

潘伟杰, 冲蕾, 李霖奇, 等. 基于深度学习与百度 AI 的菜谱推荐与健康分析微信小程序设计[J]. 智能计算机与应用, 2024, 14(5): 257-264. DOI: 10.20169/j.issn.2095-2163.240537

基于深度学习与百度 AI 的菜谱推荐与健康分析微信小程序设计

潘伟杰, 冲蕾, 李霖奇, 王唯佳, 施倩倩, 陈思枫

(上海工程技术大学 城市轨道交通学院, 上海 201620)

摘要: 本程序主要任务是在前作的基础上做出一些改良。前作^[1]是以深度学习为核心, 开发了一个可以进行食谱推荐和健康分析的微信小程序。本论文中, 将利用基于 TensorFlow 搭建的神经网络模型; 通过使用微信的相应接口, 在百度 AI 的基础上, 实现对用户的卡路里摄入动态监控。本文旨在用丰富的组件来拓展小程序的相关页面。其中, Vant 组件完善菜品界面; ECharts 组件完善卡路里曲线; 利用 Python 中重量级框架 Scrapy 和 Django, 实现搜索关键词、词云图以及其他可视化图标的功能; 同时对小程序的界面进行了一定程度的美化。

关键词: 微信小程序; 云函数; 深度学习; TensorFlow

中图分类号: U491.17 **文献标志码:** A **文章编号:** 2095-2163(2024)05-0257-08

WeChat applet design for menu recommendation and health analysis based on in-depth learning and Baidu AI

PAN Weijie, CHONG Lei, LI Linqi, WANG Weijia, SHI Qianqian, CHEN Sifeng

(School of Urban Rail Transportation, Shanghai University of Engineering Science, Shanghai 201620, China)

Abstract: The main task of this work is to make some improvements on the basis of the previous work. The previous work is to build a WeChat applet that can implement menu recommendation and health analysis with in-depth learning as the core. In this research, the paper uses the neural network model based on TensorFlow; by using the corresponding interface of WeChat, on the basis of Baidu AI, the paper realizes the dynamic monitoring of the user's calorie intake. This work aims to use rich components to improve the relevant pages of the applet; the Vant component to improve the menu interface, and the ECharts component to improve the calorie curve. Using the heavyweight framework Scrapy and Django of Python, the research could realize the function of searching keyword, word cloud map and other visual icons. Besides, the paper also beautifies the interface of the applet to a certain extent.

Key words: WeChat applet; cloud function; in-depth learning; TensorFlow

0 引言

2020年后, 考虑到新冠病毒的特点就使得许多学校和公司食堂已经投入使用无人化和自动化的餐饮服务, 这标志着智能化和数字化技术进入了一个全新的发展阶段。已有研究发现食物中的一些化学元素能够影响人体内神经递质的含量, 影响个体的情感反应, 从而导致产生多种心理状态。由此可以看出, 食物对神经递质的含量有很大的作用, 与人的情绪变化也有着密切关系, 直接影响了人的感情中枢^[2]。

Python 作为一种简单实用的编程语言, 具有很好的可读性, 对于初学者来说比较容易上手^[3]。其中

提供了高效的数据结构, 能有效地面向对象编程^[4]; 底层是用 C 语言写的, 因此运行速度非常快^[5]。现在 Python 已被广泛应用于人工智能、云计算开发、自动化运维及 Web 开发等领域。基于此, 本文拟对菜谱推荐与健康分析微信小程序设计展开深入研究, 在满足饮食健康的要求下, 取得较好的使用效果。

1 小程序中 Django 的应用

Python 下有多款开源的 Web 框架, 比如 Django、Tornado、Flask 等, 其中 Django 是最有代表性、应用范围最广泛的^[6]。

Python 开发的 Django 框架是一款免费且开源

基金项目: 国家级创新项目(202210856034)。

作者简介: 潘伟杰(2001-), 男, 本科生, 主要研究方向: 微信小程序开发, Email: 1874441736@qq.com; 冲蕾(1982-), 女, 讲师, 主要研究方向: 城市轨道交通信号自动控制。

