**数学与信息科学学院**

**课程设计报告**

**课程名称**： 大数据基础实践

**专业名称**： 数据科学与大数据技术

**班级名称**： 数信2341

**学号姓名：\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

学 期：2024 -- 2025 学年 第 一 学期

指导教师： 吴志霞

南京特殊教育师范学院

数学与信息科学学院课程设计报告

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **项目名称** | |  | | |
| **类 型** | | 口演示性 ■综合性  口验证性 口 设计性 | **同组成员** |  |
| **成 绩： 年 月 日** | | | | |
| **主要设备及器材** | |  | | |
| 项目功能概 述 | **（主要包括项目功能介绍、项目用例图等）**  1.功能介绍：  健康食谱推荐系统是一款基于用户个性化数据，提供健康食谱推荐的智能系统。系统帮助用户根据自身条件（身高、体重、性别、年龄、地区、是否可吃辣等）快速找到最适合的食谱，打造健康的生活方式。此外，系统提供 **注册** 和 **登录** 功能，保障用户数据安全，提升个性化服务体验。  2.功能结构设计图：    3.用例图： | | | |
| **详**  **细**  **设**  **计**  **阶**  **段**  **内**  **容** | 数据库设计  **1. 用户表（users）**  **功能**：存储用户的注册和登录信息。   |  |  |  |  | | --- | --- | --- | --- | | 字段名 | 数据类型 | 约束条件 | 描述 | | id | INT | PRIMARY KEY AUTO\_INCREMENT | 用户唯一标识 | | username | VARCHAR(50) | NOT NULL UNIQUE | 用户名 | | email | VARCHAR(100) | NOT NULL UNIQUE | 用户邮箱 | | password | VARCHAR(255) | NOT NULL | 加密后的用户密码 | | date\_created | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | 账号创建时间 |   （2）  **2. 食谱表（recipes）**  **功能**：存储所有健康食谱的信息，包括名称、描述、热量等。   |  |  |  |  | | --- | --- | --- | --- | | 字段名 | 数据类型 | 约束条件 | 描述 | | id | INT | PRIMARY KEY AUTO\_INCREMENT | 食谱唯一标识 | | name | VARCHAR(100) | NOT NULL | 食谱名称 | | description | TEXT | NULL | 食谱描述 | | calories | INT | NOT NULL | 每道菜的热量  (卡路里) | | suitable\_for | VARCHAR(20) | NOT NULL | 适合人群  （男、女、儿童、老人） | | region | VARCHAR(50) | NOT NULL | 地区  （华东、华南等） | | spicy | BOOLEAN | DEFAULT FALSE | 是否可以吃辣 | | date\_created | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | 数据插入时间 |   **3. 推荐记录表（recommendations）**  **功能**：记录用户每次获取健康食谱推荐的历史数据，便于分析和优化。   |  |  |  |  | | --- | --- | --- | --- | | 字段名 | 数据类型 | 约束条件 | 描述 | | id | INT | PRIMARY KEY AUTO\_INCREMENT | 推荐记录唯一标识 | | user\_id | INT | FOREIGN KEY | 关联用户表（users） | | recipe\_id | INT | FOREIGN KEY | 关联食谱表（recipes） | | height | INT | NULL | 用户身高 (cm) | | weight | INT | NULL | 用户体重 (kg) | | gender | VARCHAR(20) | NULL | 用户性别 | | age | VARCHAR(10) | NULL | 用户年龄范围 | | region | VARCHAR(50) | NULL | 用户所在地区 | | spicy | BOOLEAN | NULL | 用户是否可以吃辣 | | recommended\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | 推荐时间 |   数据表之间的关系图    -- 用户表 CREATE TABLE users (  id INT PRIMARY KEY AUTO\_INCREMENT,  username VARCHAR(50) NOT NULL UNIQUE,  email VARCHAR(100) NOT NULL UNIQUE,  password VARCHAR(255) NOT NULL,  date\_created TIMESTAMP DEFAULT *CURRENT\_TIMESTAMP* );  -- 食谱表 CREATE TABLE recipes (  id INT PRIMARY KEY AUTO\_INCREMENT,  name VARCHAR(100) NOT NULL,  description TEXT NULL,  calories INT NOT NULL,  suitable\_for VARCHAR(20) NOT NULL,  region VARCHAR(50) NOT NULL,  spicy BOOLEAN DEFAULT FALSE,  date\_created TIMESTAMP DEFAULT *CURRENT\_TIMESTAMP* );  -- 推荐记录表 CREATE TABLE recommendations (  id INT PRIMARY KEY AUTO\_INCREMENT,  user\_id INT,  recipe\_id INT,  height INT NULL,  weight INT NULL,  gender VARCHAR(20) NULL,  age VARCHAR(10) NULL,  region VARCHAR(50) NULL,  spicy BOOLEAN NULL,  recommended\_at TIMESTAMP DEFAULT *CURRENT\_TIMESTAMP*,  FOREIGN KEY (user\_id) REFERENCES users(id),  FOREIGN KEY (recipe\_id) REFERENCES recipes(id) ); | | | |
