

浙江大学

本科实验报告

课程名称: B/S 体系软件设计

姓 名: 蔡佳伟

学 院: 计算机科学与技术学院

系: 计算机科学与技术系

专 业: 软件工程

学 号: 3220104519

指导教师: 胡晓军

2024 年 11 月 11 日

浙江大学实验报告

实验名称： B/S 体系软件设计 实验类型：

实验项目名称： 商品比价网站

学生姓名： 蔡佳伟 专业： 软件工程 学号： 3220104519

同组学生姓名： 无 指导老师： 胡晓军

实验地点： 实验日期： 2024 年 11 月 11 日

1. 文档介绍

1.1. 编写目的

本文档描述商品比价网站的目的是：

- 1) 定义网站总体要求，作为用户和网站开发人员之间相互了解的基础；
- 2) 提供性能要求、初步设计和用户影响的信息，作为网站设计人员进行网站结构设计和编码的基础；
- 3) 作为网站总体测试的依据。

1.2. 文档范围

商品比价网站功能设计文档旨在为商品比价网站的具体实现提供一个完整、可行、详细的总体设计方案，并综合考虑用户需求，网站设计条件和限制，对该网站的功能目标和功能操作方式进行基本的描述。本文档将根据总体商品比价网站的系统架构图对该进行全面、细致的设计分析，其内容包括系统实现技术栈，系统功能模块设计、执行概念设计、流程处理设计、数据结构和类设计、数据库关系模型设计、数据库逻辑结构设计、数据库物理结构设计、系统内外接口设计、系统部署方案设计等。此外，本文档还会为系统出错提供一个基本的解决指导和简单说明。

本文档设计说明书的预期读者包括：

- 1) 客户
- 2) 系统开发人员
- 3) 测试人员
- 4) 需求分析人员

- 5) 项目管理人员
- 6) 系统维护人员

1.3. 读者对象

编写详细设计人员及程序开发人员、XXXX、YYYY……

1.4. 术语与缩写解释

缩写、术语及符号	解释
SOA 架构	面向服务的体系结构。
元数据 Metadata	描述数据的内容、质量、状况和其他有关特征的数据。
数据中心 Data Center	以各类数据为核心，依托成熟的存储、数据库、GIS、网络等技术，按照统一标准，建立的具有信息管理、分析、查询、统计及服务的一体化数据管理体系。
数据管理 Data Management	利用数据库、数据仓库、元数据和网络等技术，建立分布式、集中式或集中加分布式数据管理系统，开展数据接收、组织存储、运行维护、更新、共享交换等工作，实现对数据资源的有效组织和应用。
顺序图	顺序图也称为时序图，它描述了系统中，对象间通过消息 进行的交互，它强调了消息在时间轴上的先后顺序。常用 来描述用例的实现，它表明了由哪些对象，通过消息相互 协作来实现用例的功能，并标识了消息发生交互的先后顺 序。
状态图	状态图描绘一个系统或组件可能假设的状态，并且显示引起 或导致一个状态切换到另一个状态的事件或环境。
类图	类图描述了一个软件系统中的类、接口、关系以及它们之间 的静态结构。类图用于表示系统中的对象、类以及它们之间 的关系，它提供了一种直观、可视化的方式来表示系统的结 构
IPO图	IPO 是指结构化设计中变换型结构的输入（Input）、加工（Processing）、输出（Output）。IPO 图是对每个模块进行详细设计的工具，它是输入加工输出(INPUT PROCESS OUTPUT)图的简称。在系统的模块结构图形成过程中，产生了大量的模块，在进行详细设计时开发者应为每一个模块写一份说明，IPO 图就是用来说明每个模块的输入、输出数据和数据加工的重要工具。

1.5. 参考资料

序号	文档名称	文档编号	版本	发布日期
1	assignment (2024秋冬)	Doc01	1.0	2024. 9. 20

2. 项目介绍

2.1. 项目说明

项目名称：商品比价网站

任务提出者：浙江大学B/S体系软件设计任课老师-胡晓军

开发者：3220104519 蔡佳伟

用户群：网站使用者，维护者等

2.2. 项目背景

现今随着电商的发展，网络购物已经融入人们的生活。但网上商品太多，若要买到达到最好满意度的商品，用户须花费大量时间精力去对比。为了解决用户在购物时需打开多个网站，不断自行比较的繁琐问题，为了节省用户购物时间，优化用户的购物体验，给用户带来更愉悦的购物经历, 商品比价网站应运而生，致力于方便快捷地解决用户的痛点问题，提高用户的购物体验。

2.3. 需求概述

2.3.1 功能需求

2	模块名称	用户登录模块		
	模块简介	用户注册、登录模块		
模块功能列表				
序号	一级功能		二级功能	
	功能名称	功能编号	功能名称	功能编号
1	用户注册	Zc01	用户注册网站	Zc0101
2	用户登录	Zc02	用户注册时验证	Zc0201
3			用户登录验证	Zc0202

模块名称		商品查询实时价格		
模块简介		商品查询价格，多个结果处理，多平台查询比较		
模块功能列表				
序号	一级功能		二级功能	
	功能名称	功能编号	功能名称	功能编号
1	商品查询	Cx01	查询多个结果处理	Cx0101
2	商品价格比较	Cx02	分词处理优化查询	Cx0201
3			多平台查询价格比较	Cx0202

模块名称		商品库建立		
模块简介		商品库建立便于后续查询，显示历史价格		
模块功能列表				
序号	一级功能		二级功能	
	功能名称	功能编号	功能名称	功能编号
1	商品信息保存进商品库	Ku01	商品信息保存	Ku0101
2	查询商品信息	Ku02	显示商品信息	Ku0201
			显示历史价格和图表	Ku0202

模块名称		针对指定商品降价提醒，查询价格	
模块简介		支持设置降价提醒，针对指定商品定时查询最新价格，如有降价发送提醒，可以通过邮件，App 推送等方式实现	
模块功能列表			
序	一级功能		二级功能

号	功能名称	功能编号	功能名称	功能编号
1	针对指定商品定时查询价格	Ds01	定时查询商品价格	Ds0101
2	发送提醒	Ds02	邮件方式实现	Ds0201
			App方式实现	Ds0202

开发完成后补充：查询多个结果处理：可以按照价格或按照默认顺序排序，没有开发App，所以没有App方式实现价格。

2.3.2 性能需求

2.3.2.1 总体性能需求

（1）系统配置

系统要具有良好的反应速度，课题要求在良好的网络情况下，本系统应该具有如下时间特性要求：
单个用户在线时：1. Web响应用户动作时间小于1秒。2. 信息搜索操作响应用户动作时间小于2秒。500个用户同时在线时：1. Web响应用户动作时间小于2秒。2. 信息搜索操作响应用户动作时间小于5秒。

（2）访问容量

该系统至少在同一时间内支持200个用户并发访问。

（3）服务器配置最低要求

CPU4核2.6G，内存8.0G，硬盘7200转。

（4）可用性

该系统应实现多Web浏览器支持：在大多数流行的Web浏览器中正确显示和执行，包括 Firefox、Chrome、Edge 等。同时还需要在手机浏览器/微信等应用内置的浏览器中友好显示。

2.3.2.2 各部分性能需求

（1）用户登录模块

访问容量：该模块需要支持多个用户同时进行注册、登录等操作，因此需要具备高并发处理能力
响应时间：该模块需要支持多种流行的Web浏览器，以确保用户可以在不同的浏览器上正确显示和执行注册、登录功能。
数据处理效率：该模块需要具备高效的数据处理能力，包括信息展示，检索、数据导入导出等，以提高用户使用体验。

