



**本科毕业设计（论文）**

**( 2016 　届 )**

题 目： 基于自动推荐的美妆网站设计与开发

专 业： 计算机科学与技术

班 级： 16计科三班

姓 名： 王宜家

学 号： 16219112208

指导教师： 王贤明

职 称： 副教授

完成日期： 2020.3.16

目录

[摘要 1](#_Toc35303467)

[Abstract 2](#_Toc35303468)

[第1章 项目需求 3](#_Toc35303469)

[1.1 市场需求分析 3](#_Toc35303470)

[1.1.1 国内化妆品、服装市场发展现状及前景 3](#_Toc35303471)

[1.1.2 个性化推荐市场发展现状及前景 4](#_Toc35303472)

[1.2 项目可行性 5](#_Toc35303473)

[第2章 开发技术和环境 7](#_Toc35303474)

[2.1 前后端分离技术 7](#_Toc35303475)

[2.2 VUE 8](#_Toc35303476)

[2.3 Spring Boot 9](#_Toc35303477)

[2.4 MySQL 10](#_Toc35303478)

[第3章 系统设计 12](#_Toc35303479)

[3.1 设计方案概述 12](#_Toc35303480)

[3.2 前后端设计 13](#_Toc35303481)

[3.2.1 登录页面设计 13](#_Toc35303482)

[3.2.2 衣橱页面设计 15](#_Toc35303483)

[3.2.3 搭配页面设计 23](#_Toc35303484)

[3.3 类图设计 28](#_Toc35303485)

[3.3.1 Pojo 28](#_Toc35303486)

[3.3.2 Service 28](#_Toc35303487)

[3.3.3 Dao 29](#_Toc35303488)

[3.3.4 Controller 29](#_Toc35303489)

[3.4 数据库设计 30](#_Toc35303490)

[3.4.1 数据库表设计 30](#_Toc35303491)

[3.4.2 数据库关系视图 31](#_Toc35303492)

[第4章 结论 32](#_Toc35303493)

[4.1 本系统完成的工作 32](#_Toc35303494)

[4.2 项目优缺点 32](#_Toc35303495)

[4.3 个人感想 33](#_Toc35303496)

[参考文献 35](#_Toc35303497)

基于自动推荐的美妆网站设计与开发

摘要

基于现在社会上普遍存在的“选择困难症”和相比从前增长的社会购买力，本系统设计开发了一个可以整理自己衣物和化妆品，以及可以根据当天天气情况进行推荐穿搭的网站。.

本系统是基于前后端分离技术来开发的，前端运用了容易上手和开发的vue轻量级框架，并且配合element-ui框架来实现的；后端主要运用的是Spring boot框架进行开发。前后端通过api接口来进行通信。相关的数据存储在MySql数据库中。前端的开发在WebStorm中进行，后端开发在IDEA中进行。

本系统基本能够帮助用户可以更直观和高效的整理自己现阶段所拥有的衣物和彩妆物品，同时，本系统还能提供天气查询的功能，并且向用户推荐应季的穿搭方案。

关键词：美妆；整理；天气；推荐

Based on automatic recommend beauty makeup the website design and development

Abstract

Based on the widespread now "to go" and compared with the former growth of social purchasing power, the system was designed and developed a can organize their own clothing and cosmetics, and can be recommended to wear according to the weather that day web site.

This system is based on the front end separation technology to develop, front end using the vue lightweight framework, easy to use and develop and cooperate with the element - UI framework to implement;The backend is the main use of Spring boot framework for development.The front end through the API interface to communicate.Related data is stored in the MySql database.

The development of the front-end in WebStorm, back-end development in IDEA.

This system can help the user can more intuitive and efficient finishing their own clothing and cosmetics items at present, and at the same time, this system can also provide weather query function, and recommended to the user in season one outfit.

**Key words:** Beauty makeup;Sorting;The weather;recommended

# 项目需求

## 市场需求分析

### 国内化妆品、服装市场发展现状及前景

改革开放、九年制义务教育、一带一路的实行，促进了我国经济市场的快速发展，以及大大提高了我国的女性地位。受到高等教育的女性占比超过50%，这促进了女性在我国经济的参与度提高。女性的意识和经济能力的提高，代表了我国女性对于高品质生活的追求，这极大促进了我国化妆品市场和服装市场的消费和生产。

虽然美国仍是第一大消费市场，但我国在2009-2018年间，化妆品行业的年均复合增速超过全球增速水平的7.3%，我国已然成为全球化妆品行业的重要市场[[[1]](#endnote-1)]。



图1-1：全球化妆品市场规模

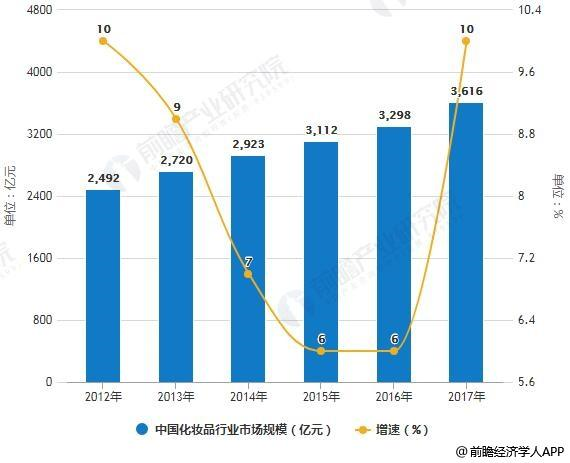


图1-2：中国化妆品行业市场规模

而我国的女装市场前景同样广阔。2011-2015 年，我国女性人均服装开支从730 元增至1262 元，预计2020年将达2369 元，女性服装开支日益增加，未来三年开支复合增速13.54%。

因此，我国的美妆市场发展迅速，市场广大，而相应的产品以及购买力也快速增长。并且中国的消费市场呈一种年轻化的趋势，20-29岁的女性虽然只占总人口的28%，但却贡献了彩妆市场销售额的45%，85后成为了化妆品消费的主力军[[[2]](#endnote-2)]。“女孩子的衣柜里永远缺一件衣服”、“女孩子永远缺一支口红”等流行语也展现在大众面前。

### 个性化推荐市场发展现状及前景

如今，各个电商平台、手机APP层出不穷，人们的选择变多了，相应的选择困难症也出现了。随着网络技术的发展，众多拥有海量数据的系统存在信息过载的问题，而推荐系统是解决这一问题的重要工具，在提高用户满意度方面，系统为其推荐可能感兴趣的事物，可以有效增加用户点击率，提高电子商务营业额，而使用标签则更能反映用户的兴趣以及物品的特征，可以大大提高推荐的准确率，也具备良好的解释说明性[[[3]](#endnote-3)]。

所以，针对这些情况，个性化推荐市场应运而生。各类个性化推荐系统的产生和使用，也促进了购物消费市场以及产品的设计生产。

在个性化推荐系统越来越普遍的今天，作为开发者，我们更应该重视用户该有的隐私权和知情权以及拒绝权，提高法律意识[[[4]](#endnote-4)]。

## 项目可行性

全世界每天都有数千万人化妆。它是古代和全世界的人类装饰面部和身体最显著的表现之一。然而，人们对化妆的心理原因，机理和后果仍然知之甚少。例如，人们对妆容如何改变人们的认知。但是，化妆对吸引力的影响已得到广泛研究。大量精心控制前后照片的研究发现，在化妆时面部图像在物理上被认为更具吸引力。

考虑到化妆可以使脸看起来更漂亮这一发现，最近人们已开始研究通过化妆修饰的视觉特征，以使脸看起来更有吸引力。众所周知，化妆可以改变至少三个视觉特征：皮肤均匀性、面部对比度和面部特征尺寸[[[5]](#endnote-5)]。

通过使用粉底或者遮瑕之类的产品，可以使肤色变得均匀，相应的对于展现出来的健康程度和对其他人的吸引力也有增加，也可使呈现出来的状态更年轻。女性化妆会比男性化妆更能有效修饰自己的健康状态以及年龄。

面部对比是通过五官与周围皮肤之间的颜色和亮度变化来体现的。面部对比越明显，会让脸部的五官看起来越立体，这也能让女性看起来更漂亮。

化妆对于面部特征尺寸的修饰也是女性选择化妆的重要原因。化妆的确可以让眼睛和眉毛显得更大，同样可以使鼻子看起来更小。通过化妆修饰面部五官的大小，使化妆对于女性有巨大的吸引力。而经过化妆修饰也能使女性对于他人有更大的吸引力。

因此，化妆会通过改变面部来刺激视觉，从而激活社交。女性通过化妆可以改变社会对女性的刻板印象和态度，而化了妆的女性在社交或工作学习时也会更加自信和成熟，有利于女性对于职业的选择。

