及收货地址, 并在 HTML 中使用插值动态生成, 商品列表使用 Element-Ui 的 Table 列表组件, 从后端取出与购物车对应的同一个 key 值的 value 值, 将 json 格式的数据转换成 JavaScript 对象并存入当前页面的 data 数据, 在 HTML 中用 v-for 动态渲染列表, 完成商品列表的展示。是否选择餐具的单选框用 Element-Ui 的 Radio 单选框实现, 用 v-model 绑定变量, 变量的值为元素的 label 属性值, 通过属性值设置的不同, 完成变量值的改变, 实现单选功能。HTML 中用 textarea 标签添加文本框, 用于给用户添加订单备注。商品总价则依旧从后台数据中

3.2.2 数据存储功能

同时每个卡片还添加了加入购物车和收藏按钮, 加购和收藏的数据存入后端数据库中, 加入购物车按钮使用 @click 绑定 addCart 方法, 该函数获取点击元素的索引, 使用 foreach 遍历数据库中 shopObj 的 value 查看是否已存在要加购的商品, 如果存在则该商品的数量值加一, 如果不存在则将该商品数据作为一个 json 对象存入 shopObj 的 value 值中。收藏按钮也是同理, 使用 @click 绑定函数, 传递点击的元素索引, 遍历查看数据库是否存在该商品, 如果存在则不添加, 不存在则添加该商品数据。

3.3 收藏模块

收藏模块从网页菜单栏的个人中心进入我的收藏, 实现了显示和删除收藏功能。

3.3.1 显示收藏功能

个人中心和首页一样采用 element-ui 的 Tabs 标签页组件来对不同信息进行展示, 每个商品的信息以卡片组件的格式显示在页面上, 收藏夹从后端中根据 key 值获取对应 value, 并存入当前网页的 data 数据, 使用 v-for 动态渲染该数据到卡片上, 包括商品的图片, 名字和评分。

3.3.2 取消收藏功能

在卡片上设置取消收藏按钮, 使用 @click 绑定 cancel 方法, 传递当前点击元素的 id, 通过 findIndex 方法找到 data 中对应 id 的数据并用 splice 方法是进行删除。

3.4 购物车模块

在首页中挑选好商品后, 可点击菜单栏中的购物车选项进入购物车界面, 实现了数量和价格变化功能。

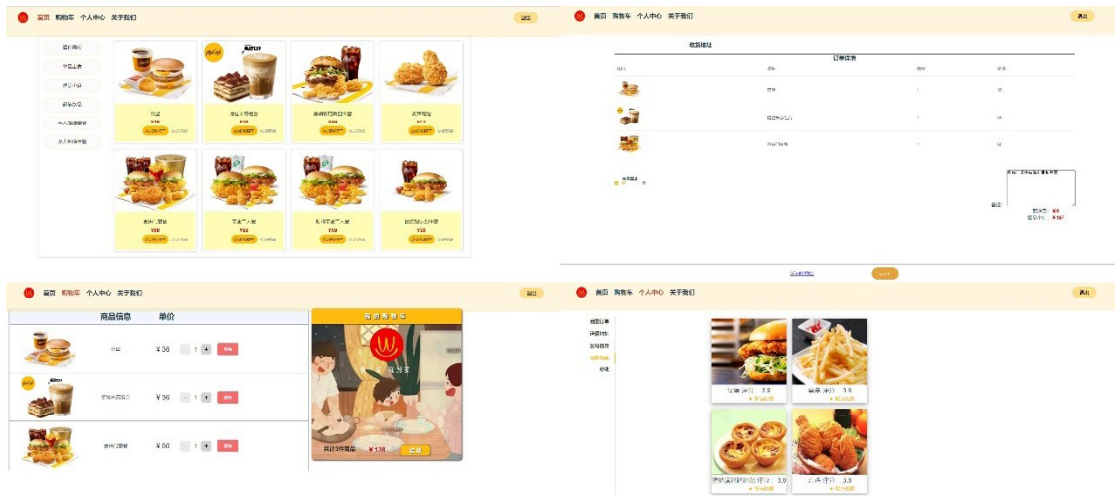
获取, 并添加规定的配送费后变成需要支付的总价, 使用数据插值展示在界面上。

3.5.2 付款功能

在结算页面点击去支付按钮可以切换路由跳转到支付页, 支付页有两个单选按钮, 支付宝和微信支付, 以及一个确认付款的提交按钮。和前页中实现餐具选择的方法相似, 对两个选项的同一个 label 属性设置不同的值, 用 v-model 绑定一个变量, 变量值为属性值, 点击不同的选项, 变量值被修改。在确认付款按钮上绑定一个点击方法, 方法内判断变量值是否与支付宝按钮的属性值相等, 如果是则切换路由到支付宝界面, 如果不是则切换路由到微信支付界面。

支付宝和微信支付界面构成大致相同, 除了固定的图标和文字等 HTML 元素, 页面中心的付款二维码会替换成两种支付方式对应的二维码, 没有对应的手机 app 时可点击二维码下载手机支付 app 按钮, 该按钮使用 Element-Ui 的 Dialog 对话框组件, 该组件由 visible 属性控制显示与否, 属性值默认为 false, 按钮被点击时属性值修改为 true, 实现弹出对话框效果, 并在对话框内放入 app 下载二维码提供下载途径。在扫码支付完成后点击二维码下方的支付完成按钮可切换路由跳转到个人中心的我的订单界面, 刚刚支付的订单内容可以显示在该界面上。

4 部分成果展示



5 结语

该项目设计并开发了基于Web基础的在线点餐系统，该系统利用了Vue搭建前端框架，利用Element-Ui的部分组件和HTML语言进行可视化界面的开发，后端采用Node.js搭建服务器，MongoDB数据库存储用户数据和商品数据。该系统目前已经实现了用户登陆注册、个人信息存储、商品数据展示、个人收藏、商品加入购物车、生成订单、扫码支付等功能，能够作为一个完整的点餐系统使用。该系统的界面美观度和用户数据的安全性保护还存在提升空间。此外可以添加定位接口，根据商家

到收货地址的距离，动态修改配送费，让系统使用起来更加方便。在后续不断的开发过程中也会针对这些不足之处进行改进，争取做出一个更加美观完善方便快捷的在线点餐系统。

【参考文献】

[1] 徐金红, 武文婧. 互联网经济对餐饮业发展的实证分析[J]. 统计与管理, 2021, 36(05): 4-9.
[2] 陈陆扬. Vue.js 前端开发快速入门与专业应用. 北京: 人民邮电出版社, 2017