（2）商品查询实时价格模块

访问容量：该模块需要支持多用户查询商品实时价格等操作，因此需要具备高并发处理能力。

响应时间：该模块需要支持多种流行的Web浏览器，以确保用户可以在不同的浏览器上正确显示和执行查询功能。

数据处理效率：该模块需要具备高效的数据处理能力，包括信息展示，检索、数据导入导出等，因此需要具备一定的并发处理能力。

(3) 商品库建立模块

访问容量：该模块需要支持商品库建立，对商品库商品查询，显示价格历史图表等，因此需要具备高并发处理能力。

响应时间：该模块需要支持多种流行的Web浏览器，以确保用户可以在不同的浏览器上正确显示和执行商品库建立和应用商品库功能。

数据处理效率：该模块需要具备高效的数据处理能力，包括信息展示，检索、数据导入导出等。

(4) 针对指定商品降价提醒，查询价格模块

访问容量：该模块需要支持针对指定商品降价提醒，查询价格模块等操作，因此需要具备高并发处理能力

响应时间：该模块需要支持多种流行的Web浏览器，以确保用户可以在不同的浏览器上正确显示和执行预订和查询功能。

数据处理效率：该模块需要具备高效的数据处理能力，包括信息展示，检索、数据导入导出等，因此需要具备一定的并发处理能力。

2.3.3.3 安全需求

用于身份验证的用户名和密码应防止未经授权的用户访问系统。应构建访问控制以防止合法用户非法使用系统资源。某些敏感数据（如用户名，密码）在交换时应加密。密码在存储之前应加密。

在用户登录期间，应该防止 SQL 注入，密码强制破解和伪造会话入侵。

1. 完整性：防止非法用户对数据进行无意或恶意的修改、插入、删除，防止数据丢失。
2. 约束性：为数据库加上一定的约束，对关键性操作如删除、修改进行限制，并对用户进行警示。不同身份所拥有的权限不同，只可以进行自己权限内的操作。
3. 账户信息安全性：着重账户信息安全性设计，做到外界人员无法入侵到系统本身。内部人员操作需要留下操作痕迹，使用权管理层可以定期或不定期地维护系统。
4. 保密性：保证没有授权的用户不能修改外币种类和汇率，也不能修改数据操作员的账号和权限，数据操作员不能修改其他操作员和数据管理员的账号和权限。
5. 数据加密：系统应该使用加密技术来保护敏感数据的机密性和完整性。例如，在传输过程中，可以使用SSL/TLS协议来加密数据，以防止未经授权的用户访问数据。在存储过程中，可以使用加密算法对密码登敏感数据进行加密，以防止数据泄露。

6. 防止攻击：系统应该实现安全措施来防止各种攻击，如SQL注入，密码强制破解，会话劫持等。这使用防火墙和入侵监测系统登记书和安全编码实践来实现。

7. 安全审计：系统应该记录所有用户的操作，以便在发生安全事件时进行调查和审计。这可以通过使用日志记录和审计工具来实现，以便管理员可以检查系统中发生的所有事件，并确定是否存在安全问题。

2.4. 条件与限制

由于实验条件有限，我们并不能提供专门的服务器运行系统，故将利用配置较高的PC作为服务器，保证服务器以及客户端间网络畅通即可。

计算机：

- 1) CPU：不小于2.0GHz
- 2) 内存：不小于2.0GB

通讯设备：

- 1) 网线：正常联通且数据传输能力好
- 2) 网卡：100M

客户端外围设备：

- 1) 键盘：可正常使用
- 2) 鼠标：可正常使用
- 3) 显示器：全彩显示，可正常使用
- 4) 硬盘：大于100GB，转速大于7200rpm

软件依赖：

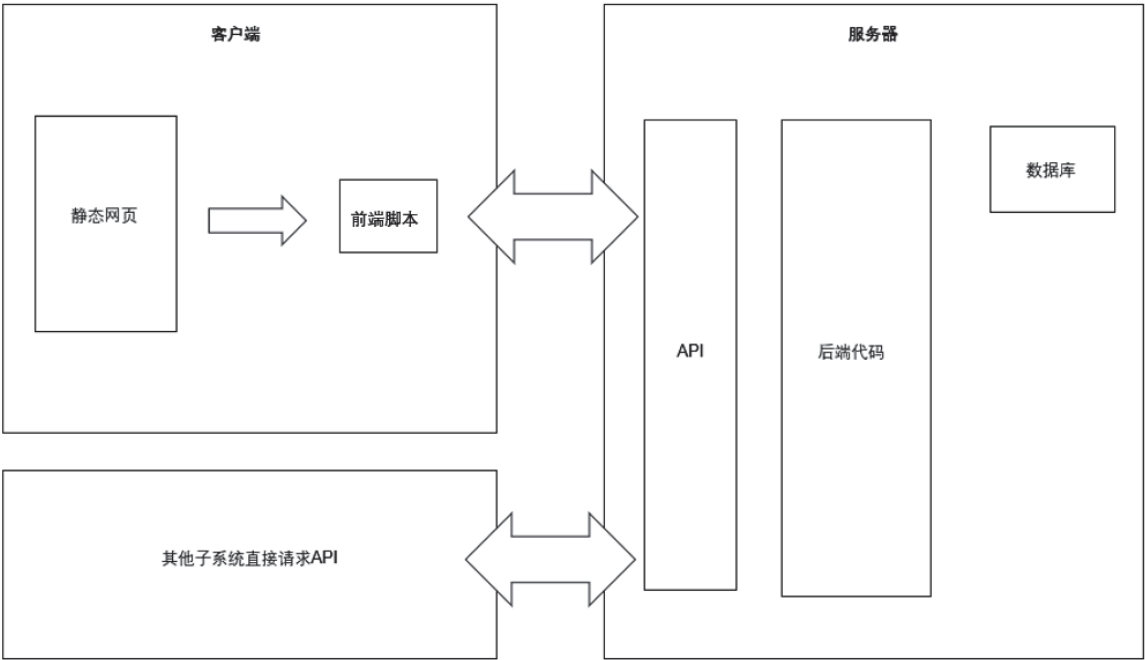
- 1) 操作系统：Windows10/11, Mac OS, Linux
- 2) 数据库平台：MySQL
- 3) 前端开发框架：Vue3
- 4) 后端语言和框架：Python3.10 FastAPI
- 5) 数据库操作：sqlite
- 6) 开发工具：能支持网页开发的工具均可（如PyCharm, vscode等）
- 7) 建模工具：Microsoft Word, Microsoft ppt, draw.io等
- 8) 浏览器：Chrome、Edge、FireFox等

3. 总体设计

3.1. 基本设计概念和流程处理

商品比价系统是一个横跨前后端的系统网站，主要任务有用户登录，商品查询实时价格，商品库建立，针对特定商品实时降价提醒等功能。

处理流程图如下：



3.2. 技术介绍

Vue3 + FastAPI + SQLite

1. Vue3 框架

Vue.js 是一款流行的前端 JavaScript 框架，它被设计用于构建交互式的 Web 用户界面。Vue3 是 Vue.js 的最新版本，具有更快的性能、更小的体积和更强大的功能。它采用了响应式数据绑定和组件化的开发模式，使得前端开发更加简单、高效和灵活。Vue3 的特点包括：

