东软教育科技集团

项目实训总结报告

项 目 名 称:“识食物者”---美食识别系统

学 校 名 称: 内蒙古大学

项 目 讲 师: 王志勇

姓 名: 王蕊

学 号: 0191120230

实 训 日 期: 2022年7月11日-8月6日

**1 引言** （一级标题，小三黑体，居中）

1.1项目概述（二级标题，四号黑体，左对齐）

项目背景：我们生活的二十一世纪，智能信息化的时代已经到来，人工智能慢慢从以前的遥不可及，变成现在的近在身边，各种服务类型的人工智能产业兴起。机器生产相比人工作业产量更稳定、效率更高并且能节省成本，机器取代人工的趋势愈加明显。人工智能如今已经成为了一个炙手可热的研究方向，而使用它的本质就是为人类服务。

项目目的：本项目应用技术属于人工智能中的神经网络分支，在图像识别，语音识别等领域都已经取得了很多成就，项目基于深度学习图像识别，主要利用深度卷积神经网络图像处理技术进行菜品的识别,结合前端，后端，数据库，制作出一个可以代替人眼实现食物识别的系统，能够随时随地，告知我们眼前的或者网络浏览的美食叫什么名字。

项目意义：就像我们经常有想听的歌曲但是不知道歌曲名字就会进行听歌识曲一样，利用深度学习训练模型从而成功识别食物名称是非常有必要的，在我们想不起来一种曾经吃过的东西的名称或者因为食物不常食用而不了解时，实时迅速地解答我们的疑惑。如果还有机会填充项目的话，我认为识别完食物之后，将识别食物的名称、热量等信息记录下来，然后统计用户的每日摄入，这样就可以作为一个减肥监控系统，辅助减肥意愿强烈的人士更加科学健康地减肥，或者可以通过用户识别的食物，为用户提供附近相似美食店铺的分享。但由于实训时间比较紧凑，还没有实践到这一步。简而言之，这个美食实时识别系统是非常有意义的。

参考资料：在实训过程中，除了老师上课时讲课的ppt、文档以及发给我们的样例以外，主要参考了几本书目：

[1] Flask Web 开发实战 ：入门、进阶与原理解析／李辉著 ．—北京：机械工业出版社，2018.8 (Web 开发技术丛书）

[2] Flask Web开发 : 基于Python的Web应用开发实战 / (美) 格林布戈 (Grinberg,M.) 著 ; 安道译. -- 北京人民邮电出版社, 2015.1 （图灵程序设计丛书）

[3] Python深度学习/(美)弗朗索瓦·肖莱 著 张亮(hysic) 译 人民邮电出版社出版时间：2018-08（图灵程序设计丛书）

说明项目目的、背景、意义及参考资料（包括企业提供的素材、DEMO）等；

1.2 需求分析 *(课程目标支撑4 ; 毕业要求支撑10.2)*

人们都知道，民以食为天，美食与我们的生活息息相关，尤其是在当下，随着祖国发展的越来越好，我们的生活条件得到了极大的改善，物质生活变得非常丰富。现在的我们想要品尝美食，也许上午刚刚下单，下午就可以品尝了，面对越来越五花八门的美食，当我们想不起眼前的食物叫什么名字时，需要食物识别，人们也可以通过简单拍照更好的了解他们不熟悉或以前从未见过的食物，并了解其细节，例如烹饪方法、食材、口味和营养成分等。系统具有良好的可扩展性，能够继续对功能进行扩展，从而创造出更高的经济效益和现实价值。现代人越来越追求健康的生活方式，当减肥人士或者特殊疾病患者需要分析食物的营养价值、热量、糖分等信息，也需要用到食物识别，本系统在健身塑形，饮食健康等领域有很大的发展空间。无人看守自助取餐需要识别食物来计算价格，从而节省人力。所以我们所编写的“识食物者”---美食实时识别系统，应用市场是非常广阔的，它在智慧健康、食品智能装备、智慧餐饮、智能零售等方面有着广泛的应用前景。

**功能性需求**

用户上传美食图片，系统将结果返回给用户，允许用户在电脑、平板、手机端正常使用，允许用户注册并登录，通过加密手段保护用户信息，允许用户进行用户反馈来进一步调整正确率，用户注册页面数据要实时验证，用户提交数据用数据库存储，用户登录采用session技术存储。用户上传食物图片后，能够实时得到图片识别结果。

**非功能性需求**

1、界面需求：

1)页面内容:排版美观简洁，布局合理，页面风格统一，富有美食主题特色。

2)导航栏:导航指明确示，易于理解，方便用户使用。

3)系统兼容：系统在web端、移动端、平板都可以正常运行。

2、实时性需求：

识别速度尽量快，最长不超过10s。系统应该尽量及时的进行反应，实时监测出各种非正常情况，进行响应。

3、开放性需求

系统应具有灵活性，以适应将来功能扩展的需求。

4、可扩展性需求

系统设计要求能够体现扩展性要求，以适应将来功能扩展的需求。

为什么做这个题目，有什么实际价值

1.3 运行环境

|  |  |
| --- | --- |
| 硬件环境 | 软件环境 |
| CPU:Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz (8 CPUs), ~2.0GHz  内存：8192MB RAM | 操作系统：Windows 10专业版64位  应用软件：Pycharm+python3.7.9  数据库：sqlite  开发工具：flask+Bootstrap框架 |

系统在web端、移动端、平板都可以正常运行。

软、硬件环境

（正文，小四号宋体，1.25倍行距）

**2 项目设计**（小三黑体，居中）*(课程目标支撑3 ; 毕业要求支撑9.2)*。

（以个人主要负责内容为主）

2.1 设计思路

在本次项目中，我主要负责通过flask框架搭建项目网站，主要负责界面以及登录、注册功能。实训讲课过程中，老师为我们介绍了Flask框架，Flask诞生于2010年，是一个用python语言基于Werkzeug工具箱编写的轻量级web开发框架，它主要面向需求简单，项目周期短的小应用。Flask可以说是最灵活的框架之一，比较适合我们的美食实时识别系统，所以后端选择Flask框架来编写。后端注册登录模块，设计采用装饰器以及Flask-WTF表单进行编写。在登陆功能模块中，要规范用户名以及密码的格式，如果不符合要求，要进行提示；没有账号选择注册，输入用户名和两次密码，以确认两次输入相同，还要输入电话号码以及邮箱，用户输入注册信息正确并注册成功后，系统将跳到登录页面令用户登录。

因为要创建主要在移动端运行的下拉式响应网页，所以界面方面采用Flask-Bootstrap。Bootstrap 是用于开发响应迅速、移动优先网站的受欢迎的集HTML、CSS和JS于一体的框架，适用于我们的美食实时识别系统。利用 Jinja2 的模板继承机制，base.html通过继承一个基模板： bootstrap/base.html，来使用 Bootstrap。其它页面通过继承base.html，沿用它的内容，并在这个基础上进行修改，即可完成新的页面，加快开发时间。并采用css，在head中连接相关需要文件，自己编写css文件，调整界面样式，使界面得到更好的优化。前端界面主要设计开始页、登录页面、注册页面、识别页面，为用户提供交互。

在编写页面时，用户首先进入开始页，已有账号可以选择登录，如果用户密码准确，进入识别页面，上传图片进行识别，识别后可以进行用户反馈，如果没有账号，可以进入注册页面注册，如果用户名或密码不符合规范会出现提醒，或者直接以游客身份进入美食识别页面，上传图片进行识别。登录和注册的前后端代码老师已经交给了我们，但是需要美化页面，设计添加css文件以及js文件修改界面样式，增加美观度。

2.2 模块功能介绍

前端界面模块：

连接后端显示信息，通过美观的、响应式良好的界面，为用户提供交互。

登录模块：

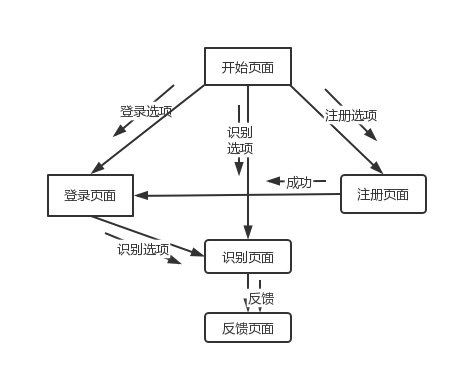
为用户提供登录功能，为前端操作提供操作响应以及跳转逻辑，对用户在前端输入的数据判断：先看用户信息是否存在，若不存在返回错误信息，若存在，看用户名与密码是否匹配，不匹配也返回错误信息，匹配则登陆成功。登录之后跳转到上传图片页面。

