

商品比价网站 设计文档

3220102185 王阳

商品比价网站 设计文档

3220102185 王阳

一、网站功能逻辑设计

二、网站数据表设计

三、网站前后端接口设计

3.1 注册

3.2 登录

3.3 修改用户信息

3.4 添加淘宝账号

3.5 添加京东账号

3.6 获取二维码

3.7 搜索商品

3.8 获取说明文档内容

3.9 获取用户搜索历史记录

3.10 订阅商品

3.11 检查订阅商品价格变动

3.12 搜索商品详细信息

四、后端框架设计

五、前端框架设计

一、网站功能逻辑设计

网站需要实现的功能：

1. 注册、登录

- 用户名、密码要求在 6 字节以上
- email 的格式验证
- 保证用户名和 email 在系统中唯一

2. 通过商品名称在主流电商平台上查询该商品实时价格

- 商品名称进行分词处理优化查询
- 查询多个结果的处理

- 需要实现购物平台自动登录查询
- 支持至少 2 个以上平台查询价格进行比较（淘宝、京东等）。本项目选取淘宝和京东两个购物平台

3. 商品库

- 将商品信息和商品价格保存在数据库中
- 商品信息包含名称、多级品类、规格、条码、图片等

4. 商品查询界面

- 显示商品信息
- 把历史价格用图表形式显示（如价格走势图）。

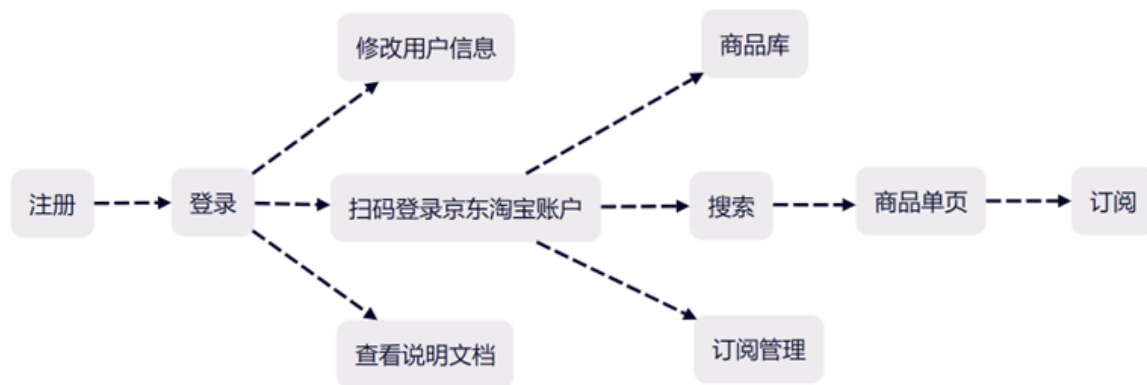
5. 设置降价提醒，针对指定商品定时查询最新价格，如有降价发送提醒，通过邮件方式实现。