响应式数据绑定：Vue3 支持响应式数据绑定，当数据发生变化时，页面自动更新，提高了开发效率。

组件化开发：Vue3 提供了强大的组件化开发能力，可以将页面拆分成多个独立的组件，提高了代码的复用性和维护性。

虚拟 DOM：Vue3 使用虚拟 DOM 技术，通过比较虚拟 DOM 和实际 DOM 的差异来更新页面，提高了性能。

生态丰富：Vue3 拥有丰富的生态系统，包括 Vue Router、Vuex 等插件和工具，可以满足各种前端开发需求。

2. FastAPI 框架

FastAPI 是一个快速、易用的 Python Web 框架

FastAPI 的优势：

高性能：FastAPI 基于 Starlette 和 Pydantic，采用了现代的 Python 异步编程模型，支持异步操作，能够高效处理高并发请求，性能非常接近 Go 和 Node.js 等语言的 Web 框架。

易用性：FastAPI 提供了简单直观的 API 设计，开发者可以快速定义接口，并且框架自动生成 OpenAPI 和 JSON Schema 文档，极大简化了接口的编写和调试过程。

类型注解支持：FastAPI 深度集成 Python 的类型注解，通过类型推导提高了代码的可读性和维护性，减少了错误并增加了开发效率。

自动化文档生成：FastAPI 自动生成交互式的 API 文档，开发者和用户可以方便地查看接口文档并进行测试，减少了文档编写的工作量。

现代异步支持：支持异步请求处理，使得 Web 应用能够更好地处理并发请求，尤其适合 IO 密集型操作的场景，比如与数据库、外部 API 的交互

3. SQLite 数据库

SQLite 则是一个轻量级的关系型数据库。

轻量级：SQLite 是一个嵌入式数据库，无需独立的服务器进程，安装简单，配置少，适合轻量级应用或嵌入式设备。它的数据库文件可以直接存储在磁盘上，便于移动和备份。

零配置：SQLite 不需要复杂的配置和管理，相比于其他关系型数据库如 MySQL 或 PostgreSQL，开发者可以更快速地部署和开始开发。

广泛兼容：SQLite 被广泛支持，可以在多种操作系统和平台上运行，包括嵌入式设备和移动设备，使其在多种应用场景中都能使用。

事务支持：SQLite 支持 ACID 事务特性，保证了数据的一致性和安全性，适合需要存储结构化数据的小型项目或开发阶段的应用。

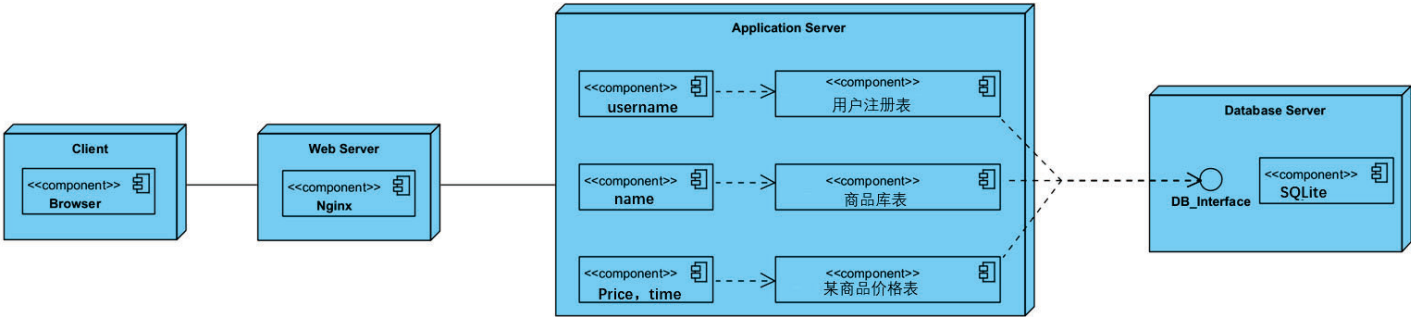
速度与效率：虽然 SQLite 是轻量级的，但它在处理小型到中型数据库时，性能非常优秀，查询和插入操作快

速高效。

总结

总的来说，Vue3 具有更快的性能、更小的体积和更强大的功能，FastAPI 提供了快速、简洁的后端开发体验，特别适合现代 Web 应用和高并发场景，而 SQLite 则是一个轻量级、高效的数据库解决方案，适合快速开发和小型项目的存储需求。三者结合使用，可以为开发者提供高效、简单的全栈开发体验。

3.3. 部署图



3.4. 接口设计

a) 硬件接口

- 1. 服务器：使用一台计算机作为服务器来托管和运行在线支付系统，需要具备至少 8G 运行内存、至少 100GB 的硬盘存储空间，具备数据库管理软件和相应的网络接口。
- 2. 网络接口：服务器通过以太网连接到网络，以便与客户端、支付网关和其他系统进行通信。因此， 需要一个以太网接口(路由器)来连接服务器和网络设备。
- 3. 客户端设备：用户需要通过某种客户端设备与在线支付系统进行交互，例如个人计算机、笔记本 电脑等。这些设备需要具备浏览器来访问和使用在线支付系统的界面。