注册模块：

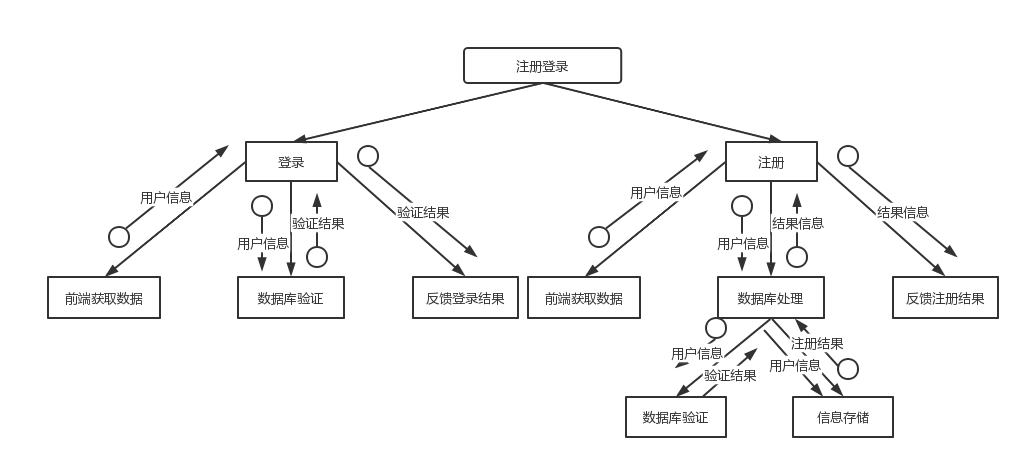
为用户提供注册功能，为前端操作提供操作响应以及跳转逻辑，对用户在前端输入的数据判断：先看看用户名、密码、邮箱、电话是否符合输入规范，不符合则返回错误信息，再用户信息是否存在，若存在返回用户已存在，若不存在， 将用户信息记录进数据库。

2.3 模块结构图

前端界面模块：



在界面模块中，在开始主页中，用户可以选择注册、登录或识别页面。如果选择登录选项，即可进入登录页面，登陆成功之后再选择识别按钮，即可进入识页面，拍照上传或者并且在识别结束后进行用户反馈。没有账号可以选择先进行注册，然后会跳到登录界面，再进行识别以及用户反馈。

注册登录模块：2.4 程序流程图

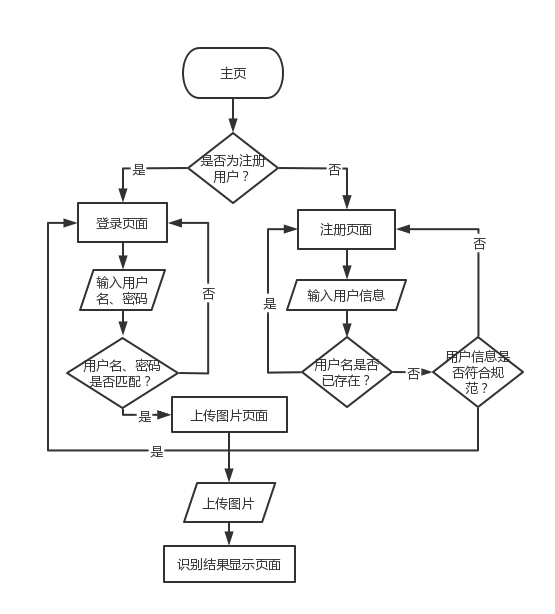


图2.4.1界面流程图

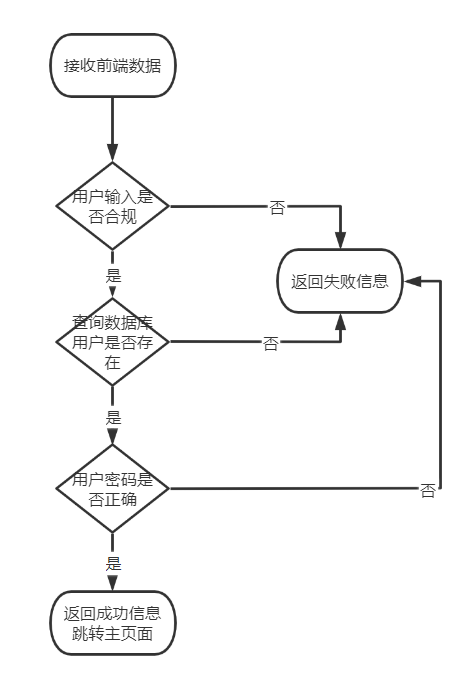
**

图2.4.2 登录流程图

为用户提供登录功能，对用户输入的用户名。密码、邮箱号进行判断：先看用户名、密码、邮箱、电话是否符合输入规范，用户名、密码都应该为3-16位，不符合规范，就返回错误信息，再进入数据库查询，先看用户信息是否存在，若不存在返回错误信息，若用户存在，看用户名与密码是否匹配，不匹配也返回错误信息，匹配则登陆成功。登录之后跳转到上传图片页面。

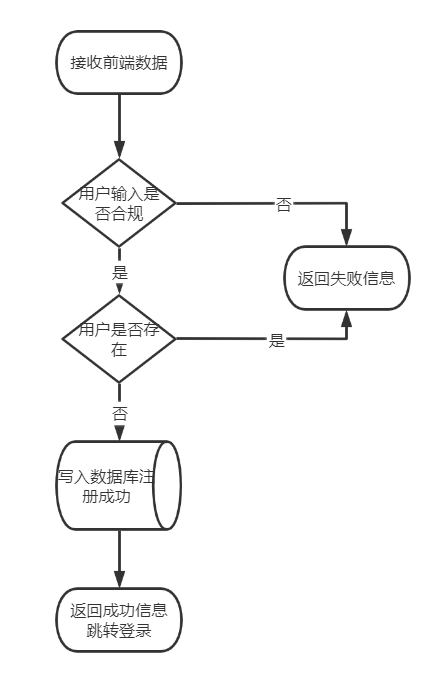
**

图2.4.3 注册流程图

为用户提供注册功能，对用户输入的用户名。密码、邮箱号进行判断：先看用户名、密码、邮箱、电话是否符合输入规范，用户名、密码都应该为3-16位，不符合规范，就返回错误信息，再看用户信息是否存在，若存在返回用户已存在，若不存在，就记录信息入数据库，然后跳转到登录页面。

2.5 功能设计分工（团队分工）

需详细标注每部分的内容，具体任务及任务安排等。

|  |  |  |
| --- | --- | --- |
| **序号** | **任务名称** | **工时** |
| 1 | 设计登录注册识别页面Ui | 0.5天 |
| 2 | 添加注册登录界面内部的函数 | 0.5天 |
| 3 | 前端业务逻辑函数 | 0.5天 |
| 4 | 添加主页内部控件的跳转 | 0.25天 |
| 5 | 连接数据库，并实现存储删除 | 0.25天 |
| 6 | 整理合并大部分代码 | 0.5天 |
| 7 | 调整修改 | 0.5天 |

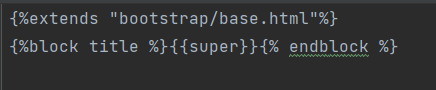
**3 详细设计**（小三黑体，居中）*(课程目标支撑3 ; 毕业要求支撑9.2)*

**系统设计：**

1. **、前端（所涉及技术以及核心功能）**

前端界面连接后端显示信息给用户，通过美观的、响应式良好的界面，为用户提供交互。前端设计部分，采用Bootstrap框架，它是用于开发响应迅速、移动优先网站的受欢迎的集HTML、CSS和JS于一体的优秀框架，比较适用于本次开发的主要在移动端运行的下拉式响应网页——美食实时识别系统。

在实训中使用Bootstrap模板，使用一个名为Flask-Bootstrap的Flask扩展，在程序中集成Bootstrap，简化集成过程。利用 Jinja2 的模板继承机制，base.htm通过extends指令继承基模板： bootstrap/base.html，即可使用 Bootstrap。Flask-Bootstrap 中的基模板提供了一个网页框架，引入了Bootstrap中的所有css和JavaScript文件，其它页面继承base.html,加快开发并统一界面风格。



使用Bootstrap模板，能够简化前端开发的复杂度，除了base.html，还需要编写主页（index.html） 、登录页面（ login.html ）、注册页面（register.html）、上传图片页面（ upload.html）以及显示页面（ show.html），为了风格统一，这些页面都继承base.html，加速开发时间。为了使界面更加美观，设计Init.css、 Login.css、 Register.css、 upload.css、 Show.css文件来修改页面表单样式、背景图片、文本框样式、组件边距等内容。