| **编**  **码**  **与**  **实**  **现** | **（主要提供****不少于三个核心模块的实现截图和核心代码或程序流程图 ）**   1. **用户登录**      1. **view.py**   from django.shortcuts import render, redirect from django.contrib.auth.hashers import make\_password, check\_password from .models import User  # 注册视图 from django.shortcuts import render, redirect from django.contrib.auth.hashers import make\_password from .models import User  def register(request):  error = None  if request.method == 'POST':  username = request.POST['username']  email = request.POST['email']  password = make\_password(request.POST['password'])   # 检查用户名或邮箱是否已存在  if User.objects.filter(username=username).exists():  error = "用户名已存在！"  elif User.objects.filter(email=email).exists():  error = "邮箱已被注册！"  else:  User.objects.create(username=username, email=email, password=password)  return redirect('login') # 注册成功后跳转到登录页面   return render(request, 'accounts/register.html', {'error': error})   # 登录视图 def login(request):  error = None  if request.method == 'POST':  username = request.POST['username']  password = request.POST['password']   user = User.objects.filter(username=username).first()  if user and check\_password(password, user.password):  request.session['user\_id'] = user.id  return redirect('home') # 登录成功跳转到首页  else:  error = "用户名或密码错误！"   return render(request, 'accounts/login.html', {'error': error})   1. **url.py**   from django.urls import path from . import views  urlpatterns = [  path('login/', views.login, name='login'),  path('register/', views.register, name='register'), ]   1. **login.html**     <!DOCTYPE html> <html lang="en"> <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>健康食谱推荐系统 - 登录</title>  <!-- 添加外部字体 -->  <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;700&display=swap" rel="stylesheet">  <style>  /\* 全局样式 \*/  \* {  margin: 0;  padding: 0;  box-sizing: border-box;  }   body {  font-family: 'Roboto', sans-serif;  background: linear-gradient(135deg, #87CEEB, #ADD8E6); /\* 渐变背景 \*/  display: flex;  justify-content: center;  align-items: center;  height: 100vh;  }   .container {  width: 100%;  max-width: 400px;  background: #ffffff;  padding: 30px;  border-radius: 10px;  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2);  text-align: center;  animation: fadeIn 1s ease-in-out;  }   /\* 系统标题 \*/  .title {  font-size: 28px;  font-weight: bold;  color: #333;  margin-bottom: 10px;  }   /\* 系统介绍 \*/  .intro {  font-size: 14px;  color: #666;  margin-bottom: 20px;  line-height: 1.6;  }   /\* 输入框和按钮样式 \*/  form label {  display: block;  text-align: left;  margin: 10px 0 5px;  color: #444;  }   form input {  width: 100%;  padding: 8px;  border: 1px solid #ccc;  border-radius: 5px;  font-size: 14px;  transition: border-color 0.3s;  }   form input:focus {  border-color: #87CEEB;  outline: none;  }   button {  width: 100%;  margin-top: 20px;  padding: 10px;  background: #87CEEB;  color: #fff;  border: none;  border-radius: 5px;  font-size: 16px;  cursor: pointer;  transition: background 0.3s, transform 0.1s;  }   button:hover {  background: #5FB3D6;  transform: scale(1.05);  }   button:active {  transform: scale(0.95);  }   /\* 注册链接 \*/  .register-link {  display: block;  margin-top: 15px;  font-size: 13px;  color: #333;  }   .register-link a {  color: #87CEEB;  text-decoration: none;  font-weight: bold;  }   .register-link a:hover {  text-decoration: underline;  }   /\* 错误提示 \*/  .error-message {  color: red;  font-size: 14px;  margin-top: 10px;  }   /\* 动画效果 \*/  @keyframes fadeIn {  from {  opacity: 0;  transform: translateY(-20px);  }  to {  opacity: 1;  transform: translateY(0);  }  }  </style> </head> <body>  <div class="container">  <!