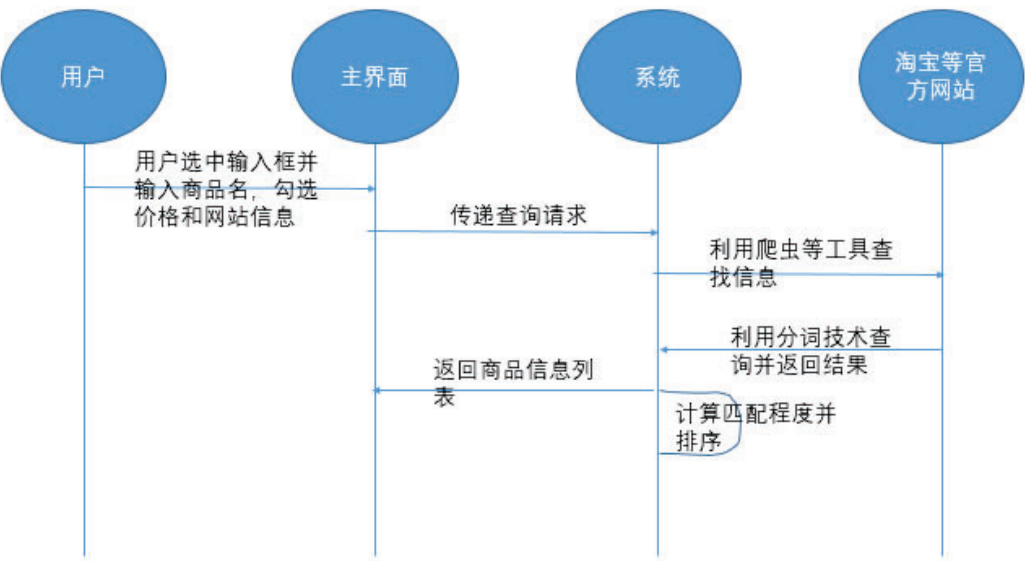
b) 软件接口

操作	接口
用户登录	POST /login
用户注册	POST /register

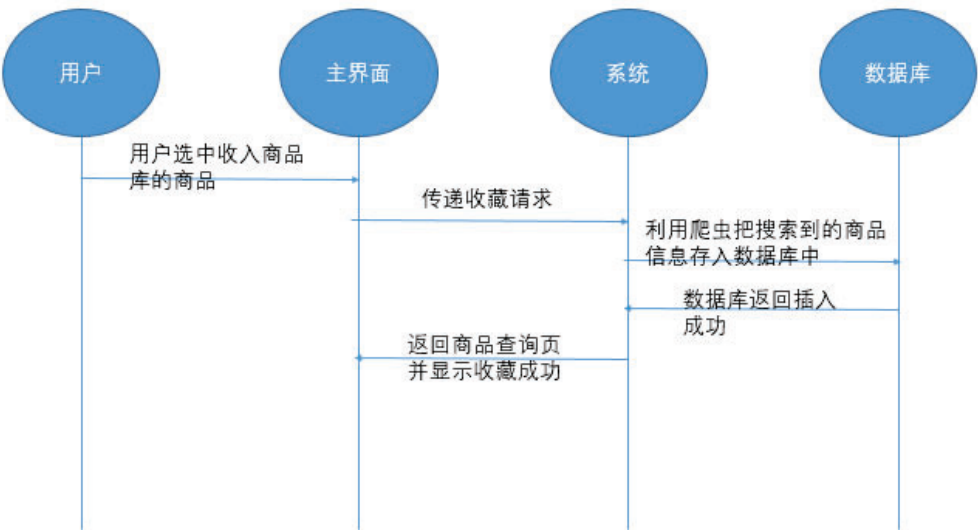
4. 详细设计

4.1 顺序图

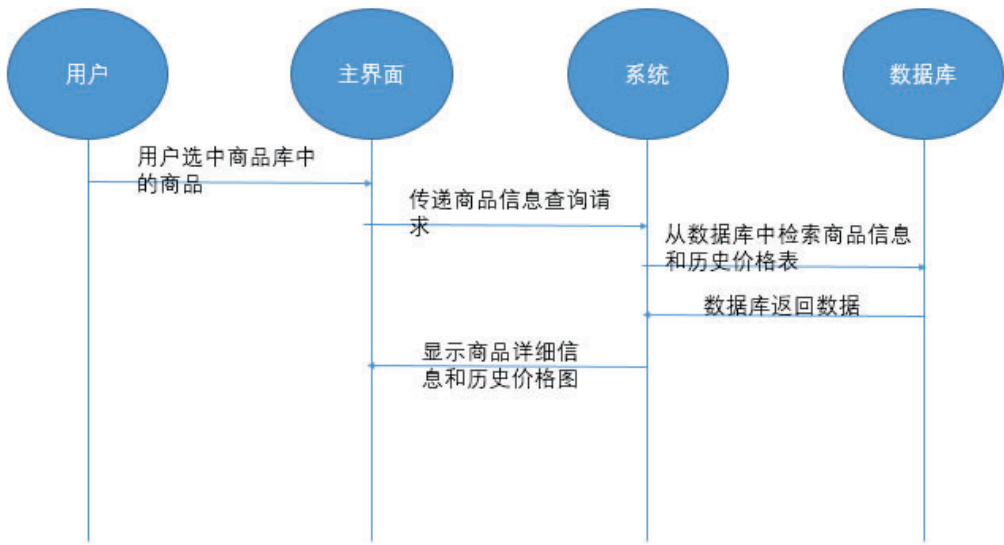
4.1.1 用户搜索商品



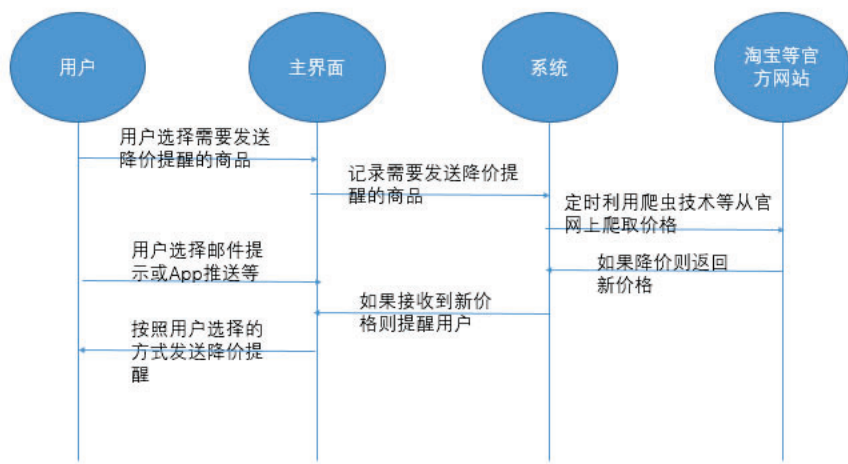
4. 1. 2 用户收藏商品进入商品库



4. 1. 3 用户查看商品详细信息



4.1.4 用户对商品设置降价提醒



4.2 执行概念

4.2.1 用户搜索商品

4.2.1.1 功能目标

在本模块中，用户在搜索框内输入商品名称，并且选择价格区间和期望查询的平台（淘宝，京东，苏宁），搜索到的相关商品信息就会显示在搜索界面。需要注意支持商品分词查询，并且很多平台需要用户登录后才能进行查询。

4.2.1.2 触发条件

以用户身份登录了商品比价网站，在搜索框中输入了内容并点击查询按钮。

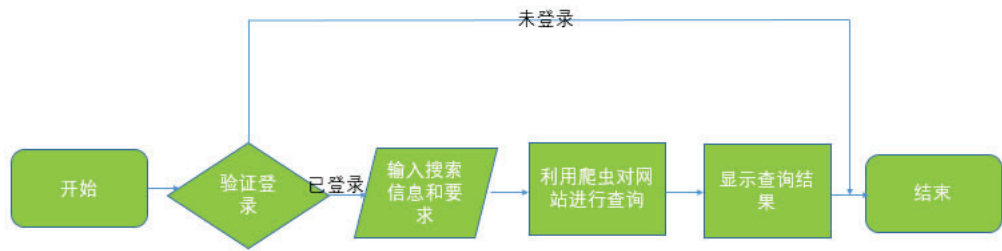
4.2.1.3 输入数据

名称	标识	类型	输入方式
商品名称	name	string	外部输入

4.2.1.4 输出结果

名称	标识	类型	输出方式
商品列表	Goods	Array	系统输出

4.2.1.5 流程图



4.2.1.6 测试计划

输入	预期结果
用户未登录	返回“请先登录再进行搜索!”, 弹出登录界面
登录, 输入商品, 勾选信息	按照匹配程度和价格高低返回商品列表

4.2.2 用户收藏商品进入商品库

4.2.2.1 功能目标

在商品比价系统中, 用户需要建立商品库。在这里我通过允许用户收藏商品的方式, 建立商品库, 用户选择收藏, 即把爬虫到的商品信息收入商品库中。注意需要存储商品的各种信息进入数据库, 以便展示商品详细信息。