1. **、注册登录后端（所涉及技术以及核心功能）**

后端采用Flask框架，客户端(例如Web浏览器)把请求发送给Web服务器，Web服务器再把请求发送给Flask程序实例。程序实例需要知道对每个URL请求运行哪些代码，所以保存了一个URL到Python函数的映射关系。处理URL和函数之间关系的程序称为路由。在Flask程序中定义路由的最简便方式，是使用程序实例提供的app.route修饰器，把修饰的函数注册为路由。把index()函数注册为程序根地址的处理程序。如果部署程序的服务器域名为www.baidu.com，在浏览器中访问htp://ww.baidu.com后，会触发服务器执行index()函数。这个函数的返回值称为响应，是客户端接收到的内容。如果客户端是Web浏览器，响应就是显示给用户查看的文档。像index()这样的函数称为视图函数。视图函数返回的响应可以是包含HTML的简单字符串，也可以是复杂的表单，我们创建login\_view视图函数和register视图函数，进行视图响应，并创建注册登录快速表单类，前端表单提供信息。

有时会发现这样的问题：用户输入名字后提交表单，然后点击浏览器的刷新按钮，会看到一个莫名其妙的警告，要求在再次提交表单之前进行确认。之所以出现这种情况，是因为刷新页面时浏览器会重新发送之前已经发送过的最后一个请求。如果这个请求是一个包含表单数据的 POST 请求，刷新页面后会再次提交表单。大多数情况下，这并不是理想的处理方式。我们最好别让 Web 程序把 POST 请求作为浏览器发送的最后一个请求。所以在视图函数里，采用重定向去解决这个问题。使用重定向作为 POST 请求的响应，而不是使用常规响应。重定 向是一种特殊的响应，响应内容是 URL，而不是包含 HTML 代码的字符串。浏览器收到 这种响应时，会向重定向的 URL 发起 GET 请求，显示页面的内容。这个页面的加载可能要多花几微秒，因为要先把第二个请求发给服务器。除此之外，用户不会察觉到有什么不同。现在，最后一个请求是 GET 请求，所以刷新命令能像预期的那样正常使用了，这个技术就是重定向，调用redirect函数。

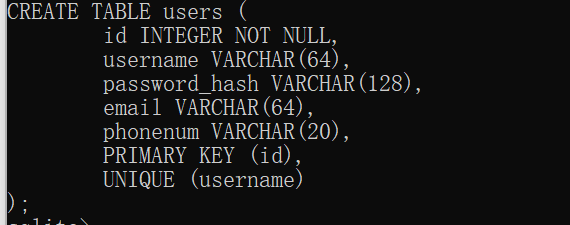
登录注册存储信息时采用session。session和cookie的作用有点类似，都是为了存储用户相关的信息。不同的是，cookie是存储在本地浏览器，而session存储在服务器。存储在服务器的数据会更加的安全，不容易被窃取。但存储在服务器也有一定的弊端，就是会占用服务器的资源，但现在服务器已经发展至今，一些session信息还是绰绰有余的。使用session，敏感数据不是直接发送回给浏览器，而是发送回一个session\_id，服务器将session\_id和敏感数据做一个映射存储在session(在服务器上面)中，更加安全。

我们在数据库中存储用户信息时，若想保证数据库中用户密码的安全，关键在于不能存储密码本身，而要存储密码的散列值。计算密码散列值的函数接收密码作为输入，使用一种或多种加密算法转换密码，最终得到一个和原始密码没有关系的字符序列。核对密码时，密码散列值可代替原始密码，因为计算散列值的函数是可复现的：只要输入一样，结果就一样。设计使用Werkzeug实现密码散列，其中：generate\_password\_hash()函数将原始密码作为输入，以字符串形式输出密码的散列值，输出的值可保存在用户数据库中。 check\_password\_hash(hash, password)函数用于登录时验证密码，它的参数是从数据库中取回的密码散列值和用户输入的密码，返回值为 True 则密码正确。

用户通过前端进行注册或登录请求完成后，有时需要让用户知道状态发生了变化。这里可以使用确认消息、警告或者错误提醒。可以用flash() 函数实现这种效果，仅调用 flash() 函数并不能把消息显示出来，还需要程序使用的模板要渲染这些消息。

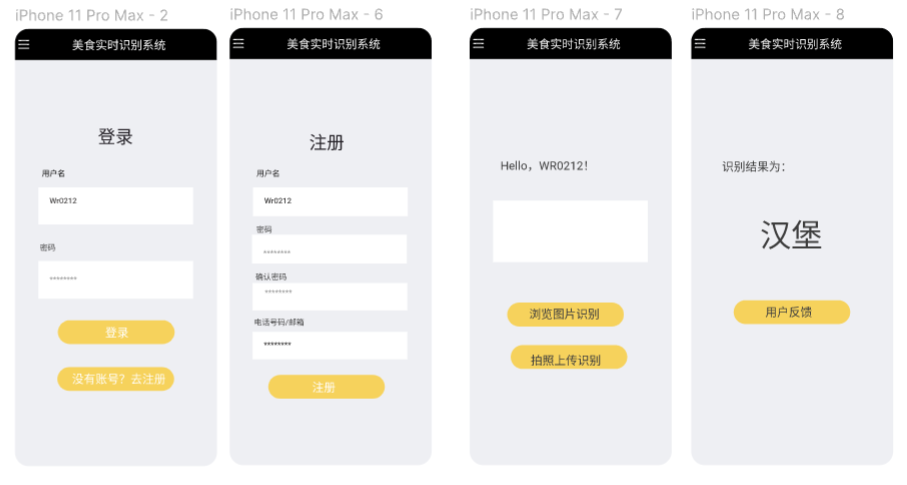
**数据库设计：**

使用Flask-SQLAlchemy框架，它是一个 Flask 扩展，简化了在 Flask 程序中使用 SQLAlchemy 的操作。 SQLAlchemy 是一个很强大的关系型数据库框架，支持多种数据库后台。SQLAlchemy 提供了高层 ORM，也提供了使用数据库原生 SQL 的低层功能。其中ORM的好处：可以让我们操作数据库跟操作对象是一样的，非常方便。因为一个表就抽象成一个类，一条数据就抽象成该类的一个对象。数据库采用不需要服务器的SQLite数据库，数据库设计如下：



其中用户id设置为主键，并且不允许用户名重复。创建User类，继承db.model，create\_ db.py创建数据库，add db\_ data.py向数据库中添加内容。

**页面设计（原型图绘制）：**



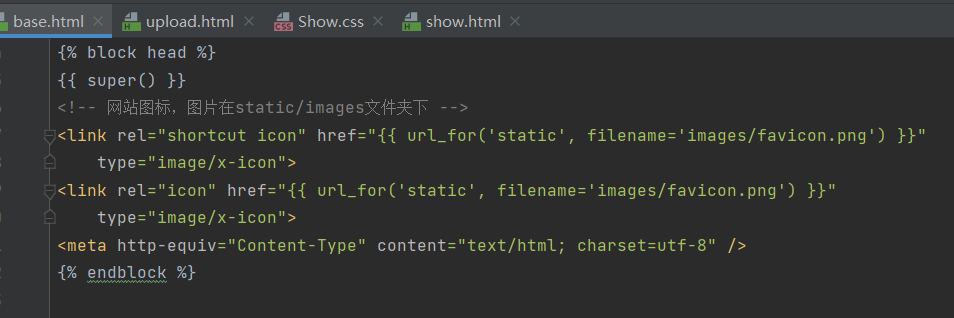
详细写明项目的设计（系统设计、数据库设计、页面等）等，并写明对所涉技术、核心功能等描述；

**4 实现方案**（小三黑体，居中）*(课程目标支撑3 ; 毕业要求支撑9.2)*

1. 、前端页面

利用 Jinja2 的模板继承机制，base.html通过继承一个基模板： bootstrap/base.html，即可使用 Bootstrap。base.html基础页面中定义导航条与图标，可以通过继承被其它页面沿用。如果程序需要向己经有内容的块中添加新内容，必须使用Jinja2提供的super(）函数。