收稿日期: 2023-04-12

的 Web 应用框架,由于 Python 语言可以跨平台使用的特性,Django 框架同样支持在 Windows、Linux 和 Mac 系统上的使用^[7]。Django 框架的设计初衷是简化并加速新闻网站的开发。2003 年,Web 开发人员使用 Python 开发新闻网站,考虑到新闻行业信息更新快的特点,就必须要提高产品的开发速度,在这种情况下开发者们则决定创建一个节省开发时间并适用于新闻业的框架,于是 Django 框架诞生了^[8]。随后,该研发小组决定把 Django 框架作为一个开源软件发布。2005 年 7 月,Django 框架首次出版,遵守伯克利软件发行(BSD)版权规定,并在 2008 年 9 月发行了第一个正式版 Django1.0^[9]。此后,大约每隔 8 个月就会推出新的版本,每个新版本都有所改进,迄至目前最新的版本是 2020 年发布的 Django3.0。经过不断的研发和更新,Django 框架有了数以万计的用户和贡献者,成为了 Web 开发中最流行的框架之一。使用 Django 框架的主要目的是为开发者节省时间,让 Web 开发变得更容易。Django 框架提供了广泛的插件和组件库来简化 Web 的开发流程,使开发者将有限的精力集中在系统更重要的功能上^[10]。

本程序所使用的 MVC 模式^[11],其核心理念是分工合作、解耦,减少了各个模块间的耦合,提高了模块的伸缩性和移植性,达到了向后兼容的目的。

MVC 是 Web 开发领域著名的设计模式,MVC 是 Model-View-Controller 的缩写,其中各部分含义具体如下^[12]。

(1) Model:表示数据存取层,处理与数据相关的所有事务,包括对数据表的定义和数据的增删改查等;

(2) View:表示视图层,是系统前端显示部分,负责显示什么和如何显示,用于向用户展示结果;

(3) Controller:表示控制层,主要根据从 View 层输入的指令来检索 Model 层的数据,并在该层编写代码产生结果并输出,用于处理请求、获取数据和返回结果^[13]。

在 Django 的 urls.py 中自定义路由,并在小程序中通过 wx.request 发送 POST 请求在 data 中传参,以此调用相关逻辑功能。

1.1 任意食物卡路里搜索功能

view 文件是视图层文件,封装业务逻辑,有函数与面向对象两种开发模式^[14]。

views.py 文件是 Django 中的核心部分,起到了项目控制的作用,其作用和以前提到的在 MVC 模式

中的“C”的功能类似。Django 中的核心部分是 views.py 文件。

本微信小程序主要通过 view 文件来实现数据挖掘、词云、数据分析功能。

如果微信小程序的数据库中有所要搜索的菜品,则直接数据库中调用,没有、发送 post 请求,data 传参调用 Django。

在 Django 路由接受到微信小程序的 post 传参后,在 views.py 中,调用 subprocess 库进行对 Scrapy 的联动,挖掘所需卡路里数据。

首先,创建 scrapy 项目,在 items.py 中设置好需要的字段,在 middleware.py 中的 process_request 函数中,修改请求头 request.headers[‘User-Agent’] 字段为 Mozilla/5.0 (Windows NT 5.1) AppleWebKit/535.11 (KHTML, like Gecko) Chrome/17.0.963.84 Safari/535.11 SE 2.X MetaSr 1.0。然后通过 lxml 获取页面内容响应, `title_href = (tree.xpath(‘//ul[@class="food-list"]//li/div[@class="text-box pull-left"]//a/@href’))` 获取菜品名以及 `title_href = (tree.xpath(‘//ul[@class="food-list"]//li/div[@class="text-box pull-left"]//a/@href’))` for i in range(len(title_href)): `title_href[i] = self.search_home_url + title_href[i]` 获取菜品详情页链接。之后在 parse 函数中解析详情页,获取热量(大卡)、脂肪(克)、纤维素(克)、碳水化合物(克)、蛋白质(克)和菜品图片链接。随后在 pipelines.py 中将数据保存至微信数据库。研究给出的微信小程序调用 Django 及 Scrapy 示意如图 1 所示。