4.2.2.2 触发条件

以用户身份登录了商品比价网站, 同时选择了收藏某件搜索到的商品。

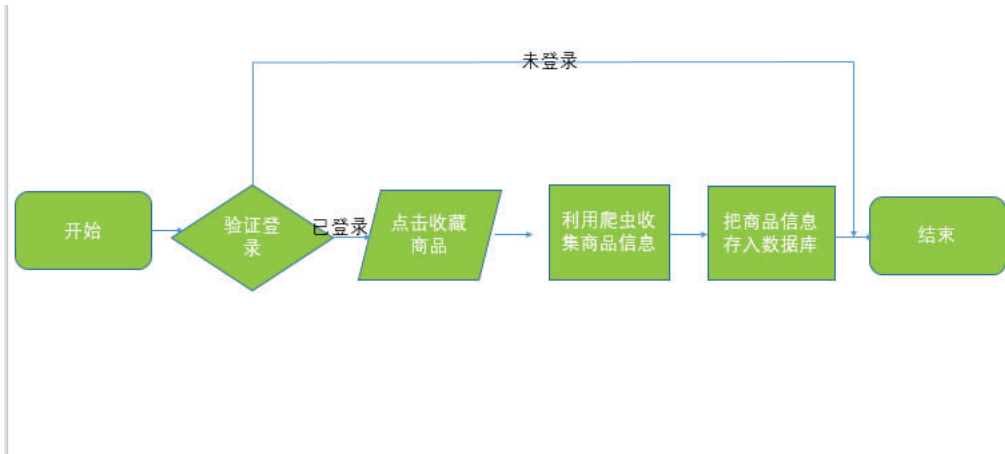
4.2.2.3 输入数据

名称	标识	类型	输入方式
商品名称	Name	String	系统输出
商品信息	Information	Array	爬虫读取
用户名	Username	String	系统读取
商品价格	Price	Float	爬虫读取

4.2.2.4 输出结果

数据库内添加商品信息, 商品进入商品库。

4.2.2.5 流程图



4.2.2.6 测试计划

输入	预期结果
用户未登录	返回“请先登录再进行搜索!”, 弹出登录界面
用户合法收藏商品	正确完成

4.2.3 用户查看商品详细信息

4.2.3.1 功能目标

用户可以在商品收藏界面，查看商品的详细信息，包括名称，多级品类，规格，条码，图片等，同时可以查询商品价格走势图。

4.2.3.2 触发条件

用户登录商品比价系统，收藏商品后进入商品收藏界面。

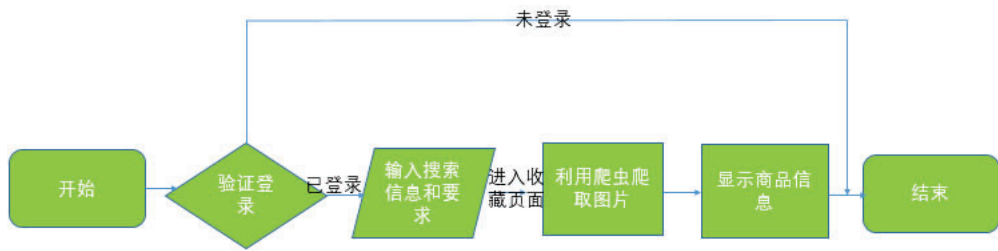
4.2.3.3 输入数据

名称	标识	类型	输入方式
商品名称	Name	String	系统输入

4.2.3.4 输出结果

名称	标识	类型	输出方式
多级品类	Series	String	系统输出
规格	Amount	String	系统输出
条码	Code	图片	系统输出
图片	Picture	图片	系统输出
价格	Price	Float	系统输出
时间	Time	Time	系统输出