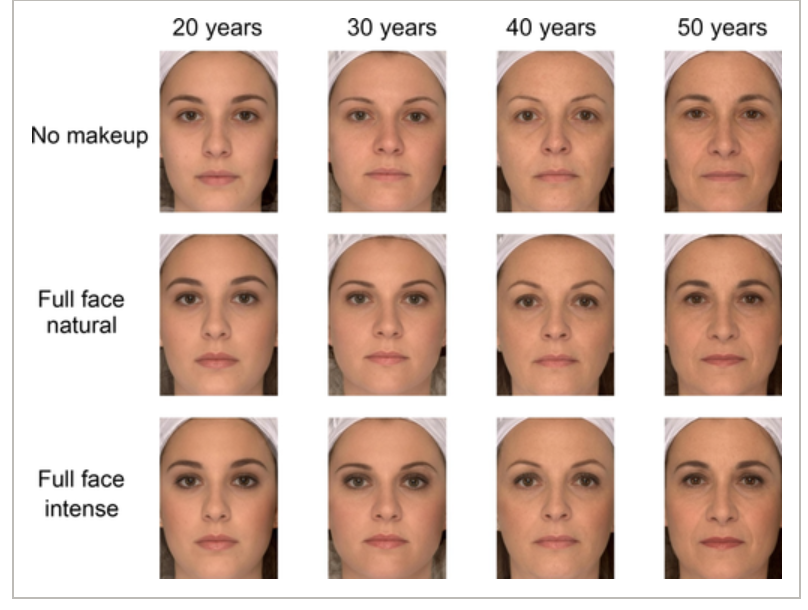


图1-3：化妆面部对比

化妆对于女性有重大的意义，但随着社会对于性别认知的开放，化妆不仅仅是独属于女性的权利，其他性别同样可以通过化妆增加吸引力。这也使化妆相关的产品和市场发展迅速、日益壮大。

和化妆品脱不开干系的是服装，好的妆容需要搭配修饰身材的服装，不然只能事倍功半。一身搭配良好的服装，可以使人显得更得体、更亮眼。优秀的服装同样可以增加吸引力和自信，也同样可以得到周围人的赞赏和好感。而拥有一身得体的服装搭配和精致的妆容，不论对于哪个性别的社交都是有益的。

人们的日常生活离不开打扮，经济能力的提高也让人们对于化妆品和服装越来越依赖。社会的快速发展、信息的碎片化让人们的生活压力日益渐增，大量信息的涌入出现了人们随大众的现象，例如，什么火买什么，网红推荐什么买什么等。经常收快递也变成了日常。

衣柜、化妆台的日渐填满，虽然满足了大多数人缺乏的安全感，但也增加了人们对于决定每天化什么妆、穿什么衣服思考犹豫的时间。

这就是开发本系统的原因。本系统可以在电脑上梳理使用者的衣柜和化妆台，并且随机推荐使用者的每日穿搭，减少使用者对于穿搭的纠结时间。本系统一大特色就是完全随机，一方面，如果按照大众的、公认的色彩或者面料搭配，就少了一些个性和特色，千篇一律的搭配会产生审美疲劳，并且不能突出个体；另一方面，系统随机出来的搭配可能会产生一些有趣的效果，会给使用者带来一些乐趣，拥有趣味性的系统会更加吸引人。

# 开发技术和环境

## 前后端分离技术

之前我做到的系统大多是基于B/S的系统，这样其实开发效率并不高，只是前段方面就要先把html页面做好之后套jsp页面，如果我更改了html页面，相关的jsp页面也要更改。在开发过程中，只能等前端做好之后开始做后端，这样的前后端强依赖，会造成如果一端发生代码的更改，另一端就要跟着改变，开发效率会比较低。而且在项目刚刚开始开发的时候，虽然有大体的框架，但是具体前端要做成什么样，后端要怎么写，其实是迷茫的，大多数都是一边开发一边把功能实现完整起来，或者对于这个功能，我先想好了前端怎么弄，但是后端还比较空白；或是我知道后端该怎么写，但是前端的设计还没想好。如果前后端没有分离的话，都得前后端的设计都想好，这样不利于我前期整理好设计的思路，还会容易产生焦虑。

这个时候我发现了前后端分离技术，开发工程师很早的就发现了前后端不分离的弊端，所以他们开发了前后端分离的技术。前后端分离最大的优点就是，前端和后端独立编写，前端只要通过ajax调用后端的api接口就可以实现数据交互。前后端分离可以大大提高开发的效率，而且一个前端可以对应多个后端。对于后端开发来说，后端的工程师可以更注重于java基础，后端代码逻辑，以及后端的java性能优化，因为java语言和java的安全体系非常适合用来开发系统[[[6]](#endnote-6)]。而对于前端开发工程师，可以更注重于前端页面的表现，页面之间的动态展现，以及用户的使用体验。

前后端分离技术可以让前后端在开发过程中进行并行的开发，如果后端的代码有变动，不需要前端进行改变，只要连接的接口和参数不变就能实现。而且前后端耦合在一起的架构模式在提交请求时，服务端的servlet或controller要先接收请求，然后调用相关代码完成业务逻辑，再返回jsp来展示动态，这样的过程会比较冗长和麻烦。前后端解耦可以在浏览器发送请求的时候直接到达html页面，html页面可以通过ajax产生json格式数据来填充html页面，以展现动态的前端页面效果。

在前后端分离开发过程中，还有一个很大的好处就是可以快速定位bug出现的地方是在前端还是后端，如果是接口出现错误或是数据应答超时那就是后端出现问题，如果是页面样式出现错误或者是页面的逻辑出现错误那就是前端的问题。这样可以提高解决bug的效率。

虽然现阶段前后端分离技术还不是很成熟，但前后端技术有利于有效提升开发的效率，在开发项目时，我们要多尝试前后端分离的模式，促进前后端分离技术的发展[[[7]](#endnote-7)]。

## VUE

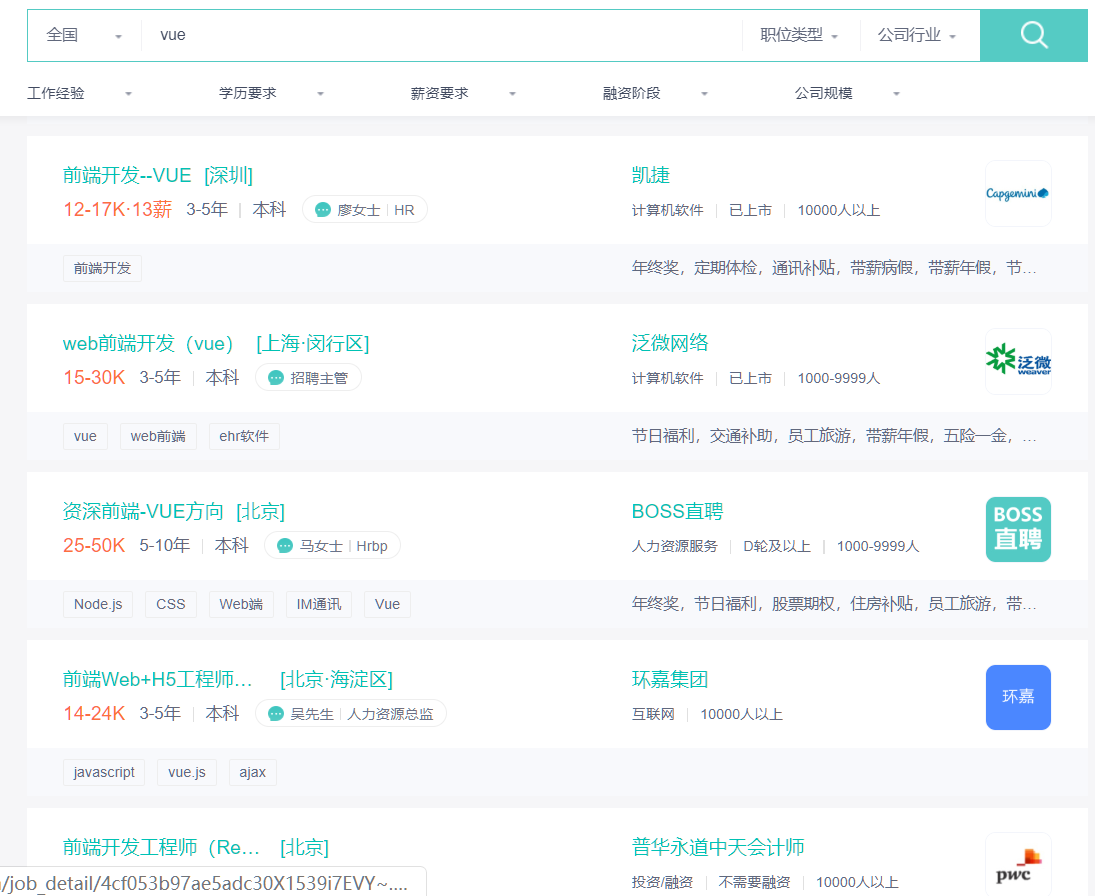


图2-1：BOSS直聘 VUE搜索结果

这是招聘网站上的vue前端工程师的薪资水平，我个人认为还是比较可观的。也说明了在现在这个阶段的前端行业里，vue并没有过时，甚至是有一定前景的。

关于前端框架的选择，之前我大多都是用的纯css，这样工作量会非常大，如果实现一些动态效果的话，代码量会大大增加，而且效果也没有特别好，后来就有接触到vue这个前端框架，vue是一套渐进式的框架，可以用来构建用户界面，而且vue采用的是从下而上的增量开发的设计。vue框架上手非常简单，再配合element-ui，就能很容易的搭建起来前端的页面。Vue具有前端工程化的自动化优势，可以自动配置WebPack、安装依赖包、生成项目目录等[[[8]](#endnote-8)]。还能对页面进行局部的无刷新替换；对于前端工程化的模块化、组件化、数据驱动、轻量级等非常友好，Vue可以解决因前端业务日益繁杂等原因引起的问题[[[9]](#endnote-9)]。