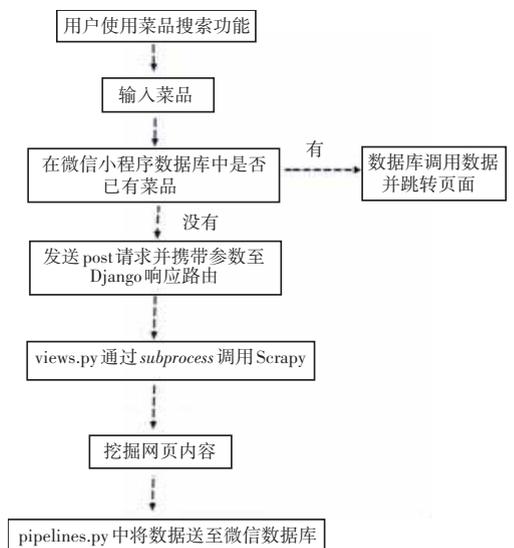


图 1 微信小程序调用 Django 及 Scrapy 示意图

Fig. 1 Flow chart of Wechat mini program calling Django and Scrapy

1.2 词云图功能实现过程

首先,微信小程序将用户近期摄入的菜品名以 json 格式数据,post 传参至 Python 的 Django 中,通过 View 视图,先导入需要用到的 Python 第三方库:jieba,matplotlib,wordcloud,PIL,numpy。然后使用 jieba 库将数据进行清洗,清洗前为 txt 文件,清洗后变为以空格分隔的词语字符串 string。接着通过函数 *tcg()* 来获得函数 *wc.generate_from_text()* 所需要的字符串 string,随后将字符串 string 传给函数 *wc.generate_from_text()* 进行绘图。这里,仍需一提的是:

(1) *jieba.cut(texts)* 的作用是将 *texts* 切分为若干组词语,*cut()* 和 *lcut()* 的不同:*cut* 返回的是生成器,*lcut* 返回的是列表;

(2) *string = ' '.join(cut)* 表示用空格将这些词语切分开,得到一个以空格分隔的词语字符串,供后面调用。

输入绘制云图所使用的函数为 *WordCloud()*,其参数见表 1,此函数接受一些参数并返回一个对象 *wc*。

表 1 WordCloud 参数表
Table 1 WordCloud parameters table

参数	描述
<i>width</i>	指定词云对象生成图片的宽度,默认 400 像素
<i>height</i>	指定词云对象生成图片的高度,默认 200 像素
<i>min_font_size</i>	指定词云中字体的最小字号,默认 4 号
<i>max_font_size</i>	指定词云中字体的最大字号,默认为 1
<i>font_step</i>	指定词云中字体字号的步进间隔,默认为 1
<i>font_path</i>	指定字体文件的路径,默认为 None
<i>max_words</i>	指定词云显示最大的单词数量,默认 200
<i>stop_words</i>	指定词云的排除词列表,即不显示的单词列表
<i>mask</i>	指定词云形状,默认为长方形,需要引用 <i>imread()</i> 函数
<i>background_color</i>	指定词云图片的背景颜色,默认为黑色

在此基础上,用前面产生的对象 *wc* 调用 *generate_from_text()* 方法即可完成绘图。实现词云图,以二进制传到微信小程序 API 中,实现两端联动。对此过程可做阐释分述如下。

(1) 数据收集与清洗:首先,确定词云图要展示的主题或内容,收集与之相关的文本数据。对数据进行清洗,去除重复、无关或格式不正确的词汇;

(2) 词频统计:对清洗后的数据进行词频统计,得出每个词汇在文本中出现的次数。可以使用现有

的分词工具和统计算法来完成这一步;

(3) 字体大小与颜色设置:根据词频来设置词汇的字体大小,词频越高,字体越大。颜色设置可以根据需要进行调整,可以选择渐变色或者与主题相关的颜色;

(4) 布局与排列:在布局上,可以采用随机分布的方式,使得词云图看起来更加生动和自然。也可以考虑加入一些动画效果,提升用户体验;

(5) 交互设计:为了增强词云图的互动性,可以加入点击或悬停提示功能,当用户点击或悬停在某个词汇上时,显示该词汇的详细信息或相关链接。还可以设计一些过滤或搜索功能,方便用户查找特定的词汇或主题;

(6) 优化与调整:在设计过程中,不断对词云图进行优化和调整,确保其视觉效果和用户体验达到最佳状态。可以根据用户的反馈和数据迭代,提升词云图的质量和实用性。

研究得到的词云图如图 2 所示。结合这张图片内容,可以看到词云图中包含了“膳食月饼”、“中国”、“Dai”、“疾病”和“公热是”等词汇。在实际设计中,可以根据具体的主题或内容,调整这些词汇的字体大小和颜色,使其更符合主题需求。同时,也可以考虑加入更多的相关词汇,丰富词云图的信息量。通过合理的布局和交互设计,可以打造出一款既美观又实用的微信小程序词云图界面。