4.2.3.5 流程图



4.2.3.6 测试计划

输入	预期结果
用户未登录	返回“请先登录再进行搜索!”, 弹出登录界面
商品名称	显示商品信息

开发完成后补充：不需要条码信息，历史价格图表同时在商品信息中显示。

4.2.4 用户对商品设置降价提醒

4.2.4.1 功能目标

在用户选择需要发送降价提醒的商品后，定时从官网上爬取价格，如果降价，则返回新价格，通过 App 或者邮件提示的方式发送降价提醒。

4.2.4.2 触发条件

用户登录，选择的商品降价。

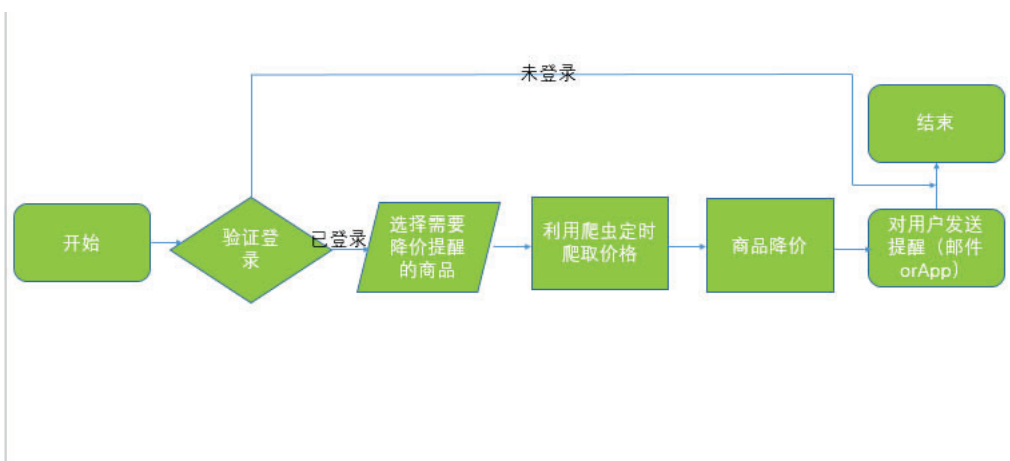
4.2.4.3 输入数据

名称	标识	类型	输入方式
商品名称	Name	String	外部输入

4.2.4.4 输出结果

名称	标识	类型	输出方式
价格	Price	Float	系统输出
时间	Time	Time	系统输出

4.2.4.5 流程图



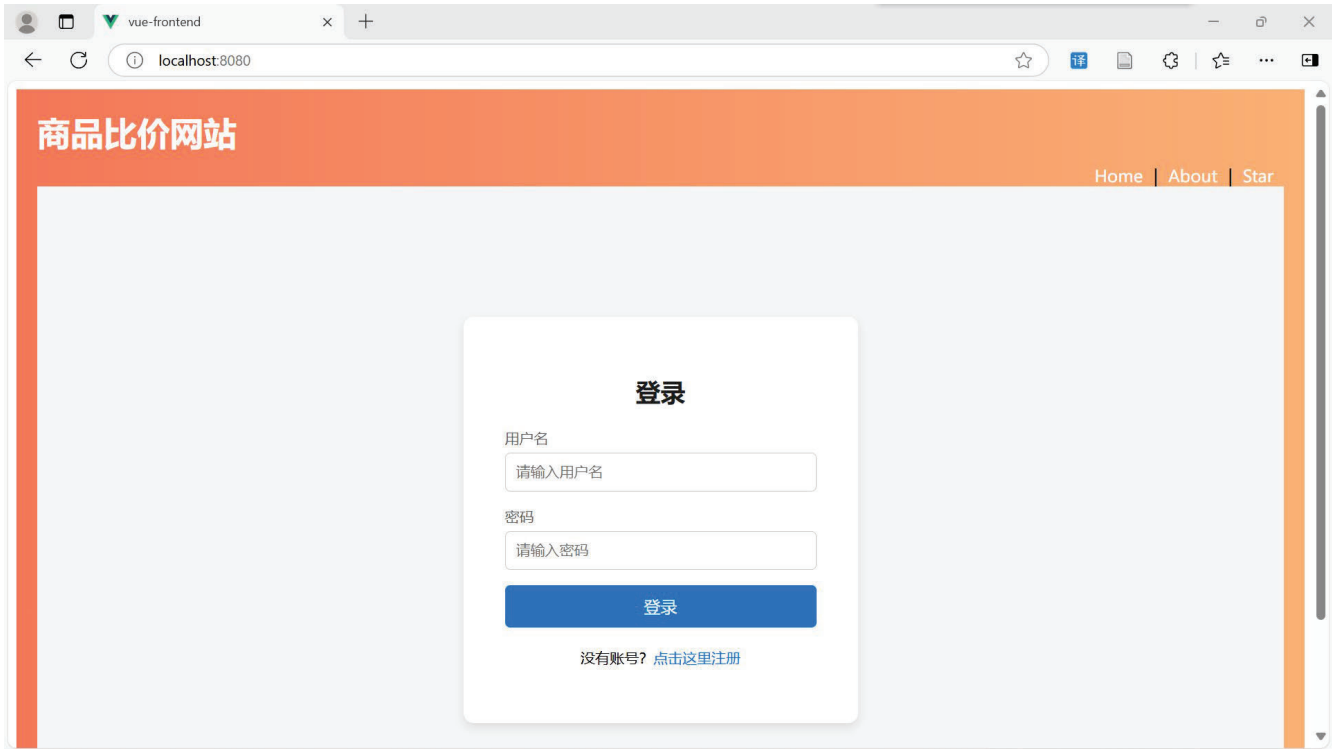
4. 2. 4. 6 测试计划

输入	预期结果
用户未登录	返回“请先登录再进行搜索!”，弹出登录界面
商品名称	正确完成降价提醒

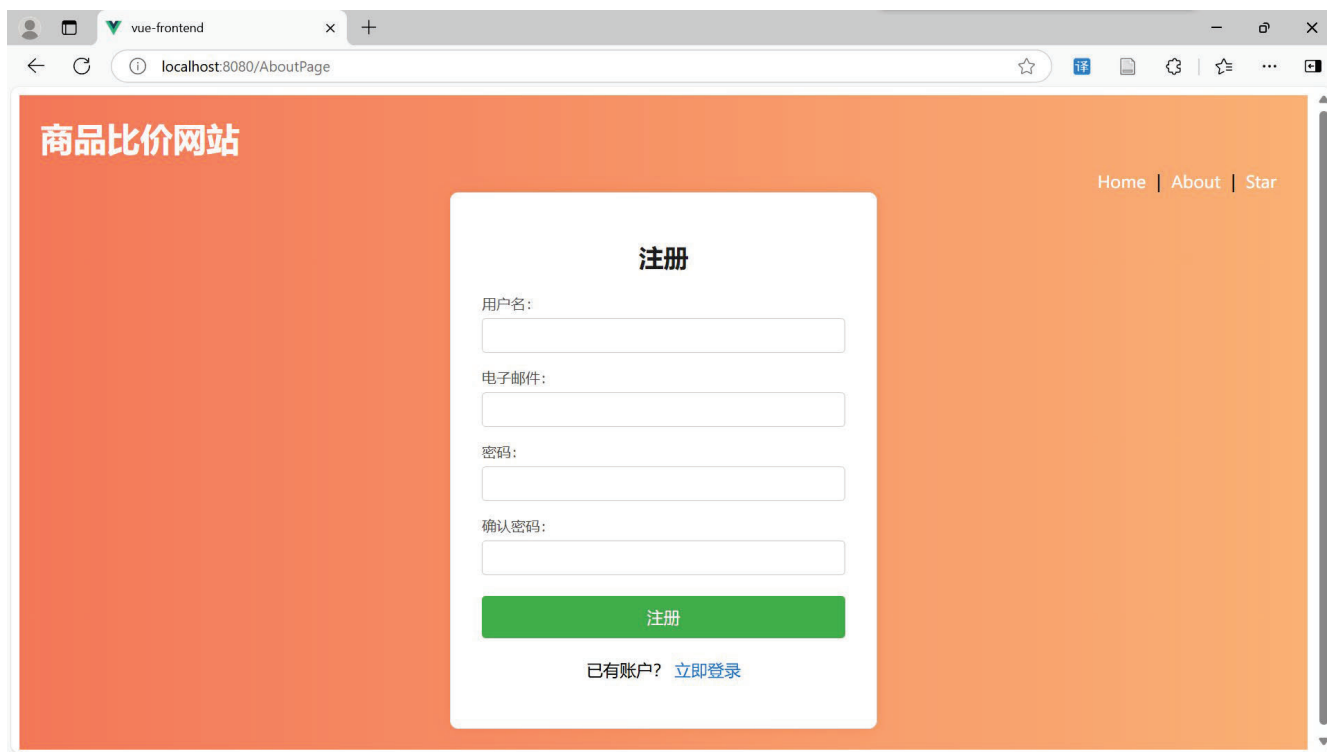
开发完成后补充：用户如果没有登录是看不到收藏商品，也无法收藏商品的，所以这个测试不需要。需要的是测试用户不登陆时，看不到收藏商品。