6. 样式适配手机，能够在手机浏览器/微信等应用内置的浏览器中友好显示

网站前后端交互逻辑图：



网站功能逻辑图：

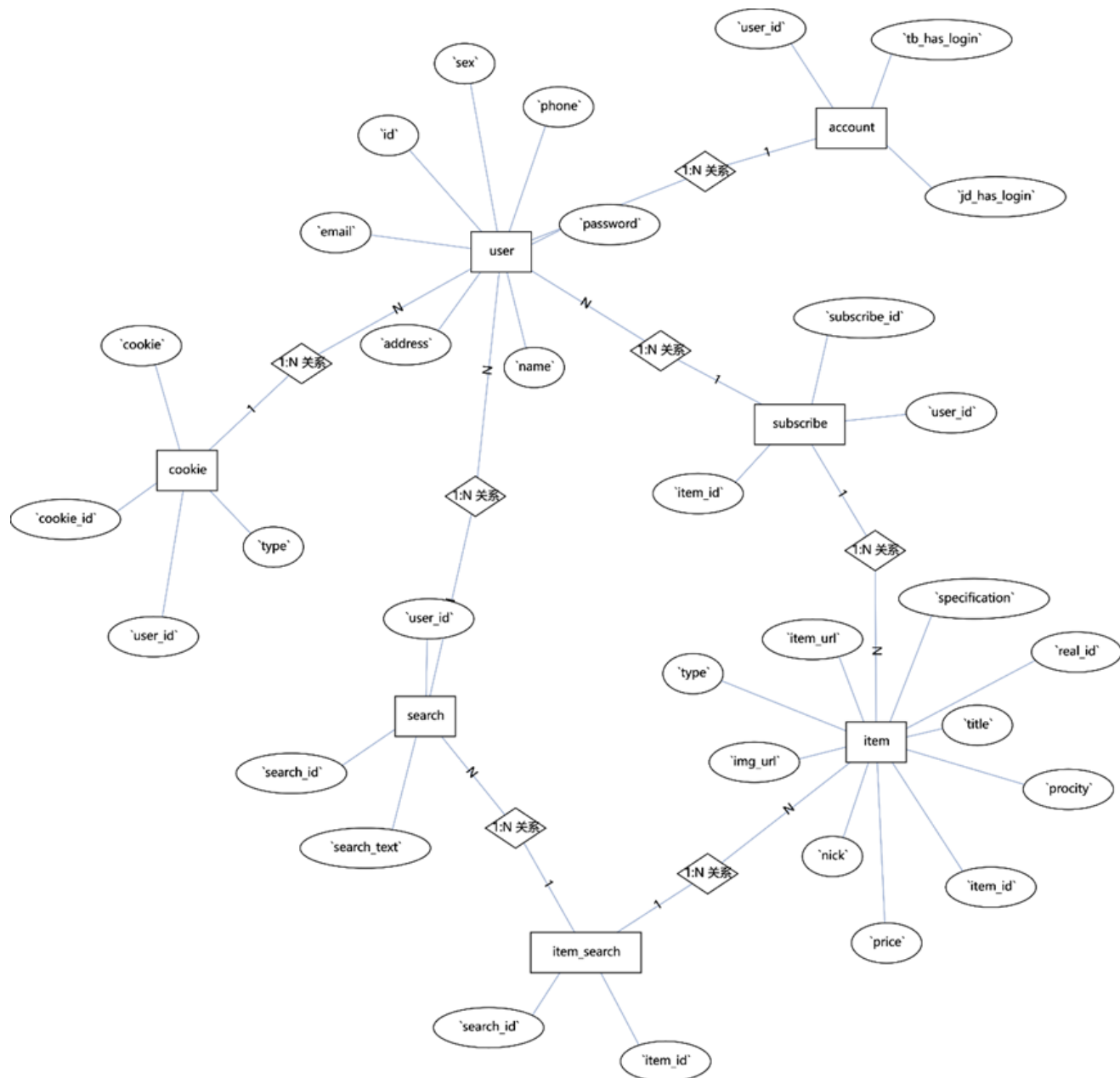


二、网站数据表设计

本项目使用MySQL数据库，共有7张表：

1. User：存储用户信息
2. Account：存储用户京东淘宝账号信息
3. Cookie：存储用户京东淘宝cookies
4. Item：商品
5. Search：搜索记录
6. item_search：search与item的关联表
7. subscribe：订阅信息

数据表的ER图如下：



三、网站前后端接口设计

3.1 注册

1. 接口信息

- 接口路径: `/register`
- 请求方法: `POST`
- 接口描述: 用户注册接口，用于创建新用户

2. 请求体格式

- JSON 格式发送请求体
- 包含以下字段

字段名	类型	必填	描述
name	string	是	用户名
password	string	是	密码
email	string	是	邮箱地址
sex	integer	是	性别
address	string	是	地址
phone	string	是	电话号码

3. 响应返回

- 成功响应

```
{
  'code': 200,
  'message': 'User created
successfully',
  'data': {
    'id': new_user.id,
    'name': new_user.name,
    'email': new_user.email,
    'phone': new_user.phone,
    'address': new_user.address
  }
}
```

- 失败响应：用户名重复

```
{
  'code': 409,
  'message': 'User created
successfully'
}
```

- 失败响应：邮箱重复

```
{
  'code': 409,
  'message': 'Email created
successfully'
}
```