图 2 词云图

Fig. 2 Word cloud map

1.3 数据可视化功能实现

本程序在原程序的基础上新增卡路里数据图标的生成,用户输入 7 日卡路里后,生成柱状图、饼图和折线图。

Pyecharts 是一个用于生成 Echarts 图表的 JS 类库,Echarts 是百度开源的一个数据可视化工具包^[15]。利用 Echarts 实现图标功能步骤如下:通过 *pages* 函数的调用,实现用户点击跳转到柱状图、饼图或折线图界面。在界面中使用 *e.detail.value.c* 获取数据,并存储到 *data* 中。调用 *ec-canvas* 中的

Echarts,根据所要制作的图表,使用 `type` 来调用 Echarts 中所需部分并进行图表制作。用 `detail link_child` 在图块上显示输入的数据。研究给出的数据可视化功能实现示意如图 3 所示。

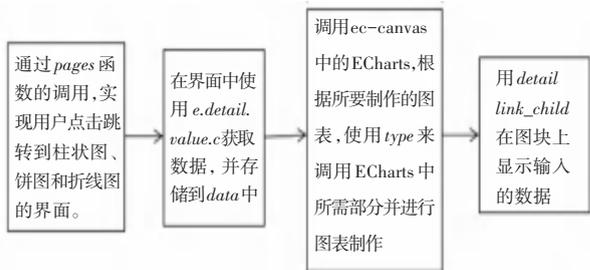


图 3 数据可视化功能实现示意图

Fig. 3 Flow chart of data visualization function implementation

1.4 深度学习

人工神经网络 (Artificial Neural Network, ANN),是一种模仿生物神经网络结构和功能的数学模型或计算模型,用于对函数进行估计或近似。神经网络主要由输入层、隐藏层、输出层构成,当隐藏层只有一层时(隐藏层层数及隐藏层神经元由人工设定)。该网络为 2 层神经网络,由于输入层未做任何变换,可以不看作单独的一层。实际中,网络输入层的每个神经元代表了一个特征,输出层个数代表了分类标签的个数。在做二分类时,如果采用 *sigmoid* 分类器,输出层的神经元个数为 1;如果采用 *softmax* 分类器,输出层神经元个数为 2;如果是多分类问题,即输出类别 ≥ 3 时,输出层神经元为类别个数。

1.4.1 搭建模型

在前文基础上,对定义模型步骤阐述如下:

首先,为指定输入资料的架构而确定了占位器的位置。

其次,对神经网络的连接模式、神经元数量、活化函数等进行界定。本文设定 2 个隐藏层,每层有 100 个神经元,使用 *sigmoid* 函数作为活化函数。通过引入权重、偏移率等可变参数,对所建立的人工神经网络进行合理性检验和分析。定义损失和优化等函数,用于对所构造的神经网络进行限制和优化。训练、保存模型的部分,其工作包括:将 `saver` 对象 `train.Saver()`,用于保存模型。建立并开启会话 `Session()`。对模型中的全部参数进行 `global_variables_initializer()` 的初始化,如模型最优函数、损失函数等。本文运用了渐变法对这一工作进行了最优,经过计算得出,在 0.15 的情况下,此时工作状态是最好的。对该方法进行循环迭代。最终模型的损耗是 2.04,具有一定的收敛性。将该模型

的 `saver.save(sess, "./src/tmp/model.ckpt")` 进行存储,并将产生一组“.ckpt”文件,用于在一个特定的目录下进行修复。

1.4.2 腾讯云上部署

将部署后端的程序编码上传到腾讯云端,并对运行入口文件 `sl.handler.py` 进行定义,以实现 `app.py` 的运行。`app.py` 包含了通过装载模型和进行预测来完成服务器请求的部件。其中,实施服务端请求的关键是:构造 Flask 实例 `app=Flask(__name__)`。

定义 `pathway operationalgorithm@ app.route()`,在此以 ANY 形式进行申请,也就是可以接受任何申请。确定一个通路运算的 `def 索引()`。

将所获得的预测参数进行正则化和转化为列表格式。在此仅需要抽取有用的数值,也就是 `data = list(re.findall(r'\d+\.\d*', user_selection))`,并且转换成 `list(map(int, data))`。