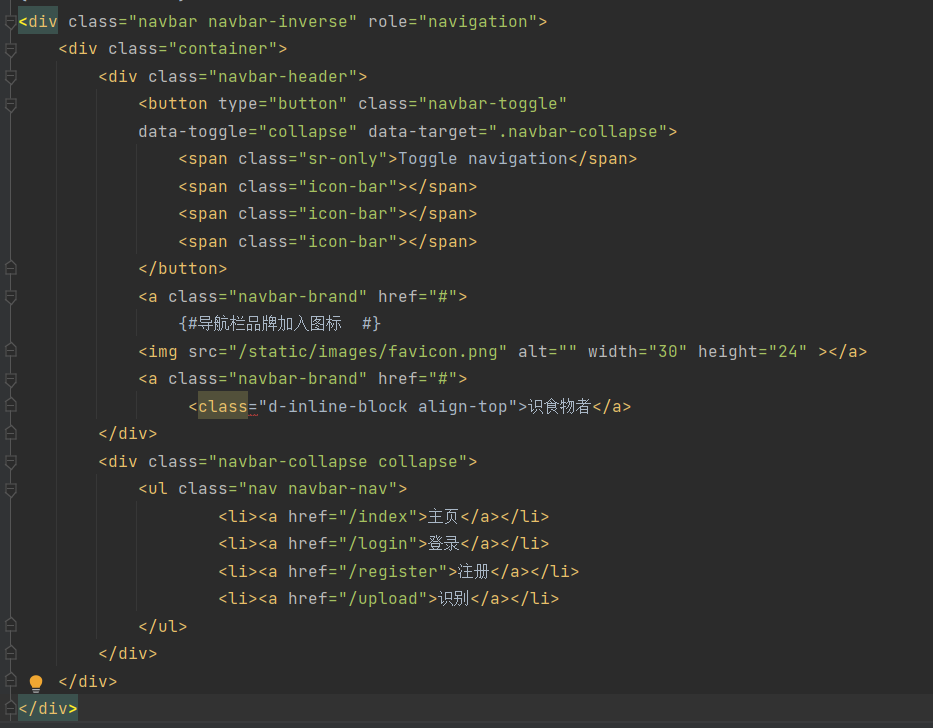
super（）中是添加了汉堡图标：



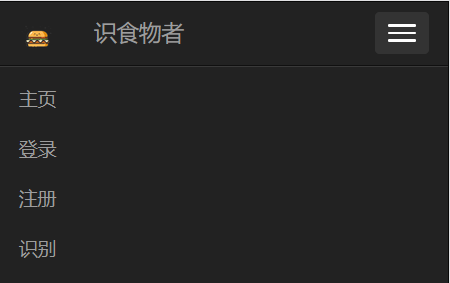
实现结果：



定义导航条：



实现结果：

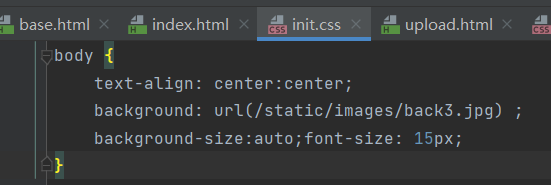


其它页面通过继承base.html，沿用它的内容，可以重新定义block模块中的内容，就可以在基础页面上进行修改，完成新的页面，并且还可以加快开发时间。

其中主页 index.html，设置主页文案：



Init.css,为主页添加背景图片以及设置背景图片铺满方式:



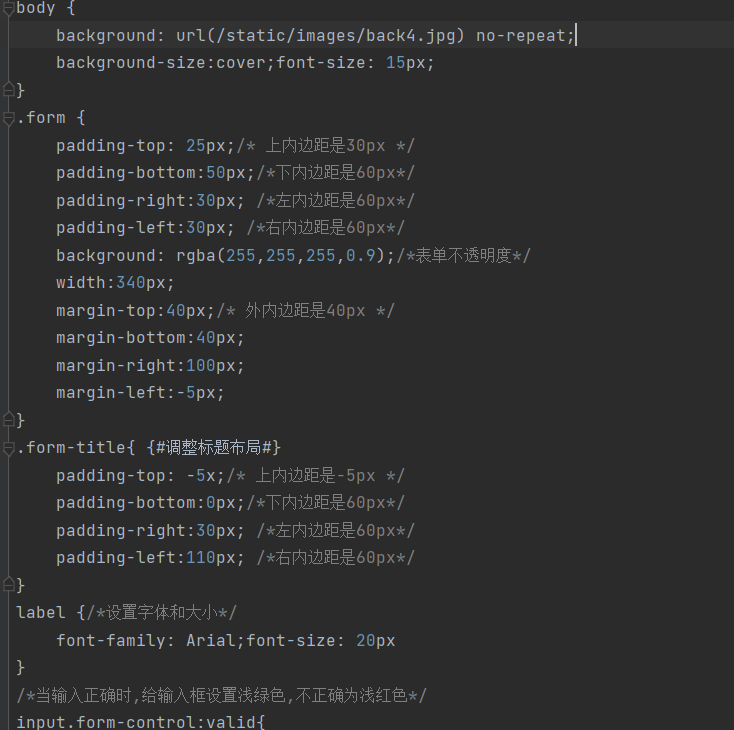
实现结果：

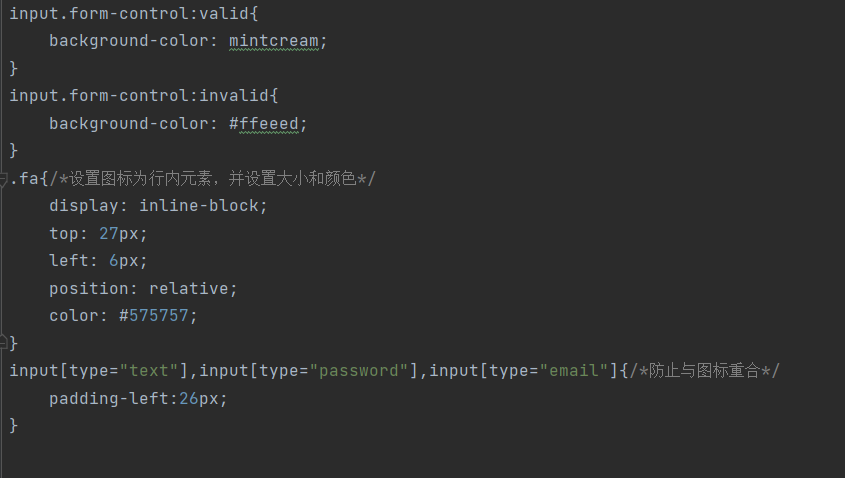


注册页面 register.html,提供注册页面显示，接收用户信息传给后端:

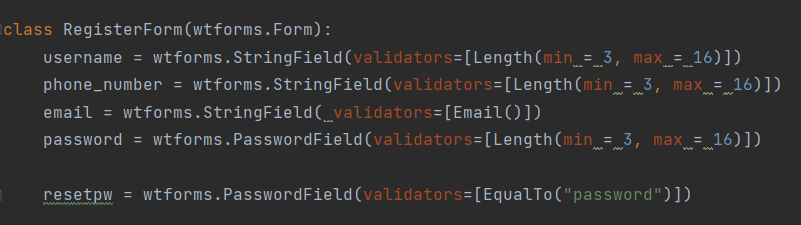


Register.css，添加背景图片，设置表单透明度，增加fafa图标，设置文字框颜色。修改组件边距：

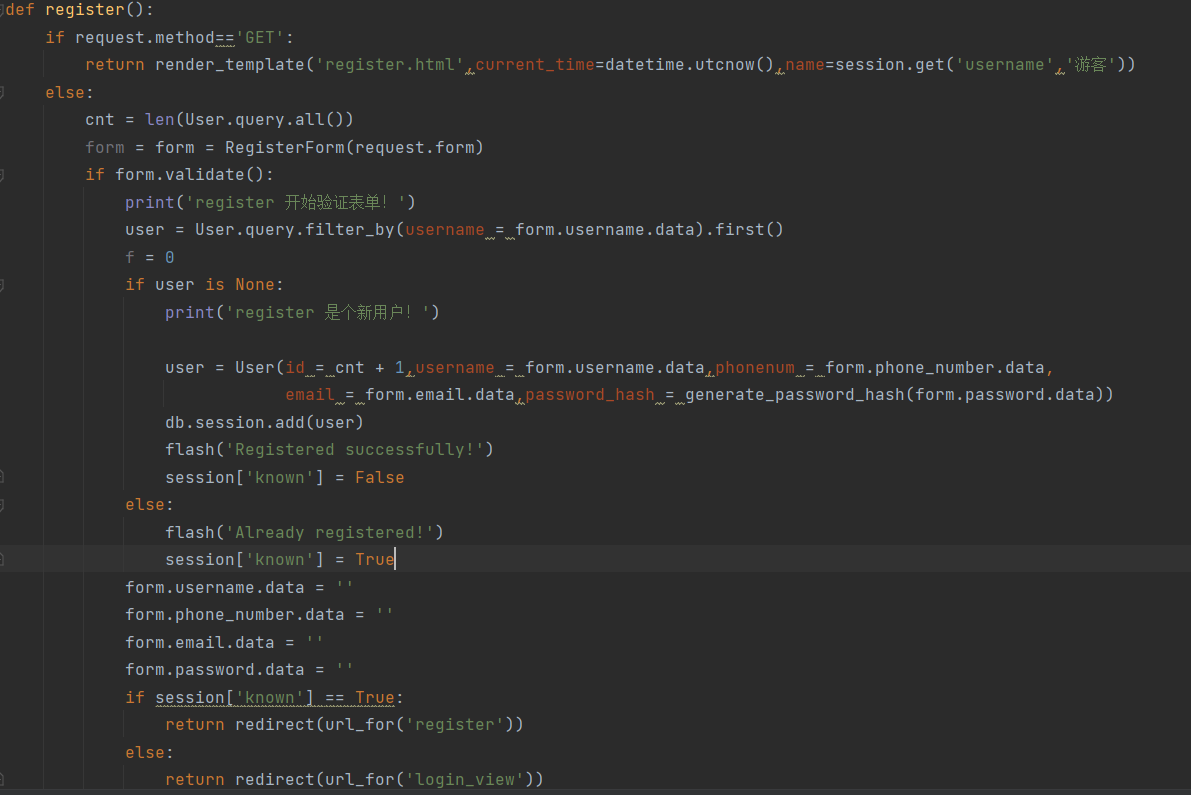




注册表单类，用到了 WTForms 提供的 Length()、Email() 验证函数，对用户信息进行规定。



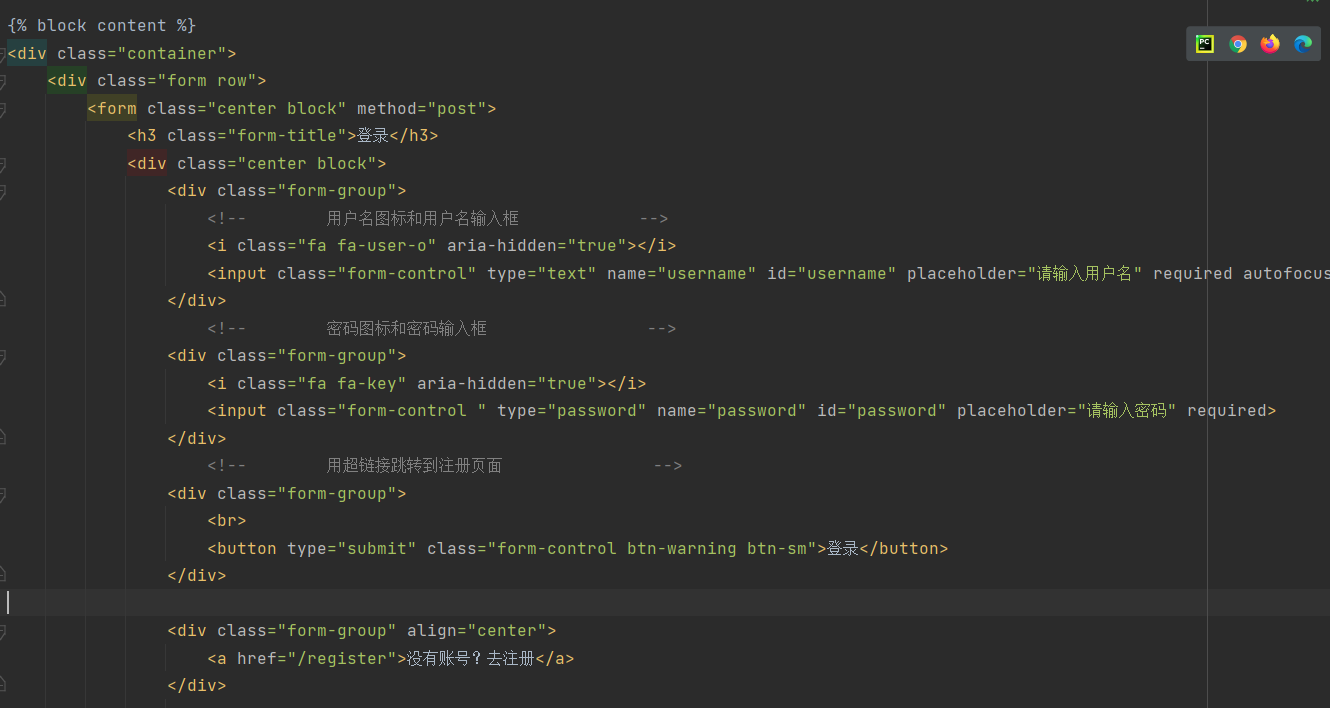
注册模块视图函数，对用户输入信息进行验证。视图函数创建了一个RegisterForm 对象，当请求类型是 GET 时，视图函数直接渲染模板，即显示表单。当表单在 POST 请求中提交时， Flask-WTF 中的 验证函数会验证表单数据，验证不通过则显示用户信息格式错误！验证通过后尝试注册用户，首先加载数据，用户id为当前所有+1。为了注册用户，视图函数首先使用表单中填写的username从数据库中加载用户。如果对应的用户不存在，将注册信息存入数据库，跳转至登录页面，如果对应的用户名存在，显示用户名已被注册！提交登录密令的 POST 请求最后也做了重定向，程序会设定Flash 消息，对用户进行各种提示，再次渲染表单，让用户重试登录。安全起见，密码要输入两次。此时要验证两个密码字段中的值是否一致，这种验证可使用 ，WTForms 提供的另一验证函数实现，即 EqualTo。这个验证函数要附属到两个密码字段中的一个上，另一个字段则作为参数传入。



实现结果：

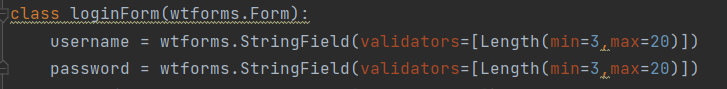


登录页面 login.html,提供注册页面显示，接收用户名与密码传给后端:

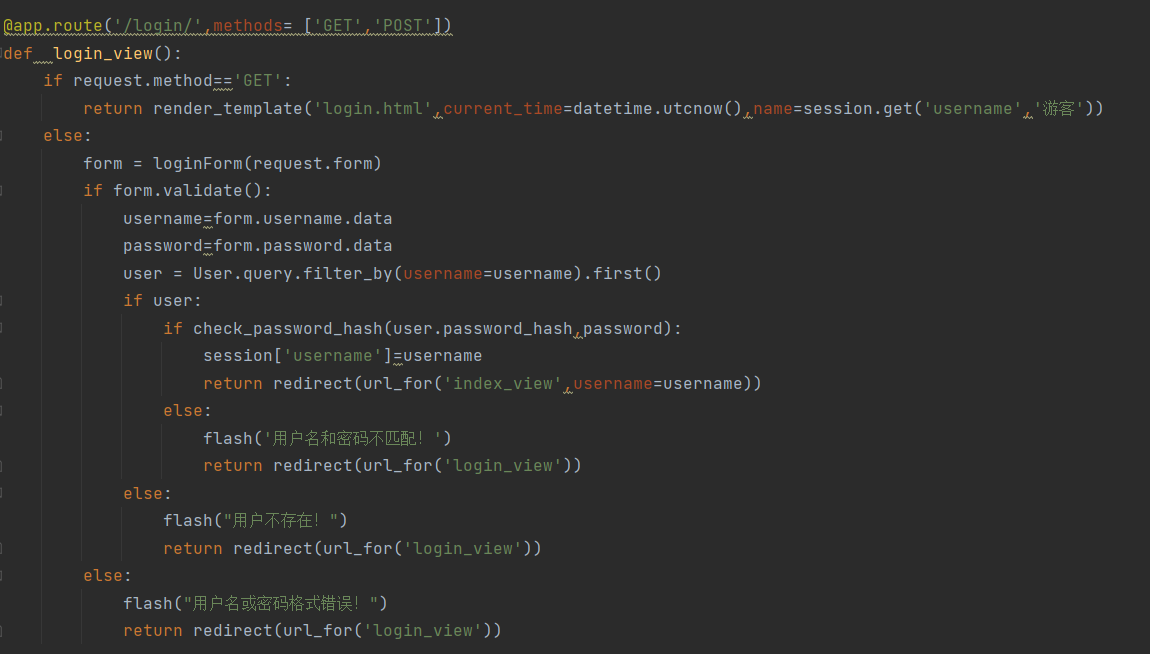


Login.css，添加背景图片，设置表单透明度，增加fafa图标，设置文字框颜色。修改组件边距，注册登录页面风格一致，所以二者的css文件相似。

登录表单类，用到了 WTForms 提供的 Length()Email() 验证函数，呈现给用户的登录表单中包含一个用于用户名字段、一个密码字段、一个提交按钮。



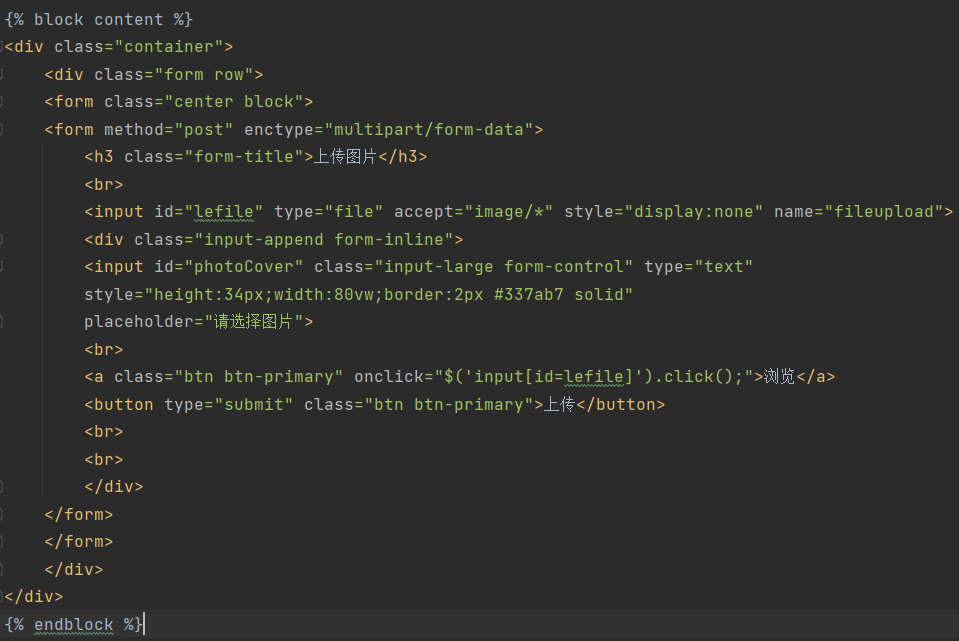
登录视图函数：这个视图函数创建了一个 loginForm 对象，当请求类型是 GET 时，视图函数直接渲染模板，即显示表单。当表单在 POST请求中提交时， Flask-WTF 中的 validate\_on\_submit() 函数会验证表单数据，验证不通过则显示用户名或密码格式错误！验证通过后尝试登入用户。为了登入用户，视图函数首先使用表单中填写的username从数据库中加载用户。如果对应的用户存在，再调用用户对象的 check\_password\_hash(user.password\_hash,password)方法，其参数是表单中填 写的用户名、密码。如果密码正确，则登录成功，用户名保存在session中，转到上传图片界面，否则显示用户名和密码不匹配！如果对应的用户不存在，显示用户不存在！提交登录密令的 POST 请求最后也做了重定向，程序会设定Flash 消息，对用户进行各种提示，再次渲染表单，让用户重试登录。



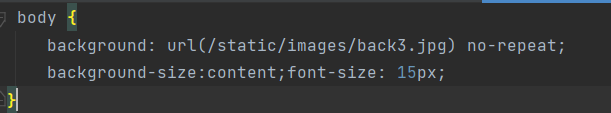
实现结果：



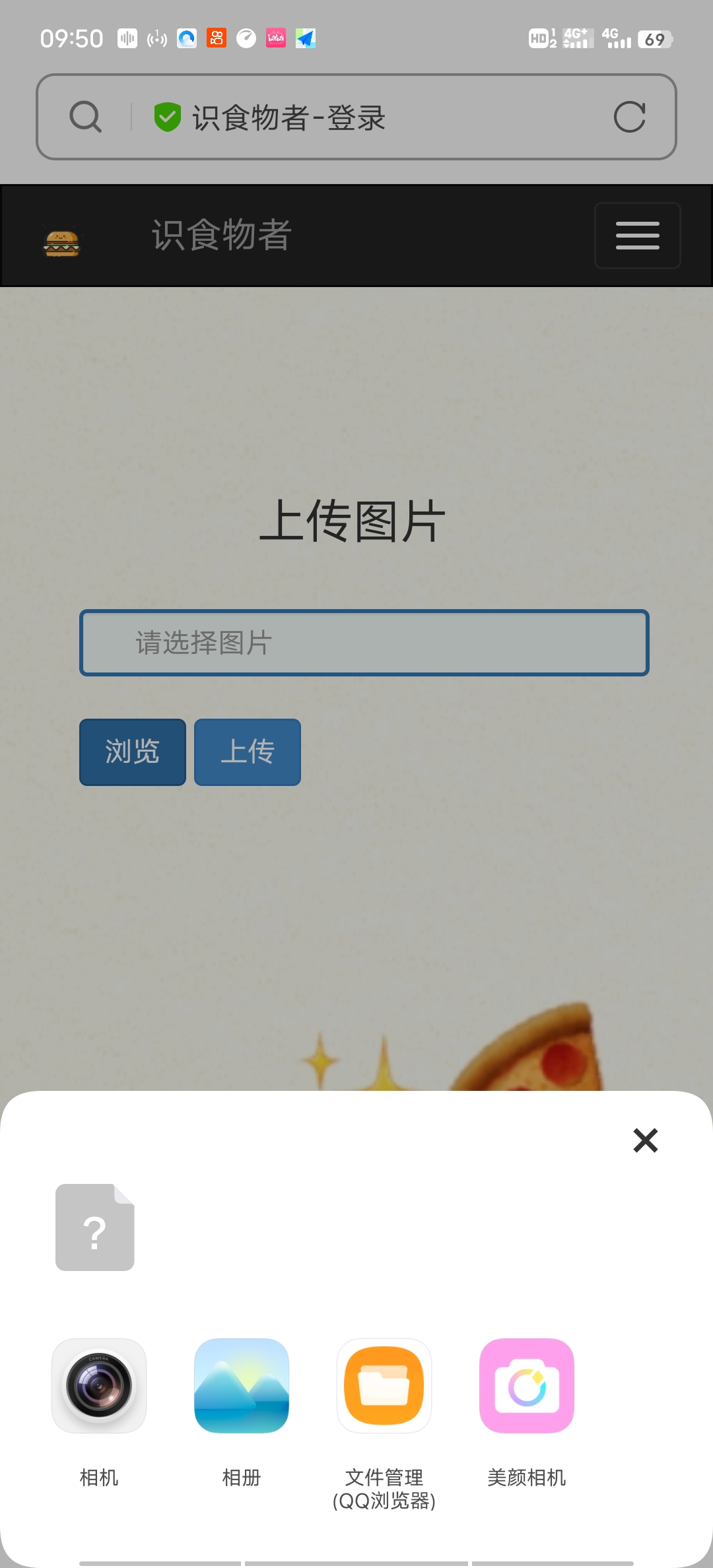
上传页面：upload.html,向用户提供上传图片功能



upload.css:为html页面提供美化，添加背景以及设定背景铺满方式



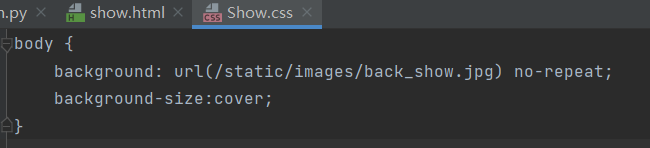
实现结果，当在手机端运行网页时，可以选择照片拍摄或者直接上传已有文件：

结果显示页面，show.html,显示图片识别结果，并提供快速表单，使用户反馈用户意见，及时收集系统反馈：



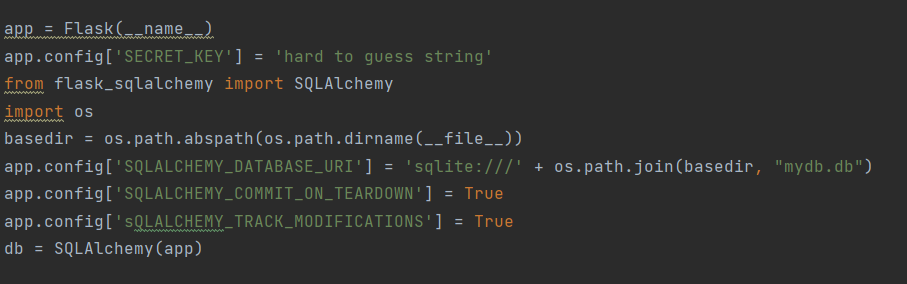
Show.css:为html页面提供美化，添加背景以及设定背景铺满方式



实现结果：



配置数据库，db对象是 SQLAlchemy 类的实例，表示程序使用的数据库，同时还获得了 Flask-SQLAlchemy 提供的所有功能：



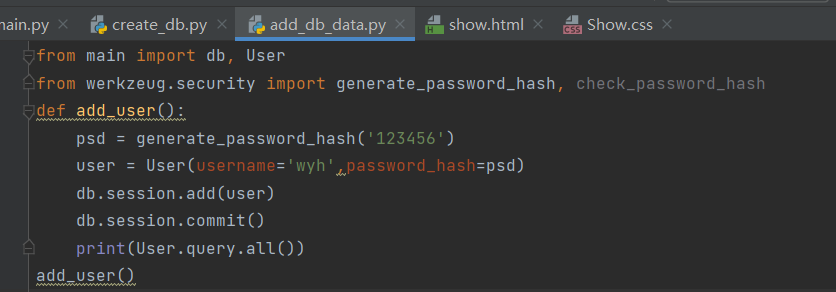
定义User模型，继承db.model，类变量 \_\_tablename\_\_ 定义在数据库中使用的表名。如果没有定义 \_\_tablename\_\_，Flask- SQLAlchemy 会使用一个默认名字，但默认的表名没有遵守使用复数形式进行命名的约定，所以我们自己来自定义表名为users。其余的类变量都是该模型的属性。 generate\_password\_hash(password)：这个函数将原始密码作为输入，以字符串形式输出密码的散列值，输出的值可保存在用户数据库中。 check\_password\_hash(hash, password)函数的参数是从数据库中取回的密码散列值和用户输入的密码，返回值为 True 表明密码正确：