5. 用户界面

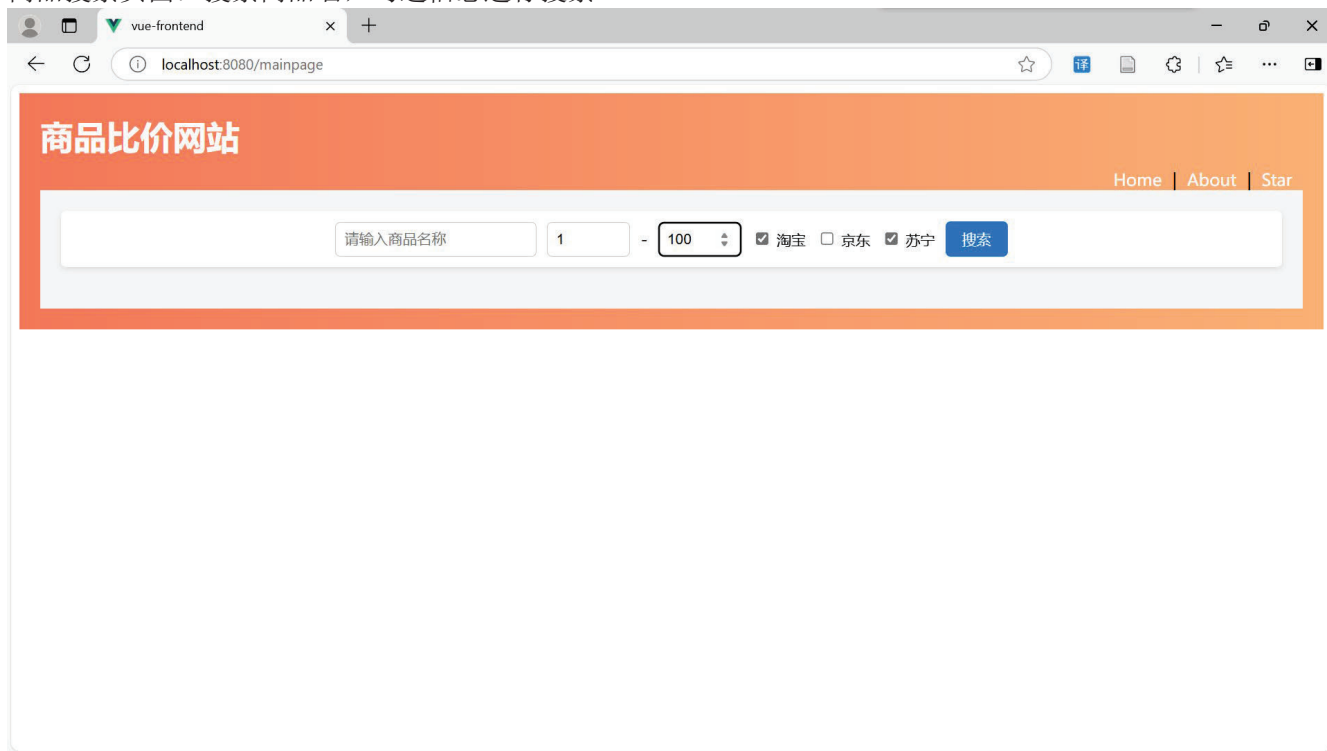
首页：可以登录或者注册



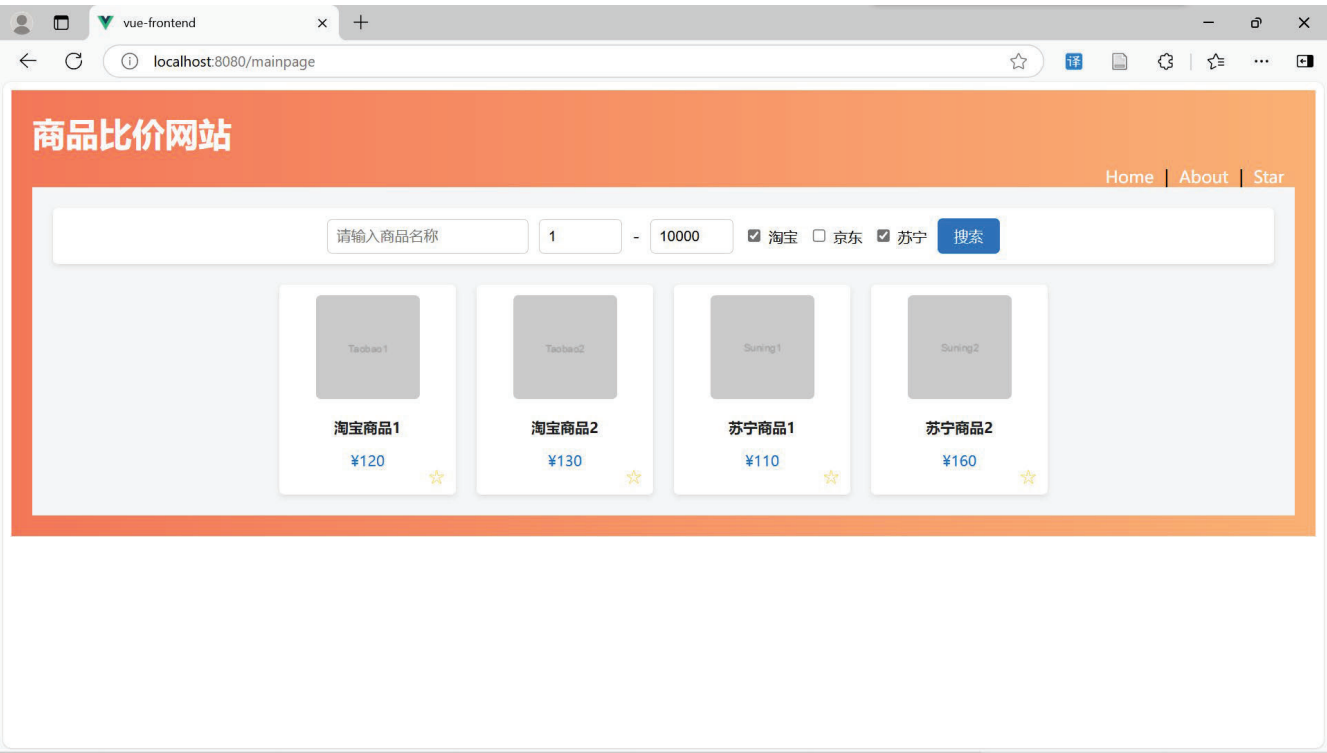
注册页面：填入信息进行注册



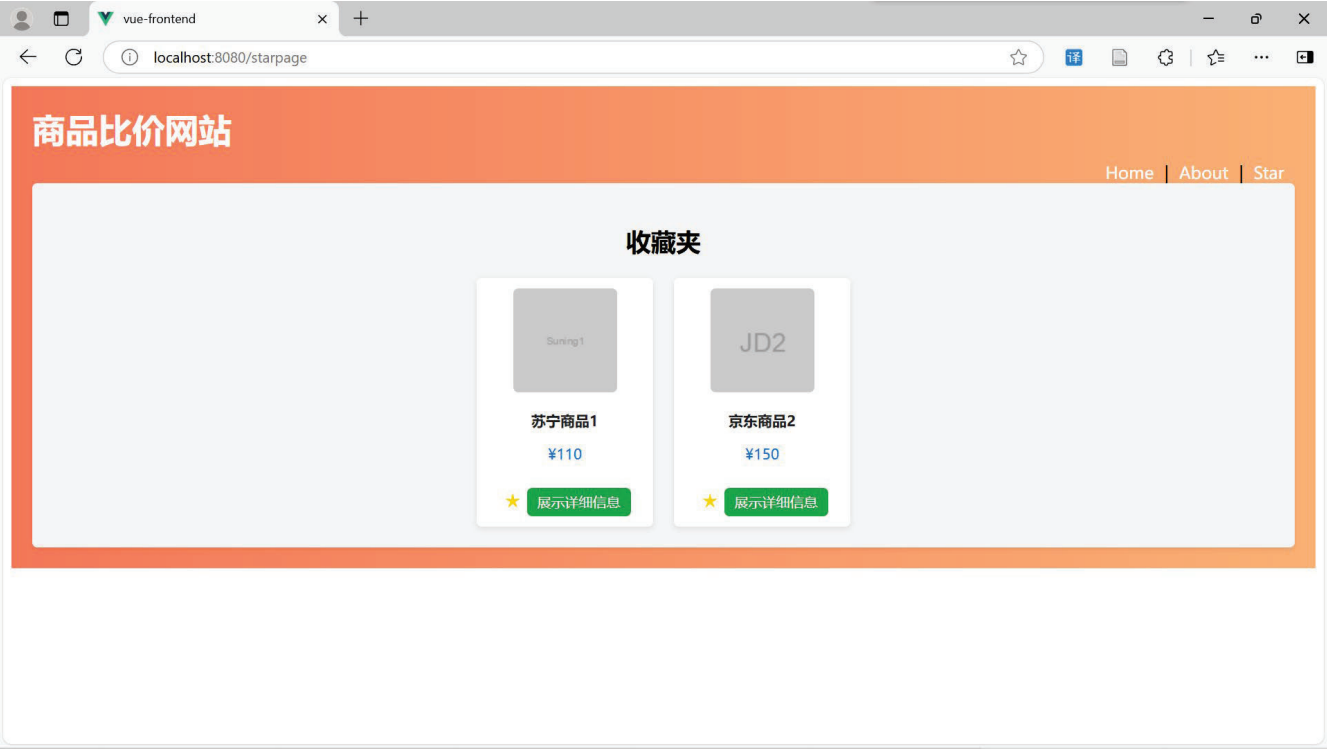
商品搜索页面：搜索商品名，勾选信息进行搜索



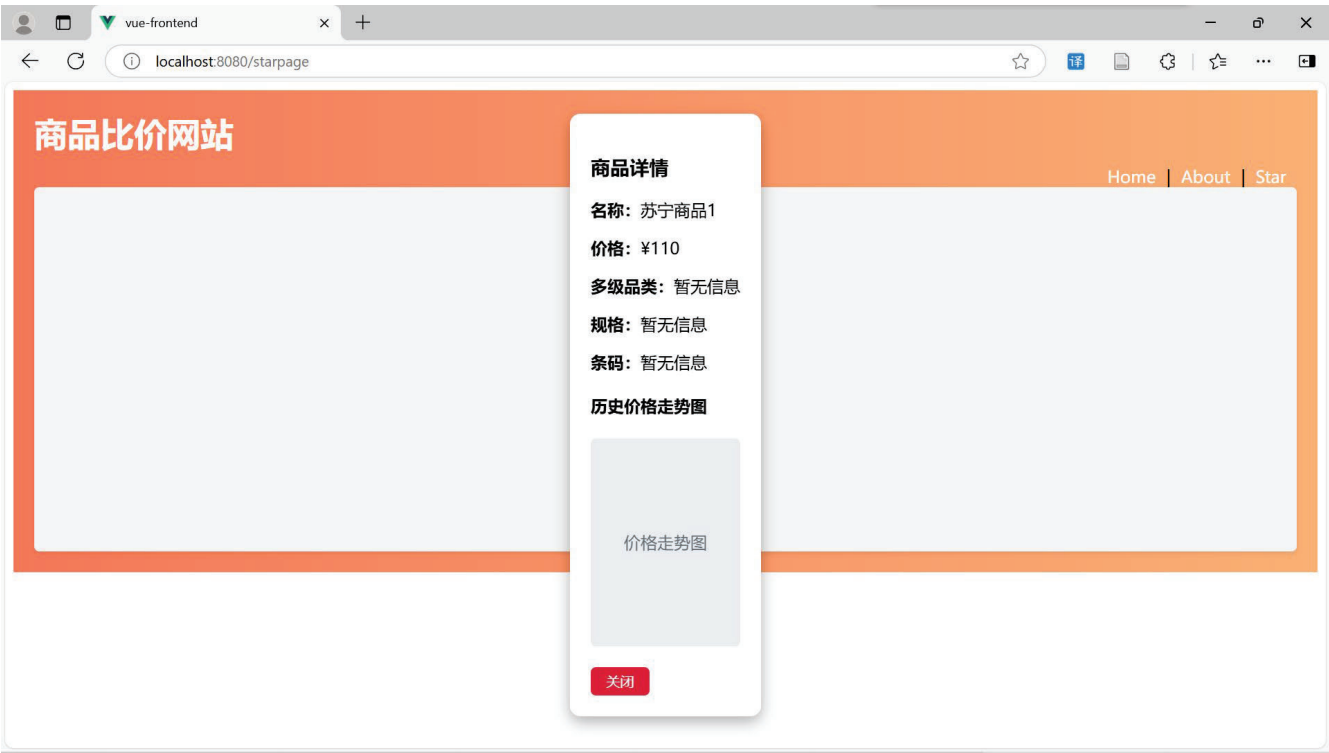
搜索结果展示页面：展示搜索到的商品名



商品库页面：进入收藏夹（即为商品库）



显示商品信息页面：详细展示选择的商品信息和价格变化图



6. 数据库设计

6.1 逻辑结构设计

- 1、用户注册表（用户名，邮箱，密码）
- 2、商品库表（用户名，商品名称，平台，商品多级品类，规格）
- 3、某商品价格表（用户名，商品名称，时间戳，价格）

6.2 物理结构设计

编号	属性名	字段名称	数据类型	长度	备注
1	用户名	Username	String		主键
2	邮箱	Email	String		不同
3	密码	Password	String		至少六字节

用户注册表

编号	属性名	字段名称	数据类型	长度	备注
1	用户名	Username	String		外键
2	商品名称	Name	String		主键
3	平台	Platform	String		苏宁/淘宝/京

					东
4	多级品类	Series	String		非空
5	规格	Amount	String		非空

商品库表

编号	属性名	字段名称	数据类型	长度	备注
1	用户名	Username	Strin		外键
2	商品名称	Name	String		外键
3	时间戳	Time	Time		主键
4	价格	Price	Float	4	非空

某商品价格表

开发完成后补充：数据库最后设计有所修改，如下：

```
class User(Base):
    __tablename__ = "users"
    username = Column(String, primary_key=True, index=True)
    email = Column(String, unique=True, index=True)
    password = Column(String)
```

```
Muzibing
class Product(Base):
    __tablename__ = "products"
    id = Column(Integer, primary_key=True, index=True, autoincrement=True)
    name = Column(String, index=True)
    price = Column(Float)
    image = Column(String)
    from_ = Column(String) # 表示商品来源