3.2 登录

1. 接口信息

- 接口路径： `/login`
- 请求方法： `POST`
- 接口描述： 用户登录接口，用于验证用户身份并返回登录信息。

2. 请求体格式

- JSON 格式发送请求体。
- 包含以下字段

字段名	类型	必填	描述
name	string	是	用户名
password	string	是	密码

3. 响应返回

- 成功响应

```
{
  "code": 200,
  "message": "Login successful",
  "data": {
    "id": user.id,
    "name": user.name,
    "email": user.email,
    "phone": user.phone,
    "address": user.address,
    "sex": user.sex,
    "cookie": a string
  }
}
```

- 失败响应：密码错误

```
{
  "code": 401,
  "message": "password error"
}
```

- 失败响应：无效凭证

```
{
  "code": 401,
  "message": "Invalid credentials"
}
```


3.3 修改用户信息

1. 接口信息

- 接口路径: `/modifyUser`
- 请求方法: `POST`
- 接口描述: 允许已认证用户修改其个人信息, 包括用户名、邮箱、电话、地址和性别。

2. 请求体格式

- JSON 格式发送请求体。
- 包含以下字段

字段名	类型	必填	描述
id	integer	是	用户ID
modifiedUser	object	是	包含修改后的用户信息

modifiedUser 对象包含:

字段名	类型	必填	描述
name	string	否	新用户名
email	string	否	新邮箱地址
phone	string	否	新电话号码
address	string	否	新地址
sex	string	否	新性别

示例请求体:

```
{
  "id": 1,
  "modifiedUser": {
    "name": "new_username",
    "email": "new_email@example.com",
    "phone": "1234567890",
    "address": "123 New Street",
    "sex": 1
  }
}
```

3. 响应返回

- 成功响应

```
{
  "code": 200,
  "message": "User modified successfully",
  "data": {
    "id": 1,
    "name": "new_username",
    "email": "new_email@example.com",
    "phone": "1234567890",
    "address": "123 New Street",
    "sex": 1
  }
}
```

- 失败响应：用户名已存在

```
{
  "code": 409,
  "message": "This name is already taken"
}
```

- 失败响应：邮箱已存在

```
{
  "code": 409,
  "message": "This email is already taken"
}
```

3.4 添加淘宝账号

1. 接口信息

- 接口路径： `/addTb`
- 请求方法： `POST`
- 接口描述： 允许用户添加他们的淘宝账号信息，包括用户名和密码，后端通过账密进行淘宝网自动登录，自动获取 cookie 并保存

2. 请求体格式

- JSON 格式发送请求体。
- 包含以下字段

字段名	类型	必填	描述
user_id	integer	是	用户ID
info_tb	object	是	淘宝账号信息

info_tb 对象包含：

字段名	类型	必填	描述
tb_name	string	是	淘宝用户名

t_name	string	是否必填	淘宝用户名
字段名	类型		描述
t_password	string	是	淘宝密码

示例请求体：

```
{
  "user_id": 1,
  "info_tb": {
    "t_name": "taobao_username",
    "t_password": "taobao_password"
  }
}
```

3. 响应返回

- 成功响应

```
{
  "code": 200,
  "message": "添加淘宝账号成功"
}
```

- 失败响应：账号密码错误

```
{
  "code": 401,
  "message": "淘宝账号添加失败，可能账号密码错误"
}
```

3.5 添加京东账号

1. 接口信息

- 接口路径: `/addJd`
- 请求方法: `POST`
- 接口描述: 允许用户添加他们的京东账号信息, 通过扫码登录并获取相关的Cookies。需要配合 `get_qrcode` 实现登录京东功能。

2. 请求体格式

- JSON 格式发送请求体。
- 包含以下字段

字段名	类型	必填	描述
<code>user_id</code>	<code>integer</code>	是	用户ID

示例请求体:

```
{
  "user_id": 1
}
```

3. 响应返回

- 成功响应

```
{
  "code": 200,
  "message": "添加京东账号成功"
}
```

- 失败响应: 超时

```
{
  "code": 401,
  "message": "超时，请重新尝试扫码登录"
}
```

3.6 获取二维码

1. 接口信息

- 接口路径: `/getQrcode`
- 请求方法: `GET`
- 接口描述: 用于获取京东账号扫码登录的二维码。接口会等待直到二维码生成或超时。

2. 请求参数

- 无需请求参数。

3. 响应返回

- 成功响应

```
{
  "code": 200,
  "message": "成功得到二维码",
  "data": "二维码Base64字符串"
}
```