因为这是个基于网站的系统，所以我选择了element-ui来作为前端的ui框架，element-ui是饿了么的前端团队开发的一款基于vue.js的桌面端ui框架，使用非常简单，而且官网提供的功能也几乎涵盖了所有的前端设计功能。

## Spring Boot

在选择后端框架时，我使用了Spring Boot，SpringBoot是Spring开发技术的升级，引入注解，简化了Web框架的配置和开发，有利于后端进行高效开发[[[10]](#endnote-10)]。所以在建立项目时，只要勾选自己需要的功能，就能快速的构建出一个项目结构。

我使用的是前后端分离技术，前后端之间的联系主要是通过api接口，而Spring Boot可以非常方便的对外输出rest api服务，这样有利于前后端的运行。Spring Boot默认使用的logback可以自动的根据环境切换日志的级别，而且Spring Boot默认会对liquibase进行配置，这样可以很好的解决数据库的版本问题。所以，Spring Boot可以让编码、配置、部署的过程变得简单，这也是我选择它的原因。

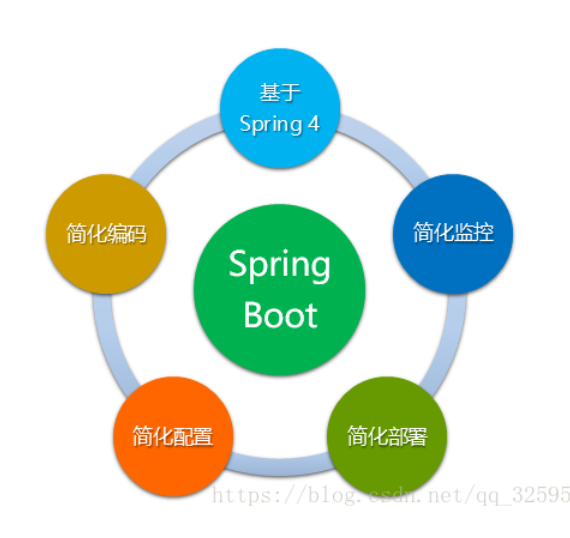


图2-2：Spring Boot优点分布图

之前在写其他系统时，我用到过Spring，在使用Spring时，需要在pom中添加很多依赖，这样会增加工作量，而Spring Boot只要在pom中添加一个starter-web依赖即可。在写项目的过程中，我感受最深的是，如果我用到了额外的依赖，只要直接在后端代码中直接使用就行，Spring Boot会自动帮我添加。

## MySQL

MySQL几乎是现在最流行的开源数据库，在全球前20大的互联网网站中，有18个网站使用的是MySQL。

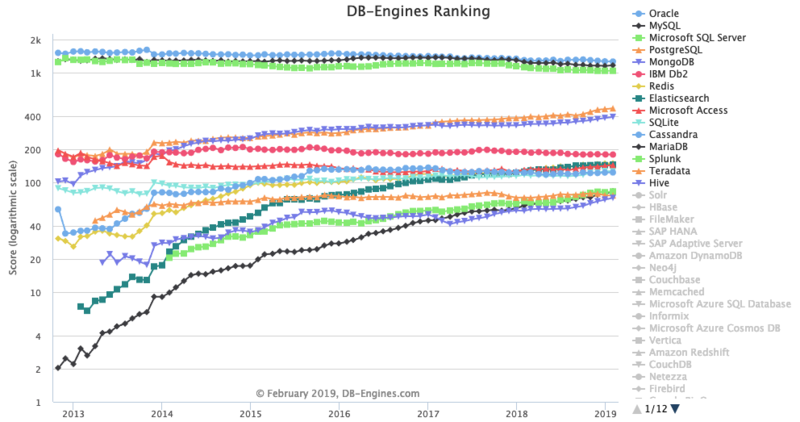


图2-3：2013-2019全球数据库的使用率

这张图显示，虽然Oracle仍在第一位，但MySQL的使用率在渐渐增长，这说明MySQL已经基本成为了流行趋势。

区别于需要收费的Oracle，MySQL是一款免费的软件。而且MySQL是一个多用户、多线程的sql数据库服务器。MySQL最大的特点就是可以快速有效的处理数据，而且MySQL使用很简单，上手很快且便捷。MySQL还支持了多种语言，常见的编码，如，中文，也可以用为对数据表和列的命名。

而MySQL特有的存储过程也有利于减少sql语句的数量，避免代码的冗长，这样可以减少系统和数据库服务器之间的流量。MySQL还大方的把数据库的接口暴露出来给所有的程序，这样可以方便在开发的时候减少开发存储功能的代码量。MySQL也保证了一定的数据安全性，因为在使用的过程中，我们的权限仅限于是MySQL赋予我们的，我们不能反过来赋予MySQL权限。

# 系统设计

## 设计方案概述

本系统是基于前后端分离的美妆整理推荐网站。所以大致的功能分为两块：整理和推荐。

在整理模块中，用户可以录入自己的物品，并且丰富物品信息，例如，图片、名称、品牌和相应适合的季节等。录入成功之后，用户可以根据分类更直观的看到自己物品的数量、类型等，用户同样可以通过搜索来实现找到某样物品。如果物品的信息有变动，用户同样可以进行修改和删除。

在推荐模块中，主要是通过获取天气信息来分类相关季节的衣物，所以这同样是一个具有天气预报功能的网站。为了解决如今社会普遍存在的“选择困难症”以及随着社会发展明显提高的购买力，在用户录入的情况下，根据相应的天气情况来随机产生今日搭配，减少纠结每天穿什么衣服，化什么妆的时间。本来我的想法是还要根据相应的衣服颜色和款式等来实现搭配，但转念一想，这样的搭配又有什么意思呢？既然已经是随机了，就不需要这么多的条条框框，千篇一律的搭配可能不会出错，但是也少了个人特色和趣味性。就像大家会觉得可乐和炸鸡是最配的，然后所有的系统在做推荐时都把可乐和炸鸡放在一起，但是谁又能确定铁观音和炸鸡就不配呢？到底怎样穿衣才好看，只有尝试过很多种可能才知道。所以，作为21世纪的年轻人，我们要敢于尝试，敢于发现。这就是我认为我的系统不需要做以颜色和款式为基础来进行搭配的功能。也许系统出来的随机效果很棒，也许根本不能穿出去，这都是我们生活的调味剂，这也是我设计这个系统的初衷——勇于尝试，然后开怀大笑的面对生活。

## 前后端设计

### 登录页面设计



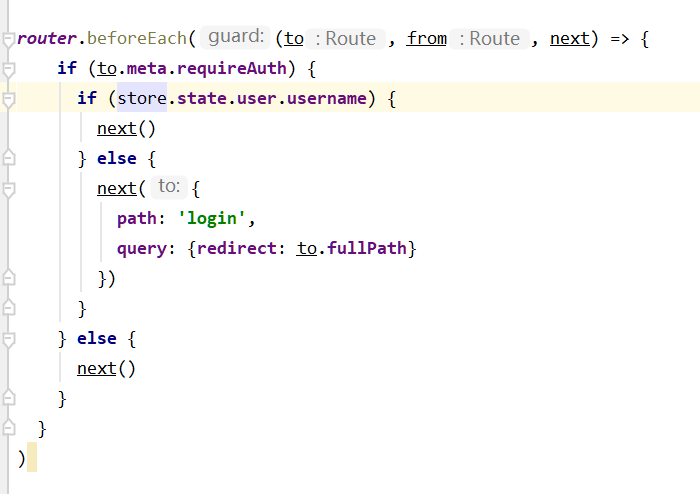
图表 3‑1：登录页面

这是一个不需要社交的网站，唯一联网的地方就是获取天气情况。所以我觉得不需要注册用户信息，因为数据绑在本地数据库上，衣柜只属于用户一人，所以我没有设计注册功能。而设计登录功能的原因是，保护一下用户的隐私，防止偶尔被其他人看到和使用，我也在前端设计了登录拦截器。

在设计登录拦截时，我引入了Vuex，可以为vue管理状态。在index.js里，用localStorage来记录用户信息的变量，判断本地存储中有没有user存在，如果这个对象存在，就获取username的值；如果不存在，则username为空。

在路由设计时，把需要登录拦截的路由的requireAuth设置为true，这样可以区分哪些页面需要拦截，哪些不需要。

在判断是否需要拦截时，还需使用钩子函数[[[11]](#endnote-11)]，钩子函数需要在访问每个页面前调用，所以我选择了router.beforeEach()，代码如下：