-- 标题 -->  <div class="title">欢迎来到健康食谱推荐系统</div>  <!-- 系统介绍 -->  <div class="intro">  本系统致力于为您提供个性化的健康食谱推荐服务。根据您的身高、体重、性别和地区，  精准推荐适合您的健康食谱，帮助您享受健康的生活方式。  </div>  <!-- 登录表单 -->  <form method="POST">  {% csrf\_token %}  <label>用户名：</label>  <input type="text" name="username" placeholder="请输入用户名" required>  <label>密码：</label>  <input type="password" name="password" placeholder="请输入密码" required>  <button type="submit">登录</button>  <!-- 注册链接 -->  <div class="register-link">  没有账号？ <a href="{% url 'register' %}">去注册</a>  </div>  <!-- 登录错误提示 -->  {% if error %}  <div class="error-message">{{ error }}</div>  {% endif %}  </form>  </div> </body> </html>   1. **个人信息填写**     **2.1views.py**  from django.shortcuts import render from .models import Recipe  def recommend(request):  if request.method == 'POST':  height = request.POST['height']  weight = request.POST['weight']  gender = request.POST['gender']  age = request.POST['age']  region = request.POST['region']  spicy = request.POST['spicy']   # 基于用户输入的条件查询推荐食谱  recipes = Recipe.objects.filter(suitable\_for=gender, region=region)   # 如果用户不能吃辣，则进一步筛选（假设 description 包含 '辣' 关键词）  if spicy == "否":  recipes = recipes.exclude(description\_\_icontains='辣')   # 将结果返回前端  return render(request, 'recipes/recommend.html', {'recipes': recipes})   return render(request, 'recipes/input.html')   1. **urls.py**   from django.urls import path from . import views  urlpatterns = [  path('', views.recommend, name='home'), ]   1. **models.py**   from django.db import models  class Recipe(models.Model):  name = models.CharField(max\_length=200)  description = models.TextField()  calories = models.IntegerField()  suitable\_for = models.CharField(max\_length=50)  region = models.CharField(max\_length=50)   def \_\_str\_\_(self):  return self.name   1. **input.html**   <!DOCTYPE html> <html lang="en"> <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>健康食谱推荐系统 - 输入个人信息</title>  <!-- 导入 Google 字体 -->  <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;700&display=swap" rel="stylesheet">  <style>  /\* 全局样式 \*/  \* {  margin: 0;  padding: 0;  box-sizing: border-box;  }   body {  font-family: 'Roboto', sans-serif;  background: linear-gradient(135deg, #FFDEE9, #B5FFFC); /\* 粉蓝渐变背景 \*/  display: flex;  justify-content: center;  align-items: center;  flex-direction: column;  min-height: 100vh;  }   h1 {  font-size: 36px;  color: #333;  text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.2);  margin-bottom: 20px;  text-align: center;  }   .intro {  font-size: 16px;  color: #555;  margin-bottom: 30px;  text-align: center;  line-height: 1.6;  max-width: 600px;  }   .form-container {  background: #ffffff;  border-radius: 10px;  box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);  padding: 30px;  width: 100%;  max-width: 500px;  animation: slideIn 1s ease-in-out;  }   form label {  display: block;  margin: 10px 0 5px;  font-size: 14px;  color: #333;  font-weight: bold;  }   form input, form select {  width: 100%;  padding: 8px;  margin-bottom: 15px;  border: 1px solid #ccc;  border-radius: 5px;  font-size: 14px;  transition: border-color 0.3s;  }   form input:focus, form select:focus {  border-color: #FF7EB3;  outline: none;  }   button {  width: 100%;  padding: 10px;  background: #FF7EB3;  color: #ffffff;  border: none;  border-radius: 5px;  font-size: 16px;  font-weight: bold;  cursor: pointer;  transition: background 0.3s, transform 0.1s;  }   button:hover {  background: #FF6699;  transform: scale(1.05);  }   button:active {  transform: scale(0.95);  }   /\* 动画效果 \*/  @keyframes slideIn {  from {  opacity: 0;  transform: translateY(-20px);  }  to {  opacity: 1;  transform: translateY(0);  }  }  </style> </head> <body>  <!-- 大标题 -->  <h1>健康食谱推荐系统</h1>  <!-- 简介 -->  <div class="intro">  欢迎来到健康食谱推荐系统！我们将根据您的个人信息，包括身高、体重、年龄和饮食偏好，  为您量身推荐健康食谱，帮助您享受美味又健康的生活方式。  </div>  <!-- 表单容器 -->  <div class="form-container">  <h2 style="text-align: center; margin-bottom: 20px; color: #444;">输入个人信息</h2>  <form method="POST">  {% csrf\_token %}   <label>身高 (cm)：</label>  <input type="number" name="height" placeholder="请输入您的身高" required>   <label>体重 (kg)：</label>  <input type="number" name="weight" placeholder="请输入您的体重" required>   <label>性别：</label>  <select name="gender" required>  <option value="" selected disabled>请选择性别</option>  <option value="男">男</option>  <option value="女">女</option>  <option value="儿童">儿童</option>  <option value="老人">老人</option>  </select>   <label>年龄：</label>  <select name="age" required>  <option value="" selected disabled>请选择年龄范围</option>  <option value="0-12">0-12岁</option>  <option value="13-18">13-18岁</option>  <option value="19-35">19-35岁</option>  <option value="36-50">36-50岁</option>  <option value="51+">51岁以上</option>  </select>   <label>地区：</label>  <select name="region" required>  <option value="" selected disabled>请选择地区</option>  <option value="华东">华东</option>  <option value="华南">华南</option>  <option value="华北">华北</option>  <option value="西南">西南</option>  <option value="西北">西北</option>  </select>   <label>是否可以吃辣：</label>  <select name="spicy" required>  <option value="" selected disabled>请选择</option>  <option value="是">是</option>  <option value="否">否</option>  </select>   <button type="submit">推荐食谱</button>  </form>  </div> </body> </html>  **3.推荐结果展示页面：**     1. **models.py**   from django.db import models  class Recipe(models.Model):  name = models.CharField(max\_length=200)  description = models.TextField()  calories = models.IntegerField()  suitable\_for = models.CharField(max\_length=50)  region = models.CharField(max\_length=50)   def \_\_str\_\_(self):  return self.name   1. **views.py**   from django.shortcuts import render from .models import Recipe  def recommend(request):  if request.method == 'POST':  height = request.POST['height']  weight = request.POST['weight']  gender = request.POST['gender']  age = request.POST['age']  region = request.POST['region']  spicy = request.POST['spicy']   # 基于用户输入的条件查询推荐食谱  recipes = Recipe.objects.filter(suitable\_for=gender, region=region)   # 如果用户不能吃辣，则进一步筛选（假设 description 包含 '辣' 关键词）  if spicy == "否":  recipes = recipes.exclude(description\_\_icontains='辣')   # 将结果返回前端  return render(request, 'recipes/recommend.html', {'recipes': recipes})   return render(request, 'recipes/input.html')   1. **urls.py**   from django.urls import path from . import views  urlpatterns = [  path('', views.recommend, name='home'), ]   1. **recommand.html**   <!DOCTYPE html> <html lang="en"> <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>推荐的健康食谱</title>  <!