- 失败响应: 未获得二维码

```
{
  "code": 401,
  "message": "没有获得到二维码"
}
```

3.7 搜索商品

1. 接口信息

- 接口路径: `/search`
- 请求方法: `GET`
- 接口描述: 根据用户ID和搜索文本, 搜索商品信息。如果搜索库中有匹配的记录, 则从库中获取商品信息; 否则, 使用用户的淘宝和京东账号爬取商品数据。

2. 请求参数

- 用户ID (`id`): 用户的唯一标识符。
- 搜索文本 (`searchText`): 用户输入的搜索关键词。

3. 响应返回

- 成功响应: 从搜索库中查询

```
{
  "code": 200,
  "message": "成功搜索到商品,从搜索库中查询",
  "data": [
    {
      "item_id": 1,
      "real_id": "123456",
      "title": "商品标题",
      "type": "商品类型",
      "price": 99.99,
      "nick": "卖家昵称",
      "item_url": "商品链接",
      "img_url": "商品图片链接",
      "procity": "商品省份"
    },
    ...
  ]
}
```

- 成功响应：爬取淘宝和京东商品数据

```
{
  "code": 200,
  "message": "搜索成功",
  "data": [
    {
      "item_id": 1,
      "real_id": "123456",
      "title": "商品标题",
      "type": "商品类型",
      "price": 99.99,
      "nick": "卖家昵称",
      "item_url": "商品链接",
      "img_url": "商品图片链接",
      "procity": "商品省份"
    },
    ...
  ]
}
```

- 失败响应：未设置淘宝账号

```
{
  "code": 401,
  "message": "您没有设置淘宝的账号，无法爬取淘宝商品数据"
}
```

- 失败响应：未设置京东账号


```
{
  "code": 401,
  "message": "您没有设置京东的账号，无法爬取京东商品数据"
}
```

3.8 获取说明文档内容

1. 接口信息

- 接口路径: `/getDoc`
- 请求方法: `GET`
- 接口描述: 获取项目相关的文档内容，通常是 `doc.md` 文件中的内容。

2. 请求参数

- 无需请求参数。

3. 响应返回

- 成功响应

```
{
  "code": 200,
  "message": "获取数据成功",
  "data": "这里是doc.md文件的Markdown内容"
}
```

3.9 获取用户搜索历史记录

1. 接口信息

- 接口路径: `/history`
- 请求方法: `GET`
- 接口描述: 获取指定用户的搜索历史记录，包括搜索内容、相关商品ID列表以及商品数量。

2. 请求参数

- 用户**ID** (`user_id`): 用户的唯一标识符, 作为查询参数。

3. 响应返回

- 成功响应

json

```
{
  "code": 200,
  "message": "查找记录成功",
  "data": [
    {
      "search_id": 1,
      "search_text": "搜索关键词",
      "item_ids": [1, 2, 3],
      "num_items": 3
    },
    ...
  ]
}
```

3.10 订阅商品

1. 接口信息

- 接口路径: `/subscribe`
- 请求方法: `POST`
- 接口描述: 允许用户订阅特定的商品, 以便在商品价格变动时收到通知。

2. 请求体格式

- JSON 格式发送请求体。
- 包含以下字段

字段名	类型	必填	描述
-----	----	----	----

字段名	类型	必填	描述
user_id	integer	是	用户ID
item_id	integer	是	商品ID

示例请求体：

```
{
  "user_id": 1,
  "item_id": 101
}
```

3. 响应返回

- 成功响应

```
{
  "code": 200,
  "message": "订阅成功"
}
```

- 失败响应：用户不存在

```
{
  "code": 409,
  "message": "该用户不存在"
}
```

- 失败响应：商品不存在

```
{
  "code": 409,
  "message": "该商品不存在"
}
```

3.11 检查订阅商品价格变动

1. 接口信息

- 接口路径: `/checkSubscribe`
- 请求方法: `POST`
- 接口描述: 检查用户订阅的商品价格是否有变动, 如果有降价, 通过邮件通知用户。