对所建立的模型和预报模块进行编程,得出预报结果。

通过使用 `jsonify()` 功能,将所获得的 JSON 转换成一个传输值。

在模型与预测方面,将重点研究如下内容:

- (1) 对一个图进行重新定义,即 `reset_default_graph`;
- (2) 确定用于指定所述输入数据的格式化的占用程序;
- (3) 建立和开启对话;
- (4) 建立储备器保存并还原该数据的 `Saver.restore(sess, "./tmp/model.ckpt")`;
- (5) 一个预测资料 `sess.run(hypothesis, feed_dict = {X:选择})`;
- (6) 用属于自己的分类对其进行匹配,形成词典形式。

以可能性的递减顺序对此词典 (`list1, key = Lambdae:e.__getitem__("probability"), 反转 = True`) 进行排序,并且返回。将此云函数的触发方式设定为 API 网关触发,使用 API 网关触发可以选择同步或异步呼叫,并且还可以使用不同的路径操作修饰器或函数来实现多个功能,充分发挥函数的作用,从而使多个功能得到充分发挥。在小程序上只需使用 `HTTPS(wx.request())` 就可以了。

2 基于百度 AI 的菜品识别功能

2.1 百度 AI

百度 AI 是中国规模最大的一家 AI 平台,可为

用户提供许多便捷快速的API界面以及特定的应用。当前,百度AI在语音、图像、文字以及客流量等方面已经获得了非常明显的结果,在准确率方面已经处于一个比较高的水平。

2.2 实现菜品识别并录入

利用百度AI的图片分析,可以快速、准确地对菜肴进行识别。其实现流程如下:

调用微信提供的(*wx.chooseImage()*)界面,指导用户选择拍照或相册上传照片,返回的结果为用户上传照片的暂时路径。

通过(*wx.getFileSystemManager().readFile()*)获得用于上传图像的暂时路径,并且通过Base64格式转换编码。

利用从百度AI官方网站上获得的*Client_id*和*Client_secret*,通过GET请求(*wx.request()*)界面,以HTTPS协议的形式,向百度AI所提供的API界面发出GET请求,从而获得一个可以被调用的独特*access_token*。

调用(*wx.showLoading()*)装载模块,以提醒使用者进行标识。

重新使用*wx.request()*界面,通过POST,向百

度AI申请API界面,将之前得到的*access_token*加入到URL中,DATA作为被识别图像的base64代码。通过使用(*wx.hideLoading()*)界面,显示被识别菜品的卡路里和可信度。当相应的菜单辨识结果出现在网页上时,就可以设定输入键,让使用者依据实际的食物摄取来进行输入。调用云函数(*wx.cloud.callFunction()*),通过*db.collection().add()*,在数据库的对应集合中,将用户*openid*、菜品名称、数据来源、卡路里值、创建时间等信息,显示在个人中心页面上。

3 微信小程序

3.1 幸运大转盘实现

本程序在原程序的基础上新增幸运大转盘的功能。幸运大转盘旨在通过趣味的方式,解决用户对于菜品的选择困难。具体实现过程如下:

使用*e.detail.value.c*获取数据,并存储到*data*中。将输入的菜品显示在转盘的每个部分中。调用*wx.random*函数,随机生成指针角度,并通过微信小程序(*wx.createAnimation*)呈现旋转动画。*wx.createAnimation*参数见表2。

表2 *wx.createAnimation*(Object object)参数表

Table 2 *wx.createAnimation*(Object object) parameters table

属性	类型	默认值	必填	说明
duration	number	400	否	动画持续时间(单位:ms)
timingFuntion	string	'linear'	否	动画效果
delay	number	0	否	动画延迟时间(单位:ms)
transformOrigin	string	'50%50%0'	否	

3.2 点赞功能实现

本程序在原程序的基础上新增点赞系统,通过数据库中用户点击数据量,可以直观地推荐用户喜欢的菜品。实现过程具体如下:

首先,云函数从数据库中获取了关键数据,如“likelikenulikepeople”等,这些数据被存储在一个名为*newList*的二维数组中。这一过程确保了数据的一致性和准确性,为后续的处理奠定了坚实基础。