-- 添加CSS样式 -->  <style>  /\* 页面整体样式 \*/  body {  font-family: Arial, sans-serif;  margin: 0;  padding: 0;  background: linear-gradient(to bottom, #ffecd2, #fcb69f); /\* 渐变背景 \*/  color: #333;  text-align: center;  }   h2 {  margin-top: 20px;  font-size: 36px;  color: #444;  text-shadow: 1px 1px 3px rgba(0, 0, 0, 0.1);  }   ul {  list-style: none;  padding: 0;  margin: 20px auto;  max-width: 600px;  }   li {  background: #ffffff;  border: 1px solid #ddd;  margin: 10px 0;  padding: 15px;  border-radius: 8px;  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);  font-size: 16px;  line-height: 1.6;  transition: transform 0.3s ease, box-shadow 0.3s ease;  }   li:hover {  transform: scale(1.02);  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);  }   strong {  color: #f77a52;  font-size: 18px;  }   p {  margin-top: 20px;  font-size: 18px;  color: #555;  }   /\* 动画效果 \*/  @keyframes fadeIn {  from {  opacity: 0;  transform: translateY(-20px);  }  to {  opacity: 1;  transform: translateY(0);  }  }   ul li {  animation: fadeIn 0.8s ease forwards;  }  </style> </head> <body>  <!-- 页面标题 -->  <h2>推荐的健康食谱</h2>   {% if recipes %}  <!-- 食谱列表 -->  <ul>  {% for recipe in recipes %}  <li>  <strong>{{ recipe.name }}</strong> - {{ recipe.description }}，  热量: {{ recipe.calories }} 卡路里  </li>  {% endfor %}  </ul>  {% else %}  <!-- 无数据提示 -->  <p>抱歉，没有找到符合条件的食谱！请修改筛选条件。</p>  {% endif %} </body> </html> | | | |
| **小结（分析遇到问题、作品创新亮点、总结）** | **小结**  **1. 遇到的问题与解决方法**  在开发健康食谱推荐系统的过程中，我遇到了不少挑战，但通过逐步分析和解决，这些问题成为了我宝贵的学习经验。  **（1）数据库设计与外键关系的理解**   * **问题**：在设计数据库表时，最初对于外键关系的实现理解不够深入，导致无法准确地建立表之间的联系。 * **解决方法**：通过学习数据库的一对多关系的概念，结合 Django ORM 外键（ForeignKey）的使用，将 recommendations 表中的 user\_id 和 recipe\_id 分别与 users 表和 recipes 表建立关联，实现了数据的完整性与可扩展性。   **（2）前端页面美化与交互优化**   * **问题**：最初的页面设计较为简单，布局不够美观，交互性也较弱，难以吸引用户。 * **解决方法**：   + 通过学习 HTML 和 CSS，使用渐变背景、卡片式布局、按钮悬停特效等技术优化页面的视觉效果。   + 添加动画效果（如淡入、悬浮放大）增强页面的交互体验，提升系统的整体用户友好度。   **（3）数据获取与推荐逻辑实现**   * **问题**：如何根据用户输入的个人信息（身高、体重、性别、年龄、地区等）智能筛选健康食谱是一个核心难点。 * **解决方法**：   + 在后端实现筛选逻辑，使用 Django 的 ORM 对数据库进行查询和过滤。   + 针对用户的个性化需求（例如是否能吃辣），通过条件筛选来推荐最符合用户需求的食谱。   + 为验证推荐功能的准确性，手动插入多条模拟食谱数据，测试筛选效果。   **（4）用户注册与登录功能**   * **问题**：实现用户注册和登录时，密码的安全性和账户验证是必须解决的问题。 * **解决方法**：   + 使用 Django 自带的用户认证模块，通过密码加密（make\_password）和验证机制确保数据安全。   + 提供友好的错误提示（如用户名已存在、密码错误等），提高系统的用户体验。   **2. 作品的创新亮点**  在开发健康食谱推荐系统时，我融入了一些创新思考，使作品更具实用性与吸引力：  **（1）个性化食谱推荐**   * 基于用户输入的 **身高、体重、性别、年龄、地区、是否吃辣** 等多维度信息，系统智能筛选健康食谱，提供精准化推荐。 * 用户可以根据自身需求修改筛选条件，获得更符合个人口味的食谱。   **（2）现代化用户界面设计**   * 使用了 **卡片式布局**、**动态悬浮效果** 和 **渐变背景** 等前端设计元素，打造简洁美观的页面。 * 添加了动画效果（如淡入、悬停放大），增强了用户的视觉体验和交互感。   **（3）用户友好的交互流程**   * 注册与登录功能完善，提供详细的错误提示，保证数据安全和操作的流畅性。 * 当没有符合条件的食谱时，系统会友好提示用户调整筛选条件，提升系统的容错性和用户体验。   **（4）可扩展的数据管理**   * 通过 **推荐记录表** 记录每次推荐的历史数据，便于后续分析用户偏好，进一步优化推荐算法。 * 数据库设计清晰，关系合理，支持后续食谱数据的扩展与管理。   **3. 总结与收获**  通过本次健康食谱推荐系统的开发，我在多个方面取得了成长与收获：  **（1）技术能力的提升**   * 熟练掌握了 **Django 框架** 进行后端开发，学会了如何通过 ORM 实现数据库查询和操作。 * 熟悉了 **HTML、CSS** 基础前端技术，能够独立美化页面布局，提升用户体验。 * 学会了数据库设计与管理，理解了表之间的 **一对多关系** 和外键的实际应用。   **（2）解决问题的能力**   * 在开发过程中，我遇到了诸多技术难题，如数据库设计、数据筛选逻辑和页面优化等，但通过不断查找资料、调试代码和请教同学，我逐步解决了这些问题，积累了宝贵的开发经验。   **（3）系统思维与团队合作**   * 开发过程中，我不断完善系统功能，优化代码逻辑，提升系统的整体结构和可维护性。 * 如果团队协作开发，我也学会了如何分工协作、协调代码的版本管理。   **（4）实践与理论的结合**   * 将课堂所学的 **Web 开发技术、数据库管理知识** 应用到实际项目中，加深了对理论知识的理解。 * 开发过程中暴露的问题让我明白了实际项目与理论学习的差距，促使我更加注重细节与规范。 | | | |