图表 3‑2：前端登录拦截代码

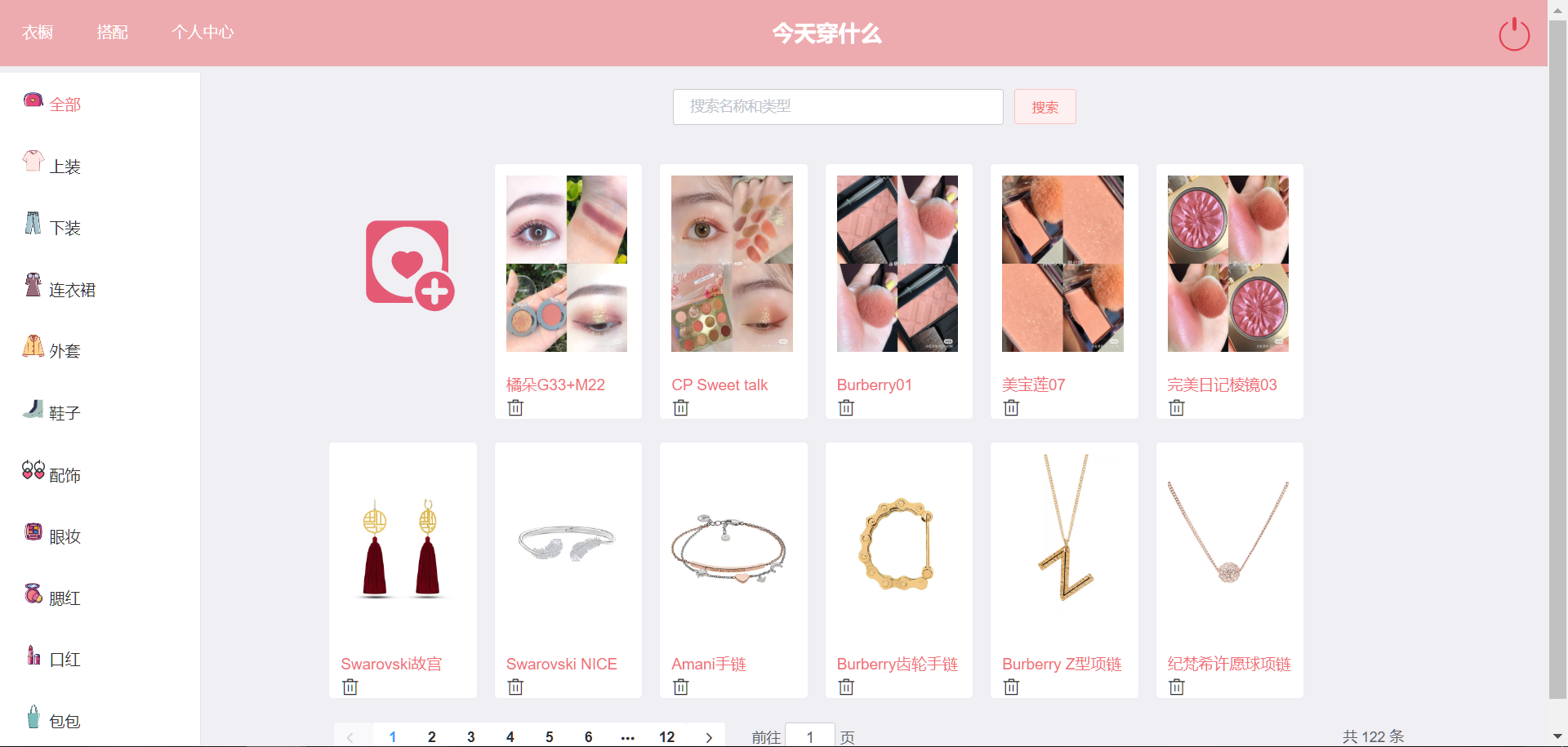
函数的逻辑是，首先需要判断现在访问的页面是否需要登录，如果需要，则要判断在store里面有没有存入user的信息，如果有，则进入该页面，如果没有，则跳转到登录页面，登录之后再进入选择的页面。

在login.vue页面中，需要加入登录的相关逻辑。在点击登录的按钮时，需要向后端发送数据；在收到后端返回的成功登录的代码时，可以把loginForm对象传递给user对象；登录时，还需要获取登录前的页面路径，登录成功之后即可跳转。相关代码如下：



图表 3‑3：登录页面代码

### 衣橱页面设计



图表 3‑4：衣橱页面

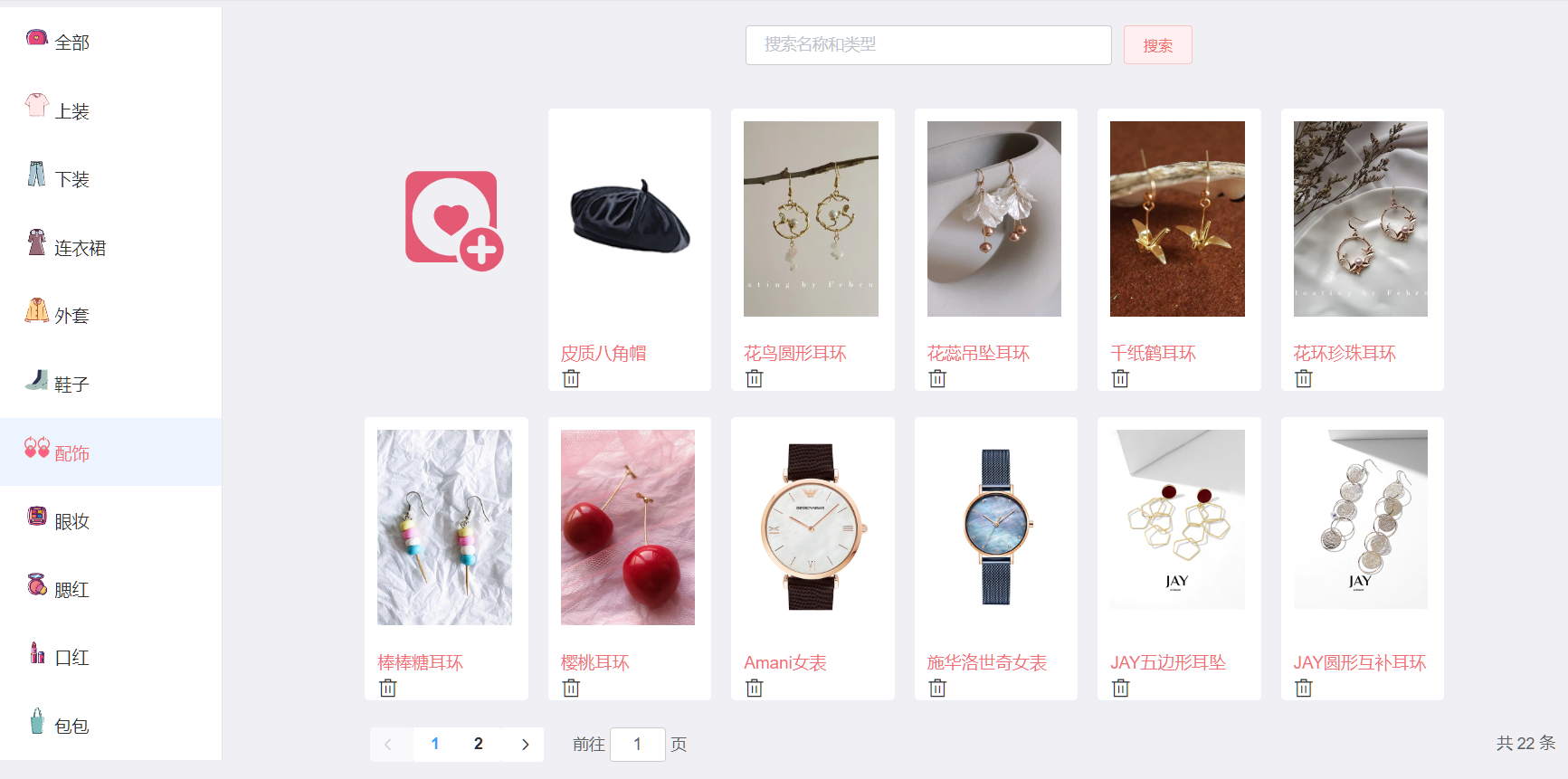
书橱的设计主要有8个功能：

1. 打开页面时，默认显示全部物品
2. 根据侧栏分类情况显示相关物品
3. 在搜索框中通过输入相关信息可以模糊查询出相关物品
4. 实现添加功能
5. 点击图片可实现修改物品信息
6. 物品可删除
7. 显示物品的图片和名称
8. 鼠标移动可显示物品的详细信息

#### 页面初始化

当用户进入“衣橱”页面时，默认显示所有的物品，这就需要在进入页面时就自动触发相关代码发送请求并且渲染页面。所以我使用了钩子函数mounted，可以通过元素的双向绑定来实现页面的动态渲染。在编写方法时，我的逻辑是，用axios发送get请求，收到后端返回的成功信息之后，把data里的数据替换为后端返回的数据。

#### 侧栏分类



图表 3‑5：侧栏分类显示

侧栏分类实际就是根据分类的id实现查询功能。在前端代码里，用户点击相关分类，即向后端发送带有参数的get请求，和显示全部信息的逻辑一样，修改data里的数据来实现动态渲染。但和全部显示不一样的是，分类显示还需要组件之间的通信，即在MyWardrobe这个组件的方法里，需要获取SideMenu组件里的data。因为SideMenu是MyWardrobe的子组件，所以在MyWardrobe中引入即可。