接下来,云函数开始对*newList*中的;“likepeople”数组进行遍历,逐一检查每个用户ID是否已存在于另一个名为*izan*的数组中。这一步的关键在于实现用户信息的精准匹配,为后续的点赞、取消点赞等操作提供准确的数据支持。

如果某个用户ID在*izan*数组中已存在,云函数会将该用户ID对应的“food_id”添加到*izan*数组中。这一步骤实现了用户点赞信息的聚合,便于后

续的数据分析和处理。

对于未在*izan*数组中找到的用户ID,云函数会创建一个新的数组来存储该用户ID及其对应的“food_id”。这一机制确保了每个用户的点赞信息都能得到妥善处理,避免了信息的遗漏和丢失。

在完成上述步骤后,云函数会进一步检查*izan*数组中是否存在该食物ID或该食物的“openid”。如果存在,则表示该用户已对该食物进行过点赞操作,云函数将进行相应的处理;否则,云函数将继续处理下一个用户ID。这一过程确保了用户点赞信息的准确性和完整性。

整个处理流程中,云函数会在每次迭代时更新新的*data_list*数组,以确保数据的实时性和一致性。这种动态更新的机制使得云函数能够迅速响应用户的操作,提升了数据处理效率。

最后,云函数与后台进行交互,将处理结果同步

到后台数据中。通过使用 *update* 等操作,确保了数据的及时更新和同步,为用户提供了更好的使用体验。

3.3 评论区功能实现

本程序在原程序的基础上新增加评论功能,通过页面初始化时,在 *download* 获取评论数据并传至云数据库,实时反馈至前端界面。实现步骤如下:

首先,系统会获取用户的 *open_id*,作为后续查询和操作的标识。紧接着,系统将在数据库中查询是否存在与该用户 *open_id* 一致的评论。这一步的目的是为了检查用户之前是否已有过评论记录,以便进行后续操作。

如果查询结果显示数据库中存在与用户 *open_id* 一致的评论,系统会显示相应的标志,这些标志可以是颜色、图标或文字等,方便用户辨认和查看自己的历史评论。反之,如果查询结果为否,系统将不会显示相应的标志。

接下来,系统会获取数据库中关于该菜品的所有评论,并将其显示在页面上。这样,用户不仅可以查看自己的评论,还可以浏览其他用户的评论,从而获取更全面的信息。

最后,如果用户决定发表评论,系统会将用户的评论上传至数据库,并立即刷新页面。这样,新发表的评论就能立即显示在页面上,供其他用户查看。

整个流程设计简洁明了,步骤清晰,既考虑到了用户的浏览需求,又兼顾了评论的上传和展示功能。通过这一流程,用户可以轻松发表自己的评论,并随时查看和管理自己的评论记录。

3.4 Vant 组件介绍

Vant 是一个轻量、可靠的移动端组件库,于 2017 年开源。本程序使用 Vant 组件在前作的基础上对点餐界面进行一定程度的美化。

3.4.1 前端页面的设计研发

图 4 和图 5 是使用 Vant 组件实现的前端页面。对此可做阐释剖析如下。

(1)图 4 中界面优化的设计研发。首先,整体设计以橙色和白色为主色调,这种色彩搭配既醒目又简洁,能够迅速吸引用户的注意力。前端代码中,通过 CSS 样式定义了界面的背景色、字体色和按钮色,实现了这一视觉效果。

其次,菜单布局采用了清晰的列表形式,每个菜品都以图文结合的方式呈现,包括菜品的图片、名称和价格。在前端代码中,这些元素被组织成有序的 HTML 列表结构,通过 CSS 进行样式控制,使得每个

菜品项都有一致的布局 and 外观。

另外,菜品下方设置了数量选择器,用户可以方便地增减所点菜品的数量。这部分功能是通过 JavaScript 代码实现的,监听用户的点击事件,动态地修改菜品数量并计算总价。

界面的下方有一个红色的“结账”按钮,作为用户完成点餐操作的关键入口。在前端代码中,这个按钮的样式和交互效果是通过 CSS 和 JavaScript 共同完成的,使其在页面上格外醒目,并且能够响应用户的点击操作。

除此之外,图片还展示了一些搜索、商品详情和卡路里显示等功能。这些功能的实现也需要前端代码的支持,如使用搜索输入框和搜索按钮来完成搜索功能,使用详情页面来展示商品的详细信息,通过动态计算来显示卡路里信息等。

