数据库期末大作业报告

图片

应用介绍

我们小组基于 OpenGauss 实现了中大校园食堂的菜品评价系统——**中大食评**。经过前期小组讨论和调研，我们的数据库包括 5 个实体，并基于 Django 的框架实现 Web 的交互界面，旨在提供涵盖五个校园的食堂菜品信息查看、菜品评价等功能。

需求分析

作为大学生，食堂的菜品是我们一日三餐的选择之一。而由于每个校区存在多个食堂，每个食堂又存在许多窗口，因此对于许多人来说在众多菜品中做选择是一件困难的事情，既要考虑价格，又想避免踩坑。因此我们希望通过设计中大食评这样一个应用，帮助我们学校的学生和教职工，能够更好地了解菜品的信息，也为食堂提供了改进菜品和服务的方向和依据。因此我们认为该数据库应用应该具有如下的功能：

用户角色

1. **学生和教职工**：浏览菜品、搜索菜品、评价菜品、查看评价
2. **食堂管理员**：添加菜品、管理评论、管理菜品信息

功能需求

1. **用户管理**：用户注册功与登录功能，学生和管理员需要不同的登录权限
2. **菜品展示**：对各个菜品的详细信息进行显示，包括菜品所在校区、所在食堂、供应时段、一些评价信息。
3. **菜品搜索**：用户应该能够通过菜品名字进行模糊搜索，或者根据校区、食堂、供应时段等进行筛选。
4. **菜品评价**：用户应该能够添加自己对某个菜品的评价，包括评分信息和评价内容，方便其它用户根据菜品评分选择菜品。
5. **数据管理**：食堂管理人员应该能够对菜品信息进行增删、管理菜品信息。

数据库设计

【TODO: kjx】介绍每一个表（简要介绍与创建语言），E-R 图。要详细。

功能实现

开发平台和框架

- 数据库：OpenGauss，OpenEuler
- 前端：Bootstrap
- 后端：Django

前端网页设计

【TODO: xjw】网页跳转逻辑图，每个网页的简要介绍。

后端功能实现

我们的后端使用 Django 框架的 psycopg2 模块实现与 OpenGauss 的连接，并利用 Django 框架处理前端请求。Django 基于 MVC 模型，因此后端需要处理好 Model+View+Controller 的关系，对应到具体实现就是在 model.py 中按照 E-R 图定义数据库的 5 个实体，在 view.py 中处理前端请求，在 urls.py 定义好路由信息。这里主要介绍 view.py 中主要的函数，更多实现可见代码注释。

搜索功能

搜索界面支持基于菜名的模糊搜索和基于食堂、供应时间的组合搜索。
首先需要为前端提供可用来组合搜索的条件，直接返回食堂和供应时段的信息；

```
# 获取所有食堂、用餐时间，用于搜索界面的筛选
campus_list = Campus.objects.all()
cafeteria_list = Cafeteria.objects.all()
served_time_list = ServedTime.objects.all()
```

接着获取前端传过来的请求的内容：

```
if request.method == 'GET':
    query = request.GET.get('q', '')                        # 获取搜索的菜名
    campus = request.GET.getlist('campus', [])              # 获取筛选条件的校区
    cafeteria = request.GET.getlist('cafeteria', [])        # 获取筛选条件的食堂
    served_time = request.GET.getlist('served_time', [])    # 获取筛选条件的用餐时间
```

执行搜索时需要对搜索结果进行过滤：
模糊搜索使用 Django 内置的过滤方法：“dish_name__icontains” 表示基于指定字符串的模糊搜索。
组合搜索使用 Django 内置的过滤方法：“cafeteria_id__in” 表示存在一个条件满足所给列表，因为勾选的食堂可能不止一个。

```
# 构建查询条件
filters = {}
if query:                   # 基于菜名的模糊查询
    filters['dish_name__icontains'] = query
if len(cafeteria) > 0:      # 基于食堂的查询
    filters['cafeteria_id__in'] = cafeteria
    # served_time_list = ServedTime.objects.filter(cafeteria_id=cafeteria)
if len(served_time) > 0:    # 基于用餐时间的查询
    filters['served_time_id__in'] = served_time

# 执行查询
search_result = Dish.objects.filter(**filters)
```

最后返回前端信息：

```
return render(request, 'search.html', {'search_result': search_result, 'context':
context})
```

菜品信息展示与评论功能

菜品详情页需要菜品的各种信息。这里涉及到多个表的连接，以根据外键依赖获得对应的菜品所在食堂、供应时间的信息。
首先需要进行菜品表和食堂表、供应时间表的自然连接，获取相应菜品信息。

```
dish = Dish.objects.get(dish_id=dish_id)
dish_name = dish.dish_name
dish_price = dish.dish_price
dish_cafe = dish.cafeteria_id.cafeteria_name
dish_served_time = dish.served_time_id.served_time_period
comments = Comments.objects.filter(dish_id=dish_id)
```

如果使用了 POST 方法，说明用户添加了评论，此时需要在 Comments 表执行插入操作：

```
comment = Comments.objects.create(
    comments_id=next_id, score=score, content=content, dish_id=dish,
username_id=username)
comment.save()
```

登录注册功能

用户登录和注册的请求均为 POST 请求，从 POST 请求获取用户输入的用户名和密码：

```
username = request.POST.get('username')
passward = request.POST.get('password')
```

对于注册则根据用户名和密码尝试在用户表中添加用户，默认为普通用户。如果添加失败，如用户名已存在，则会返回错误代码，由前端解析。

```
try:
    # 创建用户
    user = User.objects.create_user(username=username, password=passward)
    user.save()
    if user:    # 如果注册成功，跳转到登录页面
        return_code = 1
        return redirect('log_in')
except IntegrityError:
    return_code = 2          # 用户名已存在
    return render(request, 'sign_up.html', {'return_code':return_code})
```

对于登录功能则验证输入的用户名和密码是否正确，正确则登录，跳转至主界面，否则返回错误代码，如用户不存在或者密码错误，由前端解析。

```
try:
    # 验证用户名、密码是否正确
    user = auth.authenticate(username=username, password=password)
    if user:
        auth.login(request, user)   # 正确则登录
        all_dish = Comments.objects.select_related('dish_id').all()
        return_code = 1
        return render(request, 'index.html', {'comment_data':
all_dish,'return_code':return_code})
    else:
        raise auth.PermissionDenied
except auth.PermissionDenied:       # 用户名不存在或密码错误
    return_code = 2
    return render(request, 'log_in.html', {'return_code':return_code})
```

功能展示

主界面——评论展示

登录注册界面

搜索界面

菜品详情页

管理界面

分工

姓名	学号	负责的工作
许志勇（组长）	21307169	后端功能实现
许嘉瑋	21307275	前端功能实现
邝佳轩	21307177	数据库设计