图表 3‑6：子组件使用方式

这种方法可以大大减少主页面代码的复杂度，也可以让我在写代码时更能理清思路，在出现bug时也能及时发现问题出在哪个地方。

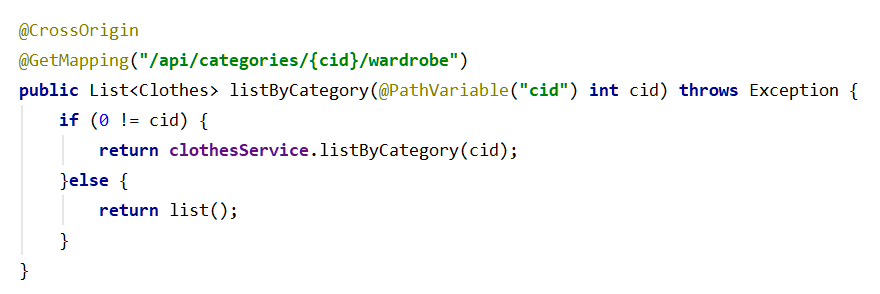
在写SideMenu组件时，需要加入@select事件：



图表 3‑7：加入事件

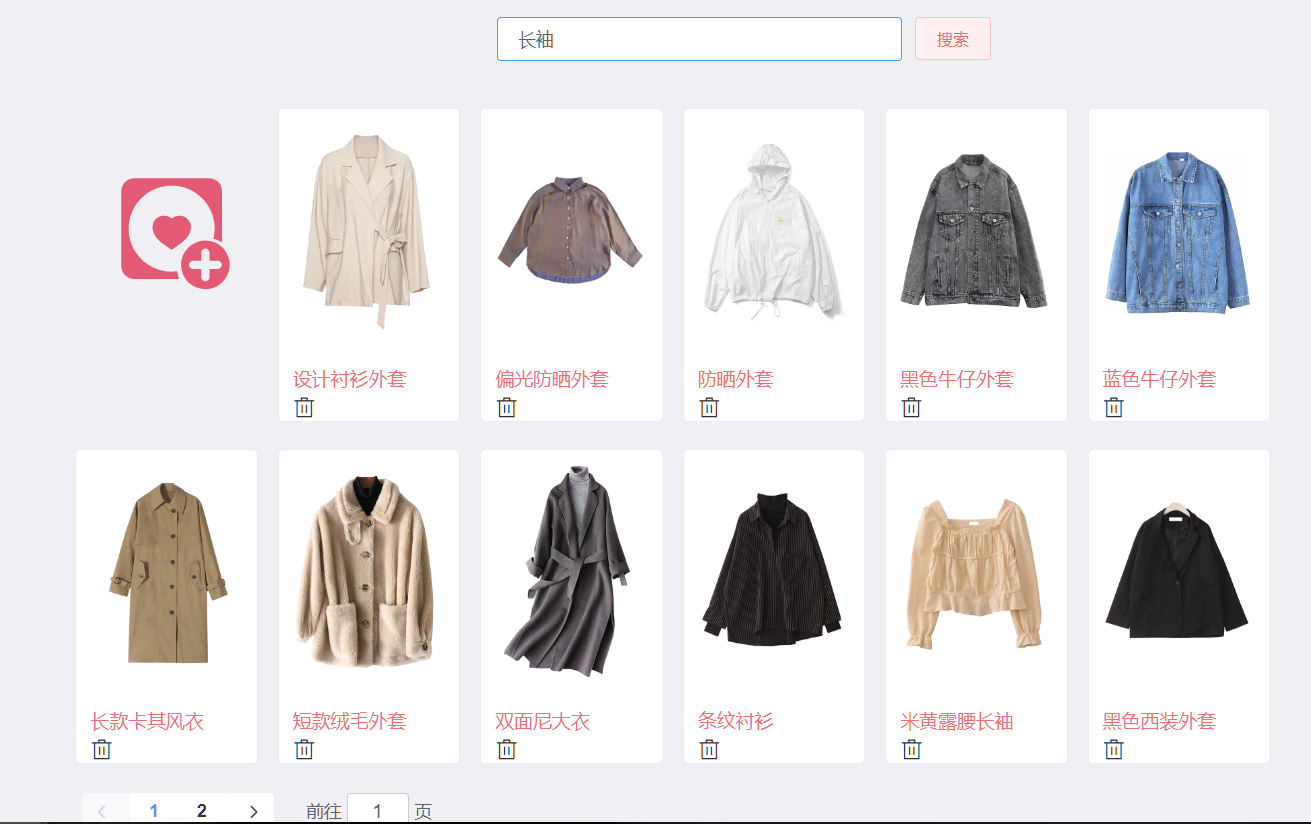
这样可以实现，在用户点击侧边栏其中一个的标签时，执行handleSelect方法，然后触发indexSelect事件，把key的值赋值给data中的分类码，这时，MyWardrobe组件就能收到相关信息然后执行listByCategory 方法，这个方法则发送请求给后端的查询代码，后端查询代码再把相关数据返回给前端，前端通过refs来修改相关data，以达到动态渲染页面的效果，即实现分类查询。

后端查询代码如下：



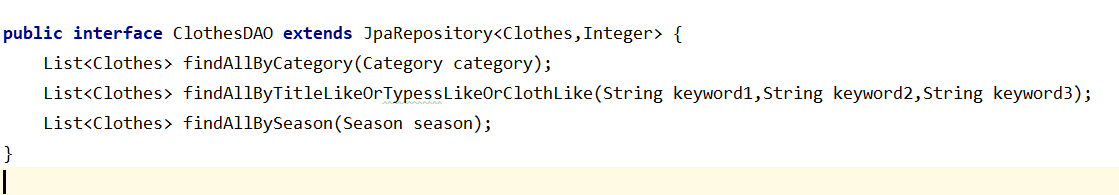
图表 3‑8：后端查询代码

#### 搜索功能



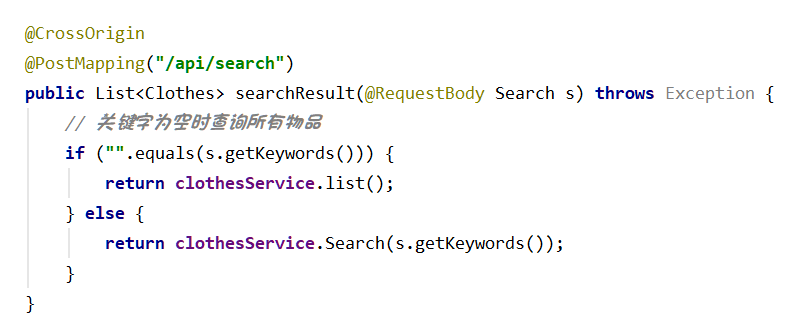
图表 3‑9：搜索功能显示

关于查询，我设计的是可以通过三个参数进行查询，分别是：title、typess、cloth，使用者可以通过搜索物品的名称、品牌或面料、类型来进行搜索。后端代码如下：