总的来说,这个微信小程序餐饮点餐应用的界面设计简洁明了,易于操作,前端代码在其中发挥了关键作用,通过合理的 HTML 结构、CSS 样式和 JavaScript 逻辑,实现了界面的美观性和交互性。



图 4 界面优化

Fig. 4 UI optimization

(2)图 5 中转盘的设计研发。在前端代码中,要实现这样的雷达图,通常会使用 Canvas 绘图 API 或者 SVG 技术。前端开发者需要编写 JavaScript 代码来处理数据,计算每个菜品在各个方面的得分,并根据得分绘制出相应的点和箭头。同时,CSS 样式将被用于设置图表的布局、颜色、字体等视觉属性,确保图表的美观性和可读性。

此外,考虑到用户体验,前端代码还需要处理用户的交互操作,如点击、滑动等。在这张图片中,看到了“确定”和“重置”两个按钮,前端代码需要为这些按钮添加事件监听器,以便在用户点击时执行相应的操作,如提交表单或重置图表。

总结来说,微信小程序相关的界面设计在雷达

图的应用中,前端代码发挥着至关重要的作用。可以负责处理数据、绘制图表、设置样式以及处理用户交互,从而为用户提供一个直观、易用且美观的数据可视化界面。

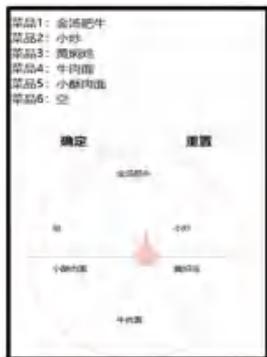


图 5 转盘

Fig. 5 Turn plate

3.4.2 用户行为记录的设计研发

图 6 和图 7 是使用 Vant 组件实现的用户行为记录页面。本次研发中,增加了用户显性及隐性行为的记录:收藏、录入、点赞、点踩、评论(显性),点击(隐性)。通过调用微信云数据库实现点赞,通过根据菜品创建新数据库,赋予用户写入权限,并且显示在评论区。对此可做阐释解析如下。



图 6 点赞

Fig. 6 Tags



图 7 评论区

Fig. 7 Comments section

(1)图 6 中点赞的设计研发。首先,图片中的手工面条是主角,因此前端代码需要确保面条、蔬菜和肉块的图片能够清晰、逼真地展示出来。这通常涉及到使用高质量的图片资源,并通过 CSS 样式调整图片的大小、位置和透明度,以达到最佳的视觉效果。

其次,卡路里信息(260 cal)的显示也十分重要。前端代码会采用一种清晰的字体和大小,将这一信息放置在显眼的位置,以使用户能够快速了解菜品的热量情况。同时,可能还会使用不同的颜色或背景来突出这一信息,增加其可读性。

图片下方的“第一食堂”字样可能是品牌标识或者地点提示。前端代码需要确保这一文本的字体、大小和位置与整体设计风格保持一致,同时也要考虑到不同屏幕尺寸下的适配问题,确保在各种设备上都能良好地显示。

在图片的左下角,有一个星星图标,表示用户可以对这道菜进行评分。这涉及到前端代码中的交互设计。当用户点击这个星星图标时,前端代码需要发送一个请求到后端服务器,记录用户的评分行为,并可能更新界面上的评分显示。同时,为了保证用户体验,前端代码还需要处理各种异常情况,如网络错误、评分失败等。

最后,还有一个“点击查看评论区”的提示,这通常是一个按钮或者链接。前端代码需要为这个元素添加点击事件监听器,当用户点击时跳转到相应的评论区页面或者弹出评论框。在这个过程中,前端代码还需要处理页面跳转的逻辑、加载数据的请求以及评论区的界面渲染等任务。

(2)图 7 中评论区的设计研发。首先,整个界面采用了橙色和白色为主色调,这种配色方案给人一种温馨、舒适的感觉,有助于提升用户体验。前端代码通过 CSS 样式控制,实现了这种颜色的搭配和布局,确保界面在视觉上的一致性和美观性。