    link = Column(String)
    user_username = Column(String, ForeignKey("users.username"))
    created_at = Column(DateTime, default=func.now())

    user = relationship( argument: "User", backref="products")
```

没有单独设计某商品价格表，不然表格数目会很多，删除，创建表格操作也会很麻烦。

7. 运行设计

用户可以注册账号再进行登录，注册时需要输入邮箱，登录时输入账号密码进行登录，登录成功后在页面上会显示用户名。

在商品搜索界面，用户可以搜索需要的商品，并且勾选价格区间和选择网站（淘宝、京东、苏宁），搜索后出现价格，可以进行商品价格比较。支持分词查询。

用户登录后，可以选择将搜索到的商品收藏进入商品库，在商品库中可以查看商品详细信息，也可以选择商品设置成需要降价提醒，提醒可以选择邮件提醒或者 App 提醒。

当选中的商品降价后，系统会将降价商品提醒用户。

8. 系统出错设计

8.1 出错信息

- 1) 数据库连接错误：由于网络问题、数据库服务器故障或配置错误等原因，无法连接到数据库，导致无法访问用户账户信息或执行交易操作。
- 2) 数据库访问权限错误：某些用户可能无法访问其账户信息或执行特定操作，因为他们缺乏相应的数据库访问权限。
- 3) 数据库事务错误：在执行银行交易时，如果发生数据库事务失败，可能会导致交易中断或部分执行，这可能会导致用户账户余额不一致或交易丢失。
- 4) 数据库死锁：如果多个用户同时访问数据库并尝试执行事务，可能会导致死锁，使某些事务无法继续执行，从而影响系统的正常运行。
- 5) 数据库数据完整性错误：如果数据不一致或违反了数据库的完整性约束（如唯一性约束或外键约束），可能会导致系统错误或数据丢失。
- 6) 数据库性能问题：如果数据库设计不佳或者查询语句效率低下，可能会导致系统响应缓慢，影响用户体验。
- 7) 数据库备份和恢复错误：如果数据库备份或恢复过程出现错误，可能会导致数据丢失或不一致，对用户账户信息造成损害。

8.2.补救措施

1) 对于数据库连接错误

检查网络连接，确保网络连接稳定，并且数据库服务器可访问。

检查数据库服务器状态，确保数据库服务器正在运行，并且服务正常。

检查连接字符串，确保连接字符串中包含正确的数据库服务器地址、端口号和凭据信息。

2) 对于数据库访问权限错误

更新权限设置，确保每个用户都具有适当的数据库访问权限，包括读取、写入和执行权限。

使用角色管理权限，通过数据库角色管理权限，以便更轻松地管理和分配权限。

3) 对于数据库事务错误

实施事务管理。使用数据库事务来确保在交易执行期间的数据一致性和完整性，并在事务失败时进行回滚或处理异常。

优化事务。尽量减少事务持续时间，避免长时间持有数据库锁或资源。