图表 3‑10：后端查询关键字范围代码

但是当用户在搜索框输入为空时可能会出现问题，所以我又加了一个方法，用来实现，当输入为空时，默认搜索的是全部的物品。后端代码如下：



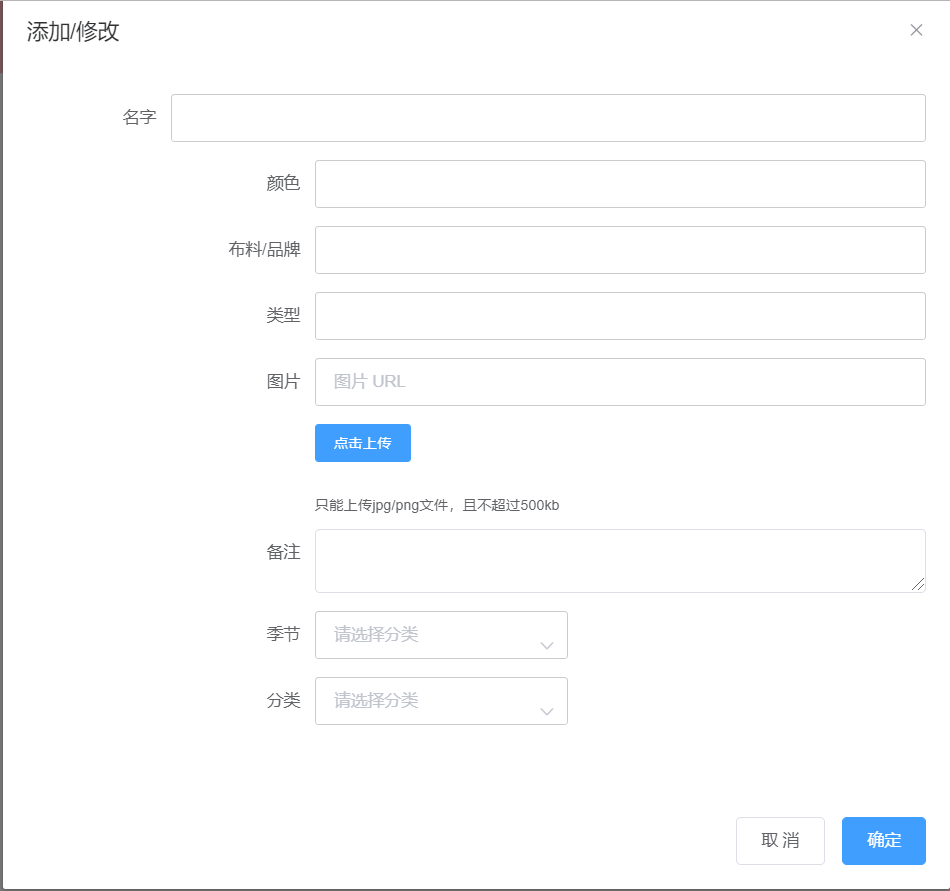
图表 3‑11：输入为空查询逻辑代码

在前端代码实现时，还是使用了和上面两个功能一样的axios请求。

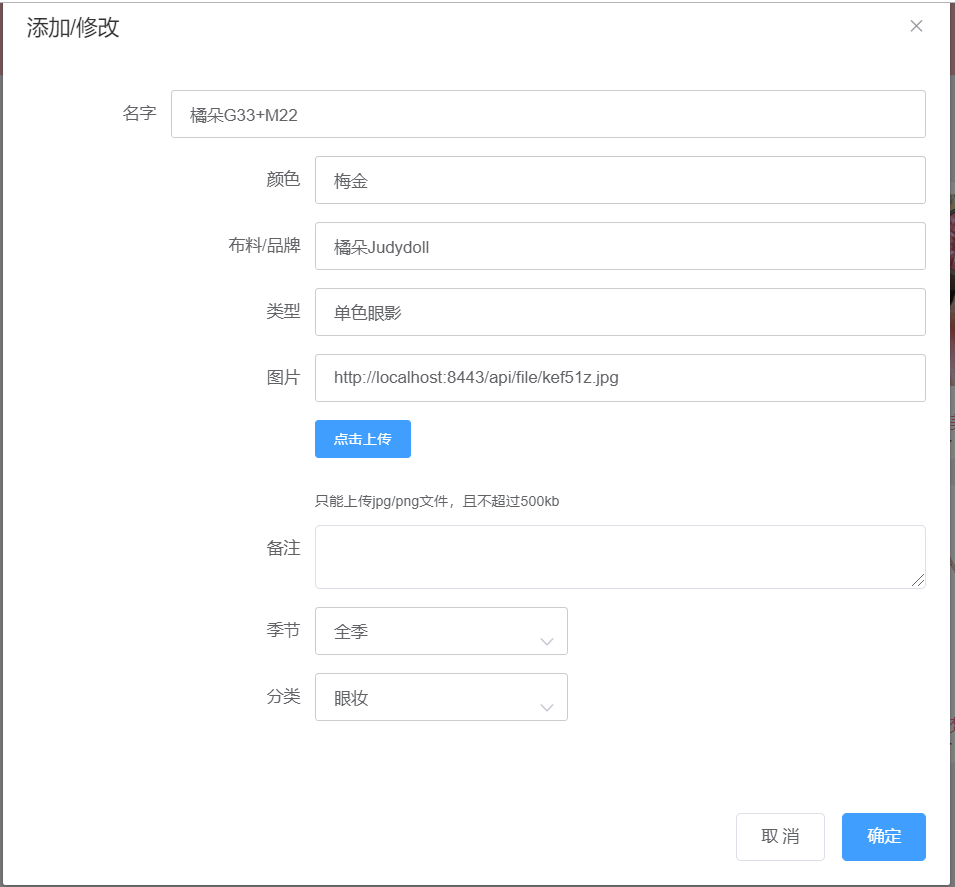
#### 添加修改功能



图表 3‑12：添加按钮



图表 3‑13：添加弹出界面



图表 3‑14：修改弹出界面

点击添加按钮，就会出现相关的表单。添加和修改使用的是同一个表单，为了减少页面的数量，所以对于这个表单，我使用了:visble.sync 属性来实现这个表单的显示或者隐藏。

在代码运行时我发现，修改内容之后再添加物品，展现的表单是有上一次修改的内容的，所以我又加了一个clear方法，这样就可以实现在关闭时清空表单的内容，在下一次点击添加时，就不会有还存在内容的情况了。

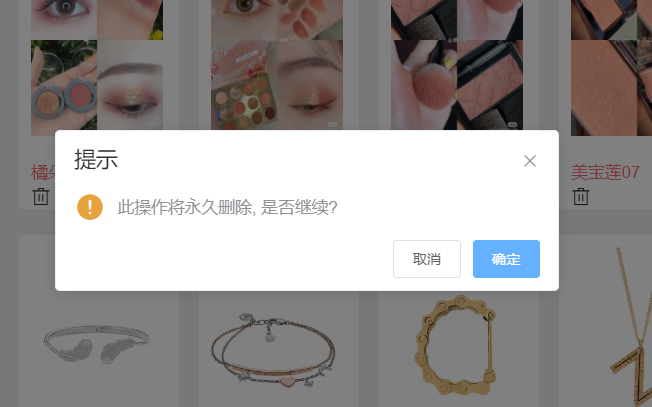
添加功能只要在数据库中直接插入数据即可，而修改时，我在物品的图片部分加了点击事件：



图表 3‑15：点击事件

editClothes这个方法可以弹出表单并且渲染上相关数据。

#### 删除功能



图表 3‑16：删除功能



图表 3‑17：取消删除

点击删除图标时，跳出一个确认对话框，如果取消删除，则从顶部弹出一个提示框，我使用了post请求，通过键值对来实现。前端代码如下：



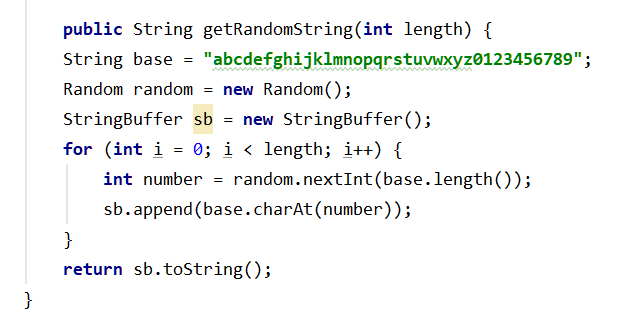
图表 3‑18：前端删除代码

#### 图片上传功能

关于实现图片上传的功能，逻辑依然是前端通过post向后端发送请求，后端对接收到的数据进行相应的处理，并保存在服务器相应的位置里，再把对应的url返回给前端。

前端设计部分我用到了element-ui中的upload，可以比较快速方便的实现前端的图片上传功能。在upload中，我用到了action属性和multiple属性，这两个属性分别能实现指定api和检测上传文件的数量。

图片部分的后端设计就比较重要，首先，在保存文件的命名时，我选择了在英文字母和阿拉伯数字中的指定长度的随机命名，后端代码如下：



图表 3‑19：后端图片命名逻辑代码

在对文件，即图片的操作中，需要对后端接收到的文件进行重命名，但得保留文件的原格式，所以我在后端添加了PostMapping，后端代码如下：



图表 3‑20：后端图片存入路径

### 搭配页面设计



图表 3‑21：搭配页面

在搭配页面中，主要实现的是两个功能：

1. 天气查询
2. 物品随机搭配

#### 天气查询功能

这个功能需要先选择位置信息，获取实时的时间，然后根据位置信息来获得相应的天气情况以及未来两天的天气信息。

关于天气预报，我用到了高德地图[[[12]](#endnote-12)]，需要在高德中申请自己的key并且在项目中引入：



图表 3‑22：高德地图key引用

高德平台给了我很大的帮助，让我提高了写代码的效率，所以根据官网给我的教程，再加自己的想法，我的逻辑是通过选择定位获取位置信息，再通过获得的城市名称获取相应的天气信息，这时，我只要在前端设计部分调用我需要的数据即可。前端代码如下：



图表 3‑23：选择城市查询功能



图表 3‑24：获取天气信息

#### 物品随机搭配功能

这是本系统最重要的功能，首先读取当前所在城市的气温情况，根据温度判断需要随机的物品是哪些，比如，在夏天时如果随机出现冬天的羽绒服的话，这个系统的参考性就降低了。但我把天气相关代码作为一个子组件，把随机相关的代码作为另一个子组件，所以两个部分并没有在同一个页面中，但两个子组件在同一个父组件中。我需要把温度信息传递给父组件，在从父组件传递给随机相关的子组件即可。相关代码如下：



图表 3‑25：WeatherIndex.vue数据传出



图表 3‑26：子组件传入



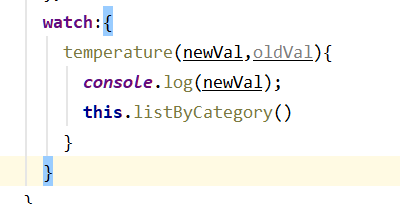
图表 3‑27：子组件传入



图表 3‑28：详细页面逻辑判断

首先，在WeatherIndex中，因为temperature这个子组件外面套有两个父组件，再往外套的父组件才是“搭配”页面HomeContainer，所以在往外传递temperature值的时候，需要连续套三个$parent，才能把温度的数据传递给“搭配”页面，然后实现随机搭配的子组件CollIndex在HomeContainer中，所以需要在<CollIndex></CollIndex>中加入:temperature="temperature"才能成功传递。但是我在设计时，是按分类组成的CollIndex的子组件，所以在CollIndex.vue页面中，还需要加入props来实现父子组件之间的通信，并且同样需要在子组件引入时添加temperature属性。最后，才能在相关的子组件页面中，加入props再和上一级组件实现通信，来调用相应的temperature值。

在代码运行时，我发现读取温度，随机搭配好了之后，如果我换了一个地方，温度变了，但随机出来的物品和这时的温度不匹配，所以我又加了一个监听温度的方法，一旦左边温度发生变化，就替换原来的温度值。这样就不会出现上述问题了。代码如下：