在界面的顶部,可以看到时间戳“23:02”以及“Weixin”和“评论区”的标签。这些元素的显示也是由前端代码控制的,确保其在界面上的准确位置和清晰可读。

聊天对话框是界面的核心部分,其中包含了用户之间的评论互动。前端代码负责接收后端数据,并动态地渲染出这些评论内容。例如,“很好吃”、“太实惠了。”和“下次再来。”这些评论消息都是通过前端代码展示在界面上的。同时,代码还处理了表情符号的显示,使得界面更加生动和有趣。

在对话框的底部,有一个蓝色的圆形按钮,上面写着“很好吃”。这个按钮通常用于用户快速发送评论或点赞。前端代码为这个按钮添加了点击事件监听器,当用户点击时,会触发相应的 JavaScript 逻辑,向服务器发送请求或执行其他操作。

此外,前端代码还需要考虑界面的响应式设计,以确保在不同尺寸和分辨率的屏幕上都能良好地显示。这包括调整界面元素的布局、字体大小等,以适应各种设备。

图6和图7展示了通过调查分析完善当前食堂的新菜品,并添加至数据库,提高相应菜品的精确性,达到精确至每种菜品的目的。

4 结束语

基于现在菜品越来越丰富,身体健康指标多元化的背景下,本程序主要是在原有程序上进行对于前端的菜单界面进行细化与再设计、交互增强和功能创新,以及后端的云数据库的更新与云函数的完善,使得其能更好地够实现菜谱推荐与健康分析两大功能。

本程序使用 Django 来调用相关功能, MongoDB 作为数据库对于原微信小程序进行修改。在能够实现菜谱推荐与健康分析两大功能的基础上,丰富了录入菜品的种类除此之外,还新增了转盘、点赞和评论区等功能,同时对于小程序的界面进行一定的美化。

参考文献

[1] 陈沁仪, 冲蕾, 于泽昊, 等. 基于深度学习与百度 AI 的菜谱推荐

与健康分析微信小程序设计[J]. 智能计算机与应用, 2022, 12(6): 146-154.

- [2] 黄琳. 揭秘饮食与情绪的心理机制—评《食物与情绪: 食欲背后的心理学》[J]. 中国酿造, 2020, 39(3): 219.
- [3] 郭鹤楠. 基于 Django 和 Python 技术的网站设计与实现[J]. 数字通信世界, 2023(6): 60-62.
- [4] Python 官方. Python 教程—Python3.9.1 文档[EB/OL]. [2020-10-05]. https://baike.baidu.com/reference/407313/5b37f0qgw_UbgSv6r-ci9fX3V94HLy3YOj6_kSa1FnTMOCLsSaf8naGWHHqv3hPLITah2-EKtH_uj0wb0iVB0WgtuypJk_i0Zx3Aco-Y7RCs.
- [5] 自强学堂. Python CGI 编程[EB/OL]. [2021]. <https://code.ziqiangxuetang.com/python/python-cgi.html>.
- [6] 韦平安. 基于 Django 的网络学习平台的开发与应用[J]. 山西电子技术, 2023(2): 71-73.
- [7] 周晓梅, 段红秀. 基于 Django 的就业推荐系统的设计与实现[J]. 电脑知识与技术, 2021, 17(27): 75-77.
- [8] CUEVA F M D, MENDOZA J S, BALENDRES M A. A new *Coleotrichum* species causing anthracnose of chilli in the Philippines and its pathogenicity to chilli cultivar Django[J]. *Crop Protection*, 2018, 2018(12): 112.
- [9] 齐俊峰. 基于 Django 的校园管理系统的分析与开发[D]. 兰州: 兰州大学, 2013.
- [10] 童乐. 基于 Django 开发框架的在线教育系统设计及实现[D]. 武汉: 华中师范大学, 2020.
- [11] 王冉阳. 基于 Django 和 Python 的 Web 开发[J]. 电脑编程技巧与维护, 2009(2): 56-58.
- [12] 杨洪涛. 基于 Django 的 MVC 框架设计与实现[J]. 电脑知识与技术, 2023, 19(4): 62-65.
- [13] 夏帮贵. Python Web 开发基础教程: Django 版[M]. 北京: 人民邮电出版社, 2020.
- [14] 林彬, 杨彬彬, 孙芳. 基于 Django 框架的高校就业信息可视化平台的设计与实现[J]. 辽宁师范大学学报(自然科学版), 2022, 45(2): 201-208.
- [15] 张玉叶. 基于 PyEcharts 的数据可视化[J]. 电脑知识与技术, 2022, 18(2): 24-27.