2. 请求体格式

- JSON 格式发送请求体。
- 包含以下字段

字段名	类型	必填	描述
<code>user_id</code>	<code>integer</code>	是	用户ID

示例请求体:

```
{
  "user_id": 1
}
```

3. 响应返回

- 成功响应: 没有检测到商品降价

```
{
  "code": 200,
  "message": "没有检测到商品降价"
}
```

- 成功响应: 检测到商品降价, 邮件已发送

```
{
  "code": 200,
  "message": "检测到商品降价，具体信息已发送给您的邮箱"
}
```

- 失败响应：未设置淘宝账号

```
{
  "code": 401,
  "message": "您没有设置淘宝的账号，无法爬取淘宝商品数据"
}
```

- 失败响应：未设置京东账号

```
{
  "code": 401,
  "message": "您没有设置京东的账号，无法爬取京东商品数据"
}
```

3.12 搜索商品详细信息

1. 接口信息

- 接口路径： `/searchItem`
- 请求方法： `GET`
- 接口描述：根据用户ID和商品ID，获取商品的详细信息，包括价格历史和价格走势图。

2. 请求参数

- 用户ID (`user_id`): 用户的唯一标识符，作为查询参数。

- 商品**ID** (**item_id**): 商品的唯一标识符, 作为查询参数。

3. 响应返回

- 成功响应

```
{
  "code": 200,
  "message": "详情获取成功",
  "data": {
    "id": 1,
    "real_id": "商品真实ID",
    "title": "商品标题",
    "price": 99.99,
    "nick": "卖家昵称",
    "item_url": "商品链接",
    "img_url": "商品图片链接",
    "procity": "商品省份",
    "specification": "商品规格",
    "history_image": "价格走势图的Base64字符串"
  }
}
```

- 失败响应: 商品不存在

```
{
  "code": 401,
  "message": "没有找到该商品"
}
```

- 失败响应: 未设置淘宝账号

```
{
  "code": 401,
  "message": "您没有设置淘宝的账号，无法爬取淘
宝商品数据"
}
```

四、后端框架设计

```
bs_backend
| app.log          调试用：输出日志
| app.py           flask项目入口
| Dockerfile
| requirements.txt python环境
| start.sh         Docker打包该组件命令行执行脚本
└─myapp            flask应用
    | config.py     配置类
    | doc.md        说明文档
    | models.py     数据类
    | plugin.py     插件
    | __init__.py
    |
    └─routes        路由视图文件夹
        | views.py  主要功能，实现各个接口
        | __init__.py
        |
        └─utils
            ApiResponse.py 返回响应类
            drission.py    爬虫部分
            email.py       发送邮件
```

五、前端框架设计

```
bs_frontend
| .gitignore
| Dockerfile
| index.html
| nginx.conf
| package-lock.json
| package.json
| README.md
| vite.config.js

├─public
|   | vite.svg
|   |
|   └─static
|       login-background.png
|       QrcodeFailed.png
|
└─src
    | App.vue
    | axios.js
    | main.js           前端入口
    | permission.js     路由守卫
    | style.css
    |
    ├─api
    |     api.js         axios请求函数
    |
    ├─assets
    |     vue.svg
    |
    ├─composables
    |     cookie.js      实现cookie相关功能
```


	utils.js	实现消息窗口
	└─layouts	
	Layout.vue	布局
	└─components	
	FHeader.vue	网页布局头部
	FMenuH.vue	网页布局菜单水平
	FMenuV.vue	网页布局菜单垂直
	└─router	
	index.js	路由
	└─store	
	index.js	存储信息
	└─views	
	Document.vue	教程
	History.vue	商品库
	Login.vue	登录
	LoginTJ.vue	登录京东淘宝
	Register.vue	注册
	Search.vue	搜索与商品详情
	Subscribe.vue	订阅管理
	User.vue	用户管理