图表 3‑29：温度值监听

在编写随机选择的逻辑时，首先需要判断温度的值适合哪些季节的衣服。我把季节分为简单的3类：春秋、夏和冬，因为春秋的气温接近，所以我放在了同一类中。当温度小于10摄氏度时，读取的是冬季的服装；当温度大于10摄氏度小于20摄氏度时，读取的是春秋的服装；当温度大于20摄氏度时，读取的是夏季的服装。读取季节分类的逻辑和读取物品分类的逻辑一样，都是向后端传递，再由后端处理之后返回相关的数据。成功读取季节分类之后，再读取相应的物品分类即可，这里我用的是相关的子组件，所以可以直接写死，不需要判断。

关于随机搭配的功能，其实就是类似于抽奖或者点名。读取符合的分类之后，把这些数据存入数组，让相关函数进行处理。随机选择时，点击开始就可反复开始，但以读取过的物品就会被系统记住，防止第二次开始时，出现的物品还是第一次随机出来的物品。我还设置了重置按钮，点击重置，系统就会删除之前以及被随机抽到过的物品记录，这样可以让之前出现过的物品有可能再次出现。代码如下：



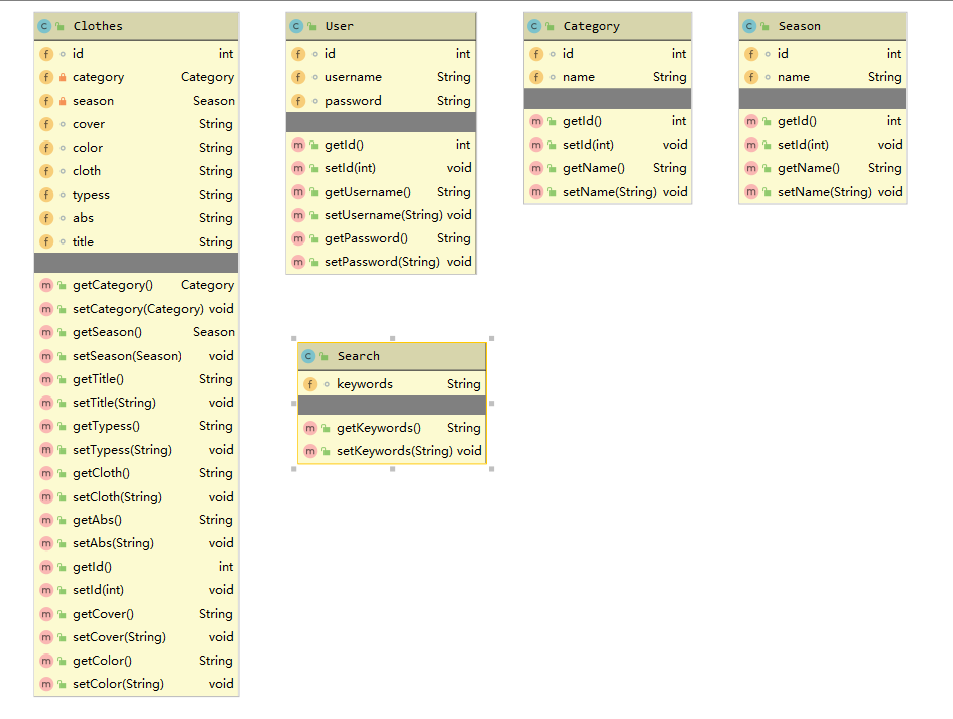
图表 3‑30：随机抽取代码



图表 3‑31：重置按钮逻辑

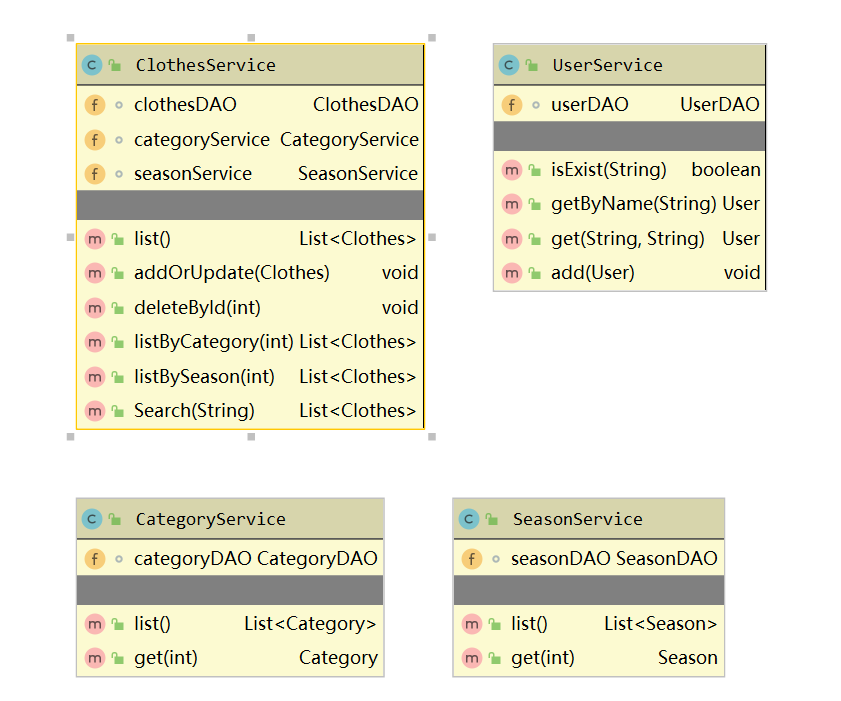
## 类图设计

### Pojo



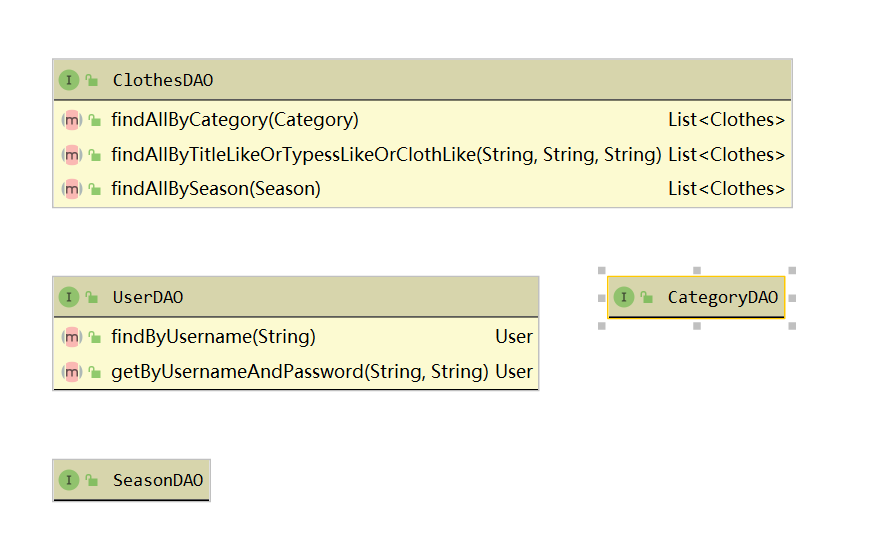
图表 3‑32：实体类

### Service



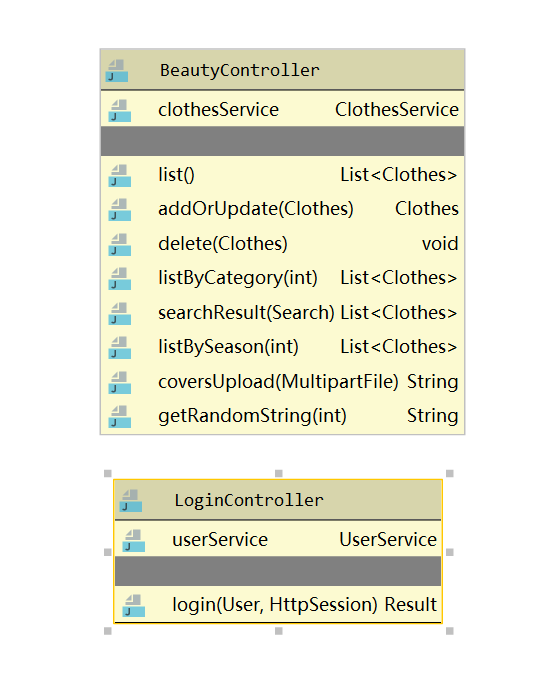
图表 3‑33：服务层

### Dao



图表 3‑34：数据层

### Controller



图表 3‑35：接口层

## 数据库设计

关于数据库，我设计了四个表，分别是category、clothes、season和user。其中，category表和season表比较类似，category表主要用于区分物品的类别，比如，眼影和上衣，这个表的id和clothes表中的cid对应；season表主要用于在选择温度随机时，判断当季衣物的，这个表的id对应clothes表中的sid。clothes表是最重要的一张表，这个表存入了所有物品的信息，分别对应了物品的id、名称、颜色、布料或品牌、类型、图片、备注以及相关的季节和分类。user表存入的是用户的用户名和密码。

### 数据库表设计

类目表：category



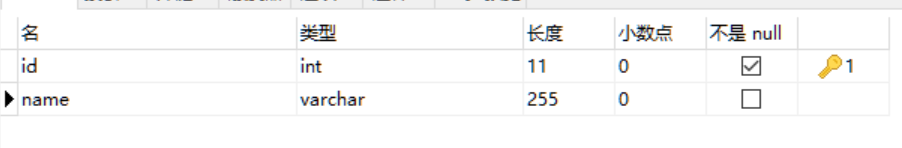
图表 3‑36：分类表设计

类目表：clothes



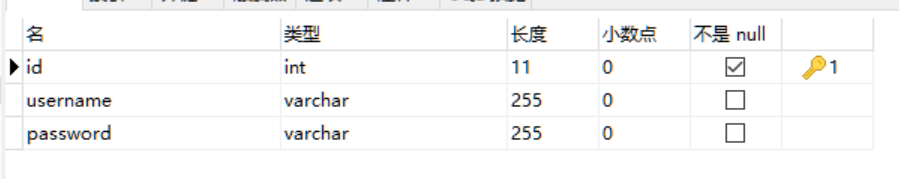
图表 3‑37：衣橱表设计

类目表：season



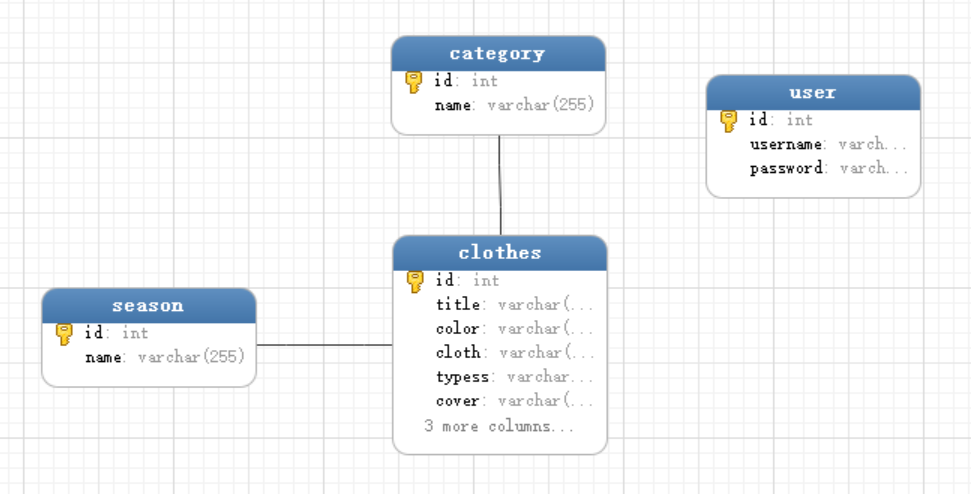
图表 3‑38：季节表设计

类目表：user



图表 3‑39：用户表设计

### 数据库关系视图



图表 3‑40：数据库表关系视图

# 结论

## 本系统完成的工作

本系统完成的功能有：

1. 登录页面开发和登录拦截；
2. 导航栏切换页面；
3. 衣橱页面的侧边栏导航、物品显示、分页、搜索、物品删除以及物品的增加和删除；
4. 搭配页面的天气查询、根据查询到的天气温度对物品进行分类然后随机推荐。

作为一个基于美妆推荐的网站，最基本的功能是在有一定数据的基础下给用户提供推荐。所以我开发了衣橱页面，主要用于用户存储和整理自己的物品。因为主要推荐的是衣服的穿搭，衣服是分季节的，春秋时期的衣服穿搭会比较随意，但在夏天穿羽绒服，或是在冬天只穿一件吊带。虽然我觉得这个系统的随机搭配不需要完全遵循大部分人认可的颜色或者面料、款式的搭配，但是如果在夏季出现冬天衣服的搭配，那这个系统的实用性就会大大降低，没有参考度，那就失去了这个推荐系统的意义。

## 项目优缺点

每个系统都会有优缺点，我这个系统的优点可能就是现在市面上关于这部分的推荐系统比较少，所以有一定的开发价值。作为一个有很多衣服、很多化妆品的用户，我在使用那么多app时，发现这些app只能让我不断的种草，但是我应该如何安排我的衣柜，每天应该穿哪些衣服，化怎样的妆，用哪个眼影盘，涂哪只口红，都是我会花很多时间来决定的。所以这个系统可以解决我现在迫切的需求，也能帮我节省很多时间。

关于系统的缺点，其实还有很多。比如，搜索框只能实现模糊查询，搜索时只能搜索出完全包含输入内容的关键字，而且搜索关键字的顺序不能变，所以在搜索框的部分还比较欠缺动态性的搜索，不能让用户在搜索时使用更自由；在图片上传部分，命名的方式我选择了简单粗暴的在26个英文字母+10个阿拉伯数字中，随机选择6位数来为图片文件命名，这样可能会出现随机出来的命名是一样的，虽然这种情况概率很小但也存在；在前端的设计方面，我觉得还是欠缺一些动态的实现效果，所以我的网站在显示时会觉得有些木讷，而且作为一个女生，能给网页换皮肤的话就太好了，但我暂时还没能实现这个功能；我写这个系统时，考虑到的是网站上的应用，所以如果在手机上使用这个系统，element-ui可能就无法适应手机的尺寸，这样网站的使用端就会有限制，不太利于网站的传播和推广使用。

## 个人感想

这是一个我大学四年来，第一个在意义上几乎是我独立完成的系统。从一开始的确定题目和系统方向，到后来的设计前端，部署后端。前前后后一个月的时间，终于把我一开始想要做的系统给初步的实现了。

作为一个热爱美妆的女生，我一直都希望可以推广美妆。而且我还是一个工科的女生，之所以坚持要做有关美妆的系统，也是因为我想稍稍的改变一下这个社会对于理工科女生的刻板印象。大家听说理工科女生，可能马上就想到了戴眼镜、不化妆、穿着中性，但是，就算是学理工科的女生，那也是女生。既然是女孩，就没有不爱美、不爱打扮的。而我们理工科的女生，不仅同样热爱美妆，还能以自己的需求出发，做出一个和美妆有关的系统。

在做这个系统的过程中，有很多知识都是新学的，比如前后端分离，比如vue框架。在学习过程中，我也收获到了很多。也再一次认识到了编程写代码的魅力。我竟然可以通过自己的双手，把自己想要的画面用代码敲出来。虽然我的水平还有很多不足，不能够进一步实现我的天马行空的想象，但我仍在学习。IT行业有一个非常大的特点就是，全行业的人都在不断学习。因为每天都有新的编程语言出现，每天我们所用的框架都在更新，甚至今天我们用的框架，明天就会被另一个更好、更高效能的框架替代，只有不断学习，不断接受新的语言系统，不断使用新的框架，才能不被这个行业淘汰。

IT行业是欣欣向上的，整个行业都在为人们提供便利。对于普通使用者来说，日新月异、层出不穷的各个网站和APP丰富了我们的生活，提高了我们的效率。就比如，因为疫情的关系，很多中小学的学生都需要在家上网课来学习，这个时候，钉钉、腾讯会议等APP为老师和学生提供了很大的帮助。如果没有这些直播软件，可能会有其他的方法来让孩子们在家学习，但效率也许就没有那么高了。我觉得，编程的意义可能就在这里吧。我们可以把自己的所思所想展现出来，并且帮到他人甚至帮助这个社会，让社会可以更高效、快捷的发展。虽然做IT会头秃，但我们用发际线来替换的是整个社会的便利。这就是我选择这个专业的原因。

参考文献

1. [] 南方医药经济研究所经济研究室 汇编. 比全球增速高7.3%中国化妆品空间诱人[N]. 医药经济报,2019-11-11(F02). [↑](#endnote-ref-1)
2. [] 朱彦婕.化妆品：市场空间可观 电商渠道助力[J].股市动态分析,2019(41):42-43. [↑](#endnote-ref-2)
3. [] 马闻锴,李贵,李征宇,韩子扬,曹科研.一种基于标签的Top-N个性化推荐算法[J].计算机科学,2019,46(S2):224-229. [↑](#endnote-ref-3)
4. [] 丁晓东.用户画像、个性化推荐与个人信息保护[J].环球法律评论,2019,41(05):82-96. [↑](#endnote-ref-4)
5. [] Color Cosmetics Products Market 2018 Global Top players, Sales, Supply and Consumption Forecast to 2025[J]. M2 Presswire,2018. [↑](#endnote-ref-5)
6. [] Java Cryptography Architecture and Information Security[J].International Journal of Systems and Control,2008(02):111-120. [↑](#endnote-ref-6)
7. [] 李宇,刘彬.前后端分离框架在软件设计中的应用[J].无线互联科技,2018,15(17):41-42. [↑](#endnote-ref-7)
8. [] 吕英华.渐进式JavaScript框架Vue.js的全家桶应用[J].电子技术与软件工程,2019(22):39-40 [↑](#endnote-ref-8)
9. [] 李广宏.vue.js前端应用技术分析[J].中国新通信,2019,21(20):115 [↑](#endnote-ref-9)
10. [] 陈倩怡,何军.Vue+Springboot+MyBatis技术应用解析[J].电脑编程技巧与维护,2020(01):14-15+28. [↑](#endnote-ref-10)
11. [] 张云超. Windows下钩子函数的应用[J]. 新浪潮, 1995, 000(012):1-2. [↑](#endnote-ref-11)
12. [] 赵炳义.高德联手中国气象局 助力雨天安全出行[J].计算机与网络,2018,44(13):35. [↑](#endnote-ref-12)