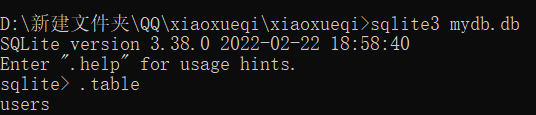
创建表：



增加用户：

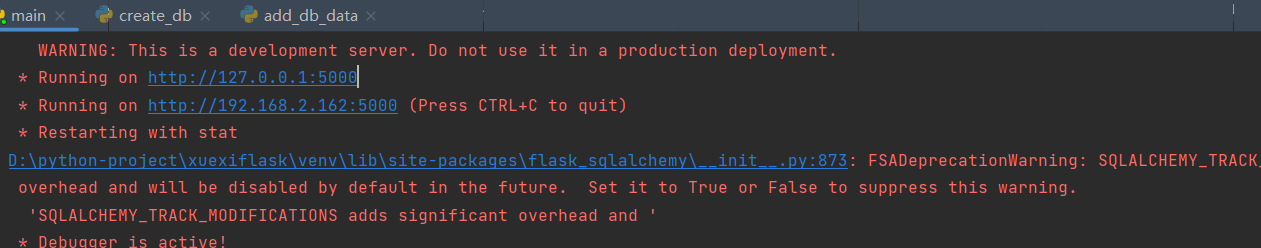


实现结果：



系统截图：

从运行程序开始：



网站图标;



主页，显示系统名称，欢迎用户使用：

接着点击注册，输入用户信息进行注册：

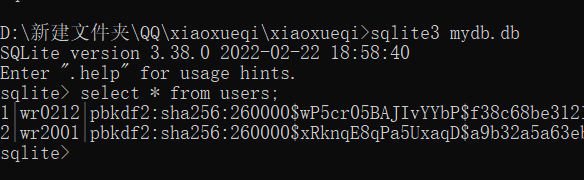


当输入数据符合系统规定时，文本框显示为浅绿色，不符合或未输入时为浅红色：

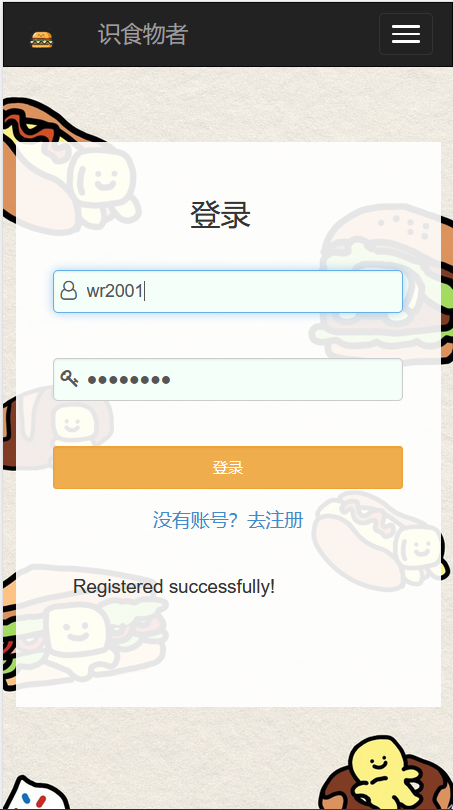
 

注册成功后，数据库内部显示用户的注册信息已被存储：





点击注册，显示注册成功并跳转到登录页面：



若用户名或密码错误，点击登录，显示错误信息，跳转回登录页面：

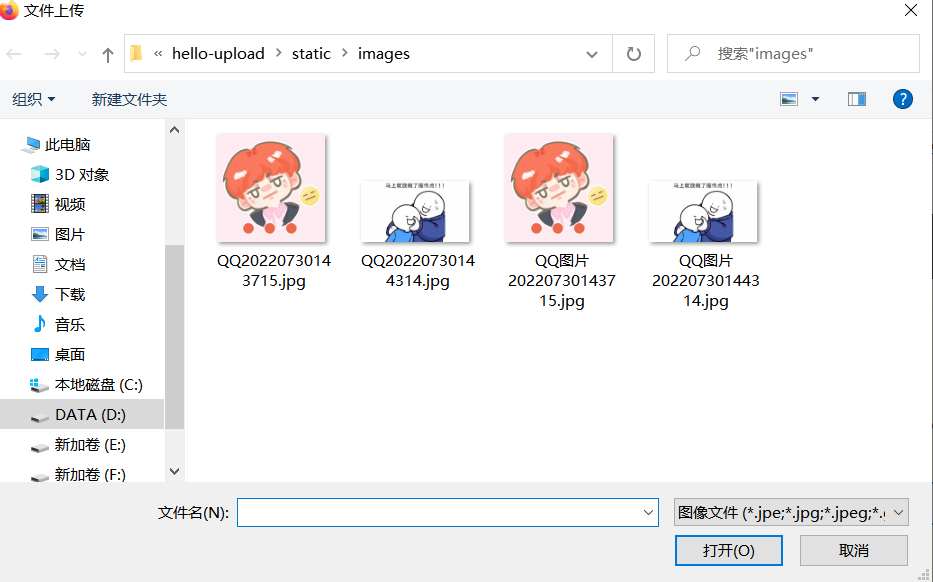


若用户名及密码正确，点击登录，跳转到上传页面：

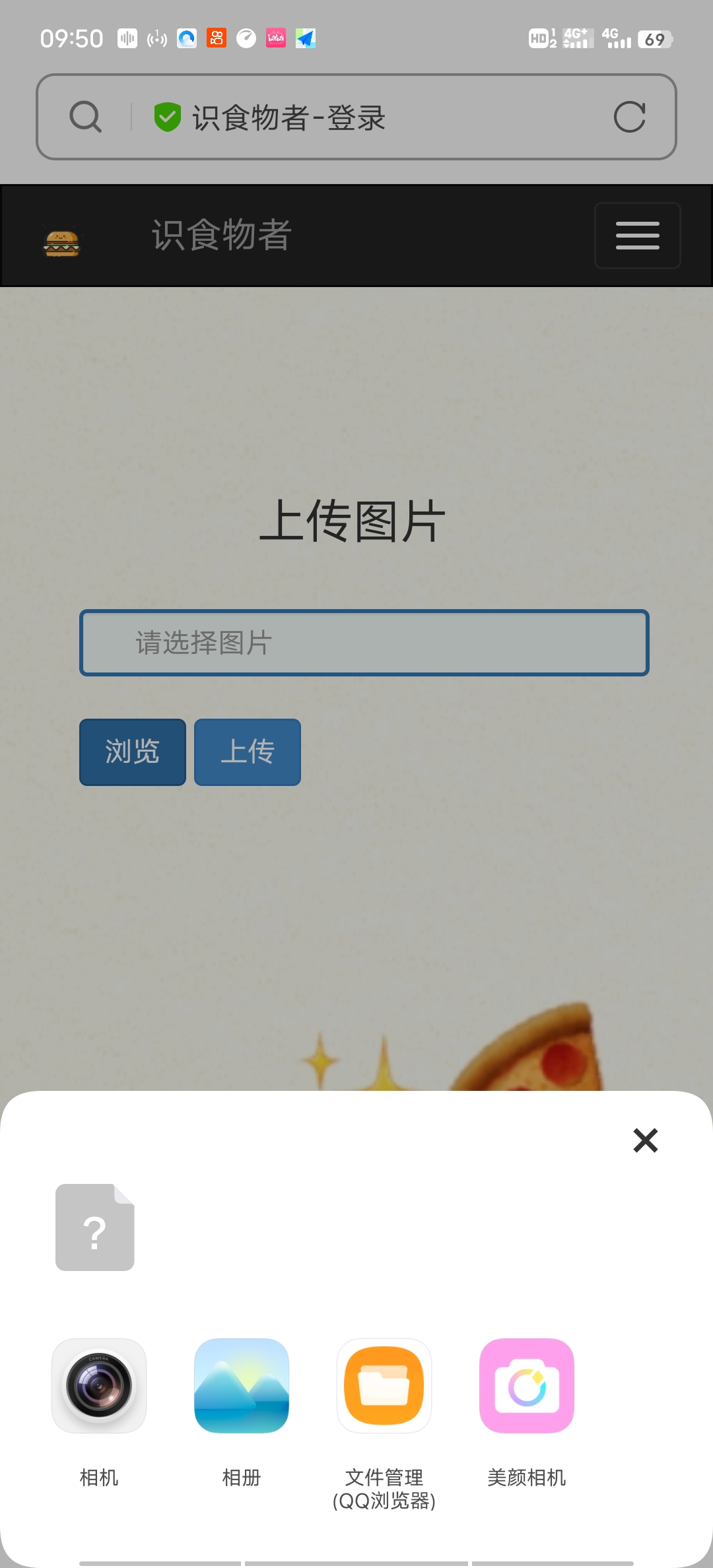


点击浏览:

电脑端弹出为：



手机端运行为：



点击上传，系统进行识别并跳转到识别显示页面，可以对结果进行用户反馈：



详细写明项目功能实现描述、核心代码及实现结果等；

并对调试好的程序，从执行程序弹出界面开始，每一步操作截一个图，并附加说明（图片格式为JPG格式）。

**5 系统测试**（小三黑体，居中）*(课程目标支撑3 ; 毕业要求支撑9.2)*

测试方法：

对系统进行功能测试，也就是我们常说的黑盒测试，它是通过测试来检测每个功能是否都能正常使用。在测试中，把程序看作一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，在程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息。黑盒测试着眼于程序外部结构，不考虑内部逻辑结构，主要针对软件界面和软件功能进行测试。

预期结果为：

|  |  |
| --- | --- |
| 系统功能 | 预期成果 |
| 注册功能 | 系统能够对用户输入信息进行验证，判断是否符合规范，不符合规范则向用户反馈信息；后台能在数据库中查询用户是否存在，不存在则存入数据库；存在则反馈给用户。 |
| 登录功能 | 系统能够对用户输入信息进行验证，判断是否符合规范，不符合规范则向用户反馈信息；后台能在数据库中查询用户是否存在，存在则判断用户名与密码是否匹配；不存在则反馈给用户。 |
| 页面显示 | 页面显示简洁美观，无论在web端还是移动端都显示正常，具有响应式。 |

测试结果：

对于“识食物者美食识别系统”，运行了多次，进行反复输入与显示，测试报告为：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试功能 | 测试次数 | 通过次数 | 失败次数 | 是否达到预期 |
| 注册功能 | 10 | 10 | 0 | 是 |
| 登录功能 | 10 | 10 | 0 | 是 |
| 页面显示 | 10 | 7 | 3 | 否 |

在进行测试的过程中，注册、登录功能以及数据库显示都非常正常，但是一开始导航栏品牌的汉堡图标与文字一上一下，不在同一条水平线上，点击导航栏的主页，页面会一片空白，注册登录的页面显示在电脑上居中且美观。但是在移动端，表单不居中且有部分无法显示：

1. bug：导航栏品牌的汉堡图标与文字一上一下，不在同一条水平线上：

修正方法：

在inline-block 后面添加 align: top; 让图标与文字基于同一个baseline 值对齐。



2、主页显示一片空白：

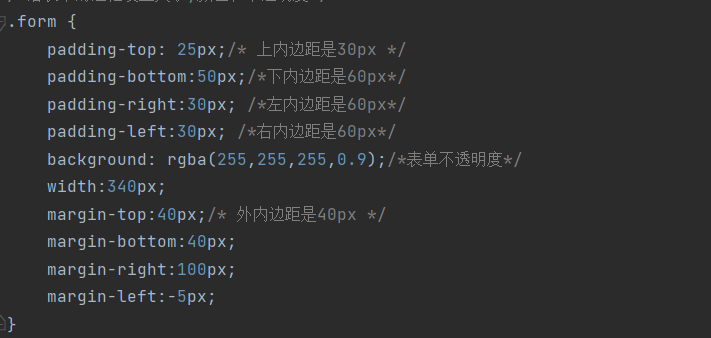


修正方法：这个问题非常容易解决，只是在视图函数中，将主页的html错写成了base.html，改正后就正常显示了。

1. 在移动端表单显示异常：

修正方法：不断尝试调整表单的内外边距，经过半天的尝试，解决了这个问题。



最终得到了在web端与移动端都能正常显示的比较美观的注册登录页面。

详细写明测试方法、测试描述（bug与修正方法等）、预期结果等；

**6 前沿分析**（小三黑体，居中）*(课程目标支撑6 ; 毕业要求支撑12.2)*

食物图像识别的研究最早可以追溯到1977年，Parrish等人最先开展基于视觉的果蔬识别研究，应用于水果采摘。2008年，Kitamura等人提出了多媒体饮食日志系统该系统通过菜品图像和非菜品图像的分类得到用户的菜品图像，然后对其进行成分和热量分析以提供饮食建议.2009年，Chen等人发布了第一个快餐食品数据集PFID，之后于2010年提出了-种联合视觉词袋模型和颜色直方图的特征提取方法在该数据集上进行了测试...... 随着深度学习技术的快速发展和大规模食品图像数据集的不断增多，食物图像识别的相关研究逐渐多了起来。

这些年来，食品图像识别技术的发展十分迅速、

研究人员不断钻研发展食物图像识别技术，是因为它有着广阔的应用前景。

系统通过自动识别食品类别，可进一步实现后续的食物分析与热量估算，并且可以有针对性地给用户提出意见。在自助餐厅中，食品图像识别可以实现自动计费.在生活中，遇到不熟悉或以前从未见过的食物，可以通过食物识别知道食物名称，并了解其细节。食物识别系统可以应用在饮食质量评估、食品自动结算、食品组织检索和推荐、餐饮机器人、食品质量监管等方面。

本文设计了一种美食实时识别系统，对于后端登录注册部分和前端界面进行了分析与设计，运用深度学习图像处理技术实现了食物识别的任务，并通过不断训练提升了系统的准确性，但该系统仍可以继续改进。在未来，也许会构建大规模食品图像识别基准数据集，定会有更多基于食品图像识别的新应用面世，方便我们的生活。

心得：

为了提升我们的是实践能力，学校为我们开设了小学期实训课程。在本次实训中，我们的实训项目是利用深度学习进行食物图像识别，最终制作出一个美食实时识别系统。在四周中，我们学习了机器学习、数据可视化matplotib、深度学习、tensorlow.keras、CNN卷积神经网络、python web的flask框架、python爬虫等内容，并小组合作设计开发了名为“识食物者”的美食实时识别系统。

在实训过程中，学习到了很多方面的知识，比如机器学习，广义上来说，机器学习是一种能够赋予机器学习的能力以此让它完成直接编程无法完成的功能的方法。但从实践的意义上来说，机器学习是一种通过利用数据，训练出模型，然后使用模型预测的一种方法。机器学习可以分为三种：监督学习、半监督学习和无监督学习。我们这次的实训是分类问题，属于监督学习。还学习了相关的的决策树、朴素贝叶斯、支持向量机、KNN等各种算法，并基于每种算法进行了实践理解，还做出了可视化，更加直观清晰。

深度学习是[机器学习](https://so.csdn.net/so/search?q=%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/epubit17/article/details/_blank)的子集，它使用级联的多层处理单元，称为神经网络，类似于大脑结构中的[神经元](https://so.csdn.net/so/search?q=%E7%A5%9E%E7%BB%8F%E5%85%83&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/epubit17/article/details/_blank)，每个连续层使用前一层的输出作为输入，它不需要我们自己去提取特征，而是自动地对数据进行筛选，自动地提取数据高维特征。深度学习也有许多相关算法，比如卷积神经网络CNN、循环神经网络RNN等。我们主要学习CNN卷积神经网络，它由卷积层（主要作用是保留图片的特征）、池化层（主要作用是把数据降维，可以有效的避免过拟合）、全连接层（根据不同任务输出我们想要的结果）组成，能够保留图片的特征，类似人类的视觉原理。还学习了预训练模型，VGG模型，和如何进行模型微调等内容。

老师带着我们从基础开始，一起动手实践，从了解基础概念开始讲起，再到几行简单的代码，最后到可以做出一整个后端以及前端，搭建起一个系统，到可以搭建自己的模型，自己调整参数，改变算法去提高精度，这短短四周，所学习的内容，所受到的锻炼，比我自学两个月都要丰富。

实训过程中遵从和企业一样的打卡制度，十分严谨，令我们在家就体会到了企业工作的感觉，通过学习与合作完成系统编写，锻炼了我们的代码能力、文档能力和小组合作能力。

很荣幸在实训中受到实训老师的教导，在四周中，每天在实训课堂中跟着老师一起动手实践，即时巩固刚学完的东西，这是大学生涯中的最后一个，也是学习知识最多，令人最难忘的小学期，虽然内容都是第一次学习，但是由于老师的认真负责，我感觉这次小学期对于知识的掌握程度，是最好的一次。非常感谢小学期认真负责的王老师，老师的水平非常高，并且非常耐心，面对我们的问题，他倾其所有，努力为我们解决，老师或许是实训期间最累的一个人。

在这四周我也一直在努力，认真完成老师交代的每一项任务，尽自己在所能去做项目，努力跟上老师的脚步。在项目中我主要负责前端界面开发以及登录注册的后端，这项任务让我发现了我对前端设计有着非常浓厚的兴趣，开学就是大四，面临就业，希望可以把前端作为一个未来的职业努力方向。本次实训，有很多收获，也发现了自己有很多不足，以后会更加努力，向着目标出发。

详细